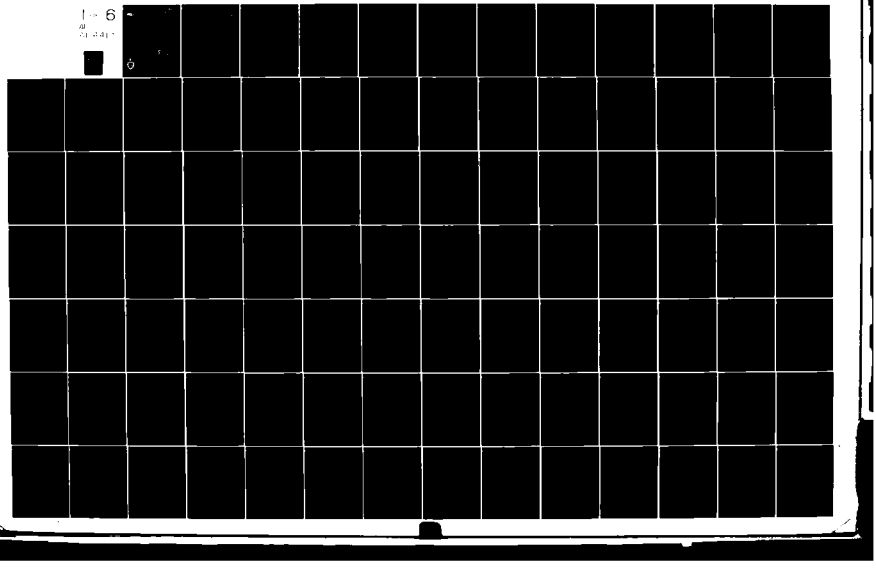
AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL





-TR-0107

AD

Reports Control Symbol
OSD - 1366

SUPPLEMENT TO EOSAEL 80
VOLUME II
USER'S MANUAL

PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND FLASH

FEBRUARY 1982

By

R. G. Steinhoff

DTIC FILE COPY

DTIC
ELECTE
MAY 14 1982
S D
D

Approved for public release; distribution unlimited.



US Army Electronics Research and Development Command

Atmospheric Sciences Laboratory

White Sands Missile Range, NM 88002

82 05 11 016

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ASL-TR-0107	2. GOVT ACCESSION NO.: AD-A114417
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: SUPPLEMENT TO EOSAEL 80 Vol II USER'S MANUAL PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND FLASH	5. TYPE OF REPORT & PERIOD COVERED: Final Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): R. G. Steinhoff	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: DA Task AH71612111 H710011
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Electronics Research and Development Command, Adelphi, MD 20783	12. REPORT DATE: February 1982
	13. NUMBER OF PAGES: 510
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

DTIC COPY INSPECTED 2

Accession For
NTIS GRA&I [X]
DTIC TAB [ ]
Unannounced [ ]
Justification
By
Distribution/
Availability Codes
Dist | Special
A

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

EOSAEL 80
Computer Codes
Transmission
Transport and diffusion

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Computer listing of the Electro-Optical Systems Atmospheric Effects Library, version 80-B.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

CONTENTS

INTRODUCTION 7

PROGRAM EOSAEL 9
SUBROUTINE RESET 18
SUBROUTINE COMPLT 19
SUBROUTINE ILLUM 20
FUNCTION SOLARS 21
FUNCTION SMOON 22
FUNCTION JPASCT 23
SUBROUTINE PFUNC 24

SUBROUTINE XSCALE 29
SUBROUTINE SLANT 33

SUBROUTINE TURB 35
FUNCTION DESUB 43
FUNCTION FALPH 44
SUBROUTINE FFT4 45
FUNCTION GAUSS 47
SUBROUTINE MEANVR 48
SUBROUTINE SPECT 49
SUBROUTINE SPREAD 50
SUBROUTINE THETO 51

SUBROUTINE BASCAT 52
SUBROUTINE PFUNC 24
SUBROUTINE BKWD 66
SUBROUTINE CONV 68
SUBROUTINE ELM 69
SUBROUTINE FIND 70
SUBROUTINE FWRD 71
SUBROUTINE GAS 72
SUBROUTINE GMAX 73
SUBROUTINE MATRX 74
SUBROUTINE ROTAT 75
SUBROUTINE SMOOZ 76
SUBROUTINE START 77
SUBROUTINE TRAVRS 79
SUBROUTINE USCA 81

SUBROUTINE SMOKE 82
FUNCTION JPASCT 23
SUBROUTINE CLSMOK 93
SUBROUTINE SCONST 96
SUBROUTINE SMASSP 98
SUBROUTINE STRANS 99
SUBROUTINE WGGEOM 100
FUNCTION QROOT 102
SUBROUTINE XYZINT 103
SUBROUTINE BRATE 108
SUBROUTINE GPUFF 109

SUBROUTINE LZTRAN 111
SUBROUTINE LZIDNM 115

SUBROUTINE DRTRAN 117
SUBROUTINE AMOUNT 130
SUBROUTINE ATMCAL 131
SUBROUTINE AVRG 135
SUBROUTINE CLIMB 136
SUBROUTINE CLDIM 138
SUBROUTINE COMPCL 140
SUBROUTINE CONLEN 143
SUBROUTINE CONVRT 145
FUNCTION CSPHER 147
FUNCTION CWAKE 149
FUNCTION CWIND 151

PRECEDING PAGE BLANK-NOT FILMED

SUBROUTINE DIFEQ 154
FUNCTION DIFFUS 156
FUNCTION DOTPRD 157
FUNCTION DTERPI 158
SUBROUTINE DTERPS 160
SUBROUTINE DUSTCL 161
FUNCTION ERF 164
SUBROUTINE FIT 165
FUNCTION FUNCT 166
SUBROUTINE GAMMA 167
FUNCTION GFUN 168
SUBROUTINE GRAD2 169
SUBROUTINE GRAND 170
SUBROUTINE GREEN 172
SUBROUTINE GREEN1 173
SUBROUTINE MOMENT 175
SUBROUTINE PATH 178
SUBROUTINE PERP 179
SUBROUTINE PRECL 180
SUBROUTINE PRETRN 183
SUBROUTINE PREVEH 185
SUBROUTINE RISE 188
SUBROUTINE RKM 190
SUBROUTINE SETUP 192
SUBROUTINE SOURCE 194
SUBROUTINE TEMP 197
FUNCTION TMPCAL 198
SUBROUTINE TRAP 200
SUBROUTINE TRNCAL 203
SUBROUTINE TRNCHK 207
SUBROUTINE TRNCLD 208
SUBROUTINE UNIT 210
SUBROUTINE VEHCL 211
SUBROUTINE VEHTRN 213
SUBROUTINE VSRC 215
SUBROUTINE VSUM 217
SUBROUTINE WIN 218
FUNCTION WNDCAL 219

SUBROUTINE NMMW 221
FUNCTION AB 224
FUNCTION DOP 225
SUBROUTINE INTRP 226
SUBROUTINE MMH2O 227
SUBROUTINE MMIDX 228
SUBROUTINE MMOXY 229
SUBROUTINE MMRAN 230
SUBROUTINE MMSNO 232
SUBROUTINE MMWFG 234
SUBROUTINE MMWGS 235
FUNCTION PFR 237
FUNCTION PSAT 238
FUNCTION SUPK 239

SUBROUTINE CLTRAN 240
SUBROUTINE LAYRXY 244
SUBROUTINE CYLXY 245
SUBROUTINE CLEXTN 247
SUBROUTINE LISOUT 248
SUBROUTINE DEFSET 250
SUBROUTINE CLREAD 251

SUBROUTINE SCREEN 253
SUBROUTINE XSCALE 29
SUBROUTINE SLANT 33
FUNCTION JPASCT 23
SUBROUTINE PFUNC 24
SUBROUTINE CWIC 254
SUBROUTINE CWIC1 261
SUBROUTINE CWIC3 264

SUBROUTINE CWIC4 265
SUBROUTINE ITAM 267
SUBROUTINE CINV 274
SUBROUTINE INTAL 279
SUBROUTINE CASE1 282
SUBROUTINE CASE2 283
SUBROUTINE CASE3 284
SUBROUTINE CYCLE 285
SUBROUTINE TREQ 286

SUBROUTINE FCLOUD 287
SUBROUTINE ILLUM 20
FUNCTION SOLARS 21
FUNCTION SMOON 22
SUBROUTINE THRMCL 293
SUBROUTINE SSCLD 294
SUBROUTINE PFNN 295
SUBROUTINE MSCLD 297
FUNCTION ETAINT 298

SUBROUTINE OVRCST 299
SUBROUTINE ILLUM 20
FUNCTION SOLARS 21
FUNCTION SMOON 22
FUNCTION G2 303
FUNCTION E1 304

SUBROUTINE GRNADE 305
SUBROUTINE CONCN 309
SUBROUTINE GOGET 311
SUBROUTINE PARMS 312
SUBROUTINE EXTIN 313
SUBROUTINE UMEAN 314
SUBROUTINE SUMA 315
SUBROUTINE LOCAT 316
SUBROUTINE DATRD 317

SUBROUTINE LT4M 321
SUBROUTINE XSCALE 29
SUBROUTINE SLANT 33
SUBROUTINE ABSORB 328
SUBROUTINE CKER 336
SUBROUTINE FREQSL 337
SUBROUTINE H2OVAP 339
SUBROUTINE H2O410 340
SUBROUTINE LTPATH 341
SUBROUTINE MOLSCT 343
SUBROUTINE NH3 344
SUBROUTINE NITRIC 345
SUBROUTINE NITRO 346
SUBROUTINE NO2 347
SUBROUTINE OZONE 348
SUBROUTINE POINT 349
FUNCTION RESFN 350
SUBROUTINE SO2 351
SUBROUTINE UNIMIX 352
SUBROUTINE UVOZNE 353

SUBROUTINE SPOT 354
SUBROUTINE LT4M 321
SUBROUTINE XSCALE 29
SUBROUTINE SLANT 33
SUBROUTINE PFUNC 24
FUNCTION SOLARS 21
FUNCTION SMOON 22
FUNCTION ALBEDO 361
FUNCTION BLACK 362
SUBROUTINE COEFS 363
SUBROUTINE DIAG 364
SUBROUTINE INDAT 366

SUBROUTINE OUTPUT 369
SUBROUTINE PATHRD 372
SUBROUTINE ZERO 374

SUBROUTINE CLIMAT 375

PROGRAM AGAUS 379
SUBROUTINE AGXP1 385
SUBROUTINE AGXP2 390
SUBROUTINE AGXP3 396
SUBROUTINE ANGLE 398
FUNCTION GAMMA 399
SUBROUTINE GUSET 400
SUBROUTINE MIEGX 401
SUBROUTINE WATER 405
SUBROUTINE GAUS 408
SUBROUTINE DIMER 409
BLOCK DATA 410

PROGRAM FLASH1 411
SUBROUTINE FLASH 412
SUBROUTINE DATRD 414
SUBROUTINE DATWT 416
SUBROUTINE GETIM 418
SUBROUTINE IRBLC 421
SUBROUTINE VSBLC 422

EOSAEL TEST DATA (NEWRUN) 423

EOSAEL OUTPUT (EOOUT) 426

INTRODUCTION

This listing of EOSAEL 80-B is a supplement to Volume II[1] and supersedes all previous listings.[2] The current listing is complete as of 8 February 1982 and has revisions one through five incorporated into it.

EOSAEL 80-B differs from EOSAEL 80 in that modules SPOT, LT4M, NMMW, CLIMAT, BASCAT, and TURB have been extensively revised and, therefore, appear with new sequence numbers. All other modules have their original sequencing, except where revisions have been inserted or deleted.

The programs are listed by module with each module followed by its subroutines. Subroutines that have been listed for prior modules in the listing are not repeated in the source listing. The table of contents lists each module along with all its corresponding subroutines and the page number of each subroutine in the listing. The elements EOMAIN, COMPLT, and RESET, which are always to be resident, appear only at the beginning of the table of contents and the source listing.

Also included herein is a sample input file, NEWRUN, and an output file, EOOUT, produced by using the aforementioned sample input file.

The supplemental codes AGAUS and FLASH are supplied with EOSAEL 80. FLASH is described in appendix A of volume II of the User's Manual[1] and is further described in the comments of the source listing. Operating instructions for the AGAUS code may be found in the comments of the source listing. Manuals for AGAUS are available upon written request from the US Army Atmospheric Sciences Laboratory, White Sands Missile Range, New Mexico.

---

[1]Shirkey, R. C., and S. G. O'Brien, EOSAEL 80, Volume II, User's Manual, ASL-TR-0073, US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM, 1981.

[2]Steinhoff, R. G., Program Listings for EOSAEL 80 and Ancillary Codes AGAUS and FLASH, ASL-TR-0073 (Supplement), US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM, 1981.

```
      PROGRAM EOSAEL
C  MAIN PROGRAM FOR EOSAEL 80                                                  EOM00010
C                                                                             EOM00020
      REAL LOTRNS,LAZTRN,LZTRN,MMTRAN,MMWTRN,IPNAM,IAL,IALB1,IALB2            EOM00030
      LOGICAL ISPOT,N16,LOREAD                                                EOM00040
      COMMON /SPOTLO/ISPOT,LOREAD,N16                                         EOM00050
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                           EOM00060
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,            EOM00070
     1       WNDVEL,WNDDIR,IPASCT                                             EOM00080
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUEOM00090
      COMMON /GEOMET/PTS(15),IGEOSW                                           EOM00100
C          IOIN - CARD READER                                                 EOM00110
C          IOOUT - PRINTER                                                    EOM00120
C          IPHFUN - UNIT UPON WHICH PHASE FUNCTION DATA RESIDES               EOM00130
C          LOUNIT - UNIT UPON WHICH LT4M ATM DATA RESIDES                     EOM00140
C          NDIRTU - UNIT UPON WHICH DRTRAN DATA RESIDES                       EOM00150
C          NCLIMT - UNIT FOR CLIMATOLOGICAL DATA                              EOM00160
C          KSTOR  - AUXILLARY START/RESTART UNIT FOR BASCAT                   EOM00170
C          NPLOTU - OPTIONAL UNIT FOR WRITING RESULTS FOR SUBSEQUENT
C                   PLOTTING PURPOSES BY THE USER.
      DIMENSION TRAN(16),RADA(16),RADG(16),IPROGN(20)                         EOM00180
      DIMENSION IDOPGM(20),IPNAM(40)                                          EOM00190
      DIMENSION IAL(12),DAT(10)                                               EOM00200
C     FOR UNIVAC                                                              EOM00210
      DATA IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR/                      EOM00220
     1     5,6,3,8,7,24,25/                                                   EOM00230
C     PROGRAM NAMES                                                           EOM00240
      DATA IPNAM/4HSPOT,4HTURB,4HBASC,4HLT4M,4HXSCA,4HSMOK,4HDRTR,            EOM00250
     1     4HLZTR,4HNMMW,4HCLTR,4HSCRE,4HFCLO,4HOVRC,4HGRNA,4H*    ,          EOM00260
     2     4H*    ,4H*    ,4H*    ,4H*    ,4HCLIM,4H     ,4H     ,4HAT   ,  EOM00270
     3     4H     ,4HLE   ,4HE    ,4HAN   ,4HAN   ,4H     ,4HAN   ,4HEN   ,  EOM00280
     4     4HUD   ,4HST   ,4HDE   ,4H     ,4H     ,4H     ,4H     ,4H     ,  EOM00290
     5     4HATE /                                                            EOM00300
C     CARD MNEMONICS                                                          EOM00310
      DATA IAL/4HEGRU,4HVIS  ,4HFREQ,4HWAVL,4HWVNU,4HRESF,                    EOM00320
     14HTARG,4HRCVR,4HDESG,4HOBSV,4HBFCL,4HGO   /                             EOM00330
      DATA PI,TORRMB,CDEGK/3.14159265,1.33322,273.16/                         EOM00340
      DATA PTS/15*0.0/                                                        EOM00350
      ISTART=0                                                                EOM00360
      CLDAMT=0.                                                               EOM00370
      CLDHYT=0.                                                               EOM00380
      FOGPRB=0.                                                               EOM00390
      PI2=PI/2.                                                               EOM00400
      PIRAD=PI/180.                                                           EOM00410
      TWOPI=2.0*PI                                                            EOM00420
      WRITE (IOOUT,1060)                                                      EOM00430
C*************************** I/O ***************************************EOM00440
C                                                                             EOM00450
C** INPUT TO EOSAEL IS CARD ORDER-INDEPENDENT, WITH EACH INPUT RECORD         EOM00460
C** HAVING A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4. THE ONLY EXCEPTION        EOM00470
C** TO THIS RULE IS THE GO SENTINEL CARD, WHICH MUST BE THE LAST RECORD       EOM00480
C** IN THE INPUT SEQUENCE. ALL RECORDS ARE READ IN UNDER THE                 EOM00490
C** FORMAT (2A4,1X,10E7.4). INTEGERS MUST BE INPUT AS REAL                   EOM00500
C** NUMBERS IN THIS COMMON FORMAT SCHEME, THEY ARE LATER FIXED TO            EOM00510
C** THE INTEGER TYPE. THE IDENTIFIERS FOR EACH OF THE INPUT                  EOM00520
C** RECORDS ARE AS FOLLOWS :                                                 EOM00530
C                                                                             EOM00540
C-------------------------------------------------------------------------  EOM00550
C     CARD IDENTIFIER : EORUN                                                 EOM00560
C     VARIABLES READ : NUMRUN                                                 EOM00570
C      NUMRUN - NUMBER OF TIMES EOSAEL DRIVER IS TO BE CYCLED                 EOM00580
C               DEFAULT IS 1.                                                 EOM00590
C-------------------------------------------------------------------------  EOM00600
C     CARD IDENTIFIER : VIS                                                   EOM00610
C     VARIABLES READ : VIS,EXTN55,EXTN                                        EOM00620
C      VIS - VISIBILITY AT WAVELENGTH OF 0.55 MICRONS (KM)                    EOM00630
C      EXTN55 - EXTINCTION COEFFICIENT AT 0.55 MICRONS (KM**-1)               EOM00640
C      EXTN - EXTINCTION COEFFICIENT AT INPUT WAVELENGTH (KM**-1)             EOM00650
C    ** NOTE : IF THE VIS CARD IS NOT INPUT, A WARNING IS PRINTED             EOM00660
C              AND THE VISIBILITY IS SET TO A DEFAULT VALUE OF 10 KM.         EOM00670
```


PRECEDING PAGE BLANK-NOT FILMED

```
C     ** NOTE : IF EXTN55 IS INPUT AS A VALUE LESS THAN 0.0001, IT        EOM00680
C               IS SET EQUAL TO THE QUOTIENT 3.912/VIS. IF VIS IS         EOM00690
C               INPUT AS A VALUE LESS THAN 0.0001, IT IS SET EQUAL TO     EOM00700
C               THE QUOTIENT 3.912/EXTN55.                                EOM00710
C      ** NOTE : EXTN IS NEEDED ONLY FOR BASCAT                           EOM00720
C-----------------------------------------------------------------------  EOM00730
C                                                                         EOM00740
C** ONLY ONE OF THE FOLLOWING THREE CARDS MAY BE INPUT FOR A GIVEN         EOM00750
C** CYCLE OF EOSAEL. IF NONE OF THESE CARDS IS PRESENT, AN ERROR           EOM00760
C** MESSAGE IS PRINTED AND EXECUTION IS TERMINATED.                       EOM00770
C                                                                         EOM00780
C                                                                         EOM00790
C      CARD IDENTIFIER : FREQ                                             EOM00800
C      VARIABLES READ : FREQ1, FREQ2, MULDV                               EOM00810
C       FREQ1 - LOWER INPUT FREQUENCY (GHZ)                               EOM00820
C       FREQ2 - HIGHER INPUT FREQUENCY (GHZ)                              EOM00830
C       MULDV - FREQUENCY INCREMENT FOR SPOT AND/OR LT4M (GHZ)            EOM00840
C                                                                         EOM00850
C      CARD IDENTIFIER : WAVL                                             EOM00860
C      VARIABLES READ : WAVE1, WAVE2, MULDV                               EOM00870
C       WAVE1 - SHORTER INPUT WAVELENGTH (MICRONS)                        EOM00880
C       WAVE2 - LONGER INPUT WAVELENGTH (MICRONS)                         EOM00890
C       MULDV - WAVELENGTH INCREMENT FOR SPOT AND/OR LT4M (MICRONS)       EOM00900
C                                                                         EOM00910
C      CARD IDENTIFIER : WVNUM                                            EOM00920
C      VARIABLES READ : WVNUM1, WVNUM2, MULDV                             EOM00930
C       WVNUM1 - LOWER INPUT WAVENUMBER (CM**-1)                          EOM00940
C       WVNUM2 - HIGHER INPUT WAVENUMBER (CM**-1)                         EOM00950
C       MULDV - WAVENUMBER INCREMENT FOR SPOT AND/OR LT4M (CM**-1)        EOM00960
C-----------------------------------------------------------------------  EOM00970
C                                                                         EOM00980
C** THE NEXT CARD DETERMINES WHETHER A SENSOR RESPONSE FUNCTION           EOM00990
C** IS DESIRED FOR BROAD BAND CALCULATIONS. THIS OPTION IS                EOM01000
C** INVOKED ONLY IF THIS CARD IS PRESENT.                                 EOM01010
C                                                                         EOM01020
C      CARD IDENTIFIER : RESF                                             EOM01030
C      VARIABLES READ : NONE HERE - SEE SPOT OR LT4M WRITEUP FOR          EOM01040
C                       PROPER PLACEMENT OF RESPONSE FUN CARDS.           EOM01050
C-----------------------------------------------------------------------  EOM01060
C                                                                         EOM01070
C** THE NEXT FIVE CARDS COMPRISE THE GEOMETRICAL OPTION OF EOSAEL.        EOM01080
C** THIS OPTION IS USEFUL FOR EOSAEL RUNS WHERE SEVERAL MODULES           EOM01090
C** EXAMINE DIFFERENT ATMOSPHERIC OBSCURATION EFFECTS ALONG THE           EOM01100
C** SAME PHYSICAL PATH. THE GEOMETRICAL OPTION ASSURES THAT THE           EOM01110
C** POINTS OF REFERENCE IN THE SCENARIO UNDER EXAMINATION ARE             EOM01120
C** CONSISTENTLY SPECIFIED FOR ALL MODULES. IT SHOULD BE NOTED            EOM01130
C** THAT THIS OPTION IS ACTIVATED WHENEVER ANY OF THE FIVE CARDS IS       EOM01140
C** ENCOUNTERED. ONCE THE OPTION IS ACTIVATED IT IS IMPORTANT THAT        EOM01150
C** AT LEAST THE FIRST TWO CARDS (TARG AND RCVR) BE INPUT TO              EOM01160
C** DEFINE THE PHYSICAL PATH, SINCE THIS OPTION WILL OVERRIDE             EOM01170
C** POSITIONS OR LENGTHS CONTAINED IN NORMAL INPUT TO ALL MODULES.        EOM01180
C** THE GEOMETRICAL INPUT CONSISTS OF FIVE SETS OF COORDINATES            EOM01190
C** WHICH OBEY THE FOLLOWING CONVENTIONS :                               EOM01200
C**       (A) ALL COORDINATES ARE DIMENSIONED IN KILOMETERS               EOM01210
C         (B) THE Z-AXIS IS POSITIVE UPWARD                               EOM01220
C         (C) THE Y-AXIS POINTS NORTH                                     EOM01230
C         (D) THE X-AXIS POINTS EAST                                      EOM01240
C                                                                         EOM01250
C** THE FIVE GEOMETRICAL CARDS ARE AS FOLLOWS :                           EOM01260
C                                                                         EOM01270
C                                                                         EOM01280
C      CARD IDENTIFIER : TARG                                             EOM01290
C      VARIABLES READ : PTS(1),PTS(2),PTS(3)                              EOM01300
C       PTS(1-3) - COORDINATES OF THE TARGET (FOR THE DRTRAN MODULE,      EOM01310
C                  THESE ARE THE COORDINATES OF THE TRANSMITTER).         EOM01320
C                                                                         EOM01330
C      CARD IDENTIFIER : RCVR                                             EOM01340
C      VARIABLES READ : PTS(4),PTS(5),PTS(6)                              EOM01350
C       PTS(4-6) - COORDINATES OF THE RECEIVER OR SEEKER                  EOM01360
C                                                                         EOM01370
```

```
C      CARD IDENTIFIER : DESG                                                  EOM01380
C      VARIABLES READ : PTS(7),PTS(8),PTS(9)                                   EOM01390
C       PTS(7-9) - COORDINATES OF THE DESIGNATOR OR SOURCE                     EOM01400
C                                                                              EOM01410
C      CARD IDENTIFIER : OBSV                                                  EOM01420
C      VARIABLES READ : PTS(10),PTS(11),PTS(12)                                EOM01430
C       PTS(10-12) - COORDINATES OF THE OBSERVER USED BY DRTRAN                EOM01440
C                                                                              EOM01450
C      CARD IDENTIFIER : BFCL                                                  EOM01460
C      VARIABLES READ : PTS(13),PTS(14),PTS(15)                                EOM01470
C       PTS(13-15) - COORDINATES OF THE CENTER OF THE CLOUD ELLIPSOID          EOM01480
C                    USED BY BASCAT AND FCLOUD                                 EOM01490
C-----------------------------------------------------------------------       EOM01500
C                                                                              EOM01510
C** THE NEXT INPUT CARD IS THE CLIMATOLOGICAL OPTION CARD. THIS                 EOM01520
C** OPTION ALLOWS USER INPUT OF METEOROLOGICAL PARAMETERS DIRECTLY              EOM01530
C** OR AUTOMATIC INPUT OF CLIMATOLOGY DATA CHARACTERISTIC OF                   EOM01540
C** WEST GERMANY.  IF THIS OPTION IS INVOKED ALL MODULES WILL USE              EOM01550
C** THIS DATA, I.E. MET DATA THAT HAS BEEN INPUT TO A SPECIFIC                 EOM01560
C** MODULE WILL BE OVERRIDDEN.                                                EOM01570
C                                                                              EOM01580
C                                                                              EOM01590
C      CARD IDENTIFIER : CLIMAT                                                EOM01600
C      VARIABLES READ : ICLMAT,LOCAT,MONTH,NHOUR,IWIND,NPRT                    EOM01610
C        ** OR **       ICLMAT,IPASCT,TEMP,PRESS,RH,AH,DP,VIS,WNDVEL,         EOM01620
C                       WINDDIR                                                EOM01630
C          LOCAT - CLIMATOLOGY REGION INDICATOR. LOCAT IS AN INTEGER
C                  (1-4) FOR CENTRAL EUROPE AND
C                  (5-10) FOR MID-EAST.
C
C                  L = 1 - EUROPEAN LOWLANDS,
C                  L = 2 - EUROPEAN RHINE VALLEY,
C                  L = 3 - EUROPEAN HIGHLANDS,
C                  L = 4 - EUROPEAN ALPINE,
C                  L = 5 - MIDEAST DESERTS,
C                  L = 6 - MIDEAST COASTAL,
C                  L = 7 - MIDEAST PERSIAN GULF,
C                  L = 8 - MIDEAST RED SEA,
C                  L = 9 - MIDEAST EASTERN MOUNTAINS, AND
C                  L = 10 - MIDEAST INDUS VALLEY.
C
C          MONTH - AN INTEGER (1-12) INDICATING THE MONTH OF THE YEAR.
C                  MONTH IS USED TO SELECT THE SEASON WHICH IS
C                  APPLICABLE TO THE REGION LOCAT.
C          NHOUR - AN INTEGER (0-23) INDICATING THE TIME OF DAY LOCAL
C                  STANDARD TIME (LST). NHOUR IS USED TO SELECT ONE OF
C                  FOUR TIME PERIODS OF THE DAY 20-02, 03-09, 10-14,
C                  AND 15-19.
C          IWIND - *** NOT USED ***
C          NPRT - A PRINT SELECTOR.
C                  NPRT LE ZERO - DO NOT PRINT CLIMATOLOGICAL DATA.
C                  NPRT GT ZERO - PRINT ALL AVAILABLE MEANS, STANDARD
C                  DEVIATIONS, AND PERCENT OCCURRENCES.
C        ICLMAT = 2.: USER INPUT QUANTITIES                                    EOM01820
C            IPASCT = PASQUILL STABILITY CATEGORY VALID RANGE =1.-6.(A-F)      EOM01830
C            TEMP = TEMPERATURE IN DEGREES C                                   EOM01840
C            PRESS = PRESSURE IN MB (SEA LEVEL IF ICLMAT=1)                    EOM01850
C            RH = RELATIVE HUMIDITY IN %                                       EOM01860
C            AH = ABSOLUTE HUMIDITY - DEFINED HERE AS THE H2O VAPOR            EOM01870
C                 DENSITY IN G/M**3.                                           EOM01880
C            DP = DEW POINT TEMPERATURE IN DEGREES C                           EOM01890
C            VIS = VISIBILITY IN KM                                            EOM01900
C            WNDVEL = WIND VELOCITY IN M/S (DEPNT UPON IWIND IF ICLMAT=1)      EOM01910
C            WINDIR = WIND DIRECTION IN DEGREES (IF ICLMAT=1 WILL BE           EOM01920
C                                    MOST PROBABLE DIRECTION)                  EOM01930
C-----------------------------------------------------------------------       EOM01940
C                                                                              EOM01950
C** THE FOLLOWING CARDS ARE ALSO READ IN UNDER THE COMMON FORMAT                EOM01960
C** USED ABOVE. THE INFORMATION ON EACH OF THESE RECORDS DETERMINES             EOM01970
C** WHICH MODULES ARE SELECTED AND HOW MANY TIMES THE MODULES ARE               EOM01980
```

```
C** TO BE CYCLED WITHIN EACH CYCLE OF THE EOSAEL DRIVER.                    EOM01990
C                                                                           EOM02000
C                                                                           EOM02010
C     CARD IDENTIFIER : (SEE MODULE IDENTIFIERS BELOW)                      EOM02020
C     VARIABLES READ : IDOPGM                                               EOM02030
C      IDOPGM - NUMBER OF TIMES THE SELECTED MODULE IS TO                   EOM02040
C               BE CYCLED WITHIN EACH EOSAEL CYCLE - DEFAULT IS ONE.        EOM02050
C                                                                           EOM02060
C                                                                           EOM02070
C            NO.  MODULE IDENTIFIER     --------- RANGES -------------     EOM02080
C            ---  -----------------      WAVE(UM)              FREQ(GHZ)    EOM02090
C             1       SPOT       .25-2.,3.-5.,8.-12.  *                     EOM02100
C             2       TURB                LT 14.      *                     EOM02110
C             3       BASCAT     ANY WAVELENGTH IN DATA FILE IPHFUN         EOM02120
C             4       LT4M       .25-2.,3.-5.,8.-12.  *                     EOM02130
C             5       XSCALE           1.06,3-5,8-12  *                     EOM02140
C             6       SMOKE          .4-1.2,3-5,8-12  *          94.        EOM02150
C             7       DRTRAN     .4-1.1,3.5-4.,8.5-12.*       94.-140.      EOM02160
C             8       LZTRAN           .8-11.         *                     EOM02170
C             9       NMMW                            *       10-350        EOM02180
C            10       CLTRAN        .2-2.,3.-5.,8.-12.*                     EOM02190
C            11       SCREEN          N/A                                   EOM02200
C            12       FCLOUD     ANY WAVELENGTH IN DATA FILE IPHFUN         EOM02210
C            13       OVRCST           ANY WAVELENGTH                       EOM02220
C            14       GRNADE        SAME AS SMOKE                           EOM02230
C**  NOTE : THE DATA SPECIFIC TO EACH MODULE MUST BE INPUT IN                EOM02240
C      THE SEQUENCE IN THE ABOVE LIST.                                      EOM02250
C-------------------------------------------------------------------------  EOM02260
C     CARD IDENTIFIER : GO                                                  EOM02270
C     VARIABLES READ : NONE                                                 EOM02280
C      END OF READ SENTINEL (MUST BE LAST CARD READ).                       EOM02290
C                                                                           EOM02300
C***************************************************************************EOM02310
      NUMRUN=1                                                              EOM02320
      IRFLAG=0                                                              EOM02330
      READ (IOIN,1000) IALB1,IALB2,(DAT(L),L=1,10)                          EOM02340
      IF (IALB1.NE.IAL(1)) GO TO 10                                         EOM02350
      NUMRUN=IFIX(DAT(1))                                                   EOM02360
      IF (NUMRUN.EQ.0) NUMRUN=1                                             EOM02370
      GO TO 20                                                              EOM02380
C     SET FLAG IF EORUN IS NOT THE FIRST CARD                               EOM02390
   10 IRFLAG=1                                                              EOM02400
   20 CONTINUE                                                              EOM02410
      DO 580 JRUN=1,NUMRUN                                                  EOM02420
C     INITIALIZATION                                                        EOM02430
        DO 30 I=1,20                                                        EOM02440
C     PROGRAM SELECTOR                                                      EOM02450
        IPROGN(I)=0                                                         EOM02460
C     PROGRAM CYCLE DEFAULT                                                 EOM02470
   30   IDOPGM(I)=1                                                         EOM02480
C     TRANSMISSIONS                                                         EOM02490
      LOTRNS=1.                                                             EOM02500
      XSTRN=1.                                                              EOM02510
      SMKTRN=1.                                                             EOM02520
      DRTRN=1.                                                              EOM02530
      LZTRN=1.                                                              EOM02540
      MMWTRN=1.                                                             EOM02550
      CLTRN=1.0                                                             EOM02560
      GRNTRN=1.0                                                            EOM02570
      IF (JRUN.GT.1) WRITE (IOOUT,1070) JRUN                                EOM02580
C     FREQUENCY, WAVELENGTH, WAVENUMBER INDICATOR                           EOM02590
      IFW=0                                                                 EOM02600
C     GEOMETRICAL OPTION SWITCH                                             EOM02610
      IGEOSW=0                                                              EOM02620
C     SENSOR RESPONSE FUNCTION OPTION SWITCH                                EOM02630
      NR=0                                                                  EOM02640
      VIS=0.0                                                               EOM02650
      EXTN55=0.0                                                            EOM02660
      EXTN=0.0                                                              EOM02670
        DO 220 J=1,25                                                       EOM02680
```

```
      IF (IRFLAG.EQ.1) WRITE (IOOUT,1010)                                   EOM02690
C     SUPRESS READ IN CASE FIRST CARD PREVIOUSLY READ WASNT EORUN.          EOM02700
      IF (IRFLAG.EQ.0) READ (IOIN,1000) IALB1,IALB2,(DAT(L),L=1,10)         EOM02710
      IRFLAG=0                                                              EOM02720
      INOPT=0                                                               EOM02730
      IF (J.EQ.25) GO TO 230                                                EOM02740
        DO 40 KK=1,12                                                       EOM02750
C     CHECK FOR CARD TYPES, NOT PROGRAM SELECTOR                            EOM02760
        IF (IALB1.NE.IAL(KK)) GO TO 40                                      EOM02770
        INOPT=KK                                                            EOM02780
        IF (INOPT.GE.3.AND.INOPT.LE.5) IFW=INOPT                            EOM02790
C     GO CARD FOUND                                                         EOM02800
        IF (INOPT.EQ.12) GO TO 250                                          EOM02810
        GO TO 80                                                            EOM02820
 40     CONTINUE                                                            EOM02830
C     SEARCH FOR PROGRAMS HERE                                              EOM02840
        DO 50 KK=1,20                                                       EOM02850
        IF (IALB1.NE.IPNAM(KK)) GO TO 50                                    EOM02860
        IPROGN(KK)=KK                                                       EOM02870
        IF (DAT(1).GT.1.0) IDOPGM(KK)=IFIX(DAT(1))                          EOM02880
        IF (KK.EQ.20) GO TO 60                                              EOM02890
        GO TO 220                                                           EOM02900
 50     CONTINUE                                                            EOM02910
      GO TO 240                                                             EOM 2920
C     CLIMATOLOGICAL OPTION INVOKED                                         EOM02930
 60     ICLMAT=IFIX(DAT(1))                                                 EOM02940
        IF (ICLMAT.EQ.2) GO TO 70                                           EOM02950
        LOCAT=IFIX(DAT(2))                                                  EOM02960
        MONTH=IFIX(DAT(3))                                                  EOM02970
        NHOUR=IFIX(DAT(4))                                                  EOM02980
        IWIND=IFIX(DAT(5))                                                  EOM02990
        NPRT=IFIX(DAT(6))                                                   EOM03000
        GO TO 220                                                           EOM03010
 70     IPASCT=IFIX(DAT(2))                                                 EOM03020
        TEMP=DAT(3)                                                         EOM03030
        PRESS=DAT(4)                                                        EOM03040
        RH=DAT(5)                                                           EOM03050
        AH=DAT(6)                                                           EOM03060
        DP=DAT(7)                                                           EOM03070
        VIS=DAT(8)                                                          EOM03080
        WNDVEL=DAT(9)                                                       EOM03090
        WNDDIR=DAT(10)                                                      EOM03100
        GO TO 220                                                           EOM03110
C     GEOMETRICAL OPTION INVOKED                                            EOM03120
 80     IF (INOPT.GT.6) GO TO 90                                            EOM03130
C     CARD SETUP SWITCHING                                                  EOM03140
        GO TO (210,160,170,180,190,200),INOPT                               EOM03150
 90     LPTSSW=INOPT-6                                                      EOM03160
        IGEOSW=1                                                            EOM03170
          DO 150 K=1,3                                                      EOM03180
          GO TO (100,110,120,130,140),LPTSSW                                EOM03190
100       PTS(K)=DAT(K)                                                     EOM03200
          GO TO 150                                                         EOM03210
110       PTS(K+3)=DAT(K)                                                   EOM03220
          GO TO 150                                                         EOM03230
120       PTS(K+6)=DAT(K)                                                   EOM03240
          GO TO 150                                                         EOM03250
130       PTS(K+9)=DAT(K)                                                   EOM03260
          GO TO 150                                                         EOM03270
140       PTS(K+12)=DAT(K)                                                  EOM03280
150       CONTINUE                                                          EOM03290
      GO TO 220                                                             EOM03300
C     VISIBILITY CARD                                                       EOM03310
160     VIS=DAT(1)                                                          EOM03320
        EXTN55=DAT(2)                                                       EOM03330
        EXTN=DAT(3)                                                         EOM03340
        GO TO 220                                                           EOM03350
C     FREQUENCY CARD                                                        EOM03360
170     FREQ1=DAT(1)                                                        EOM03370
        FREQ2=DAT(2)                                                        EOM03380
```

```
        MULDV=IFIX(DAT(3))                                                  EOM03390
        GO TO 220                                                           EOM03400
C     WAVELENGTH CARD                                                       EOM03410
  180   WAVE1=DAT(1)                                                        EOM03420
        WAVE2=DAT(2)                                                        EOM03430
        MULDV=IFIX(DAT(3))                                                  EOM03440
        GO TO 220                                                           EOM03450
C     WAVENUMBER CARD                                                       EOM03460
  190   WVNUM1=DAT(1)                                                       EOM03470
        WVNUM2=DAT(2)                                                       EOM03480
        MULDV=IFIX(DAT(3))                                                  EOM03490
        GO TO 220                                                           EOM03500
C     SENSOR OPTION INVOKED                                                 EOM03510
  200   NR=1                                                                EOM03520
        GO TO 220                                                           EOM03530
  210   WRITE (IOOUT,1010)                                                  EOM03540
  220   CONTINUE                                                            EOM03550
      IF (IFW.NE.0) GO TO 250                                               EOM03560
C     ERROR CHECK ON WAVENUMBER, WAVELENGTH, OR FREQUENCY                   EOM03570
      WRITE (IOOUT,1020)                                                    EOM03580
      GO TO 580                                                             EOM03590
  230 WRITE (IOOUT,1030)                                                    EOM03600
      GO TO 250                                                             EOM03610
C     UNKNOWN CARD TYPE                                                     EOM03620
  240 WRITE (IOOUT,1040) IALB1,IALB2                                        EOM03630
  250 CONTINUE                                                              EOM03640
C     SELECT FREQUENCY, WAVELENGTH, OR WAVENUMBER                           EOM03650
      IF (IFW-4) 260,270,280                                                EOM03660
  260 WVNUM1=FREQ1/30.                                                      EOM03670
      WVNUM2=FREQ2/30.                                                      EOM03680
      WAVE1=0.                                                              EOM03690
      IF (FREQ2.GT..0001) WAVE1=3.E+05/FREQ2                                EOM03700
      WAVE2=3.E+05/FREQ1                                                    EOM03710
      GO TO 290                                                             EOM03720
  270 WVNUM1=0.                                                             EOM03730
      IF (WAVE2.GT..0001) WVNUM1=1.E+04/WAVE2                               EOM03740
      WVNUM2=1.E+04/WAVE1                                                   EOM03750
      FREQ1=30.*WVNUM1                                                      EOM03760
      FREQ2=30.*WVNUM2                                                      EOM03770
      GO TO 290                                                             EOM03780
  280 FREQ1=30.*WVNUM1                                                      EOM03790
      FREQ2=30.*WVNUM2                                                      EOM03800
      WAVE1=0.                                                              EOM03810
      IF (WVNUM2.GT..0001) WAVE1=1.E+04/WVNUM2                              EOM03820
      WAVE2=1.E+04/WVNUM1                                                   EOM03830
  290 CONTINUE                                                              EOM03840
      IF (VIS.LT..0001.AND.EXTN55.LT.0001) WRITE (IOOUT,1050)               EOM03850
      IF (VIS.LT..0001.AND.EXTN55.LT.0001) VIS=10.                          EOM03860
      IF (EXTN55.GT..0001) VIS=3.912/EXTN55                                 EOM03870
      IF (VIS.GT..0001) EXTN55=3.912/VIS                                    EOM03880
C     OUTPUT INFORMATION                                                    EOM03890
      WRITE (IOOUT,1080)                                                    EOM03900
        DO 300 I=1,20                                                       EOM03910
  300   IF (IPROGN(I).EQ.I) WRITE (IOOUT,1090) IPNAM(I),IPNAM(I+20)         EOM03920
      WRITE (IOOUT,1100) WVNUM1,WVNUM2,WAVE1,WAVE2,FREQ1,FREQ2              EOM03930
C     CLIMAT USES UNIT NCLIMT                                               EOM03940
      IF (ICLMAT.EQ.1) CALL CLIMAT(LOCAT,MONTH,NHOUR,IWIND,NPRT,TEMP,       EOM03950
     1     PRESS,RH,AH,DP,VIS,WNDVEL,WNDDIR,IPASCT)                         EOM03960
      WRITE (IOOUT,1110) VIS                                                EOM03970
      IF (ICLMAT.EQ.2) ICLMAT=1                                             EOM03980
C***************** SPOT CONTRAST PGM ***********************************EOM03990
      IF (IPROGN(1).NE.1) GO TO 320                                         EOM04000
      IPGM1=IDOPGM(1)                                                       EOM04010
      WRITE (IOOUT,1130)                                                    EOM04020
        DO 310 I=1,IPGM1                                                    EOM04030
C     SPOT USES UNITS: IPHFUN - PHASE FUNCTION; LOUNIT - LT4M DATA          EOM04040
  310   CALL SPOT(WVNUM1,WVNUM2,VIS,NR,IERR,MULDV)                          EOM04050
      CALL RESET(IERR)                                                      EOM04060
C***************** TURBULENCE PGM ***************************************EOM04070
  320 IF (IPROGN(2).NE.2) GO TO 340                                         EOM04080
```

```
      IPGM2=IDOPGM(2)                                                    EOM04090
      WRITE (IOOUT,1140)                                                 EOM04100
        DO 330 I=1,IPGM2                                                 EOM04110
  330   CALL TURB(WAVE1,IERR)                                            EOM04120
      CALL RESET(IERR)                                                   EOM04130
C******************** LASER MULTIPLE SCATTERING PGM **********************EOM04140
  340 IF (IPROGN(3).NE.3) GO TO 360                                      EOM04150
      IPGM3=IDOPGM(3)                                                    EOM04160
      ISPOT=.FALSE.                                                      EOM04170
      WRITE (IOOUT,1150)                                                 EOM04180
        DO 350 I=1,IPGM3                                                 EOM04190
C     BASCAT USES UNIT IPHFUN FOR PHASE FUNCTION DATA                    EOM04200
  350   CALL BASCAT(WAVE1,EXTN,IERR)                                     EOM04210
      CALL RESET(IERR)                                                   EOM04220
C********************* LT4M PGM *****************************************EOM04230
  360 IF (IPROGN(4).NE.4) GO TO 380                                      EOM04240
C     LT4M READS ATM DATA FROM LOUNIT                                    EOM04250
      ISPOT=.FALSE.                                                      EOM04260
      LOREAD=.TRUE.                                                      EOM04270
      IPGM4=IDOPGM(4)                                                    EOM04280
        DO 370 I=1,IPGM4                                                 EOM04290
        CALL LT4M(H1,H2,ANGLE,ITYPE,IXY,TRAN,RADA,RADG,IEMISS,LEN,MODEL,EOM04300
     1        VIS,WVNUM1,WVNUM2,T1,ICLMAT,IERR,NR,IHAZE,MULDV)           EOM04310
  370   LOTRNS=LOTRNS*TRAN(1)                                            EOM04320
      CALL RESET(IERR)                                                   EOM04330
C******************** XSCALE EXTINCTION PGM *****************************EOM04340
  380 IF (IPROGN(5).NE.5) GO TO 400                                      EOM04350
      WRITE (IOOUT,1160)                                                 EOM04360
      IPGM5=IDOPGM(5)                                                    EOM04370
        DO 390 I=1,IPGM5                                                 EOM04380
        CALL XSCALE(WAVE1,VIS,EXTN55,XTRN,IERR,0,0,0.,0.)                EOM04390
  390   XSTRN=XSTRN*XTRN                                                 EOM04400
      CALL RESET(IERR)                                                   EOM04410
C******************** SMOKE PGM *****************************************EOM04420
  400 IF (IPROGN(6).NE.6) GO TO 420                                      EOM04430
      WRITE (IOOUT,1170)                                                 EOM04440
      IPGM6=IDOPGM(6)                                                    EOM04450
        DO 410 N=1,IPGM6                                                 EOM04460
        CALL SMOKE(WAVE1,ICLMAT,STRANS,IERR)                             EOM04470
  410   SMKTRN=SMKTRN*STRANS                                             EOM04480
      CALL RESET(IERR)                                                   EOM04490
C******************** DRTRAN  PGM ***************************************EOM04500
  420 IF (IPROGN(7).NE.7) GO TO 440                                      EOM04510
      WRITE (IOOUT,1180)                                                 EOM04520
      IPGM7=IDOPGM(7)                                                    EOM04530
      HOLDWV=WAVE1                                                       EOM04540
      IF (IFW .EQ.1) WAVE1=WAVE2                                         EOM04550
        DO 430 N=1,IPGM7                                                 EOM04560
C      DRTRAN USES NDIRTU FOR DATA                                       EOM04570
        IF (N.GT.1) WRITE (IOOUT,1120)                                   EOM04580
        CALL DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)                            EOM04590
  430   DRTRN=DRTRN*TRNLOS                                               EOM04600
      WAVE1=HOLDWV                                                       EOM04610
      CALL RESET(IERR)                                                   EOM04620
C******************** LASER TRANSMISSION PGM ****************************EOM04630
  440 IF (IPROGN(8).NE.8) GO TO 460                                      EOM04640
      WRITE (IOOUT,1190)                                                 EOM04650
      IPGM8=IDOPGM(8)                                                    EOM04660
        DO 450 I=1,IPGM8                                                 EOM04670
        CALL LZTRAN(WAVE1,ICLMAT,LAZTRN,IERR)                            EOM04680
  450   LZTRN=LZTRN*LAZTRN                                               EOM04690
      CALL RESET(IERR)                                                   EOM04700
C******************** NEAR MILLIMETER WAVE PGM **************************EOM04710
  460 IF (IPROGN(9).NE.9) GO TO 480                                      EOM04720
      WRITE (IOOUT,1200)                                                 EOM04730
      IPGM9=IDOPGM(9)                                                    EOM04740
        DO 470 I=1,IPGM9                                                 EOM04750
        CALL NMMW(FREQ1,ICLMAT,MMTRAN,IERR)                              EOM04760
  470   MMWTRN=MMWTRN*MMTRAN                                             EOM04770
      CALL RESET(IERR)                                                   EOM04780
```

```
C***************** CLOUD TRANSMISSION PGM *********************************EOM04790
  480 IF (IPROGN(10).NE.10) GO TO 500                                        EOM04800
      IPGM10=IDOPGM(10)                                                      EOM04810
        DO 490 I=1,IPGM10                                                    EOM04820
        WRITE (IOOUT,1210)                                                   EOM04830
        CALL CLTRAN(CTRANS,WAVE1,I,IERR)                                     EOM04840
  490   CLTRN=CLTRN*CTRANS                                                   EOM04850
      CALL RESET(IERR)                                                       EOM04860
C**** CWIC MUNITION EXPENDITURES/INVERSE STATIC TARGET DETECTION PGM ***EOM04870
  500 IF (IPROGN(11).NE.11) GO TO 520                                        EOM04880
      WRITE (IOOUT,1220)                                                     EOM04890
      IPGM11=IDOPGM(11)                                                      EOM04900
        DO 510 I=1,IPGM11                                                    EOM04910
  510   CALL SCREEN(IERR,ICLMAT)                                             EOM04920
      CALL RESET(IERR)                                                       EOM04930
C***************** FINITE CLOUD RADIATIVE TRANSFER PGM *********************EOM04940
  520 IF (IPROGN(12).NE.12) GO TO 540                                        EOM04950
      WRITE (IOOUT,1230)                                                     EOM04960
      IPGM12=IDOPGM(12)                                                      EOM04970
C     FCLOUD USES IPHFUN FOR PHASE FUNCTION DATA                             EOM04980
        DO 530 I=1,IPGM12                                                    EOM04990
  530   CALL FCLOUD(WAVE1,FTRANS,IERR)                                       EOM05000
      CALL RESET(IERR)                                                       EOM05010
C***************** OVERCAST SKY RADIATIVE TRANSFER PGM ********************* EOM05020
  540 IF (IPROGN(13).NE.13) GO TO 560                                        EOM05030
      WRITE (IOOUT,1240)                                                     EOM05040
      IPGM13=IDOPGM(13)                                                      EOM05050
        DO 550 I=1,IPGM13                                                    EOM05060
  550   CALL OVRCST(WAVE1,OTRANS,IERR)                                       EOM05070
      CALL RESET(IERR)                                                       EOM05080
C**************** SELF-SCREENING SMOKE GRENADE PGM ************************ EOM05090
  560 IF (IPROGN(14).NE.14) GO TO 575                                        EOM05100
      WRITE (IOOUT,1250)                                                     EOM05110
      IPGM14=IDOPGM(14)                                                      EOM05120
        DO 570 I=1,IPGM14                                                    EOM05130
        CALL GRNADE(WAVE1,ICLMAT,GRTRAN,IERR)                                EOM05140
  570   GRNTRN=GRNTRN*GRTRAN                                                 EOM05150
      CALL RESET(IERR)                                                       EOM05160
C*********************************************************************  EOM05170
  575 IF(IPROGN(4).GT.0.OR.IPROGN(5).GT.0.OR.IPROGN(6).GT.0.OR.IPROGN(7)
     + .GT.0.OR.IPROGN(8).GT.0.OR.IPROGN(9).GT.0.OR.IPROGN(10).GT.0.OR.
     + IPROGN(14).GT.0) GO TO 576
      GO TO 580
  576 CALL COMPLT(LOTRNS,XSTRN,SMKTRN,DRTRN,LZTRN,MMWTRN,GRNTRN,CLTRN)  EOM05180
  580 CONTINUE                                                               EOM05190
      WRITE (IOOUT,1260)                                                     EOM05200
      STOP                                                                   EOM05210
C                                                                            EOM05220
C                                                                            EOM05230
 1000 FORMAT(2A4,1X,10E7.4)                                                  EOM05240
 1010 FORMAT(1H0,20X,75H***EOMAIN WARNING*** EORUN CYCLE CARD OUT OF SEQEOM05250
     1UENCE, DEFAULT TO ONE CYCLE //)                                        EOM05260
 1020 FORMAT(1H0,20X,74H***EOMAIN ERROR*** FREQ, WAVL, OR WVNUM CARD WASEOM05270
     1 NOT INPUT, RUN TERMINATED //)                                         EOM05280
 1030 FORMAT(1H0,20X,46H***EOMAIN ERROR*** END OF READ SENTINEL ABSENT /EOM05290
     1/,1X,20X,28HRESULTS MAY BE UNPREDICTABLE)                              EOM05300
 1040 FORMAT(1H0,20X,80H***EOMAIN ERROR*** INPUT CARD DETECTED WHICH DOEEOM05310
     1S NOT MATCH CORRECT INPUT FORMAT//,1X,20X,13HTHE CARD WAS:,2X,2A4)EOM05320
 1050 FORMAT (1H0,20X,24H**** EOSAEL WARNING ****,/,1X,20X,15HVISIBILITYEOM05330
     1 AND , 47HEXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM/)        EOM05340
 1060 FORMAT (1H1,///////,1X,50X,30(1H*),/,1X,50X,1H*,28X,1H*,/          EOM05350
     1          ,1X,50X,30H*   ELECTRO-OPTICAL SYSTEMS   *,/,1X,50X,1H*,  EOM05360
     2         28X,1H*,/,1X,50X,30H*     ATMOSPHERIC EFFECTS     *,/     EOM05370
     3         ,1X,50X,1H*,28X,1H*,/,1X,50X,25H*          LIBRARY       EOM05380
     4         5H    *,/,1X,50X,1H*,28X,1H*,/,1X,50X,30(1H*))            EOM05390
 1070 FORMAT (1H1,/////,58X,11HRUN NUMBER ,I2)                            EOM05400
 1080 FORMAT (///1X,51X,28HINDIVIDUAL MODULES  SELECTED)                  EOM05410
 1090 FORMAT (1X,62X,2A4)                                                 EOM05420
 1100 FORMAT (1H0,63X,9HBEGINNING,12X,6HENDING,//,39X,14HWAVENUMBER(CM*   EOM05430
     1         4H*-1),6X,F10.3,10X,F10.3,//,39X,19HWAVELENGTH(MICRONS),   EOM05440
```

```
     2        5X,F10.3,10X,F10.3,//,39X,14HFREQUENCY(GHZ),5X,F15.3,5X,   EOM05450
     3        F15.3,//)                                                  EOM05460
 1110 FORMAT (1H0,62X,10HVISIBILITY,/,62X,F5.2,3H KM)                   EOM05470
 1120 FORMAT(1H1)                                                       EOM05480
 1130 FORMAT(1H1,40X,20HSPOT CONTRAST MODULE //)                        EOM05490
 1140 FORMAT(1H1,40X,17HTURB LASER MODULE //)                           EOM05500
 1150 FORMAT(1H1,40X,30HBASCAT LASER SCATTERING MODULE //)              EOM05510
 1160 FORMAT (1H1,40X,46HXSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE EOM05520
     1     ///)                                                         EOM05530
 1170 FORMAT (1H1,45X,19HSMOKE MODEL MODULE ////)                       EOM05540
 1180 FORMAT (1H1,40X,26H DIRT TRANSMISSION MODULE ///)                 EOM05550
 1190 FORMAT (1H1,40X,28H LASER TRANSMITTANCE MODULE //)                EOM05560
 1200 FORMAT (1H1,45X,29H NEAR MILLIMETER WAVE MODULE ///)              EOM05570
 1210 FORMAT(1H1,40X,27HCLOUD TRANSMITTANCE MODULE ///)                 EOM05580
 1220 FORMAT(1H1,20X,43HCWIC MUNITION EXPENDITURES / INVERSE STATIC     EOM05590
     1 24H TARGET DETECTION MODULE )                                    EOM05600
 1230 FORMAT(1H1,40X,38HFINITE CLOUD RADIATIVE TRANSFER MODULE ///)     EOM05610
 1240 FORMAT(1H1,40X,38HOVERCAST SKY RADIATIVE TRANSFER MODULE ///)     EOM05620
 1250 FORMAT(1H1,40X,35HSELF-SCREENING SMOKE GRENADE MODULE ///)        EOM05630
 1260 FORMAT (1X,/////,1X,50X,14HEND EOSAEL RUN)                        EOM05640
      END                                                               EOM05650
```

```
      SUBROUTINE RESET (IERR)                                           RES00010
C     THE PURPOSE OF THIS ROUTINE IS: (1) TO RESET THE SEQUENCING OF    RES00020
C     DATA CARDS DUE TO AN ERROR IN A PREVIOUS MODULE OR (2) TO READ    RES00030
C     A SENTENAL CARD THAT DELINEATES THE END OF A DATA SET (SEE        RES00040
C     BELOW FOR DEFINITION OF A DATA SET) OR (3) TO STOP THE PROGRAM -   RES00050
C     THIS LAST MODE IS USALLY FOR DEBUGGING OR TO ONLY CHANGE A CARD   RES00060
C     IN A COMPLETE RUN.                                                RES00070
C                                                                       RES00080
C     TO DELINEATE THE END OF A DATA SET A CARD THAT HAS JUST END       RES00090
C     ON IT MUST BE INSERTED AS A SENTINEL CARD: A DATA SET IS          RES00100
C     DEFINED AS THAT COMPLETE SET OF CARDS NECESSARY TO RUN THE        RES00110
C     CALLED MODULE THE NUMBER OF TIMES AS SPECIFIED ON THE IDOPGM(I)   RES00120
C     CARD. STOP MAY ALSO BE INSERTED AS A SENTINAL CARD, IN WHICH      RES00130
C     CASE THE PROGRAM WILL BE TERMINATED AT THAT POINT - THIS IS       RES00140
C     NOT THE NORMAL TERMINATION.                                       RES00150
C                                                                       RES00160
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTURES00170
      DATA ICHCK1,ICHCK2,ISTOP1,ISTOP2 /2HEN,2HD ,2HST,2HOP/            RES00180
      IF(IERR.EQ.1) GO TO 1                                             RES00190
5     CONTINUE                                                          RES00200
C     FOR UNIVAC AND IBM                                                RES00210
C     READ (IOIN,100,END=2) ISNTL1,ISNTL2                               RES00220
      READ (IOIN,100) ISNTL1,ISNTL2                                     RES00230
C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE            RES00240
C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.            RES00250
C3    READ(IOIN,100) ISNTL1,ISNTL2                                      RES00260
C4    IF(EOF(IOIN)) 2,10                                                RES00270
 10   IF((ISNTL1.NE.ICHCK1.AND.ISNTL2.NE.ICHCK2).AND.                   RES00280
     1     (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 5             RES00290
      IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP                   RES00300
      RETURN                                                            RES00310
1     WRITE(IOOUT,102)                                                  RES00320
C     FOR UNIVAC AND IBM                                                RES00330
C6     READ (IOIN,100,END=2) ISNTL1,ISNTL2                              RES00340
6     READ (IOIN,100) ISNTL1,ISNTL2                                     RES00350
C TO EXECUTE THIS ROUTINE ON A 'CDC' MACHINE COMMENT OUT THE            RES00360
C PRECEDING LINE AND UNCOMMENT THE NEXT TWO LINES C3 AND C4.            RES00370
C3    READ(IOIN,100) ISNTL1,ISNTL2                                      RES00380
C4    IF(EOF(IOIN)) 2,20                                                RES00390
20    CONTINUE                                                          RES00400
      IF ((ISNTL1.NE.ICHCK1.AND.ISENTL.NE.ICHCK2).AND.                  RES00410
     1  (ISNTL1.NE.ISTOP1.AND.ISNTL2.NE.ISTOP2)) GO TO 6                RES00420
      IF (ISNTL1.EQ.ISTOP1.AND.ISNTL2.EQ.ISTOP2) STOP                   RES00430
      IERR=0                                                            RES00440
      RETURN                                                            RES00450
2     WRITE (IOOUT,101) IOIN                                            RES00460
100   FORMAT (2A2)                                                      RES00470
101   FORMAT (1X,120(1H*),/,1X,29H ERROR IN INPUT CONTROL FILE ,I4,     RES00480
     + 21H - PROGRAM TERMINATED,/,1X,120(1H*))                          RES00490
102   FORMAT(1H0,50H**** CARD SEQUENCE RESET DUE TO ERROR IN PREVIOUS ,  RES00500
     1      15HMODULE (IERR=1)//)                                      RES00510
      STOP                                                              RES00520
      END                                                               RES00530
```

```
      SUBROUTINE COMPLT(LOTRNS,XSTRN,                                   COM00010
     +                  SMKTRN,DRTRN,LZTRN,MMWTRN,GRNTRN,CLTRN)         COM00020
      REAL LOTRNS,LZTRN,MMWTRN                                          COM00030
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCOM00040
C     LOWTRAN - LOTRNS                                                  COM00050
C      XSCALE - XSTRN                                                   COM00060
C     SMOKE - SMKTRN                                                    COM00070
C     DRTRAN - DRTRN                                                    COM00080
C     LZTRAN - LZTRN                                                    COM00090
C     NMMW - MMWTRN                                                     COM00100
C     CLTRAN - CLTRN                                                    COM00110
C     GRNADE - GRNTRN                                                   COM00120
      TRAN=LOTRNS*XSTRN*SMKTRN*DRTRN*CLTRN*GRNTRN*MMWTRN*LZTRN          COM00130
      WRITE (IOOUT,100) TRAN                                            COM00140
100   FORMAT (///1X,20X,24HCOMBINED TRANSMISSION FO,15HR THE SELECTED , COM00150
     1  10HMODULES = ,E10.4)                                            COM00160
      RETURN                                                            COM00170
      END                                                               COM00180
```

```
      SUBROUTINE ILLUM(LAMBDA,LD,E0)                                          ILL00010
      REAL LAMBDA,LUNPHA                                                      ILL00020
      COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU      ILL00030
C*************************************************************************ILL00040
C                                                                           ILL00050
C  SUBROUTINE ILLUM RETURNS THE EXTRATERRESTRIAL IRRADIANCE E0              ILL00060
C  AT WAVELENGTH LAMBDA.  IF LD = 0, THE VALUE GIVEN IS SOLAR               ILL00070
C  IRRADIANCE.  IF 1 < LD < 28 THE VALUE GIVEN IS LUNAR IRRAD-              ILL00080
C  IANCE ON LUNAR DAY LD, WITH DAY 28 CORRESPONDING TO FULL                 ILL00090
C  MOON AND DAY 14 BEING NEW MOON.                                          ILL00100
C                                                                           ILL00110
C  SUBROUTINE COMPUTES VALUE OF LUNAR PHASE ANGLE, IF REQUIRED,             ILL00120
C  AND CALLS ONE OF THE EOSAEL ROUTINES SOLARS OR SMOON.                    ILL00130
C                                                                           ILL00140
C*************************************************************************ILL00150
      IF(ILD.GT.0) GO TO 10                                                   ILL00160
      E0=SOLARS(LAMBDA)                                                       ILL00170
      GO TO 100                                                               ILL00180
C                                                                             ILL00190
C  LD GT 0 => E0 = LUNAR VALUE                                              ILL00200
C                                                                           ILL00210
   10 ILD=LD                                                                  ILL00220
      IF(ILD.GT.14) ILD=28-ILD                                                ILL00230
      LUNPHA=180.0*FLOAT(ILD)/14.0                                            ILL00240
      E0=SMOON(LAMBDA,LUNPHA)                                                 ILL00250
  100 WRITE(IOOUT,1000) E0                                                    ILL00260
      RETURN                                                                  ILL00270
 1000 FORMAT(32H0  EXTRATERRESTRIAL IRRADIANCE= ,1PE10.4,11H W/M2-SR-MU)      ILL00280
      END                                                                     ILL00290
```

```
      FUNCTION SOLARS(WAVL)                                                  SOL00010
C                                                                            SOL00020
C     CALCULATE THE INTENSITY OF THE SOLAR SPECTRUM FOR WAVELENGTH (WAVL     SOL00030
C                                                                            SOL00040
C     UNITS:                                                                 SOL00050
C           SOLARS ... WATTS M-2 MICRON-1                                    SOL00060
C           WAVL ..... MICRONS                                               SOL00070
      IF (WAVL.LT.0.15.OR.WAVL.GE.100.) GO TO 100                            SOL00080
      IF (WAVL.GE.0.15.AND.WAVL.LE.0.43) GO TO 200                           SOL00090
      IF (WAVL.GT.0.43.AND.WAVL.LT.0.58) GO TO 300                           SOL00100
      GO TO 400                                                              SOL00110
  100 SOLARS=0.0                                                             SOL00120
      RETURN                                                                 SOL00130
  200 Z=((WAVL-.415)/0.68)**2                                                SOL00140
      SOLARS=1775.*EXP(Z/2)                                                  SOL00150
      RETURN                                                                 SOL00160
  300 CONTINUE                                                               SOL00170
      SOLARS=(-61142.*WAVL**4)+(1344477.*WAVL**3)-(110296.*                  SOL00180
     1        WAVL**2)                                                       SOL00190
      SOLARS=(SOLARS+39952.7*WAVL-5371.)/100.                                SOL00200
      RETURN                                                                 SOL00210
  400 IF (WAVL.GE.2.5) GO TO 500                                             SOL00220
      SOLARS=5331.9*EXP(-1.959*WAVL)                                         SOL00230
      RETURN                                                                 SOL00240
  500 SOLARS=2288.38*(WAVL**(-3.9765))                                       SOL00250
      RETURN                                                                 SOL00260
      END                                                                    SOL00270
```

```
      FUNCTION SMOON(WLAM,ANGLE)                                        SMN00010
C                                                                       SMN00020
CCC   CALCULATE THE INTENSITY OF MOONLIGHT FOR WAVELENGTH (WLAM)        SMN00030
CCC   AND PHASE ANGLE (ANGLE)                                           SMN00040
CCC                                                                     SMN00050
CCC   UNITS:                                                            SMN00060
CCC         SMOON ... WATTS M-2 MICRON-1                                SMN00070
CCC         ANGLE ... DEGREES                                           SMN00080
C           WLAM .... MICRONS                                           SMN00090
      SMOON=0.0                                                         SMN00100
      IF (ANGLE.GT.160.) RETURN                                         SMN00110
      SMOON=(3.426E-9*ANGLE**4-1.63E-6*ANGLE**3+3.01E-4*                SMN00120
     1       ANGLE**2-.0266*ANGLE+.9882)*100.                           SMN00130
      ALBED=0.4                                                         SMN00140
      IF (WLAM.GE.5.) GO TO 200                                         SMN00150
      IF (WLAM.GT.2.8) GO TO 100                                        SMN00160
      IF (WLAM.LE.1.) ALBED=3.9633*WLAM**4-10.7306*WLAM**3+             SMN00170
     1   10.2188*WLAM**2-3.8208*WLAM+.5512                              SMN00180
      IF (WLAM.GT.1.) ALBED=.0482*WLAM**4-.3283*WLAM**3+                SMN00190
     1    .7584*WLAM**2-.5745*WLAM+.2808                                SMN00200
      GO TO 200                                                         SMN00210
  100 ALBED=.350+(.500-.350)*(WLAM-2.8)/2.2                             SMN00220
  200 SMOON=2.04472E-07*SOLARS(WLAM)*ALBED*SMOON                        SMN00230
      RETURN                                                            SMN00240
      END                                                               SMN00250
```

```
      FUNCTION JPASCT(ICAT)                                              JPAC0010
C     THIS FUNCTION CONVERTS THE INTEGER CODE FOR PASQUILL CATEGORY      JPA00020
C     TO THE ALPHA CHARACTER                                             JPA00030
      DIMENSION NPASCT(6)                                                JPA00040
      DATA NPASCT/1HA,1HB,1HC,1HD,1HE,1HF/                               JPA00050
      JPASCT=NPASCT(ICAT)                                                JPA00060
          RETURN                                                         JPA00070
      END                                                                JPA00080
```

```
      SUBROUTINE PFUNC(IDN)                                                 PFU00010
      LOGICAL ISPOT,LOREAD,N16                                              PFU00020
      DIMENSION PFSPOT(16),PFH(65)                                          PFU00030
      COMMON /SPOTLO/ISPOT,LOREAD,N16                                       PFU00040
      COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU     PFU00050
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                         PFU00060
      COMMON /CGEOM/COSGM,COSBT,COSIN                                       PFU00070
      COMMON /BASPOT/ANG(65),SUM(65),WVL(16),NWVL,ALB(16),BS(16),           PFU00080
     +BE(16),SINGWV,PF(65),LMAX                                             PFU00090
C                                                                           PFU00100
C     THIS SUBROUTINE PERFORMS PHASE FUNCTION READING AND INTERPOL-         PFU00110
C     ATION OPERATIONS FOR THE SPOT AND BASCAT MODULES. THE FILE IN         PFU00120
C     WHICH THE PHASE FUNCTION DATA RESIDES IS PFNDAT. USERS MAY IN-        PFU00130
C     SERT PHASE FUNCTIONS OF THEIR OWN SPECIFICATION INTO PFNDAT           PFU00140
C     UNDER AN ID NUMBER 0. THIS PHASE FUNCTION MAY HAVE ANY AR-            PFU00150
C     BITRARY NORMALIZATION, SINCE PFUNC WILL RENORMALIZE IT TO             PFU00160
C     CONFORM TO THE NORMALIZATIONS USED IN SPOT AND BASCAT.                PFU00170
C     FOR FURTHER DETAILS ON THIS PROCEDURE,                                PFU00180
C     THE USER IS REFERRED TO CHAPTER 16 OF THE EOSAEL 80 TECHNICAL         PFU00190
C     DOCUMENTATION MANUAL, WHERE THE STRUCTURE, USE, AND MODIFICATION      PFU00200
C     OF THE PFNDAT FILE IS DISCUSSED.                                      PFU00210
C                                                                           PFU00220
C***  NOTE *** FOR USER-DEFINED PHASE FUNCTIONS (IDN=0), THIS ROUTINE       PFU00230
C              WILL INTERPOLATE OVER WAVELENGTH AND ANGLE FOR THE SPOT       PFU00240
C              MODULE. FOR THE BASCAT MODULE, HOWEVER, NO WAVELENGTH         PFU00250
C              INTERPOLATION IS PERFORMED FOR THE USER-DEFINED PHASE         PFU00260
C              FUNCTION. ONLY RENORMALIZATION IS PERFORMED IN THIS           PFU00270
C              LATTER CASE.                                                  PFU00280
C                                                                           PFU00290
C***  MAXID IS THE NUMBER OF DIFFERENT DISTRIBUTIONS - (PHASE FUNCTIONS)    PFU00300
C                                                                           PFU00310
      MAXID=12                                                              PFU00320
C                                                                           PFU00330
C***      CHECK THE ALLOWABLE RANGE OF DISTRIBUTIONS                        PFU00340
C                                                                           PFU00350
      IF((IDN.GT.MAXID).OR.(IDN.LT.0))GO TO 491                             PFU00360
C                                                                           PFU00370
C***     DATA INITIALIZATIONS                                               PFU00380
C                                                                           PFU00390
      DO 40 I=1,65                                                          PFU00400
   40 PF(I)=0.                                                              PFU00410
      ALBE=0.                                                               PFU00420
      BEX=0.                                                                PFU00430
      BSC=0.                                                                PFU00440
C                                                                           PFU00450
C***            ANGULAR READ BLOCK COMMON TO SPOT AND BASCAT                PFU00460
C                                                                           PFU00470
      I=0                                                                   PFU00480
C                                                                           PFU00490
C**** READ IN ANGLES AT WHICH PHASE FUNCTION IS DEFINED                     PFU00500
C                                                                           PFU00510
   50 I=I+1                                                                 PFU00520
      L1=(I-1)*11+1                                                         PFU00530
      L11=L1+10                                                             PFU00540
      IF(L11.EQ.66)L11=65                                                   PFU00550
      IF(L11.GT.66)GO TO 492                                                PFU00560
      READ(IPHFUN,60)(ANG(L),L=L1,L11)                                      PFU00570
   60 FORMAT(11(F6.2,1X))                                                   PFU00580
C                                                                           PFU00590
C**** CHECK THIS ROW OF DATA FILE FOR TERMINATION SENTINEL                  PFU00600
C                                                                           PFU00610
      DO 100 K=L1,L11                                                       PFU00620
      IF((ANG(K).GE.999.99).AND.(L11.LT.65))GO TO 200                       PFU00630
      IF(K.EQ.65)GO TO 200                                                  PFU00640
  100 CONTINUE                                                              PFU00650
      GO TO 50                                                              PFU00660
C                                                                           PFU00670
C***  LMAX IS THE NBR OF ANGLE AND NBR OF PHASE FUNCTION VALUES+1 HERE.     PFU00680
C                                                                           PFU00690
  200 LMAX=K                                                                PFU00700
```

```
      DO 250 L=1,LMAX                                                   PFU00710
  250 ANG(L)=COS(ANG(L)*PIRAD)                                          PFU00720
C                                                                       PFU00730
C***  REDUCE DUE TO SENTINEL OF 999.99                                  PFU00740
C                                                                       PFU00750
      IF(L11.LT.65)LMAX=K-1                                             PFU00760
      IDNM=IDN-1                                                        PFU00770
      KMAX=IFIX(ALOG(FLOAT(LMAX-1))/ALOG(2.0)+0.1)                      PFU00780
C                                                                       PFU00790
C***  RESET PARAMETERS FOR BASCAT PROCESSING IF APPROPRIATE.            PFU00800
C                                                                       PFU00810
      IF(ISPOT) GO TO 260                                               PFU00820
      NWVL=1                                                            PFU00830
      WVL(1)=SINGWV                                                     PFU00840
  260 CONTINUE                                                          PFU00850
      IF((IDN.EQ.1).OR.(IDN.EQ.0.)) GO TO 1050                          PFU00860
C                                                                       PFU00870
C***  READ PAST AEROSOL DATA NOT OF CURRENT INTEREST                    PFU00880
C                                                                       PFU00890
      DO 1000 I=1,IDNM                                                  PFU00900
      DO 1000 II=1,16                                                   PFU00910
      READ(IPHFUN,300) IANG,ID,WAVE,ALBE,BEX,BSC                        PFU00920
  300 FORMAT(2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))                      PFU00930
      IF(IANG.NE.LMAX) GO TO 493                                        PFU00940
      READ(IPHFUN,400) (PF(L),L=1,LMAX)                                 PFU00950
  400 FORMAT(6(E12.6,1X))                                               PFU00960
 1000 CONTINUE                                                          PFU00970
 1050 CONTINUE                                                          PFU00980
C                                                                       PFU00990
C***  OMIT WAVELENGTH CHECKS FOR USER-DEFINED PHASE FUNCTION.           PFU01000
C                                                                       PFU01010
      IF(IDN.EQ.0) GO TO 1070                                           PFU01020
C                                                                       PFU01030
C***  THE NEXT LOOP PERFORMS THE FOLLOWING OPERATIONS :                 PFU01040
C       (A) IT VERIFIES WHETHER OR NOT ALL INPUT WAVELENGTHS LIE        PFU01050
C           WITHIN THE 0.2-12.0 MICROMETER RANGE (WITH LIMITS           PFU01060
C           EXTENDED TO PLUS OR MINUS 5%).                              PFU01070
C       (B) IF THE PHASE FUNCTION IS NOT USER-SPECIFIED AND THE INPUT   PFU01080
C           WAVELENGTH BAND LIES WITHIN THE 0.2-2.0 BAND, THEN          PFU01090
C           INTERPOLATION IS NOT POSSIBLE DUE TO THE PRESENCE OF        PFU01100
C           ONLY TWO DATA POINTS (AT 0.55 AND 1.06) IN THIS REGION.     PFU01110
C           AS A RESULT, THE 0.55 AND 1.06 VALUES ARE ASSIGNED TO       PFU01120
C           INDIVIDUAL POINTS IN THE INPUT WAVELENGTH BAND. THOSE       PFU01130
C           POINTS WITH WAVELENGTH VALUES LESS THAN OR EQUAL TO 0.8     PFU01140
C           MICROMETERS ARE ASSIGNED THE 0.55 DATA. ALL OTHER WAVE-     PFU01150
C           LENTHS ARE ASSIGNED THE 1.06 DATA.                          PFU01160
C       (C) IF AN INPUT WAVELENGTH LIES OUTSIDE OF THE 3-5 OR 8-12      PFU01170
C           MICROMETER BANDS, BUT IS WITHIN 5% OF AN EXTREMUM FOR       PFU01180
C           THESE BANDS, IT IS RESET TO THE EXTREMUM WAVELENGTH         PFU01190
C           VALUE.                                                      PFU01200
C       (D) IF AN INPUT WAVELENGTH LIES BETWEEN BANDS, EXECUTION IS     PFU01210
C           TERMINATED AND AN ERROR MESSAGE IS PRINTED.                 PFU01220
C                                                                       PFU01230
      DO 1060 I=1,NWVL                                                  PFU01240
      IF((WVL(I).LT.0.19).OR.(WVL(I).GT.12.049)) GO TO 504              PFU01250
      IF(WVL(I).LE.0.8) WVL(I)=0.55                                     PFU01260
      IF((WVL(I).GT.0.8).AND.(WVL(I).LE.2.1)) WVL(I)=1.06               PFU01270
      IF(WVL(I).GT.12.0) WVL(I)=12.0                                    PFU01280
      IF((WVL(I).GT.5.0).AND.(WVL(I).LE.5.25)) WVL(I)=5.0               PFU01290
      IF((WVL(I).GE.7.6).AND.(WVL(I).LT.8.0)) WVL(I)=8.0                PFU01300
      IF((WVL(I).GE.2.85).AND.(WVL(I).LT.3.0)) WVL(I)=3.0               PFU01310
      IF((WVL(I).GT.2.1).AND.(WVL(I).LT.2.85)) GO TO 498                PFU01320
      IF((WVL(I).GT.5.25).AND.(WVL(I).LT.7.6)) GO TO 498                PFU01330
 1060 CONTINUE                                                          PFU01340
 1070 CONTINUE                                                          PFU01350
C                                                                       PFU01360
C***  MAIN INTERPOLATION LOOP                                           PFU01370
C                                                                       PFU01380
      DO 2000 I=1,NWVL                                                  PFU01390
      IF(I.GT.1) GO TO 1260                                             PFU01400
```

```
 1100 CONTINUE                                                          PFU01410
      READ(IPHFUN,300) IANG,ID,WAVEH,ALBH,BEXH,BSCH                     PFU01420
      READ(IPHFUN,400) (PFH(L),L=1,LMAX)                                PFU01430
      IF((IDN.EQ.0).AND.(.NOT.ISPOT)) GO TO 1280                        PFU01440
      IF(WVL(I).LT.WAVEH) GO TO 1100                                    PFU01450
 1150 CONTINUE                                                          PFU01460
      READ(IPHFUN,300) IANG,ID,WAVE,ALBE,BEX,BSC                        PFU01470
      READ(IPHFUN,400) (PF(L),L=1,LMAX)                                 PFU01480
      IF(WVL(I).LE.WAVE) GO TO 1260                                     PFU01490
 1160 CONTINUE                                                          PFU01500
      WAVEH=WAVE                                                        PFU01510
      ALBH=ALBE                                                         PFU01520
      BEXH=BEX                                                          PFU01530
      DO 1240 L=1,LMAX                                                  PFU01540
 1240 PFH(L)=PF(L)                                                      PFU01550
      GO TO 1150                                                        PFU01560
 1260 CONTINUE                                                          PFU01570
C                                                                       PFU01580
C***  GO TO NEXT WAVELENGTH INTERPOLATION INTERVAL IF INPUT WAVE-       PFU01590
C***  LENGTH IS GREATER THAN THE MAXIMUM OF THE CURRENT ONE.            PFU01600
C                                                                       PFU01610
      IF(WVL(I).GT.WAVE) GO TO 1160                                     PFU01620
C                                                                       PFU01630
C***  RENORMALIZE LOWER END OF INTERPOLATION INTERVAL                   PFU01640
C                                                                       PFU01650
 1280 CONTINUE                                                          PFU01660
      SUM(1)=0.0                                                        PFU01670
      DO 1200 L=2,LMAX                                                  PFU01680
 1200 SUM(L)=((ANG(L-1)-ANG(L))*(PFH(L-1)+PFH(L))/4.0)+SUM(L-1)         PFU01690
      SUMT=SUM(LMAX)                                                    PFU01700
      DO 1250 L=1,LMAX                                                  PFU01710
 1250 PFH(L)=PFH(L)/SUMT                                                PFU01720
C                                                                       PFU01730
C***  BRANCH TO FINAL PROCEDURE FOR BASCAT USER-DEFINED PHASE           PFU01740
C***  FUNCTION IF APPROPRIATE.                                          PFU01750
C                                                                       PFU01760
      IF((IDN.EQ.0).AND.(.NOT.ISPOT)) GO TO 2500                        PFU01770
C                                                                       PFU01780
C***  RENORMALIZE UPPER END OF INTERPOLATION INTERVAL.                  PFU01790
C                                                                       PFU01800
      SUM(1)=0.0                                                        PFU01810
      DO 1400 L=2,LMAX                                                  PFU01820
 1400 SUM(L)=((ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0)+SUM(L-1)           PFU01830
      SUMT=SUM(LMAX)                                                    PFU01840
      DO 1450 L=1,LMAX                                                  PFU01850
 1450 PF(L)=PF(L)/SUMT                                                  PFU01860
C                                                                       PFU01870
C***  BRANCH TO BASCAT WAVELENGTH INTERPOLATION PROCEDURE IF            PFU01880
C***  APPROPRIATE.                                                      PFU01890
C                                                                       PFU01900
      IF(.NOT.ISPOT) GO TO 2500                                         PFU01910
C                                                                       PFU01920
C***  PERFORM HALVING SEARCH FOR COSINES IN PHASE FUNCTION DATA FILE    PFU01930
C***  WHICH BRACKET COSINE INPUT FROM SPOT.                             PFU01940
C                                                                       PFU01950
      L=1                                                               PFU01960
      LL=LMAX-1                                                         PFU01970
      DO 1300 K=1,KMAX                                                  PFU01980
      LL=LL/2                                                           PFU01990
      L=L+LL                                                            PFU02000
      AT=COSIN-ANG(L)                                                   PFU02010
      IF(AT.GT.0.) L=L-LL                                               PFU02020
 1300 CONTINUE                                                          PFU02030
C                                                                       PFU02040
C***  PERFORM SPOT EXTINCTION COEFFICIENT AND PHASE FUNCTION            PFU02050
C***  INTERPOLATIONS OVER WAVELENGTH AND ANGLE.                         PFU02060
C                                                                       PFU02070
      FACANG=(COSIN-ANG(L))/(ANG(L+1)-ANG(L))                           PFU02080
      FACWVL=(WVL(I)-WAVEH)/(WAVE-WAVEH)                                PFU02090
      BE(I)=BEXH+(BEX-BEXH)*FACWVL                                      PFU02110
```

```
      PFSPOT(I)=PFH(L)*(1.-FACANG-FACWVL+FACANG*FACWVL)+                 PFU02120
     +PFH(L+1)*(FACANG-FACANG*FACWVL)+PF(L)*(FACWVL-FACANG*FACWVL)+     PFU02130
     +PF(L+1)*(FACANG*FACWVL)                                           PFU02140
 2000 CONTINUE                                                          PFU02150
C                                                                       PFU02160
C***  LOAD FIRST NWVL ANGLES OF OUTPUT ARRAY PF() WITH INTERPOLATED     PFU02170
C***  RESULTS FOR SPOT.                                                 PFU02180
C                                                                       PFU02190
      DO 2200 N=1,NWVL                                                  PFU02200
 2200 PF(N)=PFSPOT(N)/(4.*PI)                                           PFU02210
C                                                                       PFU02220
C***  FINAL EXIT FOR SPOT PROCESSING.                                   PFU02230
C                                                                       PFU02240
      GO TO 500                                                         PFU02250
 2500 CONTINUE                                                          PFU02260
      FACWVL=0.                                                         PFU02270
      IF(IDN.EQ.0) GO TO 2700                                           PFU02280
C                                                                       PFU02290
C***  BASCAT ALBEDO, EXTINCTION COEFFICIENT, AND PHASE FUNCTION         PFU02300
C***  INTERPOLATION OVER WAVELENGTH.                                    PFU02310
C                                                                       PFU02320
      FACWVL=(WVL(1)-WAVEH)/(WAVE-WAVEH)                                PFU02330
 2700 CONTINUE                                                          PFU02340
      DO 2800 L=1,LMAX                                                  PFU02350
 2800 PF(L)=PFH(L)+(PF(L)-PFH(L))*FACWVL                                PFU02360
      ALB(1)=ALBH+(ALBE-ALBH)*FACWVL                                    PFU02370
      BE(1)=BEXH+(BEX-BEXH)*FACWVL                                      PFU02380
C                                                                       PFU02390
C***  FINAL EXIT FOR BASCAT USER-DEFINED PHASE FUNCTION PROCEDURE.      PFU02400
C                                                                       PFU02410
      IF(IDN.EQ.0) GO TO 500                                            PFU02420
C                                                                       PFU02430
C***  FINAL BASCAT PHASE FUNCTION RENORMALIZATION.                      PFU02440
C                                                                       PFU02450
      SUM(1)=0.0                                                        PFU02460
      DO 2900 L=2,LMAX                                                  PFU02470
 2900 SUM(L)=((ANG(L-1)-ANG(L))*(PF(L-1)+PF(L))/4.0)+SUM(L-1)           PFU02480
      SUMT=SUM(LMAX)                                                    PFU02490
      DO 2950 L=1,LMAX                                                  PFU02500
 2950 PF(L)=PF(L)/SUMT                                                  PFU02510
C                                                                       PFU02520
C***  FINAL EXIT FOR BASCAT PROCESSING.                                 PFU02530
C                                                                       PFU02540
      GO TO 500                                                         PFU02550
C                                                                       PFU02560
C***            ERROR EXIT BLOCK COMMON TO SPOT AND BASCAT              PFU02570
C                                                                       PFU02580
  491 CONTINUE                                                          PFU02590
      WRITE(IOOUT,495)                                                  PFU02600
  495 FORMAT(1H0,20X,58H***PFUNC ERROR*** AEROSOL ID NUMBER OUT OF ALLOWPFU02610
     +ABLE RANGE //)                                                    PFU02620
      STOP                                                              PFU02630
  492 CONTINUE                                                          PFU02640
      WRITE(IOOUT,496)                                                  PFU02650
  496 FORMAT(1H0,20X,83H***PFUNC ERROR*** READ TERMINATION SENTINEL NOT PFU02660
     +FOUND OR NUMBER OF ANGLES EXCEED 65 //)                           PFU02670
      STOP                                                              PFU02680
  493 CONTINUE                                                          PFU02690
      WRITE(IOOUT,497)                                                  PFU02700
  497 FORMAT(1H0,20X,93H***PFUNC ERROR*** NUMBER OF SPECIFIED ANGLES ANDPFU02710
     + NUMBER OF PHASE FUNCTION VALUES DO NOT MATCH //)                 PFU02720
      STOP                                                              PFU02730
  498 CONTINUE                                                          PFU02740
      WRITE(IOOUT,499)                                                  PFU02750
  499 FORMAT(1H0,10X, 79H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WVPFU02760
     +L INPUT ARRAY DO NOT LIE WITHIN /1H ,45HWAVELENGTH BANDS COVERED BPFU02770
     +Y PFNDAT DATA BASE /)                                             PFU02780
      STOP                                                              PFU02790
  504 CONTINUE                                                          PFU02800
      WRITE(IOOUT,505)                                                  PFU02810
```

```
  505 FORMAT(1H0,10X,109H***PFUNC ERROR*** SOME OR ALL WAVELENGTHS IN WVPFU02820
     +L ARRAY DO NOT LIE WITHIN OVERALL ACCEPTABLE RANGE OF 0.2-12.0 /) PFU02830
      STOP                                                             PFU02840
  500 RETURN                                                           PFU02850
      END                                                              PFU02860
```

```
      SUBROUTINE XSCALE(WAVE,VIS,EXT55,XSTRN,IERR,ISLT,IFOG,RANGE,ANGLE)XSC00010
C     THE PURPOSE OF THIS ROUTINE IS TO A) FIND THE HORIZONTAL EXTINCTIONXSC00020
C     IN FOG AT THE WAVELENGTHS SPECIFIED BELOW FROM THE EXTINCTION      XSC00030
C     AT .55 UM OR B) THE EXTINCTION ALONG A SLANT PATH AT ALLOWED       XSC00040
C     WAVELENGTHS FROM THE EXTINCTION AT .55 UM: FOG TYPE 1, 2, OR 3,    XSC00050
C     MUST BE SPECIFIED FOR SLANT PATHS.                                 XSC00060
C  *** VISIBILITY = 88. OR 89. IS NOT ALLOWED AS THIS IS USED AS AN      XSC00070
C     INDICATOR THAT XSCALE IS BEING CALLED AS A SUBROUTINE FROM EITHER XSC00080
C     SPOT(88.), LOWTRAN(88.), OR CWIC(89.), NOT EOMAIN!                 XSC00090
C     WAVE=LAMDA IN UM - MUST BE .55, 1.06, 3.0-5.0, 8.0-12.05.          XSC00100
C     ***  ALL EXTN'S ARE IN KM**-1                                      XSC00110
C     EXT55 = EXTINCTION AT .55 UM                                       XSC00120
C     EXT106 = EXTINCTION AT 1.06 UM                                     XSC00130
C     EXT35 = EXTINCTION FROM 3.0 TO 5.0 UM                              XSC00140
C     EXT812 = EXTINCTION FROM 8.0 TO 12.0 UM                            XSC00150
C     VIS= VISIBILITY IN KM -OR- EXT55 IN KM**-1                         XSC00160
C     EXT55 IS ** NOT ** CHANGED BY THIS ROUTINE.                        XSC00170
C*****************************************************************     XSC00180
C     INPUT:  THERE IS A MAXIMUM OF 3 CARDS TO EXECUTE THIS MODULE       XSC00190
C     THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OF       XSC00200
C     THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN.          XSC00210
C     THE CARDS ARE INPUT WITH FORMAT (A4,6X,5(F6.2,1X)).                XSC00220
C     EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4           XSC00230
C     FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 6 COL PER             XSC00240
C     FIELD BEGINNING IN COL 11, WITH A BLANK BETWEEN EACH SUBSEQUENT    XSC00250
C     FIELD.  THE CARDS ARE NOT ORDER DEPENDENT.                         XSC00260
C     IF GEOMET OPTION IS BEING USED, THEN ONLY THE IDENTIFIER HORZ,
C     SLNH, OR SLNS IS TO BE READ IN (NO ADDITIONAL PARAMETERS NEEDED).
C     FOG        FOG TYPE, RAIN RATE (MM/HR): RAIN RATE ONLY NEEDED      XSC00270
C                WHEN FOG TYPE=4.                                        XSC00280
C     HORZ       HORDIS (KM)                ;HORIZONTAL PATH CALCULATION XSC00300
C     SLNH       HORDIS (KM), ANGLE (DEGREES); SLANT PATH CALCULATION    XSC00310
C     SLNS       SLTDIS (KM), ANGLE (DEGREES); SLANT PATH CALCULATION    XSC00320
C     PLOT       WRITE SLANT PATH EXTINCTION, AT INPUT WAVELENGTH, AND
C                ALTITUDE TO NPLOTU (SEE COMMON BLOCK IOUNIT); THE
C                FIRST RECORD WILL BE THE NUMBER OF POINTS TO BE WRITTEN.
C                FORMATS: RECORD 1 (I5), SUBSEQUENT RECORDS (2(E10.4,1X))
C     GO        SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FOR          XSC00330
C               THIS CALL.  NOTE THAT IF A DATA CARD IS NOT READ         XSC00340
C               THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLS          XSC00350
C               TO THE MODULE ARE STILL IN EFFECT.                       XSC00360
C     ALL THE FOLLOWING FOG TYPES ARE RELEVANT TO HORIZONTAL PATHS,
C     BUT ONLY FOG TYPES 1, 2, OR 3 ARE ALLOWED FOR SLANT PATH CALCULATIONS
C     FOG TYPE=1. FOR MARITIME ARTIC                                     XSC00380
C             =2. FOR MARITIME POLAR                                     XSC00390
C             =3. FOR CONTINENTAL POLAR                                  XSC00400
C             =4. FOR RAIN                                               XSC00410
C             =5. FOR SNOW                                               XSC00420
C     HORDIS - HORZONTAL DISTANCE IN KM.                                 XSC00430
C     SLTDIS - SLANT PATH DISTANCE IN KM.                                XSC00440
C     ANG= LOOK ANGLE FROM HORIZONTAL IN DEGREES                         XSC00450
C  N.B.   ONE OF THE FOLLOWING COMBINATIONS MUST BE SUPPLIED             XSC00460
C         FOR SLANT PATH CALCULATIONS.                                    XSC00470
C     HORDIS AND ANG ** OR ** SLTDIS AND ANG                             XSC00480
C*****************************************************************     XSC00490
C  OUTPUT                                                                XSC00500
C    TRANSMISSION AT APPROPRIATE WAVELENGTH FOR SLANT OR HORIZONTAL PATHXSC00510
C                                                                        XSC00520
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     XSC00530
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUXSC00540
      COMMON /GEOMET/PTS(15),IGEOSW                                     XSC00550
      DIMENSION TYPE(6),DAT(6)                                          XSC00560
      LOGICAL NOLO                                                      XSC00570
      DATA TYPE /4HFOG ,4HHORZ,4HSLNH,4HSLNS,4HPLOT,4HGO  /             XSC00580
      DATA A0,A1,A2,NPLT/0.1425,0.1475,-0.0017,0/                       XSC00590
C     USE VIS=88., OR 89. AS AN INDICATOR THAT XSCALE HAS BEEN CALLED   XSC00600
C     AS A SUBROUTINE FROM OTHER PROGRAMS - NOT EOMAIN!                 XSC00610
      IF (VIS.LT.87.9.OR.VIS.GT.89.1) GO TO 8                           XSC00620
      ANG=ANGLE                                                         XSC00630
C     FIND ELEVATION ANGLE FROM ZENITH ANGLE IN SPOT AND LOWTRAN        XSC00640
```

```
      IF (VIS.GT.87.9.AND.VIS.LT.88.1) ANG=90.-ANGLE                    XSC00650
      IF (VIS.GT.87.9.AND.VIS.LT.88.1.AND.ANGLE.GT.90.) ANG=ANGLE-90.   XSC00660
      HORDIS=0.                                                         XSC00670
      SLTDIS=0.                                                         XSC00680
      NOLO=.FALSE.                                                      XSC00690
C     ISLT=0=HORIZONTAL; ISLT GT 0 = SLANT.                             XSC00700
      IF (ISLT.EQ.0) HORDIS=RANGE                                       XSC00710
      IF (ISLT.EQ.0) ISLANT=0                                           XSC00720
      IF (ISLT.GT.0) SLTDIS=RANGE                                       XSC00730
      IF (ISLT.GT.0) ISLANT=1                                           XSC00740
      GO TO 6                                                           XSC00750
8     CONTINUE                                                          XSC00760
      NOLO=.TRUE.                                                       XSC00770
      DO 9 I=1,3                                                        XSC00780
      READ (IOIN,500) (DAT(J),J=1,5)                                    XSC00790
      IF (DAT(1).EQ.TYPE(1)) GO TO 1                                    XSC00800
      IF (DAT(1).EQ.TYPE(2)) GO TO 2                                    XSC00810
      IF (DAT(1).EQ.TYPE(3)) GO TO 3                                    XSC00820
      IF (DAT(1).EQ.TYPE(4)) GO TO 4                                    XSC00830
      IF (DAT(1).EQ.TYPE(5)) GO TO 5
      IF (DAT(1).EQ.TYPE(6)) GO TO 6                                    XSC00840
C     ERROR CHECK                                                       XSC00850
      GO TO 7                                                           XSC00860
C     ADVERSE WEATHER INDICATOR AND OPTIONAL RAIN RATE:                 XSC00870
1     IFOG=IFIX(DAT(2))                                                 XSC00880
      RNRT=DAT(3)                                                       XSC00890
      GO TO 9                                                           XSC00900
C     HORIZONTAL DISTANCE FOR HORIZONTAL PATH CALC.                     XSC00910
2     HORDIS=DAT(2)                                                     XSC00920
      ISLANT=0                                                          XSC00930
      GO TO 9                                                           XSC00940
C     HORIZONTAL DISTANCE AND ANGLE FOR SLANT PATH CALC.                XSC00950
3     HORDIS=DAT(2)                                                     XSC00960
      ANG=DAT(3)                                                        XSC00970
      ISLANT=1                                                          XSC00980
      GO TO 9                                                           XSC00990
C     SLANT DISTANCE AND ANGLE FOR SLANT PATH CALC.                     XSC01000
4     SLTDIS=DAT(2)                                                     XSC01010
      ANG=DAT(3)                                                        XSC01020
      ISLANT=1                                                          XSC01030
      GO TO 9
C     SET PLOT FLAG
5     NPLT=1
9     CONTINUE                                                          XSC01040
6     CONTINUE                                                          XSC01050
      IF (NOLO) WRITE (IOOUT,600)                                       XSC01060
      IF(IGEOSW.NE.1)GO TO 88                                           XSC01070
      HORDIS=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2)                XSC01080
      SLTDIS=SQRT(HORDIS**2+(PTS(3)-PTS(6))**2)                         XSC01090
      ANG=ACOS(HORDIS/SLTDIS)/PIRAD                                     XSC01100
   88 CONTINUE                                                          XSC01110
C     WAVELENGTH ERROR CHECK                                            XSC01120
      IF ((WAVE.GT..4.AND.WAVE.LE.2.).OR.(WAVE.GE.3..AND.WAVE.LE.5.)    XSC01130
     1  .OR.(WAVE.GE.8.AND.WAVE.LE.12.05)) GO TO 10                     XSC01140
      WRITE (IOOUT,1600) WAVE                                           XSC01150
      IERR=1                                                            XSC01160
      XSTRN=1.                                                          XSC01170
      RETURN                                                            XSC01180
10    CONTINUE                                                          XSC01190
      IF (NOLO.AND.IFOG.EQ.1) WRITE (IOOUT,800)                         XSC01210
      IF (NOLO.AND.IFOG.EQ.2) WRITE (IOOUT,900)                         XSC01220
      IF (NOLO.AND.IFOG.EQ.3) WRITE (IOOUT,950)                         XSC01230
      IF (NOLO.AND.IFOG.EQ.4) WRITE (IOOUT,1000)                        XSC01240
      IF (NOLO.AND.IFOG.EQ.5) WRITE (IOOUT,1100)                        XSC01250
      IF (ISLANT.GT.0) GO TO 11                                         XSC01200
      IF (NOLO) WRITE (IOOUT,1200)                                      XSC01260
11    IF (NOLO.AND.ISLANT.GE.1) WRITE (IOOUT,1400) WAVE                 XSC01270
      IF (ISLANT.GE.1.AND.(IFOG.LE.0.OR.IFOG.GE.4)) WRITE (IOOUT,2100)
      IF (ISLANT.GE.1.AND.(IFOG.LE.0.OR.IFOG.GE.4)) IFOG=1
      EXTN=EXT55
```

```
      IF (ISLANT.GE.1.AND.WAVE.GE..8) GO TO 101                          XSC01280
12     IF (ISLANT.GE.1) CALL SLANT(EXTN,HORDIS,SLTDIS,ANG,AVEX55,IERR,   XSC01290
     1 WAVE,NPLT)                                                        XSC01300
      IF (IERR.EQ.1) XSTRN =1.                                           XSC01310
      IF (IERR.EQ.1) RETURN                                              XSC01320
      EXTN=EXT55                                                         XSC01330
      IF (ISLANT.GE.1) EXTN=AVEX55                                       XSC01340
      XSPATH=HORDIS                                                      XSC01350
      IF (ISLANT.GE.1) XSPATH=SLTDIS                                     XSC01360
C     SLANT PATH EXTINCTION                                              XSC01370
      XSTRN =EXP(-XSPATH*EXTN)                                           XSC01380
      IF (NOLO) WRITE (IOOUT,1500) WAVE,EXTN,XSPATH,XSTRN,ANG            XSC01390
      RETURN                                                             XSC01400
100   IF (IFOG.NE.4) GO TO 101                                           XSC01410
C     RAIN - ALL WAVELENGTHS                                             XSC01420
      IF (NOLO.AND.RNRT.LE.0.0) WRITE (IOOUT,550)                        XSC01430
      IF (RNRT.LE.0.0) RNRT=1.                                           XSC01440
      RNEXTN=A0+A1*RNRT+A2*RNRT**2                                       XSC01450
      XSTRN =EXP(-XSPATH*RNEXTN)                                         XSC01460
      IF (NOLO) WRITE (IOOUT,1550) RNRT,RNEXTN,XSPATH,XSTRN              XSC01470
      RETURN                                                             XSC01480
  101 IF (ABS(WAVE-1.06).LT..01) GO TO 400                               XSC01490
      IF (WAVE.GE.3..AND.WAVE.LE.5.) GO TO 200                           XSC01500
      IF (WAVE.GE.8..AND.WAVE.LE.12.) GO TO 300                          XSC01510
      XSTRN=EXP(-XSPATH*EXTN)
      RETURN                                                             XSC01520
  200 CONTINUE                                                           XSC01530
C     3.0 TO 5.0 RANGE                                                   XSC01540
C     MA                                                                 XSC01550
      IF (IFOG.EQ.1) EXT35=10.**(0.0345+1.03*ALOG10(EXTN))               XSC01560
C     MP                                                                 XSC01570
      IF (IFOG.EQ.2) EXT35=10.**(-0.38+1.32*ALOG10(EXTN))                XSC01580
C     CP                                                                 XSC01590
      IF (IFOG.EQ.3) EXT35=10.**(-0.82+1.58*ALOG10(EXTN))                XSC01600
      IF(IFOG.NE.5) EXTN=EXT35
      IF (ISLANT.GE.1) GO TO 12
C     SNOW                                                               XSC01610
      IF (IFOG.EQ.5) EXT35=10.0**(1.05*ALOG10(EXTN)+.021)                XSC01620
      XSTRN =EXP(-XSPATH*EXT35)                                          XSC01630
      IF (NOLO) WRITE (IOOUT,1700) EXT35,XSPATH,XSTRN                    XSC01640
      RETURN                                                             XSC01650
  300 CONTINUE                                                           XSC01660
C     8.0 TO 12.0 RANGE                                                  XSC01670
C     MA                                                                 XSC01680
      IF (IFOG.EQ.1) EXT812=10.**(-.45+1.19*ALOG10(EXTN))                XSC01690
C     MP                                                                 XSC01700
      IF (IFOG.EQ.2) EXT812=10.**(-1.01+1.51*ALOG10(EXTN))               XSC01710
C     CP                                                                 XSC01720
      IF (IFOG.EQ.3) EXT812=10.**(-1.65+1.82*ALOG10(EXTN))               XSC01730
      IF(IFOG.NE.5) EXTN=EXT812
      IF (ISLANT.GE.1) GO TO 12
C     SNOW                                                               XSC01740
      IF (IFOG.EQ.5) EXT812=10.0**(.993*ALOG10(EXTN)+.114)               XSC01750
      XSTRN =EXP(-XSPATH*EXT812)                                         XSC01760
      IF (NOLO) WRITE (IOOUT,1800) EXT812,XSPATH,XSTRN                   XSC01770
      RETURN                                                             XSC01780
  400 CONTINUE                                                           XSC01790
C     1.06 RANGE                                                         XSC01800
C     MA, MP, AND CP                                                     XSC01810
      EXT106=AMIN1(10.**(-0.14+1.16*ALOG10(EXTN)),EXTN)
      EXTN=EXT106
      IF (ISLANT.GE.1) GO TO 12
C     SNOW - ASSUME THAT THE EXTINCTION AT 1.06 IS THE SAME AS AT .55 UMXSC01870
      IF (IFOG.EQ.5) EXT106=EXT55                                        XSC01880
      XSTRN =EXP(-XSPATH*EXT106)                                         XSC01890
      IF (NOLO) WRITE (IOOUT,1900) EXT106,XSPATH,XSTRN                   XSC01900
      RETURN                                                             XSC01910
7     WRITE (IOOUT,2000) (DAT(J),J=1,4)                                  XSC01920
      XSTRN=1.                                                           XSC01930
```

```
      IERR=1                                                                    XSC01940
      RETURN                                                                    XSC01950
C                                                                               XSC01960
500   FORMAT (A4,6X,5(F6.2,1X))                                                 XSC01970
550   FORMAT(1H0,50HXSCALE WARNING - RAIN RATE IS .LE. 0.0, RAIN RATE ,         XSC01980
     +20HSET EQUAL TO 1 MM/HR,/)                                                XSC01990
  600 FORMAT (1H ,///,1X,50X,14HOPTIONS CHOSEN)                                 XSC02000
  800 FORMAT (1H ,50X,14HMARITIME ARTIC)                                        XSC02010
  900 FORMAT (1H ,50X,14HMARITIME POLAR)                                        XSC02020
  950 FORMAT (1H ,50X,17HCONTINENTAL POLAR)                                     XSC02030
 1000 FORMAT (1H ,50X,4HRAIN)                                                   XSC02040
 1100 FORMAT (1H ,50X,4HSNOW)                                                   XSC02050
 1200 FORMAT (1H ,50X,15HHORIZONTAL PATH/)                                      XSC02060
 1400 FORMAT (1H ,50X,15HSLANT PATH FOR ,F8.3,8H MICRONS/)                      XSC02070
 1500 FORMAT (1X,23X,20HSLANT EXTINCTION AT ,F5.2,8H MICRONS,                   XSC02080
     1      4X,11HDISTANCE    ,12HTRANSMISSION,6X,5HANGLE/,1X,40X,              XSC02090
     2         26HKM**-1                      KM,/                            XSC02100
     3         ,1X,38X,F8.3,12X,F8.3,5X,E9.3,7X,F7.2)                          XSC02110
 1550 FORMAT (1H ,20X,27HEXTINCTION FOR RAIN RATE OF,F5.2,6H MM/HR,             XSC02120
     1      5X,11HDISTANCE    ,12HTRANSMISSION/,1X,40X,                        XSC02130
     2         26HKM**-1                      KM,/                            XSC02140
     3         ,1X,38X,F8.3,12X,F8.3,5X,E9.3)                                  XSC02150
 1600 FORMAT (1H ,18H***** WAVELENGTH (,F9.3,10H) OUTSIDE ,                     XSC02160
     1         10HALLOWABLE ,29HRANGE (1.06,3.0-5.0,8.0-12.0                   XSC02170
     2         31HMICRONS) - CONTROL RETURNED TO ,                             XSC02180
     3         17HMAIN FROM XSCALE.)                                           XSC02190
 1700 FORMAT (1H ,25X,37HEXTINCTION FROM 3.0 TO 5.0 MICRONS                     XSC02200
     1         11HDISTANCE    ,12HTRANSMISSION/,1X,40X,                       XSC02210
     2         26HKM**-1                      KM,/                            XSC02220
     3         ,1X,38X,F8.3,13X,F8.3,5X,E9.3)                                  XSC02230
 1800 FORMAT (1H ,25X,37HEXTINCTION FROM 8.0 TO 12.0 MICRONS                    XSC02240
     1         12H DISTANCE    ,12HTRANSMISSION/,1X,40X,                       XSC02250
     2         26HKM**-1                      KM,/                            XSC02260
     3         ,1X,38X,F8.3,12X,F8.3,5X,E9.3)                                  XSC02270
 1900 FORMAT (1H ,25X,37HEXTINCTION AT 1.06 MICRONS                             XSC02280
     1         11HDISTANCE    ,12HTRANSMISSION/,1X,40X,                       XSC02290
     2         26HKM**-1                      KM,/                            XSC02300
     3         ,1X,38X,F8.3,12X,F8.3,5X,E9.3)                                  XSC02310
2000  FORMAT (1H ,44HUNKNOWN CARD TYPE.  CONTROL RETURNED TO MAIN,              XSC02320
     +13H FROM XSCALE.,/,1X,A4,6X,5(F6.2,1X))                                   XSC02330
2100  FORMAT(1H ,40HINCORRECT FOG TYPE FOR SUBROUTINE SLANT,,/,1X,
     1 21HFOG TYPE CHANGED TO 1/)
      END                                                                       XSC02340
```

```
      SUBROUTINE SLANT(EXT55,HORDIS,SLTDIS,ANG,AVEX55,IERR,WAVE,NPLT)    SLNT0010
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSLNT0020
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                      SLNT0030
C     ALL QUANTITIES IN THIS ROUTINE ARE FOR .55 UM                      SLNT0040
      REAL KMTOM                                                         SLNT0050
C     KILOMETERS TO METERS                                               SLNT0060
      KMTOM=1000.                                                        SLNT0070
      IERR=0                                                             SLNT0080
      TOL=.0001                                                          SLNT0090
  100 IF (HORDIS.LT.TOL.OR.ANG.LT.TOL) GO TO 200                         SLNT0100
C     HORZONTAL DISTANCE AND ANGLE INPUT                                 SLNT0110
      VERDIS=HORDIS*TAN(ANG*PIRAD)                                       SLNT0120
      SLTDIS=SQRT(HORDIS**2+VERDIS**2)                                   SLNT0130
      GO TO 300                                                          SLNT0140
200   IF (SLTDIS.LT.TOL.OR.ANG.LT.TOL) GO TO 500                         SLNT0150
C     SLANT DISTANCE AND ANGLE INPUT                                     SLNT0160
      VERDIS=SLTDIS*SIN(ANG*PIRAD)                                       SLNT0170
      HORDIS=SQRT(SLTDIS**2-VERDIS**2)                                   SLNT0180
C     CONVERT TO 20 METER INCREMENTS                                     SLNT0190
300   VERDIS=FLOAT(IFIX((VERDIS+TOL)*KMTOM/20.))*20./KMTOM               SLNT0200
C     LIMIT ON VERTICAL HEIGHT IS 500 METERS                             SLNT0210
      IF (VERDIS*KMTOM.GT.500.) VERDIS=.5                                SLNT0220
      SD=SQRT(HORDIS**2+VERDIS**2)                                       SLNT0230
      IF(SD/SLTDIS.GT.1.01.OR.SD/SLTDIS.LT..99)WRITE(IOOUT,700)SLTDIS,SDSLNT0240
      SLTDIS=SD                                                          SLNT0250
C     FIND NBR OF 20 METER INCREMENTS                                    SLNT0260
      ITOP=IFIX((VERDIS+TOL)*KMTOM/20.)                                  SLNT0270
      IF(ITOP.LT.1) ITOP=1
      VERDIS=FLOAT(ITOP)*20./KMTOM
      EXTN=EXT55                                                         SLNT0280
C     BEGIN TRAPEZODIAL INTEGRATION FOR TAU ( OPTICAL DEPTH)             SLNT0290
      TAU=EXTN/2.                                                        SLNT0300
      ALT=0.0
      NPTS=ITOP+1
      IF (NPLT.EQ.1) WRITE (NPLOTU,352) NPTS
352   FORMAT (I5)
      IF (NPLT.EQ.1) WRITE (NPLOTU,351) EXTN,ALT,WAVE
      DO 400 I=1,ITOP                                                    SLNT0310
C     THESE FORMULAS ARE GOOD ONLY IN 20M INCREMENTS                     SLNT0320
      IF (EXTN.GE.7.0.AND.WAVE.LT.2.0) EXTN=                             SLNT0330
     1 10.**(0.55+0.72*ALOG10(EXTN))                                     SLNT0340
      IF (EXTN.LT.7.0.AND.WAVE.LT.2.0) EXTN=                             SLNT0350
     1 10.**(0.1+1.25*ALOG10(EXTN))                                      SLNT0360
      IF (EXTN.GE.3.3.AND.(WAVE.GE.3.0.AND.WAVE.LT.5.0)) EXTN=           SLNT0370
     1 10.**(0.55+.72*ALOG10(EXTN))                                      SLNT0380
      IF (EXTN.LT.3.3.AND.(WAVE.GE.3.0.AND.WAVE.LT.5.0)) EXTN=           SLNT0390
     1 10.**(0.3+1.2*ALOG10(EXTN))                                       SLNT0400
      IF (EXTN.GE.1.7.AND.(WAVE.GE.8.0.AND.WAVE.LT.12.0)) EXTN=          SLNT0410
     1 10.**(0.5+0.75*ALOG10(EXTN))                                      SLNT0420
      IF (EXTN.LT.1.7.AND.(WAVE.GE.8.0.AND.WAVE.LT.12.0)) EXTN=          SLNT0430
     1 10.**(0.4+1.2*ALOG10(EXTN))                                       SLNT0440
      ALT=FLOAT(I)*20.
      IF (NPLT.EQ.1) WRITE (NPLOTU,351) EXTN,ALT
351   FORMAT (3(E10.4,1X))
  400 TAU=TAU+EXTN                                                       SLNT0450
C     FINISH TRAP INTEGRATION                                            SLNT0460
      TAU=(TAU-EXTN/2.)*.02                                              SLNT0470
C     FIND AVERAGE EXTINCTION VALUE FOR SLANT PATH.                      SLNT0480
      AVEX55=TAU/VERDIS                                                  SLNT0490
      RETURN                                                             SLNT0500
  500 WRITE (IOOUT,600)                                                  SLNT0510
      IERR=1                                                             SLNT0520
      RETURN                                                             SLNT0530
C                                                                        SLNT0540
  600 FORMAT (1X,38HERROR - IMPROPER INPUT FOR SUBROUTINE                SLNT0550
     1        34HSLANT; TRANSMISSION SET EQUAL TO 1)                     SLNT0560
700   FORMAT (1H ,18HWARNING FROM SLANT,/,1X,22HTHE VERTICAL DISTANCE ,  SLNT0570
     +38HEXCEEDS THE 500 METER UPPER LIMIT, OR ,/,1X,10HIS NOT AN ,      SLNT0580
     +29HINTEGER MULTIPLE OF 20 METERS,/,1X,28HSLANT DISTANCE CHANGED FRSLNT0590
     +OM ,F7.4,4H TO ,F7.4,3H KM/)                                       SLNT0600
```

SLNT0610

END

```
      SUBROUTINE TURB(WAVE,IERR)                                        TUP00010
C   CALCULATES TURBULENCE INDUCED POINTING JITTER AND POWER SPECTRUM     TUR00020
C     FOR LASER TARGET DESIGNATOR AND TERMINAL HOMING SEEKER            TUR00030
C                                                                       TUR00040
C*********************************************************************  TUR00050
C       CALCULATION FOR THE DESIGNATOR PATH ARE PERFORMED EACH          TUR00060
C       TIME THIS ROUTINE IS REFERENCED.  THE CALCULATIONS FOR THE      TUR00070
C       SEEKER PATHS ARE PERFORMED ONLY WHEN THE DATA CARDS DVRV,       TUR00080
C       CN2, OR V2 ARE INCLUDED IN THE INPUT SET.                       TUR00090
C       THE INPUT IS CARD ORDER INDEPENDENT, WITH THE SINGLE            TUR00100
C       RESTRICTION THAT THE 'GO' CARD MUST BE THE LAST CARD            TUR00110
C       OF THE DATA SET.                                                TUR00120
C       THE DATA IS COMPLETELY IDENTIFIED BY THE ID IN COLUMNS          TUR00130
C       1-4 OF EACH CARD, FOLLOWED BY UP TO 7(REAL) FIELDS AS           TUR00140
C       NEEDED, WITH 10 COLUMNS PER FIELD BEGINNING IN COL. 11.         TUR00150
C       COMMENTS BELOW.                                                 TUR00160
C                                                                       TUR00170
C       THE INPUT FORMAT IS A4,6X,7(E10.4)                              TUR00180
C                                                                       TUR00190
C       THE INPUT OF A CN1, CN2, V1 AND V2 TYPE DATA CARD IS            TUR00200
C       TERMINATED WHEN THE FIRST COEFFICIENT OF VALUE ZERO             TUR00210
C       IS ENCOUNTERED.  THE REMAINING DATA COEFFICIENTS ON             TUR00220
C       THE CARD, IF ANY, ARE IGNORED.                                  TUR00230
C                                                                       TUR00240
C*********************************************************************  TUR00250
C                                                                       TUR00260
C  THE FOLLOWING ARE REQUIRED RECORDS FOR AT THE FIRST INPUT SET.       TUR00270
C                                                                       TUR00280
C  IDENT.   VARS.            DESCRIPTION                                TUR00290
C  ------   --------    ----------------------------------------       TUR00300
C                                                                       TUR00310
C  PARM                                                                 TUR00320
C           DIAM     LASER TARGET DESIGNATOR APERTURE DIAMETER IN METERS TUR00330
C           THET     LASER BEAMSPREAD ANGLE IN RADIANS                  TUR00340
C           TDOT     LASER BEAM SLUE RATE IN RADIANS/SECOND             TUR00350
C           RANG     DISTANCE FROM DESIGNATOR TO TARGET IN METERS       TUR00360
C           TIME     DURATION OF CALCULATIONS IN SECONDS                TUR00370
C           M        NO. OF FREQUENCIES FOR WHICH POINTING JITTER POWER TUR00380
C                    SPECTRUM IS TO BE CALCULATED, IF M=0, THEN DEFAULT TUR00390
C                    TO M=512.                                          TUR00400
C    ***RANG AND R1V ARE RECALCULATED IF IGEOSW=1                       TUR00410
C                                                                       TUR00420
C    ***N1 AND N2 ARE CALCULATED WITHIN THE ROUTINE.                    TUR00430
C                                                                       TUR00440
C           N1       NO OF SEGMENTS IN DESIGNATOR PATH                  TUR00450
C           N2       NO. OF SEGMENTS IN SEEKER TO TARGET PATH           TUR00460
C                                                                       TUR00470
C       N1 AND N2 ARE SET EQUAL TO THE INDEX OF THE LAST NON-ZERO       TUR00480
C       COEFFICIENT READ INTO CN1 AND CN2 RESPECTIVELY.                 TUR00490
C                                                                       TUR00500
C  CN1                                                                  TUR00510
C         IR2      STARTING INDEX VALUE OF CN1(I)                       TUR00520
C         (CN1(I), I=IR2,IR2+5)                                         TUR00530
C          CN1(I)   VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT       TUR00540
C                   WITH ONE VALUE FOR EACH SEGMENT OF RANGE FROM LASER TUR00550
C                   DESIGNATOR TO TARGET (METERS**(-2/3))               TUR00560
C                                                                       TUR00570
C  V1                                                                   TUR00580
C         IR2      STARTING INDEX VALUE OF V1(I)                        TUR00590
C         (V1(I), I=IR2,IR2+5)                                          TUR00600
C          V1(I)    SET OF VALUES OF CROSSWIND VELOCITY                TUR00610
C                   CORRESPONDING TO EACH SEGMENT OF RANGE FROM LASER   TUR00620
C                   DESIGNATOR TO TARGET (M/SEC)                        TUR00630
C                                                                       TUR00640
C  DVRV                                                                 TUR00650
C           D1V      DIAMETER OF SEEKER APERTURE IN METERS.             TUR00660
C           R1V      SEEKER RANGE TO TARGET IN METERS.                  TUR00670
C                                                                       TUR00680
C  CN2                                                                  TUR00690
C         IR2      STARTING INDEX VALUE OF CN2(I)                       TUR00700
```

```
C          (CN2(I), I=IR2,IR2+5)                                                TUR00710
C           CN2(I)  VALUES OF REFRACTIVE INDEX STRUCTURE                        TUR00720
C                   CONSTANT FOR EACH SEGMENT OF RANGE FROM TARGET TO           TUR00730
C                   SEEKER (METERS**(-2/3))                                     TUR00740
C                                                                               TUR00750
C V2                                                                            TUR00760
C          IR2      STARTING INDEX VALUE OF V2(I)                               TUR00770
C                                                                               TUR00780
C           V2(I)   VALUES OF CROSSWIND VELOCITY FOR EACH SEGMENT               TUR00790
C                   OF RANGE FROM TARGET TO SEEKER (M/SEC).                     TUR00800
C                                                                               TUR00810
C NPPS     NONE     PRINTS TABULAR VALUES OF POWER SPECTRUM VS FREQUENCY        TUR00820
C NPPJ     NONE     PRINTS TABULAR VALUES OF POINTING JITTER VS TIME            TUR00830
C NPAL     NONE     PRINTS TABULAR VALUES OF BOTH POWER SPECTRUM (PS)           TUR00840
C                   AND POINTING JITTER (PJ).                                   TUR00850
C   **IF THIS CARD IS MISSING NO TABLES WILL BE PRINTED (DEFAULT VALUE).        TUR00860
C                                                                               TUR00870
C THE FOLLOWING IDENT RECORD IS ALWAYS REQUIRED.                                TUR00880
C                                                                               TUR00890
C       GO            SIGNIFIES TO BEGIN EXECUTION FOR THIS DATA SET.           TUR00900
C                     AFTER EXECUTION, ANOTHER SET OF INPUTS MAY BE             TUR00910
C                     READ-IN FOLLOWED BY ANOTHER 'GO' CARD.                    TUR00920
C                     ANY VALUES ESTABLISHED FROM PREVIOUS INPUT SETS           TUR00930
C                     TO THE ROUTINE ARE STILL IN EFFECT.  THUS DATA            TUR00940
C                     SUCH AS FROM CARD PARM NEED NOT BE READ AGAIN IF          TUR00950
C                     THERE ARE TO BE NO CHANGES IN THE DATA ASSOCIATED         TUR00960
C                     WITH THAT IDENTIFIER.                                     TUR00970
C                                                                               TUR00980
C*****************************************************************            TUR00990
C     ++ CALLED PROGRAMS ++                                                     TUR01000
C     DESUB                                                                     TUR01010
C     FALPH                                                                     TUR01020
C     FFT4                                                                      TUR01030
C     GAUSS                                                                     TUR01040
C     MEANVR                                                                    TUR01050
C     RAND                                                                      TUR01060
C     SPECT                                                                     TUR01070
C     SPREAD                                                                    TUR01080
C     THETO                                                                     TUR01090
C*****************************************************************            TUR01100
      COMPLEX RAN                                                               TUR01110
      REAL LAMB                                                                 TUR01120
      REAL INR(7),IR1,LABEL(11)                                                 TUR01140
      LOGICAL SETUP                                                             TUR01150
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                            TUR01160
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUTUR01170
C     THIS IS A COMPLEX NUMBER  EACH NUMBER TAKES TWO WORDS                     TUR01180
      COMMON /M05/RAN(2048)                                                     TUR01190
      COMMON /M01/FR(1025),CN1(20),V1(20),FD(20),RD(20)                         TUR01200
      COMMON /LOWEX/PS(1025),V2(20),RR(10)                                      TUR01210
      DIMENSION CN2(20),PJCHAR(4),PSCHAR(4)                                     TUR01220
      COMMON /GEOMET/PTS(15),IGEOSW                                             TUR01230
      EXTERNAL DESUB,FALPH                                                      TUR01240
      DATA SETUP/.TRUE./                                                        TUR01250
      DATA PJCHAR,PSCHAR /4HPJ (,4HRAD*,4H*2/S,4HEC) ,4HPS (,4HRAD*,            TUR01260
     1                    4H*2/H,4HZ)  /
      DATA LABEL/4HGO   ,4H     ,4HPARM,4HCN1 ,4HV1   ,4HDVRV,4HCN2 ,           TUR01270
     1 4HV2   ,4HNPPS,4HNPPJ,4HNPAL/                                            TUR01280
C                                                                               TUR01290
      NPRINT=0                                                                  TUR01300
      LAMB=WAVE                                                                 TUR01310
C  SET THE SEED FOR THE RANDOM NUMBER GENERATOR                                 TUR01320
C***  NOTE, THIS SEED IS APPROPRIATE FOR THE RANDOM NUMBER GENERATOR            TUR01330
C***  USED AT THE ATMOSPHERIC SCIENCES LAB.  USERS AT OTHER                     TUR01340
C***  INSTALLATIONS WILL NEED TO SUPPLY THEIR OWN RANDOM NUMBER                 TUR01350
C***  GENERATOR.                                                                TUR01360
      IF (.NOT.SETUP) GO TO 100                                                 TUR01370
      VARX=735.34829                                                            TUR01380
      VARX=RAND(VARX)                                                           TUR01390
      SETUP=.FALSE.                                                             TUR01400
```

```
      IOPT=1                                                                    TUR01410
  100 READ(IOIN,3000) IR1,(INR(I), I=1,7)                                       TUR01420
      IR2=IFIX(INR(1))                                                          TUR01430
      IF(IR1.EQ.LABEL(1)) GO TO 180                                             TUR01440
      IF(IR1.EQ.LABEL(3)) GO TO 110                                             TUR01450
      IF(IR1.EQ.LABEL(4)) GO TO 120                                             TUR01460
      IF(IR1.EQ.LABEL(5)) GO TO 130                                             TUR01470
      IF(IR1.EQ.LABEL(6)) GO TO 140                                             TUR01480
      IF(IR1.EQ.LABEL(7)) GO TO 150                                             TUR01490
      IF(IR1.EQ.LABEL(8)) GO TO 160                                             TUR01500
C     PRINTING OPTIONS                                                          TUR01510
      IF(IR1.EQ.LABEL(9)) NPRINT=1                                              TUR01520
      IF(IR1.EQ.LABEL(10)) NPRINT=2                                             TUR01530
      IF(IR1.EQ.LABEL(11)) NPRINT=3                                             TUR01540
      IF(NPRINT.EQ.1.OR.NPRINT.EQ.2.OR.NPRINT.EQ.3) GO TO 100                   TUR01550
      WRITE(IOOUT,3001) IR1,(INR(I), I=1,7)                                     TUR01560
      GOTO 100                                                                  TUR01570
C                                                                               TUR01580
  110 DIAM = INR(1)                                                             TUR01590
      THET = INR(2)                                                             TUR01600
      TDOT = INR(3)                                                             TUR01610
      RANG = INR(4)                                                             TUR01620
      TIME = INR(5)                                                             TUR01630
      M = IFIX(INR(6))                                                          TUR01640
      IF(M .EQ. 0) M = 512                                                      TUR01650
      GOTO 100                                                                  TUR01660
C                                                                               TUR01670
  120 DO 125 I=2,7                                                              TUR01680
      IF(INR(I).NE.0.0) GOTO 121                                                TUR01690
      N1=IR2-1                                                                  TUR01700
      GOTO 100                                                                  TUR01710
  121 CN1(IR2)=INR(I)                                                           TUR01720
      IR2=IR2+1                                                                 TUR01730
      IF(IR2 .GT. 20) GOTO 126                                                  TUR01740
  125 CONTINUE                                                                  TUR01750
  126 IR1=IR2-1                                                                 TUR01760
      N1=MAX0(N1,IFIX(IR1))                                                     TUR01770
      GOTO 100                                                                  TUR01780
C                                                                               TUR01790
  130 DO 131 I=2,7                                                              TUR01800
      V1(IR2)=INR(I)                                                            TUR01810
      IR2=IR2+1                                                                 TUR01820
      IF(IR2 .GT. 20) GOTO 100                                                  TUR01830
  131 CONTINUE                                                                  TUR01840
      GOTO 100                                                                  TUR01850
C                                                                               TUR01860
  140 IOPT = 2                                                                  TUR01870
      D1V = INR(1)                                                              TUR01880
      R1V = INR(2)                                                              TUR01890
      GOTO 100                                                                  TUR01900
C                                                                               TUR01910
  150 IOPT = 2                                                                  TUR01920
      DO 155 I=2,7                                                              TUR01930
      IF(INR(I).NE.0.0) GOTO 151                                                TUR01940
      N2=IR2-1                                                                  TUR01950
      GOTO 100                                                                  TUR01960
  151 CN2(IR2) = INR(I)                                                         TUR01970
      IR2=IR2+1                                                                 TUR01980
      IF(IR2 .GT. 20) GOTO 156                                                  TUR01990
  155 CONTINUE                                                                  TUR02000
  156 IR1=IR2-1                                                                 TUR02010
      N2=MAX0(N2,IFIX(IR1))                                                     TUR02020
      GOTO 100                                                                  TUR02030
C                                                                               TUR02040
  160 IOPT = 2                                                                  TUR02050
      DO 161 I=2,7                                                              TUR02060
      V2(IR2) = INR(I)                                                          TUR02070
      IR2=IR2+1                                                                 TUR02080
      IF(IR2 .GT. 20) GOTO 100                                                  TUR02090
  161 CONTINUE                                                                  TUR02100
```

```
      GOTO 100                                                          TUR02110
C                                                                       TUR02120
  180 IF(IGEOSW.NE.1)GO TO 190                                          TUR02130
      RANG=SQRT((PTS(7)-PTS(1))**2+(PTS(8)-PTS(2))**2+                  TUR02140
     +(PTS(9)-PTS(3))**2)                                               TUR02150
      DISKTM=1000.0                                                     TUR02160
      RANG=RANG*DISKTM                                                  TUR02170
      R1V=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+(PTS(6)-PTS(3))**2)TUR02180
      R1V=R1V*DISKTM                                                    TUR02190
  190 CONTINUE                                                          TUR02200
      IF (LAMB.LE.14.) GO TO 200                                        TUR02210
      WRITE (IOOUT,2500) LAMB                                           TUR02220
      IERR=1                                                            TUR02230
      RETURN                                                            TUR02240
  200 WRITE (IOOUT,3100)                                                TUR02250
      IF (IOPT.EQ.2) WRITE (IOOUT,3200)                                 TUR02260
      WRITE (IOOUT,3300) LAMB,DIAM,THET                                 TUR02270
C  CHANGE WAVELENGTH TO METERS                                          TUR02280
      LAMB=LAMB/1.0E+6                                                  TUR02290
      IF (IOPT.EQ.2) WRITE (IOOUT,3400) D1V,R1V                         TUR02300
      WRITE (IOOUT,3500) TDOT,RANG                                      TUR02310
      WRITE (IOOUT,3600) TIME                                           TUR02320
      WRITE (IOOUT,3700) N1                                             TUR02330
      IF (IOPT.EQ.2) WRITE (IOOUT,3800) N2                              TUR02340
      WRITE (IOOUT,3900) M                                              TUR02350
      WRITE (IOOUT,4000)                                                TUR02360
      WRITE (IOOUT,4200)                                                TUR02370
      DO 400 I=1,N1                                                     TUR02380
      WRITE (IOOUT,4300) I,CN1(I),V1(I)                                 TUR02390
  400 CONTINUE                                                          TUR02400
      IF (IOPT.EQ.1) GO TO 600                                          TUR02410
      WRITE (IOOUT,4100)                                                TUR02420
      WRITE (IOOUT,4200)                                                TUR02430
      DO 500 I=1,N2                                                     TUR02440
      WRITE (IOOUT,4300) I,CN2(I),V2(I)                                 TUR02450
  500 CONTINUE                                                          TUR02460
C  COMPUTATION OF TIME, FREQUENCY AND SPATIAL INCREMENTS                TUR02470
  600 DELT=TIME/M                                                       TUR02480
      DELF=1./TIME                                                      TUR02490
      DELZ=RANG/FLOAT(N1)                                               TUR02500
      IF (N2.NE.0) DEL1V=R1V/FLOAT(N2)                                  TUR02510
      MM=M+M                                                            TUR02520
      M1=M+1                                                            TUR02530
      MM1=MM+1                                                          TUR02540
      MM2=MM+2                                                          TUR02550
      MSQ=SQRT(FLOAT(MM))                                               TUR02560
      DO 700 I=2,M1                                                     TUR02570
      FR(I)=(I-1)*DELF                                                  TUR02580
  700 CONTINUE                                                          TUR02590
      R2=DIAM/THET                                                      TUR02600
      R=RANG+R2                                                         TUR02610
      DT=THET*RANG                                                      TUR02620
      D2=DIAM+DT                                                        TUR02630
C  COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND         TUR02640
C  NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM LASER          TUR02650
C  DESIGNATOR TO TARGET                                                 TUR02660
      ZI=0.0                                                            TUR02670
      Z1=DELZ/2.                                                        TUR02680
      ROT=0.0                                                           TUR02690
      DEL=0.0                                                           TUR02700
      DO 800 I=1,N1                                                     TUR02710
      ZI=ZI+Z1                                                          TUR02720
      Z1=DELZ                                                           TUR02730
      VEI=V1(I)+TDOT*(ZI-R2)                                            TUR02740
      RO(I)=16.71*DELZ*CN1(I)*(1.-ZI/RANG)**1.66667/(LAMB*LAMB)         TUR02750
      FO(I)=VEI/(PI*D2*ZI/R)                                            TUR02760
      ROT=ROT+RO(I)                                                     TUR02770
      DEL=DEL+DELZ*CN1(I)*(RANG-ZI)/RANG                                TUR02780
      RO(I)=RO(I)**(-.6)                                                TUR02790
  800 CONTINUE                                                          TUR02800
```

```
      ROT=ROT**(-.6)                                                    TUR02810
      CALL SPREAD(DIAM,ROT,LAMB,THET,RANG,DRO,THETDL,DDL,THET12,D12,    TUR02820
     1            DTHET,D22)                                            TUR02830
      WRITE (IOOUT,4500) R2,R                                           TUR02840
      WRITE (IOOUT,4600) ROT,DRO,THET,DT,THETDL,DDL,THET12,D12,DTHET,D22TUR02850
      WRITE (IOOUT,4700)                                                TUR02860
      DO 1000 I=1,N1                                                    TUR02870
 1000 WRITE (IOOUT,4800) I,RO(I),FO(I)                                  TUR02880
C  COMPUTATION OF ANGLE OF ARRIVAL POWER SPECTRUM OF LASER  DESIGNATOR   TUR02890
      F2=0.                                                             TUR02900
      PS(1)=0.                                                          TUR02910
      DO 1100 J=2,M1                                                    TUR02920
      F=FR(J)                                                           TUR02930
      F1=0.                                                             TUR02940
      CALL SPECT(F,D2,N1,F1,LAMB)                                       TUR02950
      PS(J)=F1                                                          TUR02960
      IF (IOPT.EQ.1) PS(J)=F1*(D2/DIAM)**2                              TUR02970
      F2=F2+PS(J)*DELF                                                  TUR02980
 1100 CONTINUE                                                          TUR02990
      IF (IOPT.EQ.1) WRITE (IOOUT,4400) F2                              TUR03000
      F2SQRT=SQRT(F2)                                                   TUR03010
      AJITT=F2SQRT*RANG                                                 TUR03020
      IF (IOPT.EQ.1) WRITE (IOOUT,4900) F2SQRT,AJITT                    TUR03030
      IF (IOPT.EQ.1) GO TO 1500                                         TUR03040
C                                                                       TUR03050
      DTV=THET*R1V                                                      TUR03060
C  COMPUTATION OF EFFECTIVE WIND VELOCITY, COHERENCE LENGTH AND         TUR03070
C  NORMALIZATION FREQUENCY FOR EACH SEGMENT OF PATH FROM                TUR03080
C  TARGET TO SEEKER                                                     TUR03090
      ZI=0.                                                             TUR03100
      Z1=DEL1V/2.                                                       TUR03110
      ROT=0.0                                                           TUR03120
      DO 1200 I=1,N2                                                    TUR03130
      ZI=ZI+Z1                                                          TUR03140
      Z1=DEL1V                                                          TUR03150
      VEI=V2(I)+TDOT*(R1V-ZI)                                           TUR03160
      RO(I)=16.71*DEL1V*CN2(I)*(ZI/R1V)**1.66667/(LAMB*LAMB)            TUR03170
      ROT=ROT+RO(I)                                                     TUR03180
      RO(I)=RO(I)**(-.6)                                                TUR03190
      FO(I)=VEI/(PI*D1V*ZI/R1V)                                         TUR03200
 1200 CONTINUE                                                          TUR03210
      ROT=ROT**(-.6)                                                    TUR03220
      CALL SPREAD(D22,ROT,LAMB,THET,R1V,DRO,THETDL,DDL,THET12,D12,DTHET,TUR03230
     1            D22V)                                                 TUR03240
      WRITE (IOOUT,4550)                                                TUR03250
      WRITE (IOOUT,4600) ROT,DRO,THET,DTV,THETDL,DDL,THET12,D12,DTHET,  TUR03260
     1                   D22V                                           TUR03270
      WRITE (IOOUT,4700)                                                TUR03280
      DO 1300 I=1,N2                                                    TUR03290
 1300 WRITE (IOOUT,4800) I,RO(I),FO(I)                                  TUR03300
C  COMPUTATION OF TURBULENCE INDUCED POINTING JITTER POWER SPECTRUM     TUR03310
C  FROM TARGET SPOT TO LASER SEEKER,  COMPUTATION OF TOTAL POWER        TUR03320
C  SPECTRUM FROM LASER DESIGNATOR TO SEEKER AND POWER SPECTRUM VARIANCE  TUR03330
      CALL THETO(THETAO,FALPH,CN2,D1V,R1V,N2)                           TUR03340
      F2=0.                                                             TUR03350
      DO 1400 J=2,M1                                                    TUR03360
      F=FR(J)                                                           TUR03370
      F1=0.                                                             TUR03380
      CALL SPECT(F,D1V,N2,F1,LAMB)                                      TUR03390
      PS(J)=PS(J)+F1/(1.+(D2/(R1V*THETAO))**2)                          TUR03400
      PS(J)=PS(J)*(D2/D1V)**2                                           TUR03410
      F2=F2+PS(J)*DELF                                                  TUR03420
 1400 CONTINUE                                                          TUR03430
      WRITE (IOOUT,4400) F2                                             TUR03440
 1500 IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5000) DELF           TUR03450
      FREQ=0.0                                                          TUR03460
      FINC=DELF*10.0                                                    TUR03470
      IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5600)                TUR03480
      IF (NPRINT.EQ.1.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PSCHAR         TUR03490
      K=1                                                               TUR03500
```

```
      L=10                                                              TUR03510
 1600 IF (L.GT.M1) L=M1                                                 TUR03520
      IF(NPRINT.EQ.1.OR.NPRINT.EQ.3)WRITE (IOOUT,2800)FREQ,(PS(J),J=K,L)TUR03530
      FREQ=FREQ+FINC                                                    TUR03540
      K=L+1                                                             TUR03550
      L=K+9                                                             TUR03560
      IF (K.LE.M1) GO TO 1600                                           TUR03570
      DO 2400 L=1,IOPT                                                  TUR03580
      IF (L.EQ.1) WRITE (IOOUT,5100)                                    TUR03590
      IF (L.EQ.2) WRITE (IOOUT,5200)                                    TUR03600
C   GENERATION OF RANDOM SEQUENCE HAVING SAME POWER SPECTRUM VARIANCE    TUR03610
C   AS INDUCED BY TURBULANCE.  ADD SYMMETRIC TERMS FOR NEGATIVE         TUR03620
C   FREQUENCIES.  COMPUTE MEAN AND VARIANCE OF RANDOM ARRAY             TUR03630
      RAN(1)=(0.,0.)                                                    TUR03640
      DO 1700 I=2,M1                                                    TUR03650
      MMM=MM2-I                                                         TUR03660
C   GENERATE RANDOM NUMBER WITH ROUTINE GAUSS                           TUR03670
C       NORMAL DISTRIBUTION                                             TUR03680
C         MEAN = 0.0                                                    TUR03690
C         STANDARD DEVIATION = 1.0                                      TUR03700
C       REAL PART = RANDOM NUMBER                                       TUR03710
C       IMAG PART = 0                                                   TUR03720
      RAN(I)=CMPLX(GAUSS(12,0.0,1.0),0.0)*SQRT(PS(I)/DELT)              TUR03730
      RAN(MMM)=RAN(I)                                                   TUR03740
 1700 CONTINUE                                                          TUR03750
C   CALCULATE POWER OF 2 ,NPOW, FOR FFT4 SINCE ARRAY PASSED TO          TUR03760
C   FFT4 MUST HAVE SIZE THAT IS A POWER OF 2.  (NOTE LN(2)=0.693147.)   TUR03770
      NPOW=IFIX(ALOG(FLOAT(MM))/0.693147)                               TUR03780
      NMAX=2**NPOW                                                      TUR03790
      IF (MM.EQ.NMAX) GO TO 1900                                        TUR03800
C   IF MM IS NOT A POWER OF 2 THEN RESET THE REST OF ARRAY RAN          TUR03810
      NPOW=NPOW+1                                                       TUR03820
      ISTART=MM+1                                                       TUR03830
      NMAX=2**NPOW                                                      TUR03840
      DO 1800 I=ISTART,NMAX                                             TUR03850
      RAN(I)=(0.0,0.0)                                                  TUR03860
 1800 CONTINUE                                                          TUR03870
C   COMPUTE AND WRITE MEAN AND VARIANCE OF RANDOM ARRAY                 TUR03880
 1900 WRITE (IOOUT,5300)                                                TUR03890
      CALL MEANVR(1,M)                                                  TUR03900
C   FAST FOURIER TRANSFORM RANDOM ARRAY                                 TUR03910
C   CDC ROUTINE CALL                                                    TUR03920
C           CALL FFT(RAN,MM,+1)                                         TUR03930
      CALL FFT4(1.0,RAN,NPOW,NMAX)                                      TUR03940
      DO 2100 I=1,MM                                                    TUR03950
      RAN(I)=RAN(I)/MSQ                                                 TUR03960
 2100 CONTINUE                                                          TUR03970
C   COMPUTE AND WRITE MEAN AND VARIANCE OF TIME SEQUENCE.               TUR03980
      WRITE (IOOUT,5500)                                                TUR03990
      CALL MEANVR(M1,MM)                                                TUR04000
C   WRITE TRANSFORMED ARRAY VALUES CORRESPONDING TO TIME VALUES         TUR04010
C   OF POINTING JITTER FOR ONE DIRECTION.                               TUR04020
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5400) DELT           TUR04030
      DTIME=DELT                                                        TUR04040
      TINC=DELT*10.0                                                    TUR04050
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5700)                TUR04060
      IF (NPRINT.EQ.2.OR.NPRINT.EQ.3) WRITE (IOOUT,5800) PJCHAR         TUR04070
      DO 2300 I1=M1,MM,10                                               TUR04080
      DO 2200 I2=1,10                                                   TUR04090
      I3=I1+I2-1                                                        TUR04100
      RR(I2)=REAL(RAN(I3))                                              TUR04110
 2200 CONTINUE                                                          TUR04120
      IF(NPRINT.EQ.2.OR.NPRINT.EQ.3)WRITE(IOOUT,2800)DTIME,             TUR04130
     1    (RR(I),I=1,10)                                                TUR04140
      DTIME=DTIME+TINC                                                  TUR04150
 2300 CONTINUE                                                          TUR04160
 2400 CONTINUE                                                          TUR04170
C                                                                       TUR04180
C                                                                       TUR04190
 2500 FORMAT (1X,100(1H*),/,13H WAVELENGTH (,F10.3,7H) GREAT,           TUR04200
```

```
     1         40HER THAN 14 MICRONS: CONTROL RETURNED TO ,                    TUR04210
     2         21HMAIN FROM TURBULENCE.,/,1X,100(1H*))                         TUR04220
2700 FORMAT (5F16.6)                                                           TUR04230
2800 FORMAT (E10.4,10E12.4)                                                    TUR04240
3000 FORMAT (A4,6X,7E10.4)                                                     TUR04250
3001 FORMAT(1H0,20X,45HTHE FOLLOWING ID-FIELD NOT RECOGNIZED BY TURB,/,        TUR04260
     1 1X,A4,6X,7E10.4)                                                        TUR04270
3100 FORMAT (1H1,36H  CALCULATION OF POWER SPECTRUM AND                        TUR04280
     1         11HTURBULENCE 29HINDUCED POINTING JITTER OF A                   TUR04290
     2         23HLASER TARGET DESIGNATOR)                                     TUR04300
3200 FORMAT (1H+,101X,10HAND SEEKER)                                           TUR04310
3300 FORMAT (1H0,41X,30HLASER WAVELENGTH (MICROMETERS),8X,                     TUR04320
     1         F10.4,//,42X,26HDESIG. APERTURE DIAMETER (                      TUR04330
     2         7HMETERS),5X,F10.6,//,42X,18HBEAMSPREAD ANGLE (                 TUR04340
     3         8HRADIANS),12X,F10.6)                                           TUR04350
3400 FORMAT (1H0,41X,33HSEEKER APERTURE DIAMETER (METERS),5X,                  TUR04360
     1         F10.6,//,42X,36HRANGE FROM TARGET TO SEEKER (METERS),           TUR04370
     2         2X,F10.2)                                                       TUR04380
3500 FORMAT (1H0,41X,24HBEAM SLUE RATE (RAD/SEC),14X,F10.6,                    TUR04390
     1         //,42X,26HDESIGNATION RANGE (METERS),12X,                       TUR04400
     2         F10.2)                                                          TUR04410
3600 FORMAT (1H0,41X,26HDURATION OF TEST (SECONDS),12X,                        TUR04420
     1         F10.4)                                                          TUR04430
3700 FORMAT (1H0,41X,30HTOTAL DESIGNATOR PATH SEGMENTS,15X,                    TUR04440
     1         I3)                                                             TUR04450
3800 FORMAT (1H0,41X,26HTOTAL SEEKER PATH SEGMENTS,19X,I3)                     TUR04460
3900 FORMAT (1H0,41X,27HTOTAL FREQUENCIES FOR WHICH,/,42X,                     TUR04470
     1         22H  POWER SPECTRUM IS TO,/,42X,                                TUR04480
     2         15H  BE CALCULATED,27X,I4)                                      TUR04490
4000 FORMAT (1H0,///,26X,29H  VALUES OF REFRACTIVE INDEX                       TUR04500
     1         37HSTRUCTURE CONSTANT AND WIND SPEED IN                         TUR04510
     2         15HDESIGNATOR PATH)                                             TUR04520
4100 FORMAT (1H0,///,26X,29H  VALUES OF REFRACTIVE INDEX                       TUR04530
     1         37HSTRUCTURE CONSTANT AND WIND SPEED IN                         TUR04540
     2         11HSEEKER PATH)                                                 TUR04550
4200 FORMAT (1H0,62X,5HCN**2,9X,9HWINDSPEED,16X,/,42X,                         TUR04560
     1         11HSEGMENT NO.,5X,15H(METER**(-2/3)),3X,                        TUR04570
     2         11H(METER/SEC),//)                                              TUR04580
4300 FORMAT (1H ,46X,I2,10X,E12.6,5X,F10.2)                                    TUR04590
4400 FORMAT (1H0,//,36H THE VARIANCE OF THE POWER SPECTRUM                     TUR04600
     1         3HIS ,E12.4)                                                    TUR04610
4500 FORMAT (1H1,//,56X,20HDESIGNATOR TO TARGET,//,38X,                        TUR04620
     1         //,24X,41HVIRTUAL POINT SOURCE TO APERTURE DISTANCE,            TUR04630
     2         28X,F10.5,9H (METERS),//,24X,22HDISTANCE FROM VIRTUAL          TUR04640
     3         22HPOINT SOURCE TO TARGET,25X,F10.5,9H (METERS))               TUR04650
4550 FORMAT (1H1,//,58X,16HTARGET TO SEEKER,//)                                TUR04660
4600 FORMAT (1H0,23X,27HINTEGRATED COHERENCE LENGTH,42X,F10.6,                 TUR04670
     1         9H (METERS),//,24X,36HDIAMETER/INTEGRATED COHERENCE LENGTHTUR04680
     2         ,33X,F10.6,//,24X,31HTRANSMITTER-INDUCED BEAM SPREAD          TUR04690
     3         13X,E12.5,10H (RADIANS),3X,F10.6,9H (METERS),                 TUR04700
     4         //,24X,31HDIFFRACTION-LIMITED BEAM SPREAD,13X,E12.5,          TUR04710
     5         10H (RADIANS),3X,F10.6,9H (METERS),//,24X,                    TUR04720
     6         38HDIFFRACTION AND TURBULENCE BEAM SPREAD,6X,E12.5,            TUR04730
     7         10H (RADIANS),3X,F10.6,9H (METERS),///,24X,6HTOTAL            TUR04740
     8         19HEFFECTIVE BEAM SIZE,19X,E12.5,10H (RADIANS),3X,F10.6,      TUR04750
     9         9H (METERS),//)                                               TUR04760
4700 FORMAT (1H0,34X,11HSEGMENT NO.,9X,16HCOHERENCE LENGTH,                    TUR04770
     1         5X,27H REFERENCE FREQUENCY(HERTZ),//)                           TUR04780
4800 FORMAT (1H ,38X,I2,10X,F16.6,10X,F16.6)                                   TUR04790
4900 FORMAT (1H0,19H RMS SPOT JITTER = ,E10.4,10H RAD, OR =                    TUR04800
     1         ,E10.4,7H METERS,/)                                             TUR04810
5000 FORMAT (1H1,1X,40X,31H CALCULATED POWER SPECTRUM VS.                      TUR04820
     1         9HFREQUENCY,/,47X,16H AT INTERVALS OF,F6.3,6H HERTZ)            TUR04830
5100 FORMAT (1H0,37H OUTPUT FOR DESIGNATOR TO TARGET PATH)                     TUR04840
5200 FORMAT (1H0,33H OUTPUT FOR TARGET TO SEEKER PATH)                         TUR04850
5300 FORMAT (1H0,35X,35H  MEAN AND VARIANCE OF RANDOM ARRAY                    TUR04860
     1         ,//)                                                            TUR04870
5400 FORMAT (1H1,49X,34HVALUES OF POINTING JITTER VS. TIME /                   TUR04880
     1         52X,16H AT INTERVALS OF,F8.4,4H SEC)                            TUR04890
5500 FORMAT (1H0,//,35X,28H  MEAN AND VARIANCE OF TIME                         TUR04900
```

```
     1          8HSEQUENCE//)                                          TUR04910
5600 FORMAT (1H ,10H BEGINNING,/,10H FREQ (HZ))                        TUR04920
5700 FORMAT (1H ,2X,5H TIME,/,2X,6H (SEC))                             TUR04930
5800 FORMAT (1H+,T13,46(1H-),1H ,4A4,55(1H-),/)                        TUR04940
     RETURN                                                            TUR04950
     END                                                               TUR04960
```

```
      FUNCTION DESUB(X,DRO)                                              DES00010
C  FUNCTION USED FOR INTEGERATION  F(X)                                  DES00020
      FACTR=3.44*(DRO*X)**1.666667*(1.-X**0.333333)                      DES00030
      IF(FACTR.LT.160.) GO TO 10                                         DES00040
      DESUB=0.                                                           DES00050
      GO TO 20                                                           DES00060
   10 ARCCX=ATAN2(SQRT(1.-X**2),X)                                       DES00070
      DESUB=X*((ARCCX-X*(1.-X**2)**.5)*EXP(-FACTR))                      DES00080
   20 RETURN                                                             DES00090
      END                                                                DES00100
```

```
      FUNCTION FALPH(XI1)                                               FAL00010
      IF (XI1.GE..5623) GO TO 100                                       FAL00020
      FALPH=10.66*((XI1)**2)                                            FAL00030
      GO TO 900                                                         FAL00040
  100 IF (XI1.GE.1.0) GO TO 200                                         FAL00050
      FALPH=4.025*XI1-.00659                                            FAL00060
      GO TO 900                                                         FAL00070
  200 IF (XI1.GE.1.778) GO TO 300                                       FAL00080
      FALPH=1.8547*XI1+2.164                                            FAL00090
      GO TO 900                                                         FAL00100
  300 IF (XI1.GE.3.162) GO TO 400                                       FAL00110
      FALPH=.8475*XI1+3.955                                             FAL00120
      GO TO 900                                                         FAL00130
  400 IF (XI1.GE.5.623) GO TO 500                                       FAL00140
      FALPH=.391*XI1+5.3977                                             FAL00150
      GO TO 900                                                         FAL00160
  500 IF (XI1.GE.10.) GO TO 600                                         FAL00170
      FALPH=.1814*XI1+6.578                                             FAL00180
      GO TO 900                                                         FAL00190
  600 IF (XI1.GT.31.62) GO TO 700                                       FAL00200
      FALPH=.0534*XI1+7.95                                              FAL00210
      GO TO 900                                                         FAL00220
  700 IF (XI1.GT.1000.) GO TO 800                                       FAL00230
      FALPH=7.8*(XI1**.06)                                              FAL00240
      GO TO 900                                                         FAL00250
  800 FALPH=11.97                                                       FAL00260
  900 RETURN                                                            FAL00270
      END                                                               FAL00280
```

```
      SUBROUTINE FFT4(SIGN,X,NPOW,NMAX)                                  FFT00010
C                                                                        FFT00020
C     COOLEY-TUKEY METHOD OF FOURIER TRANSFORM                           FFT00030
C     INCLUDES SINE COSINE COMPUTATION AND                               FFT00040
C     REARRANGING DATA ACCORDING TO REVERSE BIT ADDRESSES                FFT00050
C                                                                        FFT00060
C     SIGN = FOURIER DIRECTION TRANSFORM FLAG                            FFT00070
C            -1. FOR DIRECT TRANSFORM, TO COEFFICIENTS FROM SERIES       FFT00080
C             1. FOR INVERSE TRANSFORM, TO SERIES FROM COEFFICIENTS      FFT00090
C                                                                        FFT00100
C     X      = LOC. OF FOURIER TRANSFORM BLOCK                           FFT00110
C                                                                        FFT00120
C     NPOW = POWER OF 2 TO OBTAIN NMAX                                   FFT00130
C                                                                        FFT00140
C     NMAX = LENGTH OF BLOCK X                                           FFT00150
C                                                                        FFT00160
      COMPLEX X,CXCS,HOLD,XA                                             FFT00170
      DIMENSION CS(2),MSK(13)                                            FFT00180
      DIMENSION X(1)                                                     FFT00190
      EQUIVALENCE (CXCS,CS)                                              FFT00200
      ZZ=6.283185306*SIGN/FLOAT(NMAX)                                    FFT00210
      MSK(1)=NMAX/2                                                      FFT00220
      DO 100 I=2,NPOW                                                    FFT00230
      MSK(I)=MSK(I-1)/2                                                  FFT00240
  100 CONTINUE                                                           FFT00250
      NN=NMAX                                                            FFT00260
      MM=2                                                               FFT00270
C     LOOP OVER NPOW LAYERS                                              FFT00280
      DO 800 LAYER=1,NPOW                                                FFT00290
      NN=NN/2                                                            FFT00300
      NW=0                                                               FFT00310
      DO 700 I=1,MM,2                                                    FFT00320
      II=NN*I                                                            FFT00330
C          CXCS = CEXP(2*PI*NW*SIGN/NMAX)                                FFT00340
      W=FLOAT(NW)*ZZ                                                     FFT00350
      CS(1)=COS(W)                                                       FFT00360
      CS(2)=SIN(W)                                                       FFT00370
C     COMPUTE ELEMENTS FOR BOTH HALFS OF EACH BLOCK                      FFT00380
      DO 200 J=1,NN                                                      FFT00390
      II=II+1                                                            FFT00400
      IJ=II-NN                                                           FFT00410
      XA=CXCS*X(II)                                                      FFT00420
      X(II)=X(IJ)-XA                                                     FFT00430
      X(IJ)=X(IJ)+XA                                                     FFT00440
  200 CONTINUE                                                           FFT00450
C     BUMP UP SERIES BY 2                                                FFT00460
C     COMPUTE REVERSE ADDRESS                                            FFT00470
      DO 400 LOC=2,NPOW                                                  FFT00480
      LL=NW-MSK(LOC)                                                     FFT00490
      IF (LL) 500,600,300                                                FFT00500
  300 NW=LL                                                              FFT00510
  400 CONTINUE                                                           FFT00520
  500 NW=MSK(LOC)+NW                                                     FFT00530
      GO TO 700                                                          FFT00540
  600 NW=MSK(LOC+1)                                                      FFT00550
  700 CONTINUE                                                           FFT00560
      MM=MM*2                                                            FFT00570
  800 CONTINUE                                                           FFT00580
C     DO FINAL REARRANGEMENT                                             FFT00590
      NW=0                                                               FFT00600
      DO 1600 I=1,NMAX                                                   FFT00610
      NW1=NW+1                                                           FFT00620
      HOLD=X(NW1)                                                        FFT00630
      IF (NW1-I) 1100,1000,900                                           FFT00640
  900 X(NW1)=X(I)                                                        FFT00650
 1000 X(I)=HOLD                                                          FFT00660
C     BUMP UP SERIES BY 1                                                FFT00670
C     COMPUTE REVERSE ADDRESS                                            FFT00680
 1100 DO 1300 LOC=1,NPOW                                                 FFT00690
      LL=NW-MSK(LOC)                                                     FFT00700
```

```
      IF (LL) 1400,1500,1200                                          FFT00710
1200  NW=LL                                                           FFT00720
1300  CONTINUE                                                        FFT00730
1400  NW=MSK(LOC)+NW                                                  FFT00740
      GO TO 1600                                                      FFT00750
1500  NW=MSK(LOC+1)                                                   FFT00760
1600  CONTINUE                                                        FFT00770
      IF (SIGN) 1900,1900,1700                                        FFT00780
1700  PTS=NMAX                                                        FFT00790
      DO 1800 I=1,NMAX                                                FFT00800
      X(I)=X(I)/PTS                                                   FFT00810
1800  CONTINUE                                                        FFT00820
1900  RETURN                                                          FFT00830
      END                                                             FFT00840
```

```
      FUNCTION GAUSS(N,XBAR,SIGMA)                                      GAU00010
C   GENERATE RANDOM NUMBERS WITH NORMAL DISTRIBUTION                    GAU00030
C     MEAN = XBAR                                                       GAU00040
C     STANDARD DEVIATION = SIGMA                                        GAU00050
      DATA NN /0/                                                       GAU00060
      IF (NN.GT.0) GO TO 1                                              GAU00070
      NN=1                                                              GAU00080
      C=0.                                                              GAU00090
1     CONTINUE                                                          GAU00100
      X=0.0                                                             GAU00110
      IF (C.EQ.0.0) C=735.34829                                         GAU00120
      DO 100 J=1,N                                                      GAU00130
      C=RAND(C)                                                         GAU00140
      X=X+C                                                             GAU00150
  100 CONTINUE                                                          GAU00160
      XN=N                                                              GAU00170
      X=SQRT(12.0/XN)*(X-0.5*XN)                                        GAU00180
      GAUSS=SIGMA*X+XBAR                                                GAU00190
      RETURN                                                            GAU00200
      END                                                               GAU00210
```

```
      SUBROUTINE MEANVR(N1,N2)                                          MEA00010
C                                                                       MEA00020
CCC   COMPUTE AND WRITE MEAN AND VARIANCE OF COMPLEX ARRAY              MEA00030
CCC   OVER SOME RANGE OF THE ARRAY.                                     MEA00040
CCC                                                                     MEA00050
CCC      ++ INPUT ++                                                    MEA00060
CCC                                                                     MEA00070
CCC   N1 = STARTING INDEX OF RANGE                                      MEA00080
CCC                                                                     MEA00090
CCC   N2 = ENDING INDEX OF RANGE                                        MEA00100
CCC                                                                     MEA00110
CCC      ++ COMMON ++                                                   MEA00120
CCC                                                                     MEA00130
CCC   RAN = COMPLEX ARRAY CONTAINING DATA                               MEA00140
CCC                                                                     MEA00150
CCC      ++ OUTPUT ++                                                   MEA00160
CCC                                                                     MEA00170
CCC   MEAN1 = MEAN OF REAL PART                                         MEA00180
CCC                                                                     MEA00190
CCC   MEAN2 = MEAN OF IMAGINARY PART                                    MEA00200
CCC                                                                     MEA00210
CCC   VAR1 = VARIANCE OF REAL PART                                      MEA00220
CCC                                                                     MEA00230
CCC   VAR2 = VARIANCE OF IMAGINARY PART                                 MEA00240
C                                                                       MEA00250
      COMPLEX RAN                                                       MEA00260
      REAL MEAN1,MEAN2                                                  MEA00270
C          THIS IS A COMPLEX NUMBER  EACH NUMBER TAKES TWO WORDS        MEA00280
      COMMON /M05/RAN(2048)                                             MEA00290
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUMEA00300
      MEAN1=0.                                                          MEA00310
      MEAN2=0.                                                          MEA00320
      VAR1=0.                                                           MEA00330
      VAR2=0.                                                           MEA00340
      DO 100 I=N1,N2                                                    MEA00350
      MEAN1=REAL(RAN(I))+MEAN1                                          MEA00360
      MEAN2=AIMAG(RAN(I))+MEAN2                                         MEA00370
      VAR1=(REAL(RAN(I)))**2+VAR1                                       MEA00380
      VAR2=(AIMAG(RAN(I)))**2+VAR2                                      MEA00390
  100 CONTINUE                                                          MEA00400
      MEAN1=MEAN1/FLOAT(N2-N1)                                          MEA00410
      MEAN2=MEAN2/FLOAT(N2-N1)                                          MEA00420
      VAR1=VAR1/FLOAT(N2-N1)                                            MEA00430
      VAR2=VAR2/FLOAT(N2-N1)                                            MEA00440
      WRITE (IOOUT,200) MEAN1,MEAN2                                     MEA00450
      WRITE (IOOUT,300) VAR1,VAR2                                       MEA00460
      RETURN                                                            MEA00470
C                                                                       MEA00480
  200 FORMAT (35X,20H MEAN OF REAL PART =,E12.5,10H, MEAN OF            MEA00490
     1        11HIMAG PART =,E12.5)                                     MEA00500
  300 FORMAT (35X,20H VAR. OF REAL PART =,E12.5,10H, VAR. OF            MEA00510
     1        11HIMAG PART =,E12.5)                                     MEA00520
      END                                                               MEA00530
```

```
      SUBROUTINE SPECT(F,D2,N,F1,LAMB)                                  SPE00010
C   THIS ROUTINE GENERATES THE APPROXIMATE FUNCTION                     SPE00020
C     G<ALPHA> (F/F<SUB 0,I>)                                           SPE00030
C   USED IN THE POWER SPECTRUM OF ANGLE-OF-ARRIVAL EQUATION             SPE00040
      REAL LAMB                                                         SPE00050
      COMMON /MG1/FR(1025),CN(20),V1(20),FO(20),RO(20)                  SPE00060
      FACT=1.32E-2*(LAMB/D2)**2                                         SPE00070
      F1=0.                                                             SPE00080
      DO 100 I=1,N                                                      SPE00090
      IF (F.LE..332*FO(I)) G=1.                                         SPE00100
      IF (F.GT..332*FO(I)) G=1.12-.361*F/FO(I)                          SPE00110
      IF (F.GE.3.10*FO(I)) G=0.                                         SPE00120
      F1=F1+FACT*((D2/RO(I))**5/(F*F*FO(I)))**.33333*G                  SPE00130
  100 CONTINUE                                                          SPE00140
      RETURN                                                            SPE00150
      END                                                               SPE00160
```

```
      SUBROUTINE SPREAD(DIAM,ROT,WAVE,THET,RANG,DRO,THETDL,DDL,THET12,   SPR00010
     1                  D12,DTHET,DTOT)                                 SPR00020
C                                                                       SPR00030
CC  COMPUTATION OF BEAM SPREAD ANGLE DUE TO DIFFRACTION AND DIFFRACTION SPR00040
CC  AND TURBULENCE AND SPOT DIAMETER ON TARGET (OR SEEKER).             SPR00050
C                                                                       SPR00060
      DRO=DIAM/ROT                                                      SPR00070
C   DO 1/2 SIMPSON RULE INTEGRATION                                     SPR00080
      VARX=0.0                                                          SPR00090
      DELTAX=0.01                                                       SPR00100
      RDRO=DESUB(VARX,DRO)/2.0                                          SPR00110
      DO 100 I=1,100                                                    SPR00120
      VARX=VARX+DELTAX                                                  SPR00130
      RDRO=RDRO+DESUB(VARX,DRO)                                         SPR00140
  100 CONTINUE                                                          SPR00150
      RDRO=(RDRO-DESUB(VARX,DRO)/2.0)*DELTAX                            SPR00160
      RDRO=1.0/(SQRT(5.092958*(DRO)**2*RDRO))                           SPR00170
      THETDL=1.128*WAVE/DIAM                                            SPR00180
      DDL=THETDL*RANG                                                   SPR00190
      THET12=THETDL*DRO*RDRO                                            SPR00200
      D12=THET12*RANG                                                   SPR00210
      DTHET=SQRT(THET12**2+THET**2)                                     SPR00220
      DTOT=DIAM+DTHET*RANG                                              SPR00230
      RETURN                                                            SPR00240
      END                                                               SPR00250
```

```
      SUBROUTINE THETO(THETAO,FALPH,CN2,D1V,R1V,N2)                      THE00010
C  ++ CALLED FUNCTIONS ++                                                THE00020
C     FALPH                                                              THE00030
      DIMENSION CN2(20)                                                  THE00040
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUTHE00050
      DEL1V=R1V/FLOAT(N2)                                                THE00060
C     CALCULATE D1INF                                                    THE00070
      D1INF=0.                                                           THE00080
      S1=DEL1V/2.                                                        THE00090
      S=0.                                                               THE00100
      D1V3=(D1V)*(-.3333)                                                THE00110
      DO 100 I=1,N2                                                      THE00120
      S=S+S1                                                             THE00130
      D1INF=DEL1V*CN2(I)*((S/R1V)**1.6667)+D1INF                         THE00140
      S1=DEL1V                                                           THE00150
  100 CONTINUE                                                           THE00160
      D1INF=0.5*11.97*D1V3*D1INF                                         THE00170
C  INITIAL ESTIMATE FOR THETAO                                           THE00180
C     WRITE (IOOUT,500) D1INF                                            THE00190
      THETAO=1.E-4                                                       THE00200
  200 XI1=0.                                                             THE00210
      D1THE=0.                                                           THE00220
      S=0.                                                               THE00230
      S1=DEL1V/2.                                                        THE00240
      DO 300 I=1,N2                                                      THE00250
      S=S+S1                                                             THE00260
      XI1=THETAO*(R1V-S)/D1V                                             THE00270
      D1THE=DEL1V*CN2(I)*((S/R1V)**1.6667)*FALPH(XI1)+D1THE              THE00280
      S1=DEL1V                                                           THE00290
  300 CONTINUE                                                           THE00300
      D1THE=D1THE*D1V3                                                   THE00310
      IF (ABS((D1INF-D1THE)/D1INF).LT..001) GO TO 400                    THE00320
      THETAO=THETAO*(1.+.5*(D1INF-D1THE)/D1INF)                          THE00330
C     WRITE (IOOUT,500) THETAO,D1THE                                     THE00340
      GO TO 200                                                          THE00350
  400 CONTINUE                                                           THE00360
C     WRITE (IOOUT,500) D1THE                                            THE00370
      RETURN                                                             THE00380
C                                                                        THE00390
  500 FORMAT (2E16.8)                                                    THE00400
      END                                                                THE00410
```

```
      SUBROUTINE BASCAT(WAVE,EXCO,IERR)                                 BAS000
C*****************************************************************    BAS000
C                                                                       BAS000
C     THIS VERSION OF BASCAT (20 SEP 81) DIFFERS FROM THE EOSAEL 80     BAS000
C     VERSION IN ITS INTERNAL PROGRAM STRUCTURE AND OUTPUT CAPABILITY.  BAS000
C     INPUT FORMATTING HAS NOT BEEN CHANGED. BRIEFLY, THE INTERNAL      BAS000
C     STRUCTURAL CHANGES CONSIST OF THE FOLLOWING :                     BAS000
C                                                                       BAS000
C      (A) SUBROUTINE THIT HAS BEEN ELIMINATED. THE FUNCTIONS WHICH     BAS000
C          IT ONCE PERFORMED HAVE BEEN CONSOLIDATED INTO SUBROUTINE     BAS000
C          START.                                                       BAS000
C      (B) A NEW LIDAR BIASING ALGORITHM HAS BEEN INSERTED.             BAS000
C      (C) DIRECT BEAM (I.E., UNSCATTERED) COMPUTATIONS IN              BAS000
C          SUBROUTINE START HAVE BEEN REVISED.                          BAS000
C      (D) SUBROUTINE CONV HAS BEEN MODIFIED SO THAT THE DIFFER-        BAS000
C          ENCE OF TWO NUMBERS RETAINS MORE SIGNIFICANT DIGITS.         BAS000
C      (E) NUMERICAL CHECKS FOR IMPROPER ARGUMENTS OF FUNCTIONS         BAS000
C          (DIVISIONS, SQUARE ROOTS, LOGARITHMS, ETC.) HAVE BEEN        BAS000
C          REVISED.                                                     BAS000
C      (F) ARGUMENT LISTS OF A FEW COMMON BLOCKS HAVE BEEN CHANGED.     BAS000
C      (G) WRITE STATEMENTS FOR OUTPUTTING BASCAT RESULTS TO A USER-    BAS000
C          DEFINED PLOT FILE (NPLOTU) HAVE BEEN INCLUDED. THESE STATE-  BAS000
C          MENTS MUST BE UNCOMMENTED IN ORDER TO ACTIVATE THEM.         BAS000
C*****************************************************************    BAS000
C                                                                       BAS000
C     THE BASCAT MODULE ALONE USES THE FOLLOWING SUBROUTINES:           BAS000
C      BKWD - CONTAINS BACKWARD SCATTERING ALGORITHM                    BAS000
C      CONV - CONVOLVES IMPULSE RESPONSE WITH SQUARE PULSE              BAS000
C      ELM - DETERMINES BIASING DISTANCES                               BAS000
C      FIND - DETERMINES INTERPOLATED PHASE FUNCTION VALUE              BAS000
C      FWRD - FIRST ORDER SCATTERING ALGORITHM                          BAS000
C      GAS - DETERMINES MONTE CARLO SCATTERING ANGLES FOR TRAVERSES     BAS000
C      GMAX - DETERMINES MAXIMUM OF AN INPUT ARRAY                      BAS000
C      MATRX - GENERATES ROTATION MATRICES                              BAS000
C      ROTAT - ROTATES VECTORS FROM ONE COORDINATE SYSTEM TO ANOTHER     BAS000
C      SMOOZ - DETERMINES START OF TRAILING ZEROS IN INPUT ARRAY         BAS000
C      START - INITIATES PHOTON TRAJECTORIES                            BAS000
C      TRAVRS - MOVES PHOTONS BETWEEN SCATTERING POINTS AND FINDS OB-   BAS000
C               SERVED POWER CONTRIBUTIONS AT THOSE POINTS              BAS000
C      USCA - SELECTS RANDOM ANGLES WEIGHTED BY PHASE FUNCTION          BAS000
C                                                                       BAS000
C     TWO SUBROUTINES SHARED BY BASCAT WITH OTHER EOSAEL 80 MODULES     BAS000
C     ARE THE FOLLOWING :                                               BAS000
C      PFUNC - SELECTS AND RENORMALIZES PHASE FUNCTION DATA FROM        BAS000
C              EOSAEL 80 DATA BASE                                      BAS000
C      RAND - RANDOM NUMBER GENERATOR (GENERATES UNIFORM DISTRIBUTION   BAS000
C             OF RANDOM NUMBERS BETWEEN 0 AND 1)                        BAS000
C                                                                       BAS000
C** NOTE** THE FOLLOWING ROUTINES UTILIZE THE RANDOM NUMBER GENERATOR   BAS000
C          WHICH IS INVOKED AS FUNCTION 'RAND(SEED)' :                  BAS000
C                                                                       BAS000
C           (A) BKWD - 3 OCCURRENCES OF FUNCTION RAND                   BAS000
C           (B) FWRD - 2 OCCURRENCES                                    BAS000
C           (C) GAS - 3 OCCURRENCES                                     BAS000
C           (D) START - 2 OCCURRENCES                                   BAS000
C           (E) TRAVRS - 2 OCCURRENCES                                  BAS000
C                                                                       BAS000
C          USERS OF OTHER (NON-HP) COMPUTER SYSTEMS MUST REPLACE        BAS000
C          FUNCTION RAND(SEED) WITH A UNIFORM RANDOM NUMBER GENERATOR   BAS000
C          WHICH WORKS FOR THEIR SYSTEMS. THE RANDOM NUMBER SEED IS     BAS000
C          INITIALIZED IN SUBROUTINE BASCAT. THIS AND ALL SUBSEQUENT    BAS000
C          VALUES OF THE RANDOM NUMBER SEED ARE PASSED VIA COMMON       BAS000
C          BLOCK 'RNDM'. SHOULD THE RANDOM NUMBER SEED USED HERE        BAS000
C          (735.34829) BE INAPPROPRIATE FOR THE USER'S SYSTEM, IT IS    BAS000
C          SUGGESTED THAT THE INITIALIZATION USED BELOW (SEED0=         BAS000
C          735.34829) BE CHANGED TO AN APPROPRIATE VALUE.               BAS000
C                                                                       BAS000
C     SUBSTANTIAL MODIFICATIONS HAVE BEEN MADE TO THE FOLLOWING         BAS000
C     SUBROUTINES PRESENT IN EOSAEL 80 :                                BAS000
C      CONV                                                             BAS000
```

```
C       TRAVRS                                                                  BAS00710
C       START                                                                   BAS00720
C                                                                               BAS00730
C***************************************************************************BAS00740
      COMMON/RNDM/SEED                                                          BAS00750
      COMMON /GEOMET/PTS(15),IGEOSW                                             BAS00760
      COMMON/ALL/AT,BT,CT,BLIM                                                  BAS00770
      COMMON/IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU BAS00780
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                             BAS00790
      COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,     BAS00800
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                                BAS00810
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),       BAS00820
     +BE(16),SINGWV,PF(65),LLMAX                                                BAS00830
      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),     BAS00840
     *ALB(2),ZG,DMAX                                                            BAS00850
      COMMON/FGEL/XA(3),D,NSCA                                                  BAS00860
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)                      BAS00870
      COMMON/CONB/X(100),Y(100)                                                 BAS00880
      COMMON/HIT/UDS,THSP,RS(3,3),XS(3),DSA,XV(3)                               BAS00890
      DIMENSION NM(10),TPU(7),SS(3)                                             BAS00900
      DIMENSION IAL(7),DAT(7),IOR(10)                                           BAS00910
      DATA IAL/2HPA  ,2HSO  ,2HDE  ,2HCL  ,2HGR  ,2HPU  ,2HGO  /               BAS00920
      DATA IZERO/0/                                                             BAS00930
C*****************************************************************           BAS00940
C     THIS SUBROUTINE CALCULATES STEADY STATE AND TIME-DEPENDENT DIRECT  BAS00950
C     AND MULTIPLY SCATTERED POWER INTO A DETECTOR BY AN ELLIPSOIDAL     BAS00960
C     AEROSOL CLOUD WITH GROUND PLANE, FOR A LASER SOURCE.  THE DETECTOR BAS00970
C     AND SOURCE MAY HAVE ANY LOCATIONS, LOOK ANGLES, AND CONE OF VIEW/  BAS00980
C     BEAM SPREAD/WAVELENGTH.  THE AEROSOL CLOUD MAY HAVE ANY ORIENTA-   BAS00990
C     TION, SIZE, AND SCATTERING PHASE FUNCTION (ARBITRARY NORMALIZA-     BAS01000
C     TION), IN A COORDINATE SYSTEM WITH ORIGIN AT THE CLOUD CENTER,      BAS01010
C     WITH Z-AXIS VERTICAL, X-AXIS EAST, AND Y-AXIS NORTH.  THE GROUND    BAS01020
C     PLANE, ASSUMED AN ISOTROPIC REFLECTOR, MAY HAVE ANY ALBEDO, AND     BAS01030
C     MAY OR MAY NOT INTERSECT THE AEROSOL CLOUD.                         BAS01040
C*****************************************************************           BAS01050
C                                                                               BAS01060
C** INPUT DATA CARDS ARE READ IN AN ORDER-INDEPENDENT MANNER, WITH            BAS01070
C** A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH RECORD. DATA              BAS01080
C** ON EACH CARD IS READ IN UNDER THE FOLLOWING FORMAT :                      BAS01090
C** (A4,1X,7(E9.4,1X)). NOTE THAT INTEGER VARIABLES IN THE PROGRAM           BAS01100
C** MUST BE INPUT AS REAL NUMBERS IN THIS INPUT SCHEME ... THEY ARE          BAS01110
C** LATER FIXED TO THE INTEGER TYPE.                                         BAS01120
C                                                                               BAS01130
C-----------------------------------------------------------------            BAS01140
C     CARD IDENTIFIER : PART                                                    BAS01150
C     VARIABLES READ : N1,N2,ITIME                                              BAS01160
C      N1=NUMBER OF PARTIAL OUTPUTS DESIRED, FOR A GIVEN RUN                    BAS01170
C      N2=NUMBER OF PHOTONS TO BE USED FOR EACH PARTIAL CALCULATION             BAS01180
C                                                                               BAS01190
C** NOTE ** FOR CERTAIN DETECTOR CONDITIONS, AND WITH NORMAL BIASING,      BAS01200
C           AS MANY AS 100,000 PHOTONS MAY BE NEEDED TO OVERCOME LARGE      BAS01210
C           STATISTICAL FLUCTUATIONS IN FIRST ORDER SCATTERING RETURNS.     BAS01220
C           SUCH CONDITIONS ARE DEFINED BY THE FOLLOWING CHARACTERISTICS    BAS01230
C       (A) THE DETECTOR IS IN A MONOSTATIC LIDAR CONFIGURATION.            BAS01240
C       (B) THE DETECTOR IS WITHIN 10 METERS OF THE CLOUD (OR IS INSIDE     BAS01250
C           OF IT).                                                         BAS01260
C       (C) THE CLOUD IS NOT OPTICALLY THICK ALONG THE LOOK DIRECTION       BAS01270
C           (OPTICAL DEPTHS LESS THAN 10).                                  BAS01280
C*** IN ORDER TO ATTAIN MORE RAPID CONVERGENCE OF FIRST ORDER RETURN        BAS01290
C    POWER UNDER THE ABOVE CONDITIONS, A DIFFERENT BIASING SCHEME IS        BAS01300
C    USED. THE SPECIFIC CONDITIONS WHICH TRIGGER THIS ALTERNATE MODE        BAS01310
C    ARE THE FOLLOWING :                                                    BAS01320
C       (A) THE DOT PRODUCT OF THE SOURCE APERTURE SURFACE NORMAL AND       BAS01330
C           THE DETECTOR APERTURE SURFACE NORMAL IS GREATER THAN 0.99.      BAS01340
C       (B) THE DISTANCE OF THE DETECTOR APERTURE FROM THE NEAREST          BAS01350
C           CLOUD SURFACE (AS SEEN ALONG THE DETECTOR NORMAL) IS LESS       BAS01360
C           THAN OR EQUAL TO 10 METERS (0.01 KM).                           BAS01370
C       (C) THE SEPARATION OF SOURCE AND DETECTOR APERTURE CENTERS IS       BAS01380
C           LESS THAN OR EQUAL TO 10 TIMES THE DETECTOR APERTURE            BAS01390
C           RADIUS.                                                         BAS01400
```

```
C*** THE MAIN DIFFERNCES WHICH WILL BE OBSERVED BETWEEN THE NORMAL AND  BAS01410
C    ALTERNATE BIASING MODES ARE THE FOLLOWING :                       BAS01420
C       (A) THE FIRST ORDER RETURN POWER IN THE EARLIEST TIME BOX WILL  BAS01430
C           BE EXTREMELY STABLE IN THE ALTERNATE MODE.                  BAS01440
C       (B) ONLY THE EARLIEST FIRST ORDER SCATTERING RESULTS WILL BE    BAS01450
C           STRONGLY AFFECTED.                                          BAS01460
C       (C) LATER FIRST ORDER RETURNS AND ALL HIGHER ORDER RETURNS WILL BAS01470
C           HAVE NEGLIGIBLY POORER CONVERGENCE.                         BAS01480
C*** IN ORDER TO OBSERVE CONVERGENCE OF THE FIRST ORDER STEADY-STATE    BAS01490
C    RETURN POWER TOWARD A STABLE VALUE, IT IS SUGGESTED THAT THE USER  BAS01500
C    UTILIZE THE PARTIAL OUTPUT OPTION. AS AN EXAMPLE, IF 10,000        BAS01510
C    PHOTONS ARE REQUIRED, SET N1=10 AND N2=1,000. THIS SELECTION WILL  BAS01520
C    RUN 10 X 1,000 = 10,000 PHOTONS AND WILL OUTPUT RETURN POWER       BAS01530
C    RESULTS AFTER EACH BATCH OF 1,000 PHOTONS.                         BAS01540
C                                                                       BAS01550
C      ITIME=OVERALL RUN NUMBER FOR THIS SET OF PARAMETERS. FOR EXAMPLE,BAS01560
C            ITIME=1 MEANS THE FIRST RUN, ITIME=3 MEANS THAT THE RESULTSBAS01570
C            OF THE TWO PREVIOUS RUNS WILL BE COMBINED WITH THIS 3RD RUNBAS01580
C-----------------------------------------------------------------------BAS01590
C     CARD IDENTIFIER : SORC                                            BAS01600
C     VARIABLES READ : XS(1),XS(2),XS(3),THES,PHIS,ASMM                 BAS01610
C      (XS(K),K=1,3)=SOURCE XYZ COORDINATES(KM)                         BAS01620
C      THES,PHIS=(POLAR,AZIMUTHAL)ANGLES(DEG) OF SOURCE BEAM AXIS       BAS01630
C      ASMM=RADIUS OF SOURCE APERTURE(MM).  THE SOURCE BEAM SPREAD ANGLEBAS01640
C           'THSP' IS TAKEN BY THE PROGRAM AS THE DIFFRACTION LIMIT FOR BAS01650
C           THIS APERTURE.  IF YOU SET ASMM=0., THE PROGRAM PUTS THSP=0.BAS01660
C      IF YOUR SOURCE PHOTONS WOULD NOT INTERSECT THE CLOUD OR THE      BAS01670
C      GROUND, THE SUBROUTINE NOTIFIES YOU, AND RETURNS.                BAS01680
C-----------------------------------------------------------------------BAS01690
C     CARD IDENTIFIER : DETR                                            BAS01700
C     VARIABLES READ : XD(1),XD(2),XD(3),THED,PHID,THEV,ACM             BAS01710
C      (XD(K),K=1,3)=DETECTOR XYZ COORDINATES(KM)                       BAS01720
C      THED,PHID=(POLAR,AZIMUTHAL)ANGLES(DEG) OF DETECTOR LOOK AXIS     BAS01730
C      THEV=DETECTOR CONE OF VIEW HALF-ANGLE(DEG)                       BAS01740
C      ACM=RADIUS OF DETECTOR DISK(CM)                                  BAS01750
C     IF YOUR DETECTOR POINTS SKYWARD, OR IF NEITHER YOUR SOURCE NOR    BAS01760
C     YOUR DETECTOR LOOK INTO THE CLOUD, THE SUBROUTINE NOTIFIES YOU,   BAS01770
C     AND RETURNS.                                                      BAS01780
C-----------------------------------------------------------------------BAS01790
C     CARD IDENTIFIER : CLDS                                            BAS01800
C     VARIABLES READ : A(1),A(2),A(3),THE,PHE,PSE,ISO                   BAS01810
C      (A(K),K=1,3)=ELLIPSOIDAL CLOUD PRINCIPAL HALF-AXES(KM)           BAS01820
C      THE,PHE,PSE=ELLIPSOID EULER ANGLES(DEG), WHERE 'PHE' IS THE FIRSTBAS01830
C        ROTATION, ABOUT THE Z-AXIS, 'THE' IS THE NEXT, ABOUT THE NEW   BAS01840
C        Y-AXIS, 'PSE' IS THE LAST, ABOUT THE NEW Z-AXIS.               BAS01850
C      ISO=AEROSOL IDENTIFICATION NUMBER, TO COMPARE WITH THE 'ID'      BAS01860
C          PARAMETER.                                                   BAS01870
C-----------------------------------------------------------------------BAS01880
C     CARD IDENTIFIER : GRND                                            BAS01890
C     VARIABLES READ : ZG,ALBG                                          BAS01900
C      ZG=Z-COORDINATE OF GROUND PLANE(KM)                              BAS01910
C      ALBG=GROUND PLANE REFLECTIVITY 0.<ALBG<1.                        BAS01920
C     IF YOUR ZG IS SUCH THAT THE GROUND PLANE IS ENTIRELY ABOVE THE     BAS01930
C     CLOUD, THE SUBROUTINE INFORMS YOU, AND RETURNS.  IF YOUR ZG IS    BAS01940
C     NEGATIVE, AND SO LARGE THAT NO GROUND REFLECTIONS WILL RETURN TO   BAS01950
C     THE DETECTOR WITHIN THE TIME LIMIT SET BY THE SUBROUTINE, THE SUB-BAS01960
C     ROUTINE SETS ALBG=0.  IF YOUR SOURCE IS BELOW THE GROUND PLANE,    BAS01970
C     THE SUBROUTINE PUTS THE SOURCE ON THE GROUND PLANE, AT YOUR XY     BAS01980
C     COORDINATES, AND NOTIFIES YOU.                                    BAS01990
C-----------------------------------------------------------------------BAS02000
C     CARD IDENTIFIER : PULS                                            BAS02010
C     VARIABLES READ : TPU(1),TPU(2),...,TPU(7)                         BAS02020
C      (TPU(J),J=1,7)=SOURCE PULSE DURATIONS(USEC)                      BAS02030
C       YOU CAN INPUT AS MANY AS SIX DIFFERENT PULSE LENGTHS.  THE LAST BAS02040
C       ENTRY MUST BE BLANK(ZERO).  THE SUBROUTINE CONVOLUTES THE MONTE BAS02050
C       CARLO PROBABILITY PER UNIT TIME DATA WITH EACH OF THESE SQUARE  BAS02060
C       PULSES, AND, FOR EACH PULSE, WRITES THE TIME-DEPENDENT POWER TO BAS02070
C       THE DETECTOR, FOR UNIT SOURCE PULSE POWER, FOR EACH SIGNIFICANT BAS02080
C       ORDER OF MULTIPLE SCATTERING, AND FOR THE TOTAL OF ALL ORDERS.  BAS02090
C-----------------------------------------------------------------------BAS02100
```

```
C      CARD IDENTIFIER : GO                                                    BAS02110
C      VARIABLES READ : NONE                                                   BAS02120
C       RUN TERMINATION CARD (MUST BE LAST CARD READ).                         BAS02130
C-------------------------------------------------------------------------------BAS02140
       IERR=0                                                                  BAS02150
C                                                                              BAS02160
C***   THE FOLLOWING STATEMENT MAY NEED TO BE CHANGED OR ELIMINATED            BAS02170
C***   FOR USE ON NON-HP1000 COMPUTERS.                                        BAS02180
C                                                                              BAS02190
       SEED0=735.34829                                                         BAS02200
       NDIM=100                                                                BAS02210
       ND1=NDIM-1                                                              BAS02220
       DTAUM=0.2                                                               BAS02230
       GAMMA=EXCO                                                              BAS02240
       SINGWV=WAVE                                                             BAS02250
C                                                                              BAS02260
C** INPUT DATA INITIALIZATIONS.                                                BAS02270
C                                                                              BAS02280
       IF(IZERO.NE.0) GO TO 600                                                BAS02290
       N1=0                                                                    BAS02300
       N2=0                                                                    BAS02310
       ITIME=0                                                                 BAS02320
       ISO=0                                                                   BAS02330
       DO 599 LL=1,3                                                           BAS02340
       XS(LL)=0.                                                               BAS02350
       XD(LL)=0.                                                               BAS02360
       A(LL)=0.                                                                BAS02370
       TPU(LL)=0.                                                              BAS02380
  599  TPU(LL+3)=0.                                                            BAS02390
       TPU(7)=0.                                                               BAS02400
       THES=0.                                                                 BAS02410
       PHIS=0.                                                                 BAS02420
       ASMM=0.                                                                 BAS02430
       THED=0.                                                                 BAS02440
       PHID=0.                                                                 BAS02450
       THEV=0.                                                                 BAS02460
       ACM=0.                                                                  BAS02470
       THE=0.                                                                  BAS02480
       PHE=0.                                                                  BAS02490
       PSE=0.                                                                  BAS02500
       ZG=0.                                                                   BAS02510
       ALBG=0.                                                                 BAS02520
       LLMAX=65                                                                BAS02530
       IZERO=1                                                                 BAS02540
  600  CONTINUE                                                                BAS02550
C                                                                              BAS02560
C***   READ BASCAT DATA SET RECORDS UNDER CARD-INDEPENDENT FORMAT              BAS02570
C                                                                              BAS02580
       DO 700 K=1,7                                                            BAS02590
       READ(IOIN,610) IA,IA1,(DAT(I),I=1,7)                                    BAS02600
  610  FORMAT(2A2,1X,7(E9.4,1X))                                               BAS02610
       DO 615 JJ=1,8                                                           BAS02620
       IF(IA.NE.IAL(JJ)) GO TO 615                                             BAS02630
       INOPT=JJ                                                                BAS02640
       IF(INOPT.EQ.7) GO TO 701                                                BAS02650
       GO TO 620                                                               BAS02660
  615  CONTINUE                                                                BAS02670
       IF((K.EQ.7).AND.(JJ.EQ.8)) GO TO 697                                    BAS02680
       GO TO 695                                                               BAS02690
  620  CONTINUE                                                                BAS02700
       GO TO (621,622,623,624,625,626),INOPT                                   BAS02710
  621  N1=IFIX(DAT(1))                                                         BAS02720
       N2=IFIX(DAT(2))                                                         BAS02730
       ITIME=IFIX(DAT(3))                                                      BAS02740
       GO TO 700                                                               BAS02750
  622  XS(1)=DAT(1)                                                            BAS02760
       XS(2)=DAT(2)                                                            BAS02770
       XS(3)=DAT(3)                                                            BAS02780
       THES=DAT(4)                                                             BAS02790
       PHIS=DAT(5)                                                             BAS02800
```

```
      ASMM=DAT(6)                                                       BAS02810
      GO TO 700                                                         BAS02820
  623 XD(1)=DAT(1)                                                      BAS02830
      XD(2)=DAT(2)                                                      BAS02840
      XD(3)=DAT(3)                                                      BAS02850
      THED=DAT(4)                                                       BAS02860
      PHID=DAT(5)                                                       BAS02870
      THEV=DAT(6)                                                       BAS02880
      ACM=DAT(7)                                                        BAS02890
      GO TO 700                                                         BAS02900
  624 A(1)=DAT(1)                                                       BAS02910
      A(2)=DAT(2)                                                       BAS02920
      A(3)=DAT(3)                                                       BAS02930
      THE=DAT(4)                                                        BAS02940
      PHE=DAT(5)                                                        BAS02950
      PSE=DAT(6)                                                        BAS02960
      ISO=IFIX(DAT(7))                                                  BAS02970
      GO TO 700                                                         BAS02980
  625 ZG=DAT(1)                                                         BAS02990
      ALBG=DAT(2)                                                       BAS03000
      GO TO 700                                                         BAS03010
  626 CONTINUE                                                          BAS03020
      DO 627 NN=1,7                                                     BAS03030
  627 TPU(NN)=DAT(NN)                                                   BAS03040
      GO TO 700                                                         BAS03050
C                                                                       BAS03060
C***  ERROR RETURNS                                                     BAS03070
C                                                                       BAS03080
  695 CONTINUE                                                          BAS03090
      WRITE(IOOUT,696)                                                  BAS03100
  696 FORMAT(1H0,20X,89H***BASCAT ERROR*** INPUT CARD DETECTED WHICH DOEBAS03110
     +S NOT MATCH ANY CORPECT INPUT IDENTIFIERS /)                      BAS03120
      IERR=1                                                            BAS03130
      GO TO 777                                                         BAS03140
  697 CONTINUE                                                          BAS03150
      WRITE(IOOUT,698)                                                  BAS03160
  698 FORMAT(1H0,20X,66H***BASCAT ERROR*** TOO MANY INPUT CARDS OR GO SEBAS03170
     +NTINEL NOT PRESENT /)                                             BAS03180
  700 CONTINUE                                                          BAS03190
  701 CONTINUE                                                          BAS03200
C                                                                       BAS03210
C***  GEOMETRICAL OPTION DATA TRANSFER                                  BAS03220
C                                                                       BAS03230
      IF(IGEOSW.NE.1) GO TO 111                                         BAS03240
      DO 110 I=1,3                                                      BAS03250
      XS(I)=PTS(I+6)-PTS(I+12)                                          BAS03260
  110 XD(I)=PTS(I+3)-PTS(I+12)                                          BAS03270
  111 CONTINUE                                                          BAS03280
C                                                                       BAS03290
C***  GENERATE INTERPOLATED, RENORMALIZED PHASE FUNCTION                BAS03300
C                                                                       BAS03310
      CALL PFUNC(ISO)                                                   BAS03320
      IF(GAMMA.EQ.0.0) GAMMA=BE(1)                                      BAS03330
      ALBEDO=ALBED(1)                                                   BAS03340
      REWIND IPHFUN                                                     BAS03350
      ALB(1)=ALBEDO                                                     BAS03360
      ALB(2)=ALBG                                                       BAS03370
C                                                                       BAS03380
C***  DETERMINE POWER OF 2 (KMAX) CORRESPONDING TO NUMBER OF PHASE      BAS03390
C***  FUNCTION VALUES PRESENT.                                          BAS03400
C                                                                       BAS03410
      LMAX=LLMAX                                                        BAS03420
      LMM1=LMAX-1                                                       BAS03430
      KMAX=IFIX(ALOG(FLOAT(LMM1))/0.693147)                             BAS03440
C                                                                       BAS03450
C***  CC = SPEED OF LIGHT (KM/MICROSECOND)                              BAS03460
C                                                                       BAS03470
      CC=0.3                                                            BAS03480
      THV=PIRAD*THEV                                                    BAS03490
      IF(THV.LE.1.E-30) THV=0.                                          BAS03500
```

```
      UV=COS(THV)                                                       BAS03510
C                                                                       BAS03520
C***  SET RANDOM NUMBER SEED                                            BAS03530
C                                                                       BAS03540
      SEED=(2*ITIME-1)*SEEDO                                            BAS03550
      AKM=ACM*1.E-5                                                     BAS03560
      AKSQ=AKM**2                                                       BAS03570
C                                                                       BAS03580
C***  SET MINIMUM DISTANCE ALIM (AND ITS SQUARE BLIM) SEPARATING        BAS03590
C***  THE DETECTOR DISK AND A PHOTON SCATTERING POINT.                  BAS03600
C                                                                       BAS03610
      ALIM=AKM*10.                                                      BAS03620
      BLIM=ALIM**2                                                      BAS03630
      FAC=THV*AKSQ/8.                                                   BAS03640
      BT=.61*WAVE*1.E-3                                                 BAS03650
      THSP=0.                                                           BAS03660
C                                                                       BAS03670
C***  DETERMINE DIFFRACTION-LIMITED SOURCE BEAMSPREAD                   BAS03680
C                                                                       BAS03690
      IF(ASMM.GT.0.)THSP=BT/ASMM                                        BAS03700
C                                                                       BAS03710
C***  GENERATE ROTATION MATRIX RE( ) FOR CONVERSION FROM STANDARD       BAS03720
C***  FRAME OF REFERENCE TO CLOUD FRAME OF REFERENCE                    BAS03730
C                                                                       BAS03740
      CALL MATRX(THE,PHE,SS,R)                                          BAS03750
      AT=PIRAD*PSE                                                      BAS03760
      IF(AT.LE.1.E-30) AT=0.                                            BAS03770
      DO 1 I=1,3                                                        BAS03780
      DO 1 J=1,3                                                        BAS03790
    1 RS(I,J)=0.                                                        BAS03800
      RS(1,1)=COS(AT)                                                   BAS03810
      RS(1,2)=SIN(AT)                                                   BAS03820
      RS(2,1)=-RS(1,2)                                                  BAS03830
      RS(2,2)=RS(1,1)                                                   BAS03840
      RS(3,3)=1.                                                        BAS03850
      DO 2 I=1,3                                                        BAS03860
      DO 2 J=1,3                                                        BAS03870
      RE(I,J)=0.                                                        BAS03880
      DO 2 K=1,3                                                        BAS03890
    2 RE(I,J)=RE(I,J)+RS(I,K)*R(K,J)                                    BAS03900
C                                                                       BAS03910
C***  GENERATE ROTATION MATRIX RS( ) FOR CONVERSION FROM SOURCE         BAS03920
C***  CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE            BAS03930
C                                                                       BAS03940
      CALL MATRX(THES,PHIS,SS,RS)                                       BAS03950
C                                                                       BAS03960
C***  ECHO INPUT PARAMETERS                                             BAS03970
C                                                                       BAS03980
      WRITE (IOOUT,6800)                                                BAS03990
      WRITE (IOOUT,4800)                                                BAS04000
      WRITE (IOOUT,4900)                                                BAS04010
      WRITE (IOOUT,5000)                                                BAS04020
      WRITE (IOOUT,4800)                                                BAS04030
      WRITE (IOOUT,4700)                                                BAS04040
      WRITE (IOOUT,5100)                                                BAS04050
      IF (ISO.EQ.0) WRITE (IOOUT,5200)                                  BAS04060
      IF (ISO.EQ.1) WRITE (IOOUT,5201)                                  BAS04070
      IF (ISO.EQ.2) WRITE (IOOUT,5202)                                  BAS04080
      IF (ISO.EQ.3) WRITE (IOOUT,5203)                                  BAS04090
      IF (ISO.EQ.4) WRITE (IOOUT,5204)                                  BAS04100
      IF (ISO.EQ.5) WRITE (IOOUT,5205)                                  BAS04110
      IF (ISO.EQ.6) WRITE (IOOUT,5206)                                  BAS04120
      IF (ISO.EQ.7) WRITE (IOOUT,5207)                                  BAS04130
      IF (ISO.EQ.8) WRITE (IOOUT,5208)                                  BAS04140
      IF (ISO.EQ.9) WRITE (IOOUT,5209)                                  BAS04150
      IF (ISO.EQ.10) WRITE (IOOUT,5210)                                 BAS04160
      IF (ISO.EQ.11) WRITE (IOOUT,5211)                                 BAS04170
      IF (ISO.EQ.12) WRITE (IOOUT,5212)                                 BAS04180
      WRITE (IOOUT,5600) WAVE,ALBEDO                                    BAS04190
      WRITE (IOOUT,5700) GAMMA                                          BAS04200
```

```
      WRITE(IOOUT,5601)                                                    BAS04210
      WRITE(IOOUT,5602)                                                    BAS04220
      WRITE(IOOUT,5603)                                                    BAS04230
      WRITE(IOOUT,5701)                                                    BAS04240
      WRITE(IOOUT,5702)(XS(K),K=1,3)                                       BAS04250
      WRITE(IOOUT,5704)THES                                                BAS04260
      WRITE(IOOUT,5705)PHIS                                                BAS04270
      WRITE(IOOUT,5703)ASMM                                                BAS04280
      WRITE(IOOUT,5706)THSP                                                BAS04290
      WRITE (IOOUT,5900)                                                   BAS04300
      WRITE (IOOUT,6000) THEV                                              BAS04310
      WRITE (IOOUT,6100) ACM                                               BAS04320
      WRITE(IOOUT,6201)(XD(K),K=1,3)                                       BAS04330
      WRITE (IOOUT,6400) THED                                              BAS04340
      WRITE (IOOUT,6500) PHID                                              BAS04350
      WRITE(IOOUT,6501)                                                    BAS04360
      WRITE(IOOUT,6502)ZG,ALBG                                             BAS04370
      DO 10 K=1,3                                                          BAS04380
   10 ASQ(K)=A(K)**2                                                       BAS04390
      DO 12 K=1,3                                                          BAS04400
   12 Y(K)=A(K)*2.*GAMMA                                                   BAS04410
C                                                                          BAS04420
C***  DETERMINE LARGEST OPTICAL DEPTH PRESENT IN AEROSOL CLOUD             BAS04430
C                                                                          BAS04440
      CALL GMAX(3,TAU)                                                     BAS04450
C                                                                          BAS04460
C***  SET TIME AND DISTANCE INCREMENTS AND LIMITS                          BAS04470
C                                                                          BAS04480
      NTMAX=50                                                             BAS04490
      D=TAU/GAMMA                                                          BAS04500
      DELD=5.*D/NTMAX                                                      BAS04510
      DELT=DELD/CC                                                         BAS04520
      DMAX=5.5*D                                                           BAS04530
C                                                                          BAS04540
C***  BEGIN CLOUD SUBBLOCK                                                 BAS04550
C                                                                          BAS04560
C##########################################################################BAS04570
C                                                                          BAS04580
C     THE FOLLOWING BLOCK OF WRITE STATEMENTS MUST BE UNCOMMENTED          BAS04590
C     IN ORDER TO OUTPUT DATA TO A USER-DEFINED PLOT FILE (NPLOTU).        BAS04600
C                                                                          BAS04610
C     THE OUTPUT QUANTITIES IN THIS BLOCK ARE THE FOLLOWING :              BAS04620
C       WAVE = WAVELENGTH (MICROMETERS)                                    BAS04630
C       ISO  = AEROSOL TYPE (VALID RANGE, 0-12)                            BAS04640
C       TAU  = OPTICAL DEPTH ALONG LONGEST AXIS OF CLOUD ELLIPSOID         BAS04650
C       N1   = NUMBER OF PARTIAL RUNS WITHIN THIS BASCAT RUN               BAS04660
C       XS(  )= SOURCE XYZ POSITION ARRAY (KILOMETERS)                     BAS04670
C       THES = SOURCE VECTOR POLAR ANGLE (DEGREES)                         BAS04680
C       PHIS = SOURCE VECTOR AZIMUTHAL ANGLE (DEGREES)                     BAS04690
C       ASMM = SOURCE APERTURE RADIUS (MILLIMETERS)                        BAS04700
C       THSP = HALF-ANGLE OF SOURCE DIFFRACTION CONE (RADIANS)             BAS04710
C       XD(  )= DETECTOR XYZ POSITION ARRAY (KILOMETERS)                   BAS04720
C       THED = DETECTOR VECTOR POLAR ANGLE (DEGREES)                       BAS04730
C       PHID = DETECTOR VECTOR AZIMUTHAL ANGLE (DEGREES)                   BAS04740
C       ACM  = DETECTOR APERTURE RADIUS (CENTIMETERS)                      BAS04750
C       THEV = HALF-ANGLE OF DETECTOR FIELD OF VIEW (DEGREES)              BAS04760
C       A(  ) = CLOUD ELLIPSOID PRINCIPAL HALF-AXIS ARRAY (KILOMETERS)     BAS04770
C       ALB(1)= SINGLE-SCATTERING ALBEDO OF CLOUD AEROSOL                  BAS04780
C       ALB(2)= ALBEDO OF GROUND PLANE                                     BAS04790
C                                                                          BAS04800
C     WRITE(NPLOTU,9111) WAVE,ISO,TAU,N1                                   BAS04810
C9111 FORMAT(E9.4,1X,I2,1X,E9.4,1X,2(I2,1X))                               BAS04820
C     WRITE(NPLOTU,9222)XS(1),XS(2),XS(3),THES,PHIS,ASMM,THSP              BAS04830
C     WRITE(NPLOTU,9222)XD(1),XD(2),XD(3),THED,PHID,ACM,THEV               BAS04840
C9222 FORMAT(12(E9.4,1X))                                                  BAS04850
C     WRITE(NPLOTU,9222) A(1),A(2),A(3),ALB(1),ALB(2)                      BAS04860
C##########################################################################BAS04870
      WRITE (IOOUT,6900)                                                   BAS04880
      WRITE (IOOUT,7000)(A(K),K=1,3)                                       BAS04890
      WRITE (IOOUT,7100)THE,PHE,PSE                                        BAS04900
```

```
      WRITE (IOOUT,7200)(Y(K),K=1,3)                                    BAS04910
C                                                                       BAS04920
C***  GENERATE ROTATION MATRIX R( ) FOR CONVERSION FROM DETECTOR        BAS04930
C***  CONE FRAME OF REFERENCE TO STANDARD FRAME OF REFERENCE            BAS04940
C                                                                       BAS04950
      CALL MATRX(THED,PHID,SD,R)                                        BAS04960
C                                                                       BAS04970
C***  SET LIMIT NSCAM ON HIGHEST SIGNIFICANT SCATTERING ORDER (NSCAM-1) BAS04980
C                                                                       BAS04990
      NSCAM=3.0*ALBEDO*TAU+2.0                                          BAS05000
      IF (NSCAM.GT.10) NSCAM=10                                         BAS05010
      IF(NSCAM.LT.3)NSCAM=3                                             BAS05020
      NSCA1=NSCAM-1                                                     BAS05030
      AT=0.                                                             BAS05040
      DO 4 K=1,3                                                        BAS05050
    4 AT=AT+RE(K,3)**2/ASQ(K)                                           BAS05060
      BT=1.-AT*ZG                                                       BAS05070
      IF(BT.LE.0.)GO TO 4002                                            BAS05080
      AT=ZG+D                                                           BAS05090
      IF(AT.GT.0.)GO TO 3                                               BAS05100
      ZG=-D                                                             BAS05110
      WRITE(IOOUT,9004)ZG                                               BAS05120
    3 CONTINUE                                                          BAS05130
      IF(XS(3).GE.ZG)GO TO 16                                           BAS05140
      XS(3)=ZG                                                          BAS05150
      WRITE(IOOUT,9003)XS(3)                                            BAS05160
   16 CONTINUE                                                          BAS05170
C                                                                       BAS05180
C***  DETERMINE DISTANCES ELD(1), ELD(2) FROM DETECTOR TO NEAREST AND   BAS05190
C***  FARTHEST CLOUD BOUNDARIES ALONG DETECTOR AXIS                     BAS05200
C                                                                       BAS05210
      CALL ELM(XD,SD,ELD)                                               BAS05220
C                                                                       BAS05230
C***  DETERMINE DISTANCES EL(1), EL(2) FROM SOURCE TO NEAREST AND       BAS05240
C***  FARTHEST CLOUD BOUNDARIES ALONG SOURCE AXIS                       BAS05250
C                                                                       BAS05260
      CALL ELM(XS,SS,EL)                                                BAS05270
C                                                                       BAS05280
C***  PERFORM GEOMETRICAL ERROR CHECKS                                  BAS05290
C                                                                       BAS05300
      IF((EL(2).LE.0.).AND.(ELD(2).LE.0.))GO TO 4003                    BAS05310
      IF((EL(2).GT.0.).AND.(ELD(2).GT.0.))GO TO 6                       BAS05320
      IF(EL(2).LE.0.)GO TO 7                                            BAS05330
      IF(ALB(2).LE.0.) GO TO 4004                                       BAS05340
      IF(SD(3).GE.0.)GO TO 4001                                         BAS05350
      AT=EL(1)                                                          BAS05360
      CT=(ZG-XD(3))/SD(3)                                               BAS05370
      GO TO 9                                                           BAS05380
    7 CONTINUE                                                          BAS05390
      IF(ALB(2).LE.0.) GO TO 4005                                       BAS05400
      IF(SS(3).GT.0.)GO TO 4000                                         BAS05410
      AT=(ZG-XS(3))/SS(3)                                               BAS05420
      GO TO 8                                                           BAS05430
    6 AT=EL(1)                                                          BAS05440
C                                                                       BAS05450
C***  DETERMINE MINIMUM POSSIBLE TRAVERSE DISTANCE ELMIN                BAS05460
C                                                                       BAS05470
    8 CT=ELD(1)                                                         BAS05480
    9 BT=0.                                                             BAS05490
      DO 11 K=1,3                                                       BAS05500
      X(K)=XS(K)+AT*SS(K)                                               BAS05510
      Y(K)=XD(K)+CT*SD(K)                                               BAS05520
   11 BT=BT+(X(K)-Y(K))**2                                              BAS05530
      IF(BT.LE.1.E-30) BT=0.                                            BAS05540
      ELMIN=SQRT(BT)+AT+CT                                              BAS05550
      IF(ELMIN.LT.ALIM) ELMIN=ALIM                                      BAS05560
C                                                                       BAS05570
C***  DETERMINE VIRTUAL SOURCE POINT XV( )                              BAS05580
C                                                                       BAS05590
      DSA=0.                                                            BAS05600
```

```
      IF(ASMM.GT.0.)DSA=ASMM*1.E-6/TAN(THSP)                            BAS05610
      DO 305 K=1,3                                                      BAS05620
  305 XV(K)=XS(K)-DSA*SS(K)                                             BAS05630
C                                                                       BAS05640
C***  DETERMINE WHETHER NEAR-CLOUD LIDAR BIASING IS NEEDED.             BAS05650
C                                                                       BAS05660
      DOTSD=0.                                                          BAS05670
      DOTXD=0.                                                          BAS05680
      DO 47 KL=1,3                                                      BAS05690
      DOTXD=DOTXD+(XD(KL)-XS(KL))**2                                    BAS05700
   47 DOTSD=DOTSD+SS(KL)*SD(KL)                                         BAS05710
      IF(DOTXD.GT.BLIM) DOTSD=0.                                        BAS05720
      REWIND KSTOR                                                      BAS05730
C                                                                       BAS05740
C***  START PHOTON LOOPS                                                BAS05750
C                                                                       BAS05760
      DO 1000 I1=1,NI                                                   BAS05770
C                                                                       BAS05780
C***  CHECK TO SEE IF THIS IS THE FIRST RUN WITH NO PRIOR RESULTS USED  BAS05790
C                                                                       BAS05800
      IF((I1.EQ.1).AND.(ITIME.EQ.1))GO TO 1303                          BAS05810
C                                                                       BAS05820
C***  RELOAD WITH RESULTS FROM PREVIOUS PARTIAL RUN                     BAS05830
C                                                                       BAS05840
      DO 1301 NSCA=1,NSCAM                                              BAS05850
      READ(KSTOR)ENC(NSCA)                                              BAS05860
      DO 1301 II=1,NTMAX                                                BAS05870
      READ(KSTOR)EN(NSCA,II)                                            BAS05880
 1301 CONTINUE                                                          BAS05890
      REWIND KSTOR                                                      BAS05900
      GO TO 1302                                                        BAS05910
 1303 CONTINUE                                                          BAS05920
C                                                                       BAS05930
C***  INITIALIZATION FOR FIRST RUN                                      BAS05940
C                                                                       BAS05950
      DO 1300 NSCA=1,NSCAM                                              BAS05960
      ENC(NSCA)=0.                                                      BAS05970
      DO 1300 NT=1,NTMAX                                                BAS05980
 1300 EN(NSCA,NT)=0.0                                                   BAS05990
 1302 CONTINUE                                                          BAS06000
C                                                                       BAS06010
C***  LOAD EXPECTED CUMULATIVE NUMBER OF PHOTONS                        BAS06020
C                                                                       BAS06030
      DO 1304 NS=1,NSCAM                                                BAS06040
 1304 ENC(NS)=ENC(NS)+N2                                                BAS06050
C                                                                       BAS06060
C***  DETERMINE VECTOR FROM VIRTUAL SOURCE POINT TO DETECTOR            BAS06070
C                                                                       BAS06080
      DO 555 K=1,3                                                      BAS06090
  555 Y(K)=XD(K)-XV(K)                                                  BAS06100
C                                                                       BAS06110
C***  START INNER PHOTON LOOP                                           BAS06120
C                                                                       BAS06130
      DO 900 I2=1,N2                                                    BAS06140
C                                                                       BAS06150
C***  INITIALIZE PHOTON DIRECTION AND CALCULATE DIRECT BEAM             BAS06160
C***  CONTRIBUTIONS (IF ANY)                                            BAS06170
C                                                                       BAS06180
      CALL START(SS)                                                    BAS06190
      IF(STH.GT.0.)GO TO 150                                            BAS06200
      GO TO 900                                                         BAS06210
  150 NSCA=1                                                            BAS06220
C                                                                       BAS06230
C***  START MULTIPLE SCATTERING LOOP                                    BAS06240
C                                                                       BAS06250
      I2FLG=0                                                           BAS06260
      ICOND=0                                                           BAS06270
 1700 NSCA=NSCA+1                                                       BAS06280
C                                                                       BAS06290
C***  NEAR-CLOUD LIDAR BIASING IS ACTIVE WHEN ICOND=1. 10 PERCENT OF ALLBAS06300
```

```
C**** FIRST ORDER PHOTONS USE AN ALTERNATE MODE OF BIASING IN THIS      BAS06310
C**** SITUATION. THE REMAINING 90 PERCENT ARE NORMALLY BIASED IN ALL    BAS06320
C**** ORDERS.                                                          BAS06330
C                                                                      BAS06340
      IF((NSCA.NE.2).OR.(DOTSD.LE.0.99).OR.(ELD(1).GT.0.01))GO TO 707  BAS06350
      ICOND=1                                                          BAS06360
      IF(FLOAT(I2).LE.(FLOAT(N2)/10.))I2FLG=1                          BAS06370
  707 CONTINUE                                                         BAS06380
C                                                                      BAS06390
C***  MOVE PHOTON TO NEXT EVENT POINT VIA BIASED TRAVERSE AND DETERMINE BAS06400
C***  POWER CONTRIBUTIONS                                              BAS06410
C                                                                      BAS06420
      CALL TRAVRS(JTYPE,I2FLG,ICOND)                                   BAS06430
      IF((NSCA.EQ.NSCA1).OR.(STH.LE.0.))GO TO 900                      BAS06440
C                                                                      BAS06450
C***  DETERMINE SCATTERING DIRECTION FOR NEXT TRAVERSE                 BAS06460
C                                                                      BAS06470
      CALL GAS(JTYPE)                                                  BAS06480
C                                                                      BAS06490
C***  DETERMINE BIASING DIRECTIONS FOR NEXT TRAVERSE                   BAS06500
C                                                                      BAS06510
      CALL ELM(XA,SA,EL)                                               BAS06520
      IF(EL(2).LE.EL(1))GO TO 900                                      BAS06530
      GO TO 1700                                                       BAS06540
  900 CONTINUE                                                         BAS06550
C                                                                      BAS06560
C***  END MULTIPLE SCATTERING LOOP, BEGIN CONVOLUTION BLOCK            BAS06570
C                                                                      BAS06580
C***  WRITE PARTIAL RUN RESULTS INTO STORAGE FILE FOR USE BY NEXT RUN  BAS06590
C                                                                      BAS06600
      DO 2201 NSCA=1,NSCAM                                             BAS06610
      WRITE(KSTOR)ENC(NSCA)                                            BAS06620
      DO 2201 II=1,NTMAX                                               BAS06630
      WRITE(KSTOR)EN(NSCA,II)                                          BAS06640
 2201 CONTINUE                                                         BAS06650
      REWIND KSTOR                                                     BAS06660
C                                                                      BAS06670
C***  OUTPUT PARTIAL RUN RESULTS FOR STEADY STATE POWER                BAS06680
C                                                                      BAS06690
      WRITE (IOOUT,7400)                                               BAS06700
      WRITE(IOOUT,7500)                                                BAS06710
      WRITE(IOOUT,7600)                                                BAS06720
      PTOT=0.                                                          BAS06730
C######################################################################BAS06740
C       THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO     BAS06750
C       A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.                  BAS06760
C                                                                      BAS06770
C         NSCAM= 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING           BAS06780
C                                                                      BAS06790
C     WRITE(NPLOTU,9333) NSCAM                                         BAS06800
C######################################################################BAS06810
      DO 2110 NS=1,NSCAM                                               BAS06820
      PE=0.                                                            BAS06830
      NS1=NS-1                                                         BAS06840
      DO 2115 NT=1,NTMAX                                               BAS06850
 2115 PE=PE+EN(NS,NT)                                                  BAS06860
      PE=PE/ENC(NS)                                                    BAS06870
      PTOT=PTOT+PE                                                     BAS06880
C######################################################################BAS06890
C       THE FOLLOWING STATEMENT SHOULD BE UNCOMMENTED IF OUTPUT TO     BAS06900
C       A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.                  BAS06910
C                                                                      BAS06920
C         NS1   = ORDER OF SCATTERING                                  BAS06930
C         PE    = OBSERVED STEADY STATE POWER FOR THIS ORDER           BAS06940
C         ENC(  )= TOTAL NUMBER OF PHOTONS COUNTED FOR THIS ORDER      BAS06950
C     WRITE(NPLOTU,9333) NS1,PE,ENC(NS)                                BAS06960
C######################################################################BAS06970
      WRITE(IOOUT,7700)NS1,PE,ENC(NS)                                  BAS06980
 2110 CONTINUE                                                         BAS06990
C######################################################################BAS07000
```

```
C       THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO      BAS07010
C       A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.                    BAS07020
C                                                                        BAS07030
C         PTOT = OBSERVED TOTAL STEADY STATE POWER (ALL ORDERS)          BAS07040
C                                                                        BAS07050
C9333 FORMAT(I2,1X,2(E9.4,1X))                                           BAS07060
C       WRITE(NPLOTU,   9222) PTOT                                       BAS07070
C*************************************************************************BAS07080
      WRITE(IOOUT,7701)PTOT                                              BAS07090
      DO 2500 NSCA=1,NSCAM                                               BAS07100
C                                                                        BAS07110
C***  DETERMINE INDEX OF TIME OF LAST NONZERO POWER VALUE FOR EACH       BAS07120
C***  ORDER                                                              BAS07130
C                                                                        BAS07140
      CALL SMOOZ(NSCA,NO)                                                BAS07150
      NM(NSCA)=1+NO                                                      BAS07160
      IOR(NSCA)=NSCA-1                                                   BAS07170
 2500 Y(NSCA)=NM(NSCA)                                                   BAS07180
C                                                                        BAS07190
C***  DETERMINE LATEST TIME INDEX OF NONZERO POWER FOR ALL ORDERS        BAS07200
C                                                                        BAS07210
      CALL GMAX(NSCAM,YMAX)                                              BAS07220
      NMA=YMAX                                                           BAS07230
      JP=1                                                               BAS07240
 2900 JP=JP+1                                                            BAS07250
      AT=TPU(JP)                                                         BAS07260
      IF (AT.LE.0.0) GO TO 3000                                          BAS07270
      GO TO 2900                                                         BAS07280
 3000 JPMAX=JP-1                                                         BAS07290
      WRITE (IOOUT,7400)                                                 BAS07300
      WRITE (IOOUT,7900) JPMAX                                           BAS07310
C                                                                        BAS07320
C***  BEGIN PULSE LOOP                                                   BAS07330
C                                                                        BAS07340
      DO 3800 JP=1,JPMAX                                                 BAS07350
      WRITE (IOOUT,7400)                                                 BAS07360
      TP=TPU(JP)                                                         BAS07370
      AT=(NMA+4)*DELT                                                    BAS07380
      IF(TP.GT.AT)TP=AT                                                  BAS07390
      IF(TP.LE.DELT)TP=DELT+1.E-3                                        BAS07400
      NP=1.001+TP/DELT                                                   BAS07410
      NMAX=NMA+NP                                                        BAS07420
      IF(NMAX.GT.NDIM)NMAX=NDIM                                          BAS07430
      NP=NMAX-NMA                                                        BAS07440
      TP=(NP-1)*DELT                                                     BAS07450
      TMAX=(NMAX-1)*DELT                                                 BAS07460
      WRITE (IOOUT,8100) JP,TP                                           BAS07470
      WRITE (IOOUT,8200) JP,TMAX                                         BAS07480
      IF(TP.LE.0.)GO TO 3999                                             BAS07490
      DO 3400 NSCA=1,NSCAM                                               BAS07500
      NMS=NM(NSCA)                                                       BAS07510
      NMS1=NMS-1                                                         BAS07520
C                                                                        BAS07530
C***  NORMALIZE RETURN POWER BY DIVIDING CUMULATIVE POWER BY CUMULATIVE  BAS07540
C***  NUMBER OF PHOTONS                                                  BAS07550
C                                                                        BAS07560
      DO 3200 N=1,NMS1                                                   BAS07570
 3200 EN(NSCA,N)=EN(NSCA,N)/ENC(NSCA)                                    BAS07580
C                                                                        BAS07590
C***  PERFORM SQUARE SOURCE PULSE CONVOLUTION WITH PROBABILITIES         BAS07600
C***  PER UNIT TIME                                                      BAS07610
C                                                                        BAS07620
      CALL CONV(NP,NMS,NMAX,NSCA)                                        BAS07630
 3400 CONTINUE                                                           BAS07640
      DO 3500 N=1,NMAX                                                   BAS07650
      Y(N)=0.0                                                           BAS07660
      DO 3500 NSCA=1,NSCAM                                               BAS07670
 3500 Y(N)=Y(N)+EN(NSCA,N)                                               BAS07680
      XN=-DELT                                                           BAS07690
      DO 3600 N=1,NMAX                                                   BAS07700
```

```
      XN=XN+DELT                                                        BAS07710
 3600 X(N)=XN                                                           BAS07720
C                                                                       BAS07730
C***  OUTPUT TIME-DEPENDENT RESULTS                                     BAS07740
C                                                                       BAS07750
      WRITE (IOOUT,8300)                                                BAS07760
      WRITE (IOOUT,8400)                                                BAS07770
      WRITE(IOOUT,8500)(IOR(NS),NS=1,NSCAM)                             BAS07780
      WRITE(IOOUT,8501)                                                 BAS07790
C***##################################################################  BAS07800
C       THE FOLLOWING STATEMENTS SHOULD BE UNCOMMENTED IF OUTPUT TO     BAS07810
C       A USER-DEFINED PLOT FILE (NPLOTU) IS DESIRED.                   BAS07820
C                                                                       BAS07830
C         NMAX  = NUMBER OF TIME BOXES USED FOR TIME-DEPENDENT DATA     BAS07840
C         JP    = INDEX OF INPUT PULSE                                  BAS07850
C         NSCAM = 1 + HIGHEST SIGNIFICANT ORDER OF SCATTERING           BAS07860
C         X(  ) = ARRAY OF TIME VALUES FOR EACH TIME BOX                BAS07870
C         Y(  ) = ARRAY OF OBSERVED TIME-DEPENDENT TOTAL POWER          BAS07880
C         EN(L,)= ARRAY OF OBSERVED TIME-DEPENDENT POWER FOR ORDER L+1  BAS07890
C                                                                       BAS07900
C      IF(I1.LT.N1) GO TO 9777                                          BAS07910
C      WRITE(NPLOTU,9444) NMAX,JP,NSCAM                                 BAS07920
C9444 FORMAT(3(I3,1X))                                                  BAS07930
C      WRITE(NPLOTU,9222)(X(LLL),LLL=1,NMAX)                            BAS07940
C      WRITE(NPLOTU,9222)(Y(LLL),LLL=1,NMAX)                            BAS07950
C      DO 9666 LLL=1,NSCAM                                              BAS07960
C      WRITE(NPLOTU,9222)(EN(LLL,LLX),LLX=1,NMAX)                       BAS07970
C9666 CONTINUE                                                          BAS07980
C9777 CONTINUE                                                          BAS07990
C***##################################################################  BAS08000
      DO 3700 N=1,NMAX                                                  BAS08010
      WRITE (IOOUT,8600) X(N),Y(N),(EN(NSCA,N),NSCA=1,NSCAM)            BAS08020
 3700 CONTINUE                                                          BAS08030
C                                                                       BAS08040
C***  RELOAD PARTIAL RUN RESULTS FOR CONVOLUTION WITH NEXT PULSE        BAS08050
C                                                                       BAS08060
      DO 3701 NS=1,NSCAM                                                BAS08070
      READ(KSTOR)ENC(NS)                                                BAS08080
      DO 3701 II=1,NTMAX                                                BAS08090
      READ(KSTOR)EN(NS,II)                                              BAS08100
 3701 CONTINUE                                                          BAS08110
      REWIND KSTOR                                                      BAS08120
 3800 CONTINUE                                                          BAS08130
C                                                                       BAS08140
C***  END PULSE LOOP                                                    BAS08150
C                                                                       BAS08160
 1000 CONTINUE                                                          BAS08170
C                                                                       BAS08180
C***  END OUTER PHOTON LOOP?                                            BAS08190
C                                                                       BAS08200
  777 RETURN                                                            BAS08210
C                                                                       BAS08220
C***  ERROR RETURN MESSAGES                                             BAS08230
C                                                                       BAS08240
 3999 CONTINUE                                                          BAS08250
      WRITE(IOOUT,8700)NTMAX                                            BAS08260
      RETURN                                                            BAS08270
 4000 CONTINUE                                                          BAS08280
      WRITE(IOOUT,8800)                                                 BAS08290
      RETURN                                                            BAS08300
 4001 CONTINUE                                                          BAS08310
      WRITE(IOOUT,9000)                                                 BAS08320
      RETURN                                                            BAS08330
 4002 CONTINUE                                                          BAS08340
      WRITE(IOOUT,9001)                                                 BAS08350
      RETURN                                                            BAS08360
 4003 CONTINUE                                                          BAS08370
      WRITE(IOOUT,9002)                                                 BAS08380
      RETURN                                                            BAS08390
 4004 CONTINUE                                                          BAS08400
```

```
      WRITE(IOOUT,9005)                                                 BAS0824
      RETURN                                                            BAS0826
 4005 CONTINUE                                                          BAS0843
      WRITE(IOOUT,9006)                                                 BAS0844
      RETURN                                                            BAS0851
C*****FORMATS                                                           BAS0845
 4200 FORMAT (7F10.5)                                                   BAS0847
 4400 FORMAT (7I10)                                                     BAS0848
 4500 FORMAT (I10,2F10.5,I10)                                           BAS0849
 4600 FORMAT (5E15.10)                                                  BAS0850
 4700 FORMAT (1H ,34X,51(1H*))                                          BAS0851
 4800 FORMAT (1H ,34X,1H*,49X,1H*)                                      BAS0852
 4900 FORMAT (1H ,34X,1H*,9X,31HMONTE CARLO MULTIPLE SCATTERING,9X,1H*) BAS0853
 5000 FORMAT (1H ,34X,1H*,16X,18HAEROSOL SCATTERING,15X,1H*)            BAS0854
 5100 FORMAT (1H0,48X,23HPARAMETERS FOR THIS RUN)                       BAS0855
 5200 FORMAT(1H0,46X,28HUSER SUPPLIED PHASE FUNCTION)                   BAS0856
 5201 FORMAT(1H0,43X,34HMARITIME ARCTIC, VIS=0.1 TO 2.0 KM)             BAS0857
 5202 FORMAT(1H0,47X,26HMARITIME POLAR, VIS=0.2 KM)                     BAS0858
 5203 FORMAT(1H0,47X,26HMARITIME POLAR, VIS=2.0 KM)                     BAS0859
 5204 FORMAT(1H0,42X,36HCONTINENTAL POLAR, VIS=0.2 TO 2.5 KM)           BAS0860
 5205 FORMAT(1H0,52X,16HWHITE PHOSPHORUS)                               BAS0861
 5206 FORMAT(1H0,52X,16HHEXACHLOROETHANE)                               BAS0862
 5207 FORMAT(1H0,57X,7HFOG OIL)                                         BAS0863
 5208 FORMAT(1H0,45X,31HDUST (MODERATE AEROSOL LOADING))                BAS0864
 5209 FORMAT(1H0,46X,28HDUST (HEAVY AEROSOL LOADING))                   BAS0865
 5210 FORMAT(1H0,43X,34HMARITIME MODEL B, VIS=5 KM, RH=95%)             BAS0866
 5211 FORMAT(1H0,43X,35HMARITIME MODEL B, VIS=10 KM, RH=90%)            BAS0867
 5212 FORMAT(1H0,43X,35HMARITIME MODEL B, VIS=50 KM, RH=50%)            BAS0868
 5500 FORMAT (1H0,46X,27HUSER SUPPLIED AEROSOL MODEL)                   BAS0869
 5600 FORMAT (1H ,36X,11HWAVELENGTH=,F6.3,16H MICROMETERS               BAS0870
     1        7HALBEDO=,F5.3)                                          BAS0871
 5601 FORMAT(1H0,47X,25HELLIPSOIDAL AEROSOL CLOUD)                      BAS0872
 5602 FORMAT(1H ,41X,36HCOORDINATE ORIGIN AT CENTER OF CLOUD)           BAS0873
 5603 FORMAT(1H ,38X,42HZ-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH)     BAS0874
 5700 FORMAT (1H ,36X,31HAEROSOL EXTINCTION COEFFICIENT=,               BAS0875
     1        E10.4,7H KM**-1)                                         BAS0876
 5701 FORMAT(1H0,51X,17HSOURCE PARAMETERS)                              BAS0877
 5702 FORMAT(1H ,36X,27HSOURCE XYZ COORDINATES(KM)=,3(F8.4,1X))         BAS0878
 5703 FORMAT(1H ,36X,27HSOURCE APERTURE RADIUS(MM)=,F7.3)               BAS0879
 5704 FORMAT(1H ,36X,26HSOURCE AXIS POLAR ANGLE  =,F7.3,8H DEGREES)     BAS0880
 5705 FORMAT(1H ,36X,26HSOURCE AXIS AZIMUTH ANGLE=,F7.3,8H DEGREES)     BAS0881
 5706 FORMAT(1H ,36X,26HSOURCE BEAM SPREAD ANGLE =,E10.4,8H RADIANS)    BAS0882
 5900 FORMAT (1H0,50X,19HDETECTOR PARAMETERS)                           BAS0883
 6000 FORMAT (1H ,35X,29H CONE OF VIEW HALF-ANGLE    =,F7.3,           BAS0884
     1        8H DEGREES)                                              BAS0885
 6100 FORMAT (1H ,35X,29H DETECTOR APERTURE RADIUS   =,F7.3,           BAS0886
     1        3H CM)                                                   BAS0887
 6201 FORMAT(1H ,36X,29HDETECTOR XYZ COORDINATES(KM)=,3(F8.4,1X))       BAS0888
 6400 FORMAT (1H ,36X,28HDETECTOR AXIS POLAR ANGLE  =,F7.3,             BAS0889
     1        8H DEGREES)                                              BAS0890
 6500 FORMAT (1H ,36X,28HDETECTOR AXIS AZIMUTH ANGLE=,F7.3,             BAS0891
     1        8H DEGREES)                                              BAS0892
 6501 FORMAT(1H0,47X,23HGROUND PLANE PARAMETERS,/,40X,38HISOTROPIC REFLEBAS0893
    *CTION FROM GROUND PLANE)                                         BAS0894
 6502 FORMAT(1H ,36X,33HGROUND PLANE Z-COORDINATE ZG(KM)=,F7.3,/,       BAS0895
    *37X,33HGROUND PLANE ALBEDO, ALBG,      =,F7.3)                   BAS0896
 6800 FORMAT (1H0,34X,51(1H*))                                          BAS0897
 6900 FORMAT (1H0,51X,16HCLOUD PARAMETERS)                              BAS0898
 7000 FORMAT(1H ,31X,43HELLIPSOID PRINCIPAL XYZ HALF-AXES(KM)     =,    BAS0899
    *3(F8.4,1X))                                                      BAS0900
 7100 FORMAT (1H ,31X,43HEULER ANGLES THE,PHE,PSE OF ELLIPSOID(DEG)=,   BAS0901
    *3(F8.4,1X))                                                      BAS0902
 7200 FORMAT (1H ,31X,43HOPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES   =,   BAS0903
    *3(F8.4,1X))                                                      BAS0904
 7400 FORMAT (1H0,100X)                                                 BAS0905
 7500 FORMAT (1H0,32X,53HSTEADY STATE POWER TO DETECTOR, FOR UNIT SOURCEBAS0906
    * POWER)                                                          BAS0907
 7600 FORMAT (1H0,37X,5HORDER,3X,18HSTEADY STATE POWER,3X,              BAS0908
     1        17HNUMBER OF PHOTONS)                                    BAS0909
 7700 FORMAT (1H ,39X,I2,8X,E10.5,8X,E12.6)                             BAS0910
```

```
7701 FORMAT (1H ,37X,5HTOTAL,7X,E10.5)                                  BAS09110
7900 FORMAT (1H0,32X,23HPOWER INTO DETECTOR FOR,I2,6H PULSE             BAS09120
     1        23H(S) OF DIFFERENT LENGTH)                               BAS09130
8100 FORMAT (1H0,37X,12HPULSE NUMBER,I2,12H HAS LENGTH ,                BAS09140
     1        E10.4,13H MICROSECONDS)                                   BAS09150
8200 FORMAT (1H ,23X,34HDETECTOR RESPONSE CUTOFF TIME FOR               BAS09160
     1        12HPULSE NUMBER,I2,4H IS ,E10.4,                          BAS09170
     2        13H MICROSECONDS)                                         BAS09180
8300 FORMAT (1H0,25X,68HDETECTOR RESPONSE, POWER AS A FUNCTION OF TIME,BAS09190
     * FOR UNIT PULSE POWER)                                            BAS09200
8400 FORMAT (1H0,55X,21HPOWER FROM EACH ORDER/14X,5HTOTAL)              BAS09210
8500 FORMAT (1H ,3X,4HTIME,6X,5HPOWER,2X,10(5X,I2,4X))                  BAS09220
8501 FORMAT(1H ,130(1H-))                                               BAS09230
8600 FORMAT (12(E10.4,1X))                                              BAS09240
8700 FORMAT(1H0,6HNTMAX=,I3,80H SHOULD BE DECREASED TO 46. IT IS TOO LABAS09250
     *RGE TO ALLOW CONVOLUTION WITH YOUR PULSE)                         BAS09260
8800 FORMAT(1H0,97HYOUR INCIDENT PHOTONS NEVER INTERSECT THE CLOUD OR TBAS09270
     *HE GROUND. CHECK YOUR INPUT SOURCE PARAMETERS)                    BAS09280
9000 FORMAT(1H0,72HTHE DETECTOR LOOKS ABOVE THE CLOUD. CHECK YOUR INPUTBAS09290
     * DETECTOR PARAMETERS)                                            BAS09300
9001 FORMAT(1H0,65HYOUR GROUND PLANE IS ENTIRELY ABOVE YOUR CLOUD. CHECBAS09310
     *K YOUR INPUTS)                                                   BAS09320
9002 FORMAT(1H0,76HNEITHER YOUR SOURCE NOR YOUR DETECTOR LOOK INTO THE BAS09330
     *CLOUD. CHECK YOUR INPUTS)                                        BAS09340
9003 FORMAT(1H0,71HYOUR SOURCE WAS UNDERGROUND. IT HAS BEEN PUT AT THE BAS09350
     *GROUND, WITH XS(3)=,F6.3,2HKM)                                   BAS09360
9004 FORMAT(1H0,114HYOUR GROUND PLANE WAS TOO FAR AWAY FROM THE CLOUD TBAS09370
     *O PRODUCE GROUND REFLECTIONS WITHIN THE MAX TIME DELAY ALLOWED.,/,BAS09380
     *34H THE GROUND PLANE WAS MOVED TO ZG=,F6.3,2HKM)                  BAS09390
9005 FORMAT(1H0,91HGROUND PLANE WAS ABSENT AND DETECTOR DOES NOT LOOK ABAS09400
     *T CLOUD. CHECK YOUR DETECTOR PARAMETERS)                          BAS09410
9006 FORMAT(1H0,79HGROUND PLANE WAS ABSENT AND SOURCE DOES NOT ILLUMINABAS09420
     *TE CLOUD. CHECK YOUR INPUTS)                                      BAS09430
      END                                                               BAS09440
```

```
      SUBROUTINE BKWD(JTYPE)                                                BKW00010
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                         BKW00020
      COMMON/RNDM/ SEED                                                     BKW00030
      COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,        BKW00040
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                            BKW00050
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),           BKW00060
     +BE(16),SINGWV,PF(65),LLMAX                                            BKW00070
      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),         BKW00080
     *ALB(2),ZG,DMAX                                                        BKW00090
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)                  BKW00100
      COMMON/FGEL/XA(3),D,NSCA                                              BKW00110
      DIMENSION SBC(3),SBA(3)                                               BKW00120
      NSCAP=NSCA+1                                                          BKW00130
C                                                                           BKW00140
C***  DETERMINE MONTE CARLO BACKWARD TRAVERSE ANGLES THETA (AT)             BKW00150
C***  AND PHI (BT) INSIDE DETECTOR CONE.                                    BKW00160
C                                                                           BKW00170
      BT=TWOPI*RAND  ( SEED)                                                BKW00180
      AT=THV*RAND  ( SEED)                                                  BKW00190
      IF(AT.LE.1.E-30) AT=0.                                                BKW00200
      IF(BT.LE.1.E-30) BT=0.                                                BKW00210
      V=SIN(AT)                                                             BKW00220
      U23=COS(AT)                                                           BKW00230
C                                                                           BKW00240
C***  ROTATE BACKWARD TRAVERSE VECTOR INTO STANDARD FRAME OF REFERENCE.     BKW00250
C                                                                           BKW00260
      CALL ROTAT(AT,BT,R,SBC)                                               BKW00270
C                                                                           BKW00280
C***  DETERMINE BIASING DISTANCES FOR BACKWARD TRAVERSE.                    BKW00290
C                                                                           BKW00300
      CALL ELM(XD,SBC,EL)                                                   BKW00310
C                                                                           BKW00320
C***  DETERMINE STATISTICAL WEIGHT REX OF BACKWARD TRAVERSE.                BKW00330
C                                                                           BKW00340
      REX=1.-EXP(-GAMMA*(EL(2)-EL(1)))                                      BKW00350
      REXRN=REX*RAND(SEED)                                                  BKW00360
      IF((1.-REXRN).LE.1.E-7) GO TO 15                                      BKW00370
C                                                                           BKW00380
C***  DETERMINE RANDOM DISTANCE FOR BIASED BACKWARD TRAVERSE.               BKW00390
C                                                                           BKW00400
      ELBC=-ALOG(1.-REXRN)/GAMMA+EL(1)                                      BKW00410
      CT=0.                                                                 BKW00420
      DO 4 K=1,3                                                            BKW00430
      SBA(K)=XD(K)-XA(K)+SBC(K)*ELBC                                        BKW00440
    4 CT=CT+SBA(K)**2                                                       BKW00450
      IF(CT.LE.1.E-30) CT=0.                                                BKW00460
      ELAB=SQRT(CT)                                                         BKW00470
      IF(ELAB.LT.ALIM)GO TO 15                                              BKW00480
C                                                                           BKW00490
C***  DETERMINE TIME BOX INDEX NT FOR THE COMPLETE BACKWARD TRAVERSE.       BKW00500
C                                                                           BKW00510
      NT=1.+(DTOT+ELAB+ELBC-ELMIN)/DELD                                     BKW00520
      IF(NT.LT.0) GO TO 15                                                  BKW00530
      IF(NT.EQ.0) NT=1                                                      BKW00540
      IF(NT.GT.NTMAX)RETURN                                                 BKW00550
      U1=0.                                                                 BKW00560
      U2=0.                                                                 BKW00570
      DO 5 K=1,3                                                            BKW00580
      SBA(K)=SBA(K)/ELAB                                                    BKW00590
      U1=U1+SA(K)*SBA(K)                                                    BKW00600
    5 U2=U2-SBA(K)*SBC(K)                                                   BKW00610
    6 CALL FIND(U2,PF2)                                                     BKW00620
      IF(JTYPE.EQ.2)GO TO 9                                                 BKW00630
      CALL FIND(U1,PF1)                                                     BKW00640
      GO TO 10                                                              BKW00650
    9 CONTINUE                                                              BKW00660
      CALL ELM(XA,SBA,EL)                                                   BKW00670
      ELAB=ELAB-EL(1)                                                       BKW00680
      PF1=1.                                                                BKW00690
   10 DOM=FAC*ALB(1)*STH*REX*V*U23*PF1*PF2*EXP(-GAMMA*ELAB)/CT              BKW00700
```

```
      ENC(NSCAP,NT)=ENC(NSCAP,NT)+DOM                                   BKW00710
      RETURN                                                            BKW00720
   15 ENC(NSCAP)=ENC(NSCAP)-1.                                          BKW00730
      STH=0.                                                            BKW00740
      RETURN                                                            BKW00750
      END                                                               BKW00760
```

```
      SUBROUTINE CONV(NP,NMS,NMAX,NSCA)                                 CON00010
C*****THIS SUBROUTINE CONVOLUTES A GIVEN SEQUENCE OF VALUES EN(NSCA,N),  CON00020
C     N=1,NMS, WITH THE UNIT SQUARE FUNCTION EXTENDING FROM TIME=0 TO   CON00030
C*****TIME=(NP-1)*DELT.                                                 CON00040
      DOUBLE PRECISION DBLE,XDBLE(100)                                  CON00050
      COMMON/CONB/X(100),Y(100)                                         CON00060
      COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,    CON00070
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                        CON00080
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),       CON00090
     +BE(16),SINGWV,PF(65),LLMAX                                        CON00100
      NMS1=NMS+1                                                        CON00110
      NMA=NMS+NP-1                                                      CON00120
      XDBLE(1)=0.E-00                                                   CON00130
      DO 100 N=2,NMS                                                    CON00140
  100 XDBLE(N)=XDBLE(N-1)+DBLE(EN(NSCA,N-1))                            CON00150
      NP1=NP+1                                                          CON00160
      IF (NP.LT.NMS) GO TO 600                                          CON00170
      DO 200 N=1,NMS                                                    CON00180
  200 Y(N)=XDBLE(N)                                                     CON00190
      IF (NP.EQ.NMS) GO TO 400                                          CON00200
      DO 300 N=NMS1,NP                                                  CON00210
  300 Y(N)=XDBLE(NMS)                                                   CON00220
  400 CONTINUE                                                          CON00230
      DO 500 N=NP1,NMA                                                  CON00240
  500 Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)                                     CON00250
      GO TO 1000                                                        CON00260
  600 CONTINUE                                                          CON00270
      DO 700 N=1,NP                                                     CON00280
  700 Y(N)=XDBLE(N)                                                     CON00290
      DO 800 N=NP1,NMS                                                  CON00300
  800 Y(N)=XDBLE(N)-XDBLE(N-NP+1)                                       CON00310
      DO 900 N=NMS1,NMA                                                 CON00320
  900 Y(N)=XDBLE(NMS)-XDBLE(N-NP+1)                                     CON00330
 1000 CONTINUE                                                          CON00340
      IF (NMA.EQ.NMAX) GO TO 1200                                       CON00350
      NMA1=NMA+1                                                        CON00360
      DO 1100 N=NMA1,NMAX                                               CON00370
 1100 Y(N)=0.0                                                          CON00380
      DO 1150 N=1,NMAX                                                  CON00390
 1150 EN(NSCA,N)=Y(N)                                                   CON00400
 1200 RETURN                                                            CON00410
      END                                                               CON00420
```

```
      SUBROUTINE ELM(X1,S1,EL)                                          ELM00010
C                                                                       ELM00020
C***  THIS SUBROUTINE DETERMINES THE SMALLEST (EL(1)) AND THE LARGEST   ELM00030
C***  (EL(2)) DISTANCES FROM A POINT X1() TO THE SURFACES OF AN         ELM00040
C***  ELLIPSOID ALONG A LINE OF SIGHT DEFINED BY UNIT VECTOR S1().      ELM00050
C                                                                       ELM00060
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)              ELM00070
      DIMENSION X1(3),S1(3),EL(2)                                       ELM00080
      AT=0.                                                             ELM00090
      BT=0.                                                             ELM00100
      CT=-1.                                                            ELM00110
      DO 1 K=1,3                                                        ELM00120
      X=0.                                                              ELM00130
      Y=0.                                                              ELM00140
      DO 2 L=1,3                                                        ELM00150
      X=X+RE(K,L)*X1(L)                                                 ELM00160
    2 Y=Y+RE(K,L)*S1(L)                                                 ELM00170
      AT=AT+Y**2/ASQ(K)                                                 ELM00180
      BT=BT+X*Y/ASQ(K)                                                  ELM00190
    1 CT=CT+X**2/ASQ(K)                                                 ELM00200
      DISC=BT**2-AT*CT                                                  ELM00210
      IF(DISC.LT.1.E-30)GO TO 10                                        ELM00220
      DISC=SQRT(DISC)                                                   ELM00230
      EL(1)=-(BT+DISC)/AT                                               ELM00240
      EL(2)=(-BT+DISC)/AT                                               ELM00250
      IF(CT.LE.0.)EL(1)=0.                                              ELM00260
      IF((EL(1).LT.0.).OR.(EL(2).LT.0.))GO TO 10                        ELM00270
      RETURN                                                            ELM00280
   10 EL(1)=0.                                                          ELM00290
      EL(2)=0.                                                          ELM00300
      RETURN                                                            ELM00310
      END                                                               ELM00320
```

```
      SUBROUTINE FIND(U1,PFU)                                           FIN00010
      COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,    FIN00020
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                        FIN00030
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),       FIN00040
     +BE(16),SINGWV,PF(65),LLMAX                                        FIN00050
      L=1                                                               FIN00060
      LL=LMM1                                                           FIN00070
      DO 1 K=1,KMAX                                                     FIN00080
      LL=LL/2                                                           FIN00090
      L=L+LL                                                            FIN00100
      AT=U1-U(L)                                                        FIN00110
      IF(AT.GT.1.E-7)L=L-LL                                             FIN00120
    1 CONTINUE                                                          FIN00130
      PFU=PF(L)+(U1-U(L))*(PF(L+1)-PF(L))/(U(L+1)-U(L))                 FIN00140
      RETURN                                                            FIN00150
      END                                                               FIN00160
```

```
      SUBROUTINE FWRD(SCA,DOM,ELAC)                                     FWR00010
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                      FWR00020
      COMMON/FGEL/XA(3),D,NSCA                                          FWR00030
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)              FWR00040
      COMMON/RNDM/ SEED                                                 FWR00050
      DIMENSION SCA(3)                                                  FWR00060
      PSI=TWOPI*RAND  ( SEED)                                           FWR00070
      TH=PI2                                                            FWR00080
      CALL ROTAT(TH,PSI,R,SCA)                                          FWR00090
      ANR=RAND(SEED)                                                    FWR00100
      IF(ANR.LE.1.E-30) ANR=0.                                          FWR00110
      RHO=AKM*SQRT(ANR)                                                 FWR00120
      ELSQ=0.                                                           FWR00130
      DO 1 K=1,3                                                        FWR00140
      SCA(K)=XD(K)-XA(K)+SCA(K)*RHO                                     FWR00150
    1 ELSQ=ELSQ+SCA(K)**2                                               FWR00160
      IF(ELSQ.LE.1.E-30) ELSQ=1.E-30                                    FWR00170
      ELAC=SQRT(ELSQ)                                                   FWR00180
      DO 3 K=1,3                                                        FWR00190
    3 SCA(K)=SCA(K)/ELAC                                                FWR00200
      DOM=AKSQ/(4.*ELSQ)                                                FWR00210
      RETURN                                                            FWR00220
      END                                                               FWR00230
```

```
      SUBROUTINE GAS(JTYPE)                                              GAS00010
C                                                                        GAS00020
C***  THIS SUBROUTINE DETERMINES A RANDOMLY-SELECTED SCATTERING          GAS00030
C***  DIRECTION USED IN PHOTON TRAVERSES WITHIN THE ELLIPSOIDAL          GAS00040
C***  AEROSOL CLOUD. JTYPE=1 SIGNIFIES THAT THE SCATTERING EVENT AT      GAS00050
C***  WHICH THIS ANGLE IS SELECTED IS WITHIN THE AEROSOL CLOUD.          GAS00060
C***  JTYPE=2 SIGNIFIES THAT THE SCATTERING EVENT IS ON THE GROUND       GAS00070
C***  PLANE.                                                             GAS00080
C                                                                        GAS00090
      COMMON/ALL/AT,BT,CT,BLIM                                           GAS00100
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                       GAS00110
      COMMON /MO5/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,     GAS00120
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                         GAS00130
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),        GAS00140
     +BE(16),SINGWV,PF(65),LLMAX                                         GAS00150
      COMMON/RNDM/   SEED                                                GAS00160
      COMMON/FGEL/XA(3),D,NSCA                                           GAS00170
      AT=RAND  ( SEED)                                                   GAS00180
      IF(AT.LE.1.E-30) AT=0.                                             GAS00190
      IF(JTYPE.EQ.1)GO TO 1                                              GAS00200
      SA(3)=1.-AT                                                        GAS00210
      ARG=1.-SA(3)**2                                                    GAS00220
      IF(ARG.LT.1.E-30) ARG=0.                                           GAS00230
      BT=TWOPI*RAND  ( SEED)                                             GAS00240
      IF(BT.LE.1.E-30) BT=0.                                             GAS00250
      V1=SQRT(ARG)                                                       GAS00260
      C1=COS(BT)                                                         GAS00270
      S1=SIN(BT)                                                         GAS00280
      SA(1)=V1*C1                                                        GAS00290
      SA(2)=V1*S1                                                        GAS00300
      RETURN                                                             GAS00310
    1 CALL USCA(AT,BT)                                                   GAS00320
      CT=TWOPI*RAND  ( SEED)                                             GAS00330
      IF(CT.LE.1.E-30) CT=0.                                             GAS00340
      ARG=1.-BT**2                                                       GAS00350
      IF(ARG.LT.1.E-30) ARG=0.                                           GAS00360
      V=SQRT(ARG)                                                        GAS00370
      C=COS(CT)                                                          GAS00380
      S=SIN(CT)                                                          GAS00390
      SA(1)=BT*SA(1)+V*(C*C1*SA(3)-S*S1)                                 GAS00400
      SA(2)=BT*SA(2)+V*(C*S1*SA(3)+S*C1)                                 GAS00410
      SA(3)=BT*SA(3)-V*C*V1                                              GAS00420
      AT=SA(1)**2+SA(2)**2                                               GAS00430
      IF(AT.LT.1.E-10)GO TO 3                                            GAS00440
      V1=SQRT(AT)                                                        GAS00450
      C1=SA(1)/V1                                                        GAS00460
      S1=SA(2)/V1                                                        GAS00470
      RETURN                                                             GAS00480
    3 C1=C                                                               GAS00490
      S1=S                                                               GAS00500
      V1=0.                                                              GAS00510
      SA(1)=0.                                                           GAS00520
      SA(2)=0.                                                           GAS00530
      BT=SA(3)                                                           GAS00540
      SA(3)=1.                                                           GAS00550
      IF(BT.LT.0.)SA(3)=-1.                                              GAS00560
      RETURN                                                             GAS00570
      END                                                                GAS00580
```

```
      SUBROUTINE GMAX(IMAX,YMAX)                                        GMA00010
C                                                                       GMA00020
C***  THIS SUBROUTINE DETERMINES THE MAXIMUM VALUE YMAX OF AN INPUT     GMA00030
C***  ARRAY Y( ) OF DIMENSION IMAX.                                     GMA00040
C                                                                       GMA00050
      COMMON/CONB/X(100),Y(100)                                         GMA00060
      I=0                                                               GMA00070
      YMAX=Y(1)                                                         GMA00080
    1 I=I+1                                                             GMA00090
      IF(I.EQ.IMAX)RETURN                                               GMA00100
      T=Y(I+1)-YMAX                                                     GMA00110
      IF(T.GT.0.)YMAX=Y(I+1)                                            GMA00120
      GO TO 1                                                           GMA00130
      END                                                               GMA00140
```

```
      SUBROUTINE MATRX(TH,PH,S,R)                                       MAT00010
C                                                                       MAT00020
C***  THIS SUBROUTINE GENERATES UNIT VECTOR S( ) AND ROTATION MATRIX    MAT00030
C***  R( ) FOR A SET OF INPUT POLAR ANGLES (TH,PH).                     MAT00040
C***  THE ROTATION MATRIX R( ) ROTATES A VECTOR DEFINED RELATIVE TO THE MAT00050
C***  (TH,PH) DIRECTION INTO THE STANDARD SYSTEM OF COORDINATES.        MAT00060
C***  THE UNIT VECTOR S( ) POINTS IN THE (TH,PH) DIRECTION IN THE       MAT00070
C***  STANDARD SYSTEM OF COORDINATES.                                   MAT00080
C                                                                       MAT00090
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                    MAT00100
      DIMENSION S(3),R(3,3)                                             MAT00110
      AT=PIRAD*TH                                                       MAT00120
      BT=PIRAD*PH                                                       MAT00130
      IF(AT.LE.1.E-30) AT=0.                                            MAT00140
      IF(BT.LE.1.E-30) BT=0.                                            MAT00150
      V1=SIN(AT)                                                        MAT00160
      C1=COS(BT)                                                        MAT00170
      S1=SIN(BT)                                                        MAT00180
      S(1)=V1*C1                                                        MAT00190
      S(2)=V1*S1                                                        MAT00200
      S(3)=COS(AT)                                                      MAT00210
      R(1,1)=C1*S(3)                                                    MAT00220
      R(1,2)=S1*S(3)                                                    MAT00230
      R(1,3)=-V1                                                        MAT00240
      R(2,1)=-S1                                                        MAT00250
      R(2,2)=C1                                                         MAT00260
      R(2,3)=0.                                                         MAT00270
      DO 1 K=1,3                                                        MAT00280
    1 R(3,K)=S(K)                                                       MAT00290
      RETURN                                                            MAT00300
      END                                                               MAT00310
```

```
      SUBROUTINE ROTAT(TH,PH,R,S)                                       ROT00010
      DIMENSION R(3,3),X(3),S(3)                                        ROT00020
      IF(TH.LE.1.E-30) TH=0.                                            ROT00030
      IF(PH.LE.1.E-30) PH=0.                                            ROT00040
      V=SIN(TH)                                                         ROT00050
      X(1)=V*COS(PH)                                                    ROT00060
      X(2)=V*SIN(PH)                                                    ROT00070
      X(3)=COS(TH)                                                      ROT00080
      DO 1 J=1,3                                                        ROT00090
      S(J)=0.                                                           ROT00100
      DO 1 K=1,3                                                        ROT00110
    1 S(J)=S(J)+R(K,J)*X(K)                                             ROT00120
      RETURN                                                            ROT00130
      END                                                               ROT00140
```

```
      SUBROUTINE SMOOZ(NSCA,NO)                                          SMO00010
C*****THIS SUBROUTINE DETERMINES 'NO', THAT VALUE OF I BEYOND WHICH ALL  SMO00020
C*****Y(I) IN A SEQUENCE ARE ZERO.                                       SMO00030
      COMMON /MO5/Y1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,     SMO00040
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                         SMO00050
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),        SMO00060
     +BE(16),SINGWV,PF(65),LLMAX                                         SMO00070
      I=NTMAX+1                                                          SMO00080
  100 I=I-1                                                              SMO00090
      IF(I.EQ.0)GO TO 200                                                SMO00100
      YI=EN(NSCA,I)                                                      SMO00110
      IF (YI.GT.0.0) GO TO 200                                           SMO00120
      GO TO 100                                                          SMO00130
  200 NO=I                                                               SMO00140
      RETURN                                                             SMO00150
      END                                                                SMO00160
```

```
      SUBROUTINE START(SS)                                                  STA00010
C                                                                           STA00020
C***  THIS SUBROUTINE INITIALIZES PHOTON LAUNCH DIRECTION AND CALCULATES    STA00030
C***  DIRECT BEAM (ZEROTH ORDER) CONTRIBUTIONS TO RECEIVED POWER.           ST. 00040
C                                                                           STA00050
      COMMON/ALL/AT,BT,CT,BLIM                                              STA00060
      COMMON/CONB/X(100),Y(100)                                             STA00070
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)                  STA00080
      COMMON/HIT/UDS,THSP,RS(3,3),XS(3),DSA,XV(3)                           STA00090
      COMMON/FGEL/XA(3),D,NSCA                                              STA00100
      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),         STA00110
     *ALB(2),ZG,DMAX                                                        STA00120
      COMMON/RNDM/ SEED                                                     STA00130
      COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,        STA00140
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                            STA00150
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),           STA00160
     +BE(16),SINGWV,PF(65),LLMAX                                            STA00170
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                         STA00180
      DIMENSION SS(3)                                                       STA00190
C                                                                           STA00200
C***  GENERATE RANDOM THETA (AT) AND PHI (BT) PHOTON LAUNCH ANGLES          STA00210
C***  CONFINED WITHIN SOURCE CONE.                                          STA00220
C                                                                           STA00230
      ANR=RAND(SEED)                                                        STA00240
      IF(ANR.LE.1.E-30) ANR=0.                                              STA00250
      AT=THSP*SQRT(ANR)                                                     STA00260
      BT=TWOPI*RAND  ( SEED)                                                STA00270
C                                                                           STA00280
C***  ROTATE LAUNCH VECTOR SA( ) INTO STANDARD FRAME OF REFERENCE.          STA00290
C                                                                           STA00300
      CALL ROTAT(AT,BT,RS,SA)                                               STA00310
C                                                                           STA00320
C***  DETERMINE HORIZONTAL COMPONENT CT OF LAUNCH VECTOR AND BRANCH         STA00330
C***  TO VERTICAL TREATMENT (WHICH INCLUDES GROUND PLANE) IF THIS           STA00340
C***  COMPONENT IS VERY SMALL.                                              STA00350
C                                                                           STA00360
      CT=SA(1)**2+SA(2)**2                                                  STA00370
      IF(CT.LE.1.E-24)GO TO 1                                               STA00380
      V1=SQRT(CT)                                                           STA00390
      C1=SA(1)/V1                                                           STA00400
      S1=SA(2)/V1                                                           STA00410
      GO TO 2                                                               STA00420
C                                                                           STA00430
C***  DEFINE VERTICAL UNIT VECTOR                                           STA00440
C                                                                           STA00450
    1 V1=0.                                                                 STA00460
      CT=SA(3)                                                              STA00470
      SA(3)=1.                                                              STA00480
      IF((CT.LT.0.).AND.(ALB(2).GT.0.)) SA(3)=-1.                           STA00490
      C1=1.                                                                 STA00500
      S1=0.                                                                 STA00510
      SA(1)=0.                                                              STA00520
      SA(2)=0.                                                              STA00530
C                                                                           STA00540
C***  INITIALIZE TOTAL TRAVERSE DISTANCE DTOT AND STATISTICAL               STA00550
C***  STRENGTH STH.                                                         STA00560
C                                                                           STA00570
    2 DTOT=0.                                                               STA00580
      STH=1.                                                                STA00590
      BT=0.                                                                 STA00600
      IF(AT.LE.1.E-30) GO TO 11                                             STA00610
      DAA=DSA/COS(AT)                                                       STA00620
      DAD=0.                                                                STA00630
      DO 3 K=1,3                                                            STA00640
      XA(K)=XV(K)+DAA*SA(K)                                                 STA00650
      DAD=DAD+Y(K)**2                                                       STA00660
    3 BT=BT+Y(K)*SS(K)                                                      STA00670
      IF(DAD.LE.1.E-30) DAD=0.                                              STA00680
      DAD=SQRT(DAD)                                                         STA00690
C                                                                           STA00700
```

```
C***  DETERMINE DIRECT BEAM ATTENUATION DISTANCE AND BIASING DISTANCES  STA00710
C***  FOR INITIAL PHOTON TRAVERSE.                                      STA00720
C                                                                       STA00730
      CALL ELM(XA,SA,EL)                                                STA00740
      IF(BT.LE.ALIM) GO TO 4                                            STA00750
      IF(DAD.LE.(1.001*DSA)) GO TO 4                                    STA00760
      BT=0.                                                             STA00770
      CT=0.                                                             STA00780
      DO 5 K=1,3                                                        STA00790
      CT=CT+SA(K)*SD(K)                                                 STA00800
    5 BT=BT+Y(K)*SD(K)                                                  STA00810
      CTSQ=CT**2                                                        STA00820
      IF(CTSQ.LE.1.E-30) GO TO 4                                        STA00830
      ELSD=ABS(BT/CT)-DAA                                               STA00840
      CT=0.                                                             STA00850
      DO 6 K=1,3                                                        STA00860
    6 CT=CT+(XA(K)+SA(K)*ELSD-XD(K))**2                                 STA00870
      IF(CT.GT.AKSQ) GO TO 4                                            STA00880
      IF(EL(2).LE.EL(1)) STH=0.                                         STA00890
      IF(ELD(1).LE.0.) EL(2)=ELSD                                       STA00900
      IF(DAD.LT.ALIM) GO TO 4                                           STA00910
      UDS=-BT/DAD                                                       STA00920
      IF(UDS.LT.UV)RETURN                                               STA00930
      NT=1.+(ELSD-ELMIN)/DELD                                           STA00940
      IF(NT.LT.0) GO TO 11                                              STA00950
      IF(NT.EQ.0) NT=1                                                  STA00960
      EN(1,NT)=EN(1,NT)+EXP(-GAMMA*(EL(2)-EL(1)))                       STA00970
      RETURN                                                            STA00980
    4 CONTINUE                                                          STA00990
      IF((EL(2)-EL(1)).GT.1.E-20) RETURN                                STA01000
      IF((SA(3).GE.0.).OR.(ALB(2).LE.1.E-20)) STH=0.                    STA01010
      RETURN                                                            STA01020
   11 STH=0.                                                            STA01030
      RETURN                                                            STA01040
      END                                                               STA01050
```

```
      SUBROUTINE TRAVRS(JTYPE,I2FLG,ICOND)                              TRA00010
      COMMON/RNDM/  SEED                                                TRA00020
      COMMON/ALL/AT,BT,CT,BLIM                                          TRA00030
      COMMON/FAHT/SD(3),UV,GAMMA,ELD(2),STH,FAC,ALIM,THV,TAU,EL(2),     TRA00040
     *ALB(2),ZG,DMAX                                                    TRA00050
      COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,    TRA00060
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                        TRA00070
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),       TRA00080
     +BE(16),SINGWV,PF(65),LLMAX                                        TRA00090
      COMMON/FGEL/XA(3),D,NSCA                                          TRA00100
      COMMON/FWD/AKM,R(3,3),AKSQ,XD(3),ASQ(3),RE(3,3),A(3)              TRA00110
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     TRA00120
      COMMON/CONB/X(100),Y(100)                                         TRA00130
      DIMENSION SDA(3)                                                  TRA00140
      DGAM=EL(2)-EL(1)                                                  TRA00150
      REXIT=.99999                                                      TRA00160
      RX=.99999                                                         TRA00170
      RXX=.99999                                                        TRA00180
      IF((DGAM.LE.0.).AND.(ALB(2).GT.0.))GO TO 9                        TRA00190
      IF(DGAM.LE.1.E-7) GO TO 70                                        TRA00200
      IF(I2FLG.EQ.1) GO TO 5                                            TRA00210
      IF((SA(3).GE.0.).OR.(ALB(2).LE.0.))REXIT=1.-EXP(-GAMMA*DGAM)      TRA00220
      AT=EL(1)-(ALOG(1.-REXIT*RAND  ( SEED)))/GAMMA                     TRA00230
      GO TO 4                                                           TRA00240
    5 CONTINUE                                                          TRA00250
      IF(ELD(1).GT.ALIM)ALIM=ELD(1)                                     TRA00260
      RXX=(1./GAMMA)*((1./ALIM)-(1./DGAM))                              TRA00270
      DENOM=((1./ALIM)-GAMMA*RXX*RAND(SEED))                            TRA00280
      IF(DENOM.LE.1.E-7) GO TO 70                                       TRA00290
      AT=1./DENOM                                                       TRA00300
      RX=EXP(-(GAMMA*AT))*(AT*GAMMA)**2                                 TRA00310
    4 BT=XA(3)+SA(3)*AT                                                 TRA00320
      IF((BT.GT.ZG).AND.(AT.LE.EL(2)))GO TO 2                           TRA00330
    9 JTYPE=2                                                           TRA00340
      AT=(ZG-XA(3))/SA(3)                                               TRA00350
      GO TO 1                                                           TRA00360
    2 JTYPE=1                                                           TRA00370
    1 STH=ALB(JTYPE)*REXIT*STH*RXX                                      TRA00380
      IF(STH.LE.0.)RETURN                                               TRA00390
      DO 3 K=1,3                                                        TRA00400
    3 XA(K)=XA(K)+AT*SA(K)                                              TRA00410
      DTOT=DTOT+AT                                                      TRA00420
      IF(NSCA.NE.2) GO TO 50                                            TRA00430
      AT=0.                                                             TRA00440
      DO 8 K=1,3                                                        TRA00450
      SDA(K)=XD(K)-XA(K)                                                TRA00460
    8 AT=AT+SDA(K)*SDA(K)                                               TRA00470
      IF(AT.LT.BLIM)GO TO 15                                            TRA00480
      AT=SQRT(AT)                                                       TRA00490
      BT=0.                                                             TRA00500
      DO 32 K=1,3                                                       TRA00510
      SDA(K)=SDA(K)/AT                                                  TRA00520
   32 BT=BT-SDA(K)*SD(K)                                                TRA00530
      IF(BT.LT.UV) GO TO 50                                             TRA00540
      CALL FWRD(SDA,DOM,AT)                                             TRA00550
      NT=1.+(DTOT+AT-ELMIN)/DELD                                        TRA00560
      IF(NT.LT.0) RETURN                                                TRA00570
      IF(NT.EQ.0) NT=1                                                  TRA00580
      IF((NT.EQ.1).AND.(I2FLG.EQ.0).AND.(ICOND.EQ.1))GO TO 50           TRA00590
      IF(NT.GT.NTMAX)RETURN                                             TRA00600
      CT=0.                                                             TRA00610
      BT=0.                                                             TRA00620
      DO 7 K=1,3                                                        TRA00630
      BT=BT-SDA(K)*SD(K)                                                TRA00640
    7 CT=CT+SDA(K)*SA(K)                                                TRA00650
      PFI=1.                                                            TRA00660
      IF(JTYPE.EQ.1)CALL FIND(CT,PFI)                                   TRA00670
      CALL ELM(XA,SDA,EL)                                               TRA00680
      AT=AT-EL(1)                                                       TRA00690
      IF(ELD(1).GT.0.)AT=EL(2)-EL(1)                                    TRA00700
```

```
      DOM=DOM*PFI*STH*EXP(-GAMMA*AT)*BT*RX                              TRA00710
      IF(I2FLG.NE.1) GO TO 13                                           TRA00720
      IF(NT.EQ.1) DOM=DOM*10.                                           TRA00730
      IF(NT.GT.1) DOM=0.                                                TRA00740
      GO TO 14                                                          TRA00750
   13 CONTINUE                                                          TRA00760
      IF(ICOND.EQ.1) DOM=DOM/0.9                                        TRA00770
   14 EN(NSCA,NT)=EN(NSCA,NT)+DOM                                       TRA00780
      GO TO 50                                                          TRA00790
   15 ENC(2)=ENC(2)-1.                                                  TRA00800
      STH=0.                                                            TRA00810
      RETURN                                                            TRA00820
   50 CONTINUE                                                          TRA00830
      IF(I2FLG.EQ.1) GO TO 60                                           TRA00840
      CALL BKWD(JTYPE)                                                  TRA00850
   60 RETURN                                                            TRA00860
   70 ENC(NSCA)=ENC(NSCA)-1.                                            TRA00870
      STH=0.                                                            TRA00880
      RETURN                                                            TRA00890
      END                                                               TRA00900
```

```
      SUBROUTINE USCA(SC,US)                                            USC00010
      COMMON /MOS/V1,C1,S1,SA(3),EN(10,100),ENC(10),ELMIN,DELD,DTOT,    USC00020
     +NTMAX,NSCAM,KMAX,LMAX,LMM1                                        USC00030
      COMMON /BASPOT/U(65),SUM(65),WVL(16),NWVL,ALBED(16),BS(16),       USC00040
     +BE(16),SINGWV,PF(65),LLMAX                                        USC00050
      L=1                                                               USC00060
      LL=LMM1                                                           USC00070
      DO 1 K=1,KMAX                                                     USC00080
      LL=LL/2                                                           USC00090
      L=L+LL                                                            USC00100
      IF(SUM(L).GT.SC)L=L-LL                                            USC00110
    1 CONTINUE                                                          USC00120
      US=U(L)+(SC-SUM(L))*(U(L+1)-U(L))/(SUM(L+1)-SUM(L))               USC00130
      RETURN                                                            USC00140
      END                                                               USC00150
```

```
      SUBROUTINE SMOKE(WAVE1,ICLMAT,TRANS,IERR)                          SMK00010
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSMK00020
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,              SMK00030
     1      FOGPRB,WNDVEL,WNDDIR,IPASCT                                  SMK00040
      COMMON /GEOMET/ PTS(15),IGEOSW                                     SMK00050
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                      SMK00060
      DIMENSION IR(26),ITTL(11,5),TSUB(11),QDIV(11)                      SMK00070
      EXTERNAL JPASCT                                                    SMK00080
C
C**** NEW PROGRAM OPTIONS ADDED AS REVISIONS 03 & 04
C
C     THIS REVISION TO THE MODEL ALLOWS THE OPTION OF PRODUCING A ONE
C     DIMENSIONAL "SNAPSHOT" OF A SMOKE SCREEN DUE TO ANY NUMBER OF
C     MUNITIONS AT SOME SPECIFIED TIME; OUTPUT FOR THIS OPTION IS
C     CROSSWIND TRANSMISSION AS A FUNCTION OF DOWNWIND DISTANCE AT
C     A SINGLE GIVEN TIME RATHER THAN TRANSMISSION AS A FUNCTION
C     OF TIME AT A SINGLE GIVEN LINE OF SIGHT.
C
C     NEW INPUTS ARE ENTERED THROUGH THE "NAME" CARD AS:
C     NAME
C          STIME = SINGLE GIVEN TIME AT WHICH SCREEN IS TO BE SAMPLED
C          FRONT = LENGTH OF SCREEN TO BE SAMPLED (ALONGWIND)
C          DELX = INCREMENTS BETWEEN CONTIGUOUS LINES OF SIGHT
C          MCUOPT = OPTION TO SUPRESS INTERMEDIATE OUTPUT (1 = SUPRESS)
C
C     ALSO THE TIME OF DETONATION FOR EACH MUNITION IS REQUIRED (IN
C     SECONDS) AS THE FOURTH ENTRY ON THE "MUNC" CARD
C
C**** THE CAVEATS FOR THE NEW OPTION ARE:
C     1) THE LINES OF SIGHT MUST BE CROSSWIND
C     2) STARTING POINT FOR SAMPLING IS THE OBSERVER-TARGET
C        COORDINATES ENTERED ON THE "OBSC" AND "TARC" CARDS
C     3) THE PRNT OPTION ON THE "OUTP" CARD MUST BE ZERO
C     4) A PLOT FILE OPTION HAS BEEN ADDED AS THE FOURTH ENTRY ON
C        THE "OUTP" CARD (NPLT=1 WILL CREATE A PLOT FILE OF THE
C        FINAL RESULTS ON UNIT NPLOTU - SEE EOMAIN)
C                                                                        SMK00090
C  NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION        SMK00100
C        AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)           SMK00110
C         TO ALTER PERIOD, CHANGE MAXS AND DIMENSIONS OF SMAS, PVOL AND SMK00120
C         CLTOT.                                                         SMK00130
C                                                                        SMK00140
      COMMON /MOS/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9),       SMK00150
     *EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)         SMK00160
      DATA MAXS/960/                                                     SMK00170
      DATA QDIV /0.,.3,.50,.60,.65,.70,.80,1.0,1.,1.,1./                 SMK00180
C                                                                        SMK00190
C  NOTE: FOR SHORTER OR LONGER BURNS AND OBSCURATIONS, USERS MAY         SMK00200
C      SUBSTITUTE FOR THE COMMON BLOCK AND MAXS PARAMETER AND ALTER      SMK00210
C       DIMENSIONS FOR SMAS, PVOL AND CLTOT.                             SMK00220
C                                                                        SMK00230
C     SMOKE MODEL                                                        SMK00240
C     FORMAL INPUTS:                                                     SMK00250
C          WAVE1 = WAVELENGTH IN MICROMETERS (OR 94. GHZ) FROM ALLOWED   SMK00260
C                  BANDS (SEE EXTC RECORD BELOW).                        SMK00270
C          ICLMAT = CLIMATOLOGY FLAG. IF 1 THEN /CLYMAT/ VALUES OVER-    SMK00280
C                  RIDE METR RECORD VALUES.                              SMK00290
C          IGEOSW = GEOMETRY FLAG FROM COMMON GEOMET.  IF SET            SMK00300
C                  TO 1 IN EOMAIN, THEN OBSERVER AND TARGET              SMK00310
C                  COORDINATES ARE PASSED TO SMOKE, AND ANGLE            SMK00320
C                  XNORTH SET TO 90. DEGREES.                            SMK00330
C     FORMAL OUTPUTS:                                                    SMK00340
C          TRANS = TRANSMISSION AT TIME ETO (DEFAULT) OR USER            SMK00350
C                  SPECIFIED TIME (SEE OUTP RECORD BELOW) FOR            SMK00360
C                  WAVELENGTH WAVE1 ALONG THE OBSERVER-TARGET            SMK00370
C                  LINE-OF-SIGHT.                                        SMK00380
C          IERR  = ERROR FLAG.  1 IF ERROR IN SMOKE.                     SMK00390
C     USER RECORDS INPUT:                                                SMK00400
C       EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,          SMK00410
C       FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.             SMK00420
```

```
C       PER FIELD BEGINNING IN COL 11.  THE CARDS ARE NOT ORDER          SMK00430
C       DEPENDENT ALTHOUGH VALUES ON EACH RECORD MUST FOLLOW THE ORDER   SMK00440
C       SHOWN IN COMMENTS BELOW.       FORMAT (2A2,6X,7F10.3)            SMK00450
C                                                                        SMK00460
C THE FOLLOWING ARE REQUIRED RECORDS FOR AT LEAST ONE INPUT SET.         SMK00470
C                                                                        SMK00480
C      IDENT.   VARS.            DESCRIPTION                             SMK00490
C      ------   --------  ----------------------------------------------SMK00500
C                                                                        SMK00510
C       MUNC    XM,YM,ZM,TM  MUNITION COORDINATES AND EVENT TIME         SMK00520
C               (M,M,M,SEC)                                              SMK00530
C       BART                  COMPUTATION TIMES AND X-AXIS DEFINITION:   SMK00540
C              STO           OUTPUT STARTING TIME  (SEC. SINCE IGNITION) SMK00550
C              ETO           ENDING TIME FOR CALCULATION  (SEC.)         SMK00560
C              DTO           TIME INCREMENT FOR OUTPUT TABLES (SEC)      SMK00570
C              ANGLX         ANGLE OF POSITIVE X-AXIS WRT NORTH          SMK00580
C                            (DEG. CLOCKWISE WRT NORTH) ASSUMED 90 DEG. IFSMK00590
C                            IGEOSW IS 1 FROM EOMAIN                     SMK00600
C                                                                        SMK00610
C OPTIONAL DEPENDING ON PARAMETERS CHOSEN:                               SMK00620
C                                                                        SMK00630
C      IDENT.   VARS.            DESCRIPTION                             SMK00640
C      ------   --------  ----------------------------------------------SMK00650
C       OBSC    XO,YO,ZO      OBSERVER COORDINATES (IGNORED IF IGEOSW    SMK00660
C                             IS 1 FROM EOMAIN). (M,M,M)                 SMK00670
C       TARC    XT,YT,ZT      TARGET COORDINATES (IGNORED IF IGEOSW IS   SMK00680
C                             1 FROM EOMAIN)  (M,M,M)                    SMK00690
C       OUTP                OPTIONAL RECORD TO SELECT AMOUNT OF PRINT OUTPUTSMK00700
C                           AND TO SELECT CRITERIA FOR RETURNED          SMK00710
C                           TRANSMISSION.                                SMK00720
C              PRNT         IF 0., ALL FULL OUTPUT LISTINGS CREATED.     SMK00730
C                           IF 1. IS ENTERED AT ANY POINT, THEN ALL FURTHER SMK00740
C                           OUTPUT IS SUPPRESSED, EXCEPT FOR THE FINAL   SMK00750
C                           ACCUMULATED EFFECTS LISTING OF TOTAL CL AND  SMK00760
C                           TOTAL TRANSMITTANCE FOR COMBINED MULTIPLE INPUT SMK00770
C                           SETS.  (DEFAULT IS PRNT = 0.)                SMK00780
C              CRITER       SELECTS CHOICE OF TRANSMISSION RETURNED FROM SMK00790
C                           SMOKE.                                       SMK00800
C                             0. = RETURN TOTAL TRANSMITTANCE COMPUTED AT SMK00810
C                                  LAST TIME ETO. (DEFAULT CASE)         SMK00820
C                             1. = RETURN THE MINIMUM VALUE OF TOTAL     SMK00830
C                                  TRANSMITTANCE COMPUTED FOR WAVELENGTH SMK00840
C                                  WAVE1.                                SMK00850
C                             2. = RETURN VALUE OF TOTAL TRANSMITTANCE FOR SMK00860
C                                  WAVELENGTH WAVE1 COMPUTED AT USER-    SMK00870
C                                  SPECIFIED TIME TIMTRN BELOW.          SMK00880
C              TIMTRN       REQUIRED ONLY IF CRITER IS 2., TIME FOR WHICH SMK00890
C                           TRANSMITTANCE RETURNED IS COMPUTED.  SHOULD BE SMK00900
C                           CLOSE OR EQUAL TO A TABLE TIME (AS DETERMINED BYSMK00910
C                           THE BART RECORD) FOR ACCURACY.               SMK00920
C              IPLT         PLOT CODE ADDED IN ORDER TO PLOT OUTPUT ON UNIT
C                           NPLOTU IF IPLT = 1.                          SMK00930
C       MUNT                REQUIRED IF BURN RECORD IS NOT USED. OTHERWISE, SMK00940
C                           OPTIONAL.  ANY NON-ZERO VALUES INPUT WILL OVER-SMK00950
C                           RIDE PREVIOUS SOURCE DEFINITIONS (INCLUDING  SMK00960
C                           THOSE FROM THE BURN RECORD.)                 SMK00970
C                                                                        SMK00980
C              XN             NUMBER OF MUNITIONS IGNITED AT THE SAME   SMK00990
C                             LOCATION AND AT THE SAME TIME (DIMENSIONLESS)SMK01000
C              FW             FILL WEIGHT OF ONE MUNITION (LBS.) FOR    SMK01010
C                                 WP,PWP,HC OR RP (BUT RATE OF BURN      SMK01020
C                                 IN GAL./HR. FOR FOG OIL. NOTE 1 GAL/HR=SMK01030
C                                 0.93 G/S)                              SMK01040
C              TBURN          BURN DURATION FOR THIS MUNITION (SEC)     SMK01050
C              TYPE           TYPE OF SMOKE (DIMENSIONLESS)             SMK01060
C                        1.=WP, 2.=PWP OR WP WICK/WEDGE, 3.=HC,          SMK01070
C                        4.=FOG OIL, 5.=RP                               SMK01080
C              EFF            MUNITION BURN EFFICIENCY. (PERCENT)       SMK01090
```

```
                    IF INPUT AS 0., THEN DEFAULT IS USED.
          YF        YIELD FACTOR (DIMENSIONLESS). IF 0., DEFAULTS
                    TO RELATIVE HUMIDITY DEPENDENT STRAIGHT
                       LINE FIT FOR WP,PWP,RP FROM JOHNSON
                       AND FORNEY. SIMILARLY FOR HC. FOG OIL IS
                       SET TO 1.

METR                REQUIRED IF ICLMAT IS ZERO. OTHERWISE,
                    NEEDED ONLY FOR E OR F PASQUILL CATEGORY TO
                    PROVIDE TGRAD WHICH IS NOT AVAILABLE IN
                    /CLYMAT/.

          RELHUM    RELATIVE HUMIDITY    (PERCENT)
          UW        WIND VELOCITY (M/SEC)
          WNDIR     WIND DIRECTION (USUAL MET CONVENTION,
                    ANGLE IN DEG. CLOCKWISE FROM NORTH OF
                    DIRECTION FROM WHICH WIND ORIGINATES.)
          PCAT      PASQUILL CATEGORY (DIMENSIONLESS)
                      1.-A, 2.-B, 3.-C, 4.-D, 5.-E, 6.-F
          AIRT      SURFACE AIR TEMPERATURE    (DEG C)
          TGRAD     VERT TEMP GRADIENT (C DEG/M).
                    EXAMPLE: TGRAD=(AIRT(10 M)-AIRT(.5 M))/9.5 M
                       (USED ONLY FOR PASQUILL CATEGORIES E, F)

EXTC                OPTIONAL USER OVERRIDE FOR EXTINCTION
                    COEFFICIENTS.
                     IF RECORD NOT USED, OR FOR ANY VALUES READ
                     IN AS 0., THE EXTINCTION COEFF.
                    DEFAULTS TO ALPHA ARRAY VALUE IN STRANS.

                  INPUT EXTINCTION COEFF. (M**2/G) ORDER ON CARD
                    CORRESPONDS TO THE BANDS:
                        0.4-0.7 MICROMETERS
                        0.7-1.2 MICROMETERS
                          1.06  MICROMETERS
                        3.0-5.0 MICROMETERS
                        8.0-12. MICROMETERS
                          10.6  MICROMETERS
                          94.0  GHZ.

BURN          OPTIONAL -  SELECTS BUILT-IN MUNITION CHARACTER-
              ISTICS FROM THE BRATE ROUTINE FOR ONE (XN=1)
              MUNITION. VALUES ARE FOR FILL WEIGHT (FW), BURN
              DURATION (TBURN), SMOKE TYPE (ITYPE), EFFICIENCY
              (EFF).  YIELD FACTOR IS SET TO ZERO SO THAT RH
              MODEL DEPENDENT VALUES ARE USED. ANY VALUES READ
              IN AS NON-ZERO ON A MUNT RECORD (WHICH IS
              OPTIONAL IF A BURN CARD IS USED) WILL OVERRIDE
              THE DEFAULTS STORED IN BRATE.

          TYPM      MUNITION TYPE:
                      0. = USER DEFINED MUNITION SOURCE CHAR.
                      1. = 155MM HC, M1 CANISTER.
                      2. = 155MM HC, M2 CANISTER.
                      3. = 105MM HC CANISTER.
                      4. = 155MM HC M116B1 PROJ.
                      5. = 105MM HC M84A1  PROJ.
                      6. = SMOKE POT HC M5
                      7. = 60MM  WP M302 CARTRIDGE
                      8. = 81MM  WP M375A2
                      9. = 4.2 IN WP M328A1
                     10. = 155MM WP M110E2
                     11. = 105MM WP M60A2
                     12. = 4.2 IN PWP M328A1
                     13. = 5.  IN PWP ZUNI MK4
                     14. = 2.75 IN WP WEDGE SUB-MUNITION.
                     15. = 3.   IN WP WICK  SUB-MUNITION
                     16. = 6.   IN WP WICK  SUB-MUNITION
                     17. = 155MM WP WEDGE XM825 (92 SUB-MUN.)
                     18. = 81MM RP WEDGE NAVY SUB-MUNITION
```

```
C                              19. = 81MM RP WEDGE GERMAN SUB-MUNITION    SMK01800
C                              20. = 155MM RP WEDGE XM803 (228 SUB-MUN.)  SMK01810
C                              21. = FOG OIL GENERATOR (EXAMPLE)          SMK01820
C                 BRAT1        BURN-RATE COEFF. (CONST. TERM)             SMK01830
C                 BRAT2        BURN-RATE COEFF. (LINEAR TERM)             SMK01840
C                 BRAT3        BURN-RATE COEFF. (QUADRATIC TERM)          SMK01850
C                 BRAT4        BURN-RATE COEFF. (CUBIC TERM)              SMK01860
C                                 RATE COEFF. ARE BASED ON THE SCALED     SMK01870
C                                 POLYNOMIAL DESCRIBING THE FRACTIONAL    SMK01880
C                                 MASS BURNED AT SCALED TIMES FROM 0.     SMK01890
C                                 (IGNITION) TO 1. (BURN-OUT). THE INTEGRAL SMK01900
C                                 OF THE POLYNOMIAL FROM 0. TO 1. MUST    SMK01910
C                                 BE 1., IE. THE SCALED TOTAL MASS BURNED SMK01920
C                                 BY TIME 1. IS 100 PERCENT.  THE PROGRAM SMK01930
C                                 WILL CHECK FOR THIS CONDITION AND RE-SCALESMK01940
C                                 COEFF. IF NECESSARY.                    SMK01950
C                 BRAT5        SPECIAL BURN RATE COEFFICIENT FOR ESTIMATING SMK01960
C                                 THE RAPID BURST FOLLOWED BY SLOW BURN OF SMK01970
C                                 SOME MUNITIONS.  THE FUNCTIONAL FORM OF SMK01980
C                                 THE TERM MULTIPLYIED BY BRAT5 IS        SMK01990
C                       .07818 * (358800. / (1. + 358800. T/TBURN))       SMK02000
C                                 THOSE MUNITIONS FOR WHICH THE RATE IS NOT SMK02010
C                                 MODELED WILL RECEIVE BRAT1=1., BRAT2,3,4,5 =0SMK02020
C                                 (IE., CONSTANT BURN-RATE). THE USER MAY SMK02030
C                                 OVERRIDE THE DEFAULT RATES BY ENTERING VALUESSMK02040
C                                 ON THE BURN CARD.                       SMK02050
C        NAME           SCREEN PARAMETERS
C                 STIME        TIME AT WHICH SCREEN IS EXAMINED
C                 FRONT        SCREEN FRONTAGE TO BE EXAMINED
C                 DELX         INCREMENTAL SPACING FOR SCREEN EXAMINATION
C                 MCUOPT       OPTION TO SUPRESS INTERMEDIATE OUTPUT WITH
C                                 MCU STUDIES OPTION (MCUOPT=1 WILL SUPRESS)
C                                                                         SMK02060
C  THE FOLLOWING IDENT RECORDS ARE ALWAYS REQUIRED.                       SMK02070
C                                                                         SMK02080
C        GO             SIGNIFIES TO BEGIN EXECUTION FOR THIS DATA SET.  SMK02090
C                       AFTER EXECUTION, ANOTHER SET OF INPUTS MAY BE     SMK02100
C                       READ-IN FOLLOWED BY ANOTHER 'GO' CARD.            SMK02110
C                       ANY VALUES ESTABLISHED FROM PREVIOUS INPUT SETS   SMK02120
C                       TO THE ROUTINE ARE STILL IN EFFECT.  THUS DATA    SMK02130
C                       SUCH AS FROM CARD OBSC NEED NOT BE READ AGAIN IF  SMK02140
C                       THERE ARE TO BE NO CHANGES TO OBSERVER COORD.,ETC. SMK02150
C                                                                         SMK02160
C        DONE           *** MUST *** BE THE LAST RECORD READ.  DESIGNATES SMK02170
C                       LISTING OF ACCUMULATED EFFECTS IF MORE THAN ONE   SMK02180
C                       INPUT SET (DELINEATED BY GO CARDS) WERE INPUT.    SMK02190
C                       ALSO RETURNS CONTOL TO EOSAEL EXEC MODULE.        SMK02200
C                       NOTE THAT A 'DONE' CARD CAN BE USED IN PLACE OF   SMK02210
C                       THE FINAL 'GO' CARD IF DESIRED.                   SMK02220
C                                                                         SMK02230
C       OUTPUTS                                                           SMK02240
C            LIST OF INPUT PARAMETERS AND                                 SMK02250
C            AT EACH COMPUTATION TIME, TO, THE FOLLOWING:                 SMK02260
C                                                                         SMK02270
C                 X             CLOUD LENGTH ALONG WIND VECTOR    (M)     SMK02280
C                 Z             CLOUD LEADING EDGE HEIGHT ALONG WIND VECTOR SMK02290
C                                 (M)                                     SMK02300
C                 YFULL         CLOUD LEADING EDGE FULL-WIDTH PERPENDICULAR SMK02310
C                                 TO WIND VECTOR (M).                     SMK02320
C                 PATHL         SMOKE LENGTH OF 1 MUNITION ALONG OBS-TGT LOS SMK02330
C                                 (M)                                     SMK02340
C                 CL            TOTAL CL OF SMOKE ALONG OBS-TGT LOS (GM/M**2)SMK02350
C                                 FOR XN MUNITIONS.                       SMK02360
C                 SMTRAN        TRANSMISSION IN SPECTRAL BANDS OF SMOKE ALONGSMK02370
C                                 LOS                                     SMK02380
C                                                                         SMK02390
C    SUBROUTINES CALLED...   DIRECTLY: CLSMOK, SCONST, SMASSP, JPASCT,    SMK02400
C                 WGGEOM, STRANS, BRATE..  INDIRECTLY: XYZINT, QROOT.     SMK02410
C                                                                         SMK02420
      DATA ITTL/ 2HWH,2HIT,2HE ,2HPH,2HOS,2HPH,2HOR,2HUS,2H (,2HWP,2H) ,SMK02430
```

```
     * 2HPW,2HP  ,2HOR,2H W,2HP  ,2HWI,2HCK,2H/W,2HED,2HGE,2H   ,                SMK024
     * 2HHC,2H S,2HMO,2HKE,2H   ,2H   ,2H   ,2H   ,2H   ,2H   ,2H   ,           SMK024
     * 2HFO,2HG  ,2HOI,2HL  ,2HSM,2HOK,2HE  ,2H   ,2H   ,2H   ,2H   ,           SMK024
     * 2HRE,2HD  ,2HPH,2HOS,2HPH,2HOR,2HUS,2H (,2HRP,2H) ,2H   /                SMK02470
      DATA IR /2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA,          SMK02480
     *2HRC,2HEX,2HTC,2HGO,2H   ,2HBU,2HRN  ,2HMI,2HSC,2HOU,2HTP,2HDO,           SMK02490
     *2HNE  ,2HNA,2HME /                                                        SMK02500
      DATA BRAT1,BRAT2,BRAT3,BRAT4,BRAT5/1.,0.,0.,0.,1./                        SMK02510
      DATA TYPM,XN,FW,TBURN,EFF,YF,RELHUM,UW,WNDIR,ANGLX,PCAT,                  SMK02520
     *AIRT,TGRAD /13*0./                                                        SMK02530
      DATA ITGRAD/0/                                                            SMK02540
      BFUN(T)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*BRAT4/4.)))                    SMK02550
     *+BRAT5*.07818288*ALOG(1.+358800.*T)                                       SMK02560
      NUMDIV=8                                                                  SMK02570
      NRUNS=0                                                                   SMK02580
      TIMTRN=1.                                                                 SMK02590
      ICRTR=0                                                                   SMK02600
      TRANS=1.                                                                  SMK02610
      MCUOPT=0
      NOPRNT=0                                                                  SMK02620
      NOMORE=0                                                                  SMK02630
      NCY=0                                                                     SMK02640
      DO 1 J=1,MAXS                                                             SMK02650
      CLTOT(J)=0.                                                               SMK02660
1     CONTINUE                                                                  SMK02670
      DO 2 J=1,8                                                                SMK02680
    2 EXTC(J)=0.                                                                SMK02690
 5    CONTINUE                                                                  SMK02700
      NRUNS=NRUNS+1                                                             SMK02710
      MUNRD=0                                                                   SMK02720
      KWAVE=0                                                                   SMK02730
      IF (WAVE1.GE.0.4.AND.WAVE1.LT.0.7) KWAVE=1                                SMK02740
      IF (WAVE1.GE.0.7.AND.WAVE1.LE.1.2) KWAVE=2                                SMK02750
      IF (WAVE1.GE.3.0.AND.WAVE1.LE.5.0) KWAVE=4                                SMK02760
      IF (WAVE1.GE.8.0.AND.WAVE1.LE.12.0) KWAVE=5                               SMK02770
      IF (WAVE1.GT.1.059.AND.WAVE1.LT.1.061) KWAVE=3                            SMK02780
      IF (WAVE1.GT.10.59.AND.WAVE1.LT.10.61) KWAVE=6                            SMK02790
      IF (WAVE1.GT.93.9.AND.WAVE1.LT.94.1) KWAVE=7                              SMK02800
      IF (WAVE1.GT.3188..AND.WAVE1.LT.3195.) KWAVE=7                            SMK02810
      IF (KWAVE.EQ.0) GOTO 998                                                  SMK02820
C***  BEGINNING OF READ LOOP                                                    SMK02830
      NCHK=0                                                                    SMK02840
      DO 70 I = 1, 15                                                           SMK02850
           IF(I.EQ.15) GO TO 310                                                SMK02860
      READ(IOIN,20) IR1,IR2,(R1(J),J=1,7)                                       SMK02870
 20        FORMAT(2A2,6X,7F10.3)                                                SMK02880
C***  RELATING INPUT DATA TO VARIABLE NAMES.                                    SMK02890
           IF(IR1.EQ.IR(1).AND.IR2.EQ.IR(2)) GOTO 90                            SMK02900
           IF(IR1.EQ.IR(3).AND.IR2.EQ.IR(4)) GOTO 100                           SMK02910
           IF(IR1.EQ.IR(5).AND.IR2.EQ.IR(6)) GOTO 110                           SMK02920
           IF(IR1.EQ.IR(7).AND.IR2.EQ.IR(8)) GOTO 120                           SMK02930
           IF(IR1.EQ.IR(9).AND.IR2.EQ.IR(10)) GOTO 130                          SMK02940
           IF(IR1.EQ.IR(11).AND.IR2.EQ.IR(12)) GOTO 140                         SMK02950
           IF(IR1.EQ.IR(13).AND.IR2.EQ.IR(14)) GOTO 150                         SMK02960
           IF(IR1.EQ.IR(15).AND.IR2.EQ.IR(16)) GOTO 155                         SMK02970
           IF(IR1.EQ.IR(17).AND.IR2.EQ.IR(18)) GOTO 105                         SMK02980
           IF(IR1.EQ.IR(19).AND.IR2.EQ.IR(20)) GOTO 70                          SMK02990
           IF(IR1.EQ.IR(21).AND.IR2.EQ.IR(22)) GOTO 115                         SMK03000
           IF(IR1.EQ.IR(23).AND.IR2.EQ.IR(24)) GOTO 154                         SMK03010
           IF(IR1.EQ.IR(25).AND.IR2.EQ.IR(26)) GOTO 121                         SMK03020
           WRITE(IOOUT,80)                                                      SMK03030
 80        FORMAT(1H ,72HINVALID DATA CARD-DOES NOT CONFORM TO PROPER COSMK03040
     *NVENTION IN SMOKE ROUTINE)                                                SMK03050
      WRITE(IOOUT,30) IR1,IR2,(R1(J),J=1,7)                                     SMK03060
 30        FORMAT(1H ,2A2,6X,7F10.3)                                           SMK03070
                                GO TO 999                                       SMK03080
90    IF (ICLMAT.EQ.1) GOTO 92                                                  SMK03090
       IF (R1(1).NE.0.)     RELHUM= R1(1)                                       SMK03100
       IF (R1(2).NE.0.)     UW    = R1(2)                                       SMK03110
       IF (R1(3).NE.0..OR.R1(2).NE.0.)    WNDIR = R1(3)                        SMK03120
```

```
       IF (R1(4).NE.0.)    PCAT   = R1(4)                                    SMK03130
      ICAT=IFIX(PCAT+.0001)                                                  SMK03140
           AIRT   = R1(5)                                                    SMK03150
92         TGRAD  = R1(6)                                                   SMK03160
           ITGRAD=1                                                         SMK03170
           NCHK=1                                                           SMK03180
                                GO TO 70                                    SMK03190
 100       IF (R1(1).NE.0.) XN = R1(1)                                      SMK03200
           IF (R1(2).NE.0.) FW = R1(2)                                      SMK03210
           IF (R1(3).NE.0.) TBURN = R1(3)                                   SMK03220
           IF (R1(4).NE.0.) ITYPE = IFIX(R1(4)+.0001)                       SMK03230
           IF (MUNRD.EQ.0.OR.R1(5).NE.0.) EFF    = R1(5)                    SMK03240
           YF=R1(6)                                                         SMK03250
      MUNRD=1                                                               SMK03260
           NCHK=1                                                           SMK03270
                                GO TO 70                                    SMK03280
105   TYPM=R1(1)                                                            SMK03290
      CALL BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,BRAT1,            SMK03300
     *BRAT2,BRAT3,BRAT4,BRAT5)                                              SMK03310
      IF (IERR.NE.0) WRITE (IOOUT,95) TYPM                                  SMK03320
95    FORMAT(37H IN SMOKE, ILLEGAL MUNITION TYPE READ ,F5.0)                SMK03330
      IF (IERR.NE.0) GOTO 999                                               SMK03340
      IF (R1(2).EQ.0..AND.R1(3).EQ.0..AND.R1(4).EQ.0..AND.R1(5).EQ.0.)      SMK03350
     *                                  GOTO 93                            SMK03360
      BRAT1=R1(2)                                                           SMK03370
      BRAT2=R1(3)                                                           SMK03380
      BRAT3=R1(4)                                                           SMK03390
      BRAT4=R1(5)                                                           SMK03400
      BRAT5=R1(6)                                                           SMK03410
93    IF (TYPM.GT.0.) MUNRD=1                                               SMK03420
      NCHK=1                                                                SMK03430
                                GO TO 70                                    SMK03440
 110       ISTO   = IFIX(R1(1)+.0001)                                       SMK03450
           IETO   = IFIX(R1(2)+.0001)                                       SMK03460
           IDTO   = IFIX(R1(3)+.0001)                                       SMK03470
           ANGLX=R1(4)                                                      SMK03480
      NCHK=1                                                                SMK03490
                                GO TO 70                                    SMK03500
115   NOPRNT=0                                                              SMK03510
      IF (R1(1).NE.0.) NOPRNT=1                                             SMK03520
      CRITER=R1(2)                                                          SMK03530
      ICRTR=IFIX(CRITER+.001)                                               SMK03540
      IF (ICRTR.GT.2) ICRTR=2                                               SMK03550
      IF (ICRTR.LT.0) ICRTR=0                                               SMK03560
      IF (ICRTR.EQ.2) TIMTRN=R1(3)                                          SMK03570
      IPLT=IFIX(R1(4))                                                      SMK03580
      GOTO 70                                                               SMK03590
 120       XM     = R1(1)                                                  SMK03600
           YM     = R1(2)                                                  SMK03610
           ZM     = R1(3)                                                  SMK03620
           TM     = R1(4)                                                  SMK03630
      NCHK=1                                                                SMK03640
                                GO TO 70                                    SMK03650
 121       MODE=1                                                           SMK03660
           STIME  =R1(1)                                                    SMK03670
           ISRN   =IFIX(STIME+0.0001)                                       SMK03680
           FRONT  =R1(2)                                                    SMK03690
           DELX   =R1(3)                                                    SMK03700
           IF(DELX.LE.0.0) DELX=5.0
           XXX=FRONT/DELX
           NPTS=IFIX(XXX)+1
      MCUOPT=IFIX(R1(4))
                                GO TO 70
 130       XO     = R1(1)
           YO     = R1(2)
           ZO     = R1(3)
      NCHK=1
                                GO TO 70
 140       XT     = R1(1)
           YT     = R1(2)
```

```
          ZT     = R1(3)                                                  SMK03
      NCHK=1                                                              SMK03
                                  GO TO 70                                SMK03
  150 DO 152 J=1,7                                                        SMK03
  152 EXTC(J)=R1(J+1)                                                     SMK03
      EXTC(8)=0.                                                          SMK03
   70 CONTINUE                                                            SMK03
  154 NOMORE=1                                                            SMK03
      IF (NCHK.EQ.0) NRUNS=NRUNS-1                                        SMK03
      IF (NCHK.EQ.0) GOTO 980                                             SMK03
  155 CONTINUE                                                            SMK03
C******REDEFINE MUNITION EFFICIENCY IF INPUT AS ZERO                     SMK03
      IF(EFF.GT.0.0)GO TO 11                                              SMK03
      IF(ITYPE.EQ.1)EFF=100.                                              SMK03
      IF(ITYPE.EQ.2)EFF=65.0                                              SMK03
      IF(ITYPE.EQ.3)EFF=40.0                                              SMK03
      IF(ITYPE.EQ.4)EFF=100.0                                             SMK03
      IF(ITYPE.EQ.5)EFF=50.                                               SMK03
   11 CONTINUE                                                            SMK03
      IF (ICLMAT.NE.1) GOTO 12                                            SMK03
      RELHUM=RH                                                           SMK03
      UW    =WNDVEL                                                       SMK03
      WNDIR=WNDDIR                                                        SMK03
      ICAT  =IPASCT                                                       SMK03
      PCAT=FLOAT(ICAT)                                                    SMK03
      AIRT  =TEMP                                                         SMK03
   12 CONTINUE                                                            SMK03
      IF (IGEOSW.NE.1) GOTO 13                                            SMK03
      DISKTM=1000.
C...  CONVERT UNITS FROM KM TO M.
      XT=PTS(1)*DISKTM                                                    SMK03
      YT=PTS(2)*DISKTM                                                    SMK04
      ZT=PTS(3)*DISKTM                                                    SMK04
      XO=PTS(4)*DISKTM                                                    SMK04
      YO=PTS(5)*DISKTM                                                    SMK04
      ZO=PTS(6)*DISKTM                                                    SMK04
      ANGLX=90.                                                           SMK04
13    CONTINUE                                                            SMK04
      IF (ITYPE.LT.1.OR.ITYPE.GT.5) IERR=1                                SMK04
      IF (IERR.EQ.1) WRITE (IOOUT,180) ITYPE                              SMK04
180   FORMAT(1X,31HIN SMOKE, INVALID SMOKE TYPE = ,I4)                    SMK04
      IF (IERR.EQ.1) GOTO 999                                             SMK04
C***  CHECK BURN RATE FOR 100 PERCENT BURN AT TBURN...                    SMK04
      IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0..AND.BRAT5.EQ.0.)    SMK04
     *BRAT1=1.                                                            SMK04
      VNORM=BFUN(1.)                                                      SMK04
      BRAT1=BRAT1/VNORM                                                   SMK04
      BRAT2=BRAT2/VNORM                                                   SMK04
      BRAT3=BRAT3/VNORM                                                   SMK04
      BRAT4=BRAT4/VNORM                                                   SMK04
      BRAT5=BRAT5/VNORM                                                   SMK04
      IF (XN.LE.0.) XN=1.                                                 SMK04
      IF (ITYPE.EQ.1.AND.TBURN.GT.1.) TBURN=1.                            SMK04
C***  SET UP EXTINCTION COEFF TO BE USED...                               SMK04
      CALL STRANS(CL,SMTRAN,ITYPE,EXTC,0)                                 SMK04
      CALL SMASSP(XN,EFF,FW,RELHUM, W,ITYPE,YF)                           SMK04
      TGR=TGRAD                                                           SMK04
      TGRAD=ABS(TGRAD)                                                    SMK04
      IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 255                            SMK04
      IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 255
      IF(NRUNS.GT.1) WRITE (IOOUT,172) NRUNS                              SMK04
      IF(NRUNS.EQ.1) WRITE (IOOUT,1720) NRUNS                             SMK04
172   FORMAT(1H1,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,       SMK04
     *1H*,15X,9HEXECUTION ,I3/51X,1H*,15X,1H*/51X,17(1H*)/)               SMK04
1720  FORMAT(1H0,50X,17(1H*)/51X,1H*,15X,1H*/51X,1H*,5X,5HSMOKE,5X,       SMK04
     *1H*,15X,9HEXECUTION ,I3/51X,1H*,15X,1H*/51X,17(1H*)/)               SMK04
C***  REPORTING INPUT DATA.                                               SMK04
      IF(ICAT.GE.5.AND.ITYPE.NE.4.AND.ITGRAD.EQ.0)WRITE (IOOUT,98) TGRADSMK04
   98 FORMAT(1X,44HIN SMOKE ROUTINE PASQUILL CATEGORIES E AND F,          SMK04
     *29H REQUIRE TEMPERATURE GRADIENT/10X,23HIF SMOKE IS EXOTHERMIC.,    SMK04
```

```
     *20H  A VALUE OF TGRAD = ,F7.2,25H C DEG/M WILL BE ASSUMED./)       SMK04380
      WRITE(IOOUT,170)                                                   SMK04390
  170  FORMAT(5X,15HSMOKE MUNITIONS,22X,25HMETEOROLOGICAL CONDITIONS,25X,SMK04400
     123HEXTINCTION COEFFICIENTS)                                        SMK04410
      WRITE(IOOUT,190) (ITTL(J,ITYPE),J=1,11),UW,EXTC(1)                 SMK04420
  190  FORMAT(3X,11A2,11X,10HWINDSPEED ,14X,F5.1,2X,3HM/S,16X,           SMK04430
     120H0.4-0.7 MICROMETERS ,F7.3,2X,7HM**2/GM)                         SMK04440
      WRITE(IOOUT,200) XN,EXTC(2)                                        SMK04450
  200  FORMAT(3X,10HNO. ROUNDS,1X,F5.0,17X,22HWIND DIRECTION (USUAL ,28X,SMK04460
     120H0.7-1.2 MICROMETERS ,F7.3,2X,7HM**2/GM)                         SMK04470
      IF (ITYPE.NE.4) WRITE(IOOUT,210) FW,WNDIR,EXTC(3)                  SMK04480
  210  FORMAT(3X,11HFILL WEIGHT,F8.3,3H LB,11X,23HMET CONVENTION AZIMUTH)SMK04490
     1,F6.1,2X,7HDEGREES,12X,20H1.06    MICROMETERS ,F7.3,2X,7HM**2/GM) SMK04500
      IF (ITYPE.EQ.4) WRITE(IOOUT,215) FW,WNDIR,EXTC(3)                  SMK04510
   215 FORMAT(3X,9HBURN RATE,2X,F6.1,2X,6HGAL/HR,  8X,23HMET CONVENTION AZSMK04520
     1IMUTH) ,F6.1,2X,7HDEGREES,12X,20H1.06    MICROMETERS ,F7.3,2X,     SMK04530
     27HM**2/GM)                                                         SMK04540
      WRITE(IOOUT,220)TBURN,RELHUM,EXTC(4)                               SMK04550
  220  FORMAT(3X,9HBURN TIME,F8.1,2X,3HSEC,11X,17HRELATIVE HUMIDITY,7X,  SMK04560
     1F5.1,2X,7HPERCENT,12X,20H3.0-5.0 MICROMETERS ,F7.3,2X,7HM**2/GM)   SMK04570
      LCAT=JPASCT(ICAT)                                                  SMK04580
      WRITE(IOOUT,230) EFF,LCAT,EXTC(5)                                  SMK04590
   230 FORMAT(3X,10HEFFICIENCY,1X,F6.1,2X,7HPERCENT,7X,17HPASQUILL CATEGOSMK04600
     1RY,9X,A1,23X,20H8.0-12. MICROMETERS ,F7.3,2X,7HM**2/GM)            SMK04610
      WRITE (IOOUT,235) YF,AIRT,EXTC(6)                                  SMK04620
  235  FORMAT(3X,12HYIELD FACTOR,F6.2,15X,15HAIR TEMPERATURE,8X,F6.1,2X,  SMK04630
     18HDEGREE C,11X,20H10.6    MICROMETERS ,F7.3,2X,7HM**2/GM)          SMK04640
      WRITE(IOOUT,240)TGRAD,EXTC(7)                                      SMK04650
   240 FORMAT(36X,14HTEMP. GRADIENT,10X,F6.2,9H C DEG./M,11X,            SMK04660
     111H94.0    GHZ,9X,F7.3,2X,7HM**2/GM)                               SMK04670
      IF (ICAT.GE.5.AND.TGR.LT.0.) WRITE(IOOUT,250)                      SMK04680
   250  FORMAT(36X,24H(ASSUMED POSITIVE INPUT))                           SMK04690
255    NRAT=0                                                             SMK04700
       IF (BRAT1.NE.0.) NRAT=1                                           SMK04710
       IF (BRAT2.NE.0.) NRAT=2                                           SMK04720
       IF (BRAT3.NE.0.) NRAT=3                                           SMK04730
       IF (BRAT4.NE.0.) NRAT=4                                           SMK04740
       IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 258                          SMK04750
       IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GOTO 258
       IF (NRAT.EQ.1) WRITE (IOOUT,175) BRAT1                            SMK04760
       IF (NRAT.EQ.2) WRITE (IOOUT,176) BRAT1,BRAT2                      SMK04770
       IF (NRAT.EQ.3) WRITE (IOOUT,177) BRAT1,BRAT2,BRAT3                SMK04780
       IF (NRAT.EQ.4) WRITE (IOOUT,178) BRAT1,BRAT2,BRAT3,BRAT4          SMK04790
       IF (NRAT.EQ.0) WRITE (IOOUT,181)                                  SMK04800
181    FORMAT(40X,19HBURN RATE PROFILE =)                                 SMK04810
       IF (BRAT5.NE.0.) NRAT=5                                           SMK04820
       NPWP=0                                                            SMK04830
       IF ((ITYPE.EQ.2.OR.ITYPE.EQ.5).AND.NRAT.GT.1) NPWP=1              SMK04840
       IF (NRAT.EQ.5) WRITE (IOOUT,179) BRAT5                            SMK04850
179    FORMAT(40X,2H+ ,F8.4,41H*0.0781829*(358800./(1.+358800.*T/TBURN)))SMK04860
175    FORMAT(1H0,40X,19HBURN RATE PROFILE =,F8.4)                       SMK04870
176    FORMAT(1H0,31X,19HBURN RATE PROFILE =,F8.4,2H +,                  SMK04880
     *F8.4,10H (T/TBURN))                                                SMK04890
177    FORMAT(1H0,20X,19HBURN RATE PROFILE =,F8.4,2H +,                  SMK04900
     *F8.4,12H (T/TBURN) +,F8.4,13H (T/TBURN)**2)                        SMK04910
178    FORMAT(1H0, 9X,19HBURN RATE PROFILE =,F8.4,2H+ ,                  SMK04920
     *F8.4,12H (T/TBURN) +,F8.4,15H (T/TBURN)**2 +,F8.4,                 SMK04930
     *13H (T/TBURN)**3)                                                  SMK04940
C*** PROVIDE COORDINATES...                                             SMK04950
258    ANGL=ANGLX+180.-WNDIR                                             SMK04960
       IF (ANGL.GT.360.) ANGL=ANGL-360.                                  SMK04970
       IF (ANGL.LT.0.  ) ANGL=ANGL+360.                                  SMK04980
       IF (NOPRNT.EQ.1.AND.NRUNS.GT.1) GOTO 285                          SMK04990
       XP1=XO-XM                                                         SMK05000
       XP2=XT-XM                                                         SMK05010
       YP1=YO-YM                                                         SMK05020
       YP2=YT-YM                                                         SMK05030
       ZPP1=ZO-ZM                                                        SMK05040
       ZPP2=ZT-ZM                                                        SMK05050
       CA=COS(ANGL*PIRAD)                                                SMK05060
```

```
      SA=SIN(ANGL*PIRAD)                                                SMK05070
      XPP3=0.                                                           SMK05080
      YPP3=0.                                                           SMK05090
      ZPP3=0.                                                           SMK05100
      XPP1=XP1*CA+YP1*SA                                                SMK05110
      YPP1=YP1*CA-XP1*SA                                                SMK05120
      XPP2=XP2*CA+YP2*SA                                                SMK05130
      YPP2=YP2*CA-XP2*SA                                                SMK05140
      IF (MCUOPT.EQ.1.AND.NRUNS.GT.1) GO TO 285
      WRITE (IOOUT,260)                                                 SMK05150
      WRITE (IOOUT,270) XM,YM,ZM,XPP3,YPP3,ZPP3,XO,YO,ZO,XPP1,YPP1,     SMK05160
     1ZPP1,XT,YT,ZT,XPP2,YPP2,ZPP2                                      SMK05170
  260 FORMAT(1H0,30X,17HFIELD COORDINATES,20X,43HROTATED COORD.(WIND X-ASMK05180
     1XIS,MUNITION ORIGIN)/27X,3H(X),8X,3H(Y),8X,3H(Z),20X,4H(XW),      SMK05190
     27X,4H(YW),7X,4H(ZW))                                              SMK05200
  270 FORMAT(1X,22HMUNITION COORDINATES= ,3(F9.2,2X),6HMETERS,6X,       SMK05210
     13(F9.2,2X),6HMETERS/1X,22HOBSERVER COORDINATES= ,3(F9.2,2X),      SMK05220
     26HMETERS,6X,3(F9.2,2X),6HMETERS/1X,22HTARGET   COORDINATES= ,     SMK05230
     33(F9.2,2X),6HMETERS,6X,3(F9.2,2X),6HMETERS)                       SMK05240
      WRITE (IOOUT,275) ANGLX,TM                                        SMK05250
  275 FORMAT(1X,48HANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = ,    SMK05260
     *F7.2,5H DEG.,/,5X,12HEVENT TIME =,F6.1,4H SEC)                    SMK05270
      IF(MODE.GT.0) WRITE(IOOUT,1000) MODE,STIME,FRONT,DELX
 1000 FORMAT(5X,5HMODE=,I2,/,5X,12HSCREEN TIME=,F6.1,/,5X,6HFRONT=,F6.1,
     */,5X,11HINCREMENT= ,F6.1)
      IF (NOPRNT.EQ.1.OR.MCUOPT.EQ.1) GO TO 285
      IF(MODE.EQ.0) WRITE(IOOUT,280)                                    SMK05280
      IF(MODE.GT.0) WRITE(IOOUT,281)
  280 FORMAT(1H0,6X,4HTIME,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,       SMK05290
     110HPATHLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,7X,5H(SEC),3(1X,8H(METSMK05300
     2ERS)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMETSMK05310
     3ERS),/,60X,39H0.4-0.7 0.7-1.2   1.06  3.0-5.0 8.0-12.,          SMK05320
     415H   10.6  94.GHZ,/)                                           SMK05330
  281 FORMAT(1H0,6X,5H*LOS*,3X,6HLENGTH,3X,5HWIDTH,4X,6HHEIGHT,2X,10HPAT
     1HLENGTH,4X,2HCL,24X,12HTRANSMISSION,/,4X,8H(METERS),3(1X,8H(METERS
     2)),2X,8H(METERS),2X,9H(GM/M**2),12X,28HSPECTRAL BANDS (MICROMETERS
     3),/,60X,40H0.4-0.7  0.7-1.2  1.06  3.0-5.0  8.0-12.,
     415H   10.6  94.GHZ/)
C***  BEGINNING OF CALCULATIONS.                                        SMK05340
  285 IF (W.EQ.0.) GOTO 999                                             SMK07370
      IF (ICAT.LT.1.OR.ICAT.GT.6) GOTO 999                              SMK05360
      IF(IDTO.EQ.0)   IDTO = 1                                          SMK05370
      IF (ISTO.LE.0) ISTO=IDTO                                          SMK05380
      IF (IETO.LE.ISTO) IETO=ISTO                                       SMK05390
C***  CALCULATIONS DEPENDENT ON TIME.                                   SMK05400
C***  SET COMPUTATION TIME STEP DTIME TO 1 SEC. FOR HC/FOG OIL/PWP/     SMK05410
C     RP AND WP WICKS/WEDGES                                            SMK05420
C***    BUT TO TABLE REPORT TIME INCREMENT FOR WP                       SMK05430
      DTIME=FLOAT(IDTO)                                                 SMK05440
      TRANS=1.0                                                         SMK05450
      TSUB(1)=0.                                                        SMK05460
      TSUB(2)=TBURN                                                     SMK05470
      IF (NPWP.EQ.0) GOTO 620                                           SMK05480
      IS=IFIX(5.*TBURN+.0001)                                           SMK05490
      IF (IS.LT.1) GOTO 610                                             SMK05500
      DO 286 I=2,NUMDIV                                                 SMK05510
      TSUB(I)=1./5.                                                     SMK05520
  286 CONTINUE                                                          SMK05530
      DO 600 I=1,IS                                                     SMK05540
      FI=FLOAT(I)/5.                                                    SMK05550
      T=FI/TBURN                                                        SMK05560
      TMS=BFUN(T)                                                       SMK05570
      DO 601 JDIV=2,NUMDIV                                              SMK05580
      IF (TMS.LE.QDIV(JDIV)) TSUB(JDIV)=FI                              SMK05590
  601 CONTINUE                                                          SMK05600
      TSUB(NUMDIV)=TBURN                                                SMK05610
  600 CONTINUE                                                          SMK05620
      GOTO 620                                                          SMK05630
  610 NPWP=0                                                            SMK05640
  620 CONTINUE                                                          SMK05650
```

```
      IF (ITYPE.GE.2) DTIME=1.                                          SMK05660
      NDIV=NUMDIV-1                                                     SMK05670
      DO 700 JPWP=1,NDIV                                                SMK05680
      IF (JPWP.GT.1.AND.NPWP.EQ.0) GOTO 700                             SMK05690
      TSTAGE=TSUB(JPWP)                                                 SMK05700
      TBRN=TSUB(JPWP+1)-TSTAGE                                          SMK05710
      PCNT=QDIV(JPWP+1)-QDIV(JPWP)                                      SMK05720
      IF (NPWP.EQ.0) PCNT=1.                                            SMK05730
      IF (TBRN.LE.0.) GOTO 700                                          SMK05740
      CALL SCONST(ICAT,UW,ITYPE,FW,EFF,TBRN,NPWP,AIRT,TGRAD,C1,C2,C3,   SMK05750
     *XN,XLIM,YLIM,ZLIM,HLIM,TLIM,XS,CNEUT,PCNT,W)                       SMK05760
      PCT=PCNT*100                                                      SMK05770
      IF (NPWP.NE.0.AND.NOPRNT.NE.1) WRITE (IOOUT,705) PCT,TSUB(JPWP),  SMK05780
     *TSUB(JPWP+1)                                                      SMK05790
      IF (NPWP.NE.0.AND.MCUOPT.NE.1) WRITE (IOOUT,705) PCT,TSUB(JPWP),
     *TSUB(JPWP+1)
705   FORMAT(1X,30H *** CLOUD PORTION CONTAINING ,F7.3,35H PERCENT OF SMSMK05800
     *OKE DURING BURN FROM ,F7.2,3H TO,F7.2,4H SEC//)                   SMK05810
      CL=0.                                                             SMK05820
      ICODE=0                                                           SMK05830
      TO=0.                                                             SMK05840
      NCY=0                                                             SMK05850
      ICALL=0                                                           SMK05860
      ISTT=ISRN-IFIX(TM+0.0001)
      II=IFIX(TM)
      XPP0=XPP1
      I1=ISTO
      I2=IETO
      I3=IDTO
      NNN=1
      IF(MODE.EQ.0) GO TO 2000
      I1=ISTT
      I2=ISTT
      I3=1
      II=0
      NNN=NPTS
      TO=0.
2000  CONTINUE
      DO 6 J=1,NNN
      IF (ITYPE.EQ.1) TO=FLOAT(I1)-DTIME                                SMK05870
      DO 6 I=I1,I2,I3                                                   SMK05880
      L=I-II
      IF(MODE.EQ.1) XPP1=XPP0+(J-1)*DELX
      IF(MODE.EQ.1) XPP2=XPP1
      X=0.0
      Y=0.0
      Z=0.0
      PATHL=0.0
      CL=0.0
      IF(L.LT.I1) GO TO 2001
C***  TAB IS NEXT TABLE REPORT TIME.  TO IS NEXT COMPUTATION TIME.      SMK05890
      TAB=FLOAT(L)-TSTAGE                                               SMK05900
      IF (TAB.GT.0.) GOTO 3                                             SMK05910
      IF (NCY.GE.MAXS) GOTO 6                                           SMK05920
      NCY=NCY+1                                                         SMK05930
      GOTO 6                                                            SMK05940
    3 TO=TO+DTIME                                                       SMK05950
      CALL WGGEOM(ICALL,CLGAUS,ITYPE,XPP1,YPP1,ZPP1,XPP2,YPP2,ZPP2,C1,  SMK05960
     1C2,C3,TO,UW,ICAT,HLIM,TLIM,CNEUT,XS,PATHL,X,Y,Z,XLIM,YLIM,ZLIM)   SMK05970
      CALL CLSMOK(ICODE,CLGAUS,ITYPE,CL,W,PATHL,TBURN,TBRN,PCNT,TSTAGE, SMK05980
     *NPWP,XLIM,YLIM,ZLIM,TO,TLIM,DTIME,X,Y,Z,BRAT1,BRAT2,BRAT3,BRAT4,  SMK05990
     *BRAT5)                                                            SMK06000
C***  REPORT OUTPUT DATA IF TIME TO .EQ. TAB, OTHERWISE, LOOP           SMK06010
C***  BACK FOR NEXT TIME STEP...                                        SMK06020
      IF (TO.LT.(TAB-.001)) GOTO 3                                      SMK06030
 2001 CALL STRANS(CL,SMTRAN,ITYPE,EXTC,1)                               SMK06040
      YFULL=2.*Y                                                        SMK06050
      TV=FLOAT(I)-TM                                                    SMK06060
      XWRIT=FLOAT(I)
      IF(MODE.GT.0) XWRIT=XPP2+XM
```

```
      PRNTOP=NOPRNT                                                          SMK06070
      IF (MODE.GT.0) PRNTOP=MCUOPT                                           SMK06080
      IF (PRNTOP.NE.1) WRITE(IOOUT,290) XWRIT,X,YFULL,Z,PATHL,CL,            SMK06090
     *(SMTRAN(K),K=1,7)                                                      SMK06100
  290 FORMAT(6X,F5.0,2X,F6.0,3X,F6.0,3X,F6.0,4X,F6.2,4X,F7.2,7(3X,F5.3))     SMK06110
      IF (ICRTR.EQ.2.AND.TV.GT.(TIMTRN+.5)) GOTO 7                           SMK06120
      IF (ICRTR.EQ.1.AND.SMTRAN(KWAVE).GE.TRANS) GOTO 7                      SMK06130
      TRTM=TV                                                                SMK06140
      TRANS=SMTRAN(KWAVE)                                                    SMK06150
    7 CONTINUE                                                               SMK06160
      IF (NCY.GE.MAXS) GOTO 6                                                SMK06170
      NCY=NCY+1                                                              SMK06180
      CLTOT(NCY)=CLTOT(NCY)+CL                                               SMK06190
    6 CONTINUE                                                               SMK06200
  700 CONTINUE                                                               SMK06210
C*** FINAL OUTPUT.                                                           SMK06220
      IF (NOMORE.EQ.0) GOTO 5                                                SMK06230
  980 IF (NCY.LE.0) GOTO 999                                                 SMK06240
      IF (NRUNS.LE.1.AND.NPWP.EQ.0) GOTO 997                                 SMK06250
      IF(MODE.EQ.0) WRITE(IOOUT,982) NRUNS                                   SMK06260
      IF(MODE.GT.0) WRITE(IOOUT,984) NRUNS                                   SMK06270
  982 FORMAT(1H1,40X,19HCOMBINED EFFECT OF ,I3,21H EXECUTIONS IN SMOKE:/     SMK06280
     *10X,10HTIME (SEC),5X,11HCL (G/M**2),21X,12HTRANSMISSION/               SMK06290
     *10X,10(1H-),5X,11(1H-),4X,                                             SMK06300
     *54H0.4-0.7  0.7-1.2    1.06   3.0-5.0 8.0-12.    10.6   94.GHZ/)      SMK06310
  984 FORMAT(1H1,40X,19HCOMBINED EFFECT OF ,I3,21H EXECUTIONS IN SMOKE:/     SMK06320
     *10X,11HLOS(METERS),4X,11HCL (G/M**2),21X,12HTRANSMISSION/              SMK06330
     *10X,10(1H-),5X,11(1H-),4X,                                             SMK06340
     *54H0.4-0.7  0.7-1.2    1.06   3.0-5.0 8.0-12.    10.6   94.GHZ/)      SMK06350
      TRANS=1.0                                                              SMK06360
      DO 985 I=1,NCY                                                         SMK06370
      CL=CLTOT(I)                                                            SMK06380
      TO=FLOAT(ISTO+(I-1)*IDTO)                                              SMK06390
      CALL STRANS(CL,SMTRAN,ITYPE,EXTC,1)                                    SMK06400
      YWRIT=TO                                                               SMK06410
      IF(MODE.GT.0) YWRIT=XPP0+XM+(I-1)*DELX                                 SMK06420
      WRITE (IOOUT,983) YWRIT,CL,(SMTRAN(J),J=1,7)                           SMK06430
      IF(IPLT.EQ.1) WRITE(NPLOTU,883) YWRIT,CL                               SMK06440
  883 FORMAT(1X,F6.1,2X,F8.3)                                                SMK06450
  983 FORMAT(12X,F6.0,8X,F8.2,6X,7(1X,F5.3,2X))                              SMK06460
      IF (ICRTR.EQ.2.AND.TO.GT.(TIMTRN+.5)) GOTO 985                         SMK06470
      IF (ICRTR.EQ.1.AND.SMTRAN(KWAVE).GE.TRANS) GOTO 985                    SMK06480
      TRTM=TO                                                                SMK06490
      TRANS=SMTRAN(KWAVE)                                                    SMK06500
  985 CONTINUE                                                               SMK06510
  997 CONTINUE                                                               SMK06520
      WRITE (IOOUT,3100)WAVE1,TRANS,TRTM                                     SMK06530
      TRANS=SMTRAN(KWAVE)                                                    SMK06540
      RETURN                                                                 SMK06550
  998 WRITE (IOOUT,3200)                                                     SMK06560
      GOTO 999                                                               SMK06570
 3100 FORMAT(1H0,5X,37H***TRANSMISSION RETURNED TO MAIN FOR                  SMK06580
     1       14HWAVELENGTH OF ,F8.3,16H MICROMETERS IS ,F5.3,                SMK06590
     2       8H AT TIME ,F7.0)                                               SMK06600
 3200 FORMAT(1H0,10X,35HINVALID WAVELENGTH PASSED FROM MAIN                  SMK06610
     1       ,//,10X,27H TRANS=1.0 RETURNED TO MAIN,//)                      SMK06620
  310 CONTINUE                                                               SMK06630
C***  ERROR CONDITION                                                        SMK06640
      WRITE(IOOUT,320)                                                       SMK06650
  320 FORMAT(1H ,105HMORE THAN 13 DATA CARDS HAVE BEEN INPUT . PLEASE CHSMK06660
     *ECK THAT THERE ARE NO MORE THAN 13 DATA CARDS PER RUN.)                 SMK06670
  999 CONTINUE                                                               SMK06680
      IERR=1                                                                 SMK06690
      TRANS=1.0                                                              SMK06700
      RETURN                                                                 SMK06710
      END                                                                    SMK06720
```

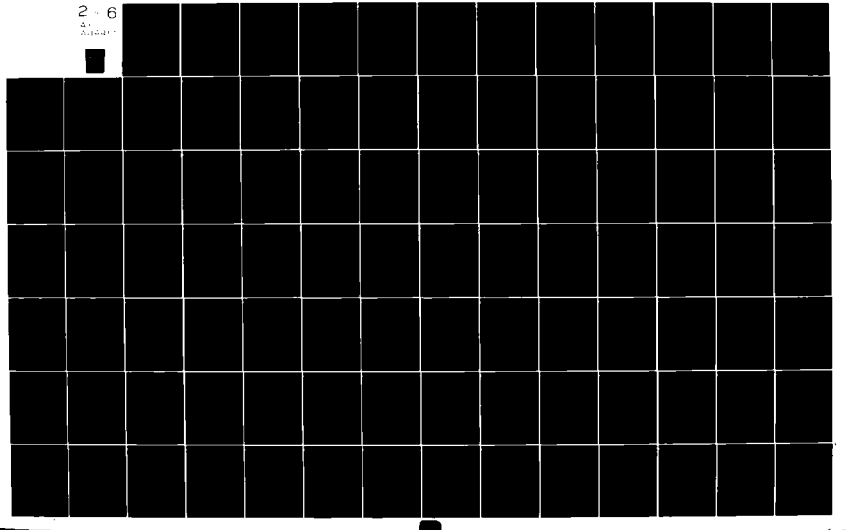
```
      SUBROUTINE CLSMOK(ICODE,CLGAUS,ITYPE,CL,W,PATHL,TBURN,TBRN,PCNT,   CLS00010
     *TSTAGE,NPWP,XLIM,YLIM,ZLIM,TO,TLIM,DTIME,X,Y,Z,BRAT1,BRAT2,        CLS00020
     *BRAT3,BRAT4,BRAT5)                                                 CLS00030
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                      CLS00040
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCLS00050
C                                                                        CLS00060
C  NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION       CLS00070
C        AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)           CLS00080
C                                                                        CLS00090
      COMMON /MOS/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9),       CLS00100
     *EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)         CLS00110
C                                                                        CLS00120
C   NOTE: TO CHANGE MAXIMUM BURN OR OBSCURATION DURATION, CHANGE        CLS00130
C      PARAMETER MAXS AND DIMENSIONS OF SMAS,PVOL,CLTOT TO MAXS         CLS00140
C                                                                        CLS00150
C      THIS SUBROUTINE CALCULATES THE VOLUME OF THE SMOKE CLOUD AND      CLS00160
C      THE CONCENTRATION * LENGTH(CL) AT THE INTERSECTION OF THE OBSERVERCLS00170
C      TARGET LOS OF THE CLOUD FORMED BY THE TOTAL NUMBER OF SMOKE       CLS00180
C      MUNITIONS WHICH WERE DETONATED AT THE SAME TIME AND FROM THE SAME CLS00190
C      POINT.  FOR CONTINUOUS TYPE BURNS (HC,FOG OIL, PWP, RP, AND      CLS00200
C      WP WICKS OR WEDGES), THE BURN MASS INCREMENT AND PATH LENGTH      CLS00210
C      TO VOLUME RATIO ARE STORED FOR EACH PUFF.  THESE PUFFS ARE        CLS00220
C      SUBSEQUENTLY ADDED TO FIND THE TOTAL EFFECT OF OBSCURANT.         CLS00230
C                                                                        CLS00240
C        INPUTS                                                          CLS00250
C                                                                        CLS00260
C               ICODE  A FLAG TO BE SET BY USER TO 0 ON FIRST CALL       CLS00270
C                      TO PROGRAM, WHICH WILL THEN RESET IT TO 1.        CLS00280
C               ITYPE  SMOKE TYPE  1=WP,  2=PWP OR WP WICK/WEDGE         CLS00290
C                                  3=HC, 4=FOG OIL, 5=RP                 CLS00300
C               W      SMOKE MASS PRODUCED BY XN  MUNITIONS (G FOR       CLS00310
C                      TYPES 1-3 AND 5, G/S FOR TYPE 4)                  CLS00320
C               PATHL   PATHLENGTH OF SMOKE CLOUD AS IT INTERSECTS THE   CLS00330
C                       OBSERVER-TARGET LINE OF SIGHT  (M)               CLS00340
C               TBURN   TOTAL LENGTH OF TIME OF BURN (S)                 CLS00350
C               NPWP    FLAG FOR PWP/RP/WP WICKS OR WEDGES. IF NON-ZERO, CLS00360
C                       THEN BURN IS IN 25 PERCENT (OF W) STAGES.        CLS00370
C               TBRN    PARTIAL BURN DURATION THIS STAGE (SEC)           CLS00380
C               TSTAGE  START OF PARTIAL BURN (SEC)                      CLS00390
C               XLIM,YLIM,ZLIM DIMENSIONS (LENGTH,HALF-WIDTH,HEIGHT)     CLS00400
C                       OF CLOUD AT END OF EXOTHERMIC RISE TIME (M)       CLS00410
C               TO      TIME AFTER IGNITION (S)                          CLS00420
C               TLIM    TIME OF TERMINATION OF HEAT RISE (S)             CLS00430
C               DTIME   TIME INCREMENT OF COMPUTATION (S)                CLS00440
C               ENDTIM ENDING TIME OF COMPUTATION (S)                    CLS00450
C               X,Y,Z   CURRENT LEADING EDGE CO-ORDINATES (M)            CLS00460
C               BRAT1,2,3,4 POLYNOMIAL BURN RATE COEFF.                  CLS00470
C               CLGAUS  UNIT CONTRIBUTION FROM GAUSSIAN PUFF.            CLS00480
C        OUTPUTS                                                         CLS00490
C               CL      COMPUTED CL IN (G/M**2) FOR THIS MUNITION SET.   CLS00500
C                                                                        CLS00510
C*** SIMPLE FOR INSTANTANEOUS BURN OF WP                                 CLS00520
      BFUN(T)=T*(BRAT1+T*(BRAT2/2.+T*(BRAT3/3.+T*BRAT4/4.)))+BRAT5*      CLS00530
     *0.07818288*ALOG(1.+358800.*T)                                      CLS00540
      MAXS=960                                                           CLS00550
      IF (ITYPE.GT.1) GOTO 100                                           CLS00560
      VOL=0.25*(4.*PI/3.) *X*Y*Z                                         CLS00570
      CL=W*(.25*PATHL/VOL  +.75  *CLGAUS)                                CLS00580
      RETURN                                                             CLS00590
C*** INITIALIZE                                                          CLS00600
  100 CL=0.                                                              CLS00610
      IF (ICODE.NE.0) GOTO 110                                           CLS00620
      ICODE=1                                                            CLS00630
      IB=0                                                               CLS00640
      IPL=0                                                              CLS00650
      START=0.                                                           CLS00660
      BRNOUT=FLOAT(IFIX(.9999+TBRN))                                     CLS00670
      NRAT=2                                                             CLS00680
      IF (BRAT2.EQ.0..AND.BRAT3.EQ.0..AND.BRAT4.EQ.0.) NRAT=1            CLS00690
      IF (BRAT5.NE.0.) NRAT=5                                            CLS00700
```

```
      KSUB=0                                                            CLS00710
      TOTPV=0.                                                          CLS00720
      BR=W                                                              CLS00730
      IF (ITYPE.NE.4) BR=W/BRNOUT                                       CLS00740
      IF (NPWP.NE.0) BR=BR*PCNT                                         CLS00750
      IF (BRNOUT.LE.1.) NRAT=1                                          CLS00760
      IF (NRAT.GT.1) BTHEN=BFUN(TSTAGE/TBURN)                           CLS00770
C***  STORE PUFF MASS EMITTED AT TIME TO                                CLS00780
110   IF (TO.GT.BRNOUT) GOTO 120                                        CLS00790
      IF (IB.GE.MAXS) GOTO 120                                          CLS00800
      IB=IB+1                                                           CLS00810
      IF (NRAT.EQ.1) GOTO 120                                           CLS00820
      IF (IB.EQ.MAXS) WRITE (IOOUT,900) IB                              CLS00830
900   FORMAT(1X,61H*** WARNING - IN SMOKE, MAXIMUM STORAGE FOR BURN DURACLS00840
     *TION OF ,I5,55H SEC IS FULL. ACCURACY BEYOND THIS POINT DECREASES.CLS00850
     * ***  )                                                          CLS00860
      TM=TO                                                             CLS00870
      IF (TO.GT.TBRN) TM=TBRN                                           CLS00880
      T=(TM+TSTAGE)/TBURN                                               CLS00890
      BNOW=BFUN(T)                                                      CLS00900
      SMAS(IB)=W*(BNOW-BTHEN)                                           CLS00910
      BTHEN=BNOW                                                        CLS00920
      IF (ITYPE.EQ.4) SMAS(IB)=SMAS(IB)*TBURN                           CLS00930
C***  COMPUTE VOLUME AT TIME TO OF FIRST PUFF                           CLS00940
120   IF (ITYPE.EQ.3) GOTO 130                                          CLS00950
      IF (ITYPE.EQ.4) GOTO 140                                          CLS00960
C***  PWP, RP OR WP WICKS/WEDGES CLOUD                                  CLS00970
      VOL=0.25*(4.*PI/3.)*X*Y*Z                                         CLS00980
      GOTO 200                                                          CLS00990
130   IF (TO.LE.TLIM) GOTO 140                                          CLS01000
C***  POST RISE REGION HC CONE.                                         CLS01010
      VOL=0.5*(PI/3.)*XLIM*YLIM*ZLIM                                    CLS01020
C***  POST-RISE FRUSTRUM OF APPROXIMATED ELLIPTIC CONE.                 CLS01030
      XZPROJ=Z*(X-XLIM)/(Z-ZLIM)                                        CLS01040
      VFRUST=0.5*(PI/3.)*(XZPROJ*(Y*Z-YLIM*ZLIM)+(X-XLIM)*YLIM*ZLIM)    CLS01050
      VOL=VOL+VFRUST                                                    CLS01060
      GOTO 200                                                          CLS01070
C***  HC BEFORE THE END OF EXOTHERMIC RISE AND FOG OIL CASE             CLS01080
140   VOL=0.5*(PI/3.)*X*Y*Z                                            CLS01090
C***  STORE PATH LENGTH/VOLUME RATIO AT TIME TO FOR FIRST EXPANDING     CLS01100
C     UNIFORM AND GAUSSIAN PUFF CONTRIBUTION.                           CLS01110
200   IF (ITYPE.NE.4) PV=.25*(PATHL/VOL)+.75*CLGAUS                     CLS01120
      IF (ITYPE.EQ.4) PV=PATHL/VOL                                      CLS01130
      IF (PV.LE.0..AND.START.EQ.0.) RETURN                              CLS01140
      IF (PV.GT.0..AND.START.EQ.0.) START=TO                            CLS01150
      IF (IPL.GE.MAXS) GOTO 300                                         CLS01160
      IPL=IPL+1                                                         CLS01170
      IF (IPL.EQ.MAXS) WRITE (IOOUT,910) TO,IPL                         CLS01180
910   FORMAT(1X,58H*** WARNING - IN SMOKE, MAX. STORAGE FOR CLOUD VOLUMECLS01190
     *S OF ,I5,34H SEC. IS FULL. COMPUTATION TIME = ,F6.0,5H SEC./      CLS01200
     *15X,40HACCURACY BEYOND THIS POINT DECREASES ***)                  CLS01210
      PVOL(IPL)=PV                                                      CLS01220
C***  SUM CL FOR PUFFS                                                  CLS01230
300   IF (NRAT.EQ.1) GOTO 400                                           CLS01240
      LMIN=MAX0(1,(IPL-IB+1))                                           CLS01250
      IF (IPL.LT.LMIN) RETURN                                           CLS01260
      N=0                                                               CLS01270
      DO 320 J=LMIN,IPL                                                 CLS01280
      K=IPL-J+1                                                         CLS01290
      CDEL=SMAS(K)*PVOL(J)                                              CLS01300
      CL=CL+CDEL                                                        CLS01310
      N=N+1                                                             CLS01320
      IF (N.LT.120) GOTO 320                                            CLS01330
      IF (ABS(CDEL).LT.1.E-5) RETURN                                    CLS01340
320   CONTINUE                                                          CLS01350
      RETURN                                                            CLS01360
C***  FAST COMPUTATION FOR CONSTANT BURN RATES                          CLS01370
400   TOTPV=TOTPV+PVOL(IPL)                                             CLS01380
C  INDEX OF LAST PUFF EMITTED                                           CLS01390
      IF (IPL.LT.MAXS) KSUB=IPL-IB                                      CLS01400
```

```
C KEEP REMOVING LAST PUFF IF POSSIBLE                                   CLS01410
      IF (IPL.EQ.MAXS) KSUB=KSUB+1                                      CLS01420
      IF (KSUB.GT.IPL) KSUB=IPL                                         CLS01430
C ONLY REMOVE PUFFS AFTER BURN HAS STOPPED (IE LAST PUFF OUT)           CLS01440
      IF (KSUB.GT.0) TOTPV=TOTPV-PVOL(KSUB)                             CLS01450
      CL=TOTPV*BR                                                       CLS01460
      RETURN                                                            CLS01470
      END                                                               CLS01480
```

```
      SUBROUTINE SCONST(ICAT,UW,ITYPE,FW,EFF,TBRN,NPWP,AIRT,TGRAD,C1,C2,SCO00010
     *     C3, XN,XLIM,YLIM,ZLIM,HLIM,TLIM,XS,CNEUT,PCNT,W)                SCO00020
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                       SCO00030
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSCO00040
C                                                                         SCO00050
C     SMOKE MODEL  PARAMETERS FOR CLOUD DIMENSIONS                        SCO00060
C     THIS SUBROUTINE CALCULATES THE PARAMETERS FOR THE CLOUD WIDTH AND   SCO00070
C     CLOUD HEIGHT(MOMENTUM) AS A FUNCTION OF PASQUILL CATEGORY AND THE   SCO00080
C     HEAT RISE PARAMETER FOR THE HEIGHT(EXOTHERMIC). HEAT RISE IS        SCO00090
C     TERMINATED AS A FUNCTION OF ATMOSHERIC STABILITY                    SCO00100
C     INPUTS    ICAT    PASQUILL CATEGORY                                 SCO00110
C                         1-A, 2-B, 3-C, 4-D, 5-E, 6-F                    SCO00120
C               UW      WIND VELOCITY  (M/S)                              SCO00130
C               ITYPE   TYPE OF SMOKE.  1=WP, 2=PWP OR WP WICK/WEDGE,     SCO00140
C                                       3=HC, 4=FOG OIL, 5=RP             SCO00150
C               FW      FILL WEIGHT  (LBS)                                SCO00160
C               EFF     EFFICIENCY (PERCENT)                              SCO00170
C               XN      NUMBER OF MUNITIONS                               SCO00180
C               TBRN    BURN TIME FOR THIS STAGE (OR ENTIRE MUNITION)SEC. SCO00190
C               NPWP    FLAG FOR LONG-BURN PHOSPHORUS. IF NON-ZERO, THEN  SCO00200
C                       BURNS ARE IN 25 PERCENT STAGES TO ALLOW SEPARATE  SCO00210
C                       BUOYANCIES FOR EACH.                              SCO00220
C               AIRT    SURFACE AIR TEMPERATURE    (DEG C)                SCO00230
C               TGRAD   VERT TEMP GRADIENT (C DEG./M) USED ONLY FOR       SCO00240
C                           CATAGORIES E,F (IE. 5,6) IN WHICH IT MUST     SCO00250
C                           BE POSITIVE.                                  SCO00260
C     OUTPUTS   C1      EQUATION PARAMETER FOR CLOUD WIDTH  Y=9.1+C1*X    SCO00270
C               C2      PARAMETER FOR HEIGHT (MOM)  Z=2.73+C2*X           SCO00280
C               C3      PARAMETER FOR HEIGHT (EXO)  Z=2.73+C2*X+C3*X**2/3 SCO00290
C               XLIM    CLOUD LENGTH ALONG WIND DIRECTION (M)             SCO00300
C                         AT RISE TERMINATION.                            SCO00310
C               YLIM    CLOUD BASE HALF-WIDTH PERP. WIND DIRECTION (M)    SCO00320
C                         AT RISE TERMINATION.                            SCO00330
C               ZLIM    TOTAL CLOUD HEIGHT AT TERMINATION OF HEAT RISE.   SCO00340
C               HLIM    TOTAL ADDED RISE AT TERMINATION OF HEAT RISE.     SCO00350
C               TLIM    TIME OF TERMINATION OF HEAT RISE                  SCO00360
C               XS      TERM FOR CALCULATION OF HEIGHT FOR NEUTRAL CONDITISCO00370
C               CNEUT   TERM FOR CALCULATION OF HEIGHT FOR NEUTRAL CONDITISCO00380
C                                                                         SCO00390
      DIMENSION C11(6),C22(6)                                             SCO00400
      DATA C11/.419,.328,.238,.2,.18,.146/                                SCO00410
      DATA C22/.137,.11,.073,.066,.055,.046/                              SCO00420
      TLIM=600.                                                           SCO00430
      XS=0.0                                                              SCO00440
      CNEUT=0.0                                                           SCO00450
      XLIM=0.                                                             SCO00460
      YLIM=0.                                                             SCO00470
      ZLIM=0.                                                             SCO00480
      HLIM=1.0                                                            SCO00490
C***  ERROR CONDITION                                                     SCO00500
      IF(ICAT.LT.1.OR.ICAT.GT.6)   WRITE(IOOUT,50)                        SCO00510
   50 FORMAT(1H ,62HERROR IN SUBROUTINE SCONST. PASQUILL CATEGORY IS NOTSCO00520
     1ACCEPTABLE)                                                         SCO00530
      IF (ICAT.LT.1.OR.ICAT.GT.6) RETURN                                  SCO00540
C***  SELECTION OF CLOUD WIDTH COEFFICIENT AS A FUNCTION OF PASQUILL      SCO00550
C***  CATEGORY.                                                           SCO00560
      C1 = C11(ICAT)                                                      SCO00570
C***  SELECTION OF CLOUD HEIGHT(MOM) COEFFICIENT AS A FUNCTION OF         SCO00580
C***  PASQUILL CATEGORY.                                                  SCO00590
      C2 = C22(ICAT)                                                      SCO00600
      IF(ICAT.GT.1)GO TO 1                                                SCO00610
      IF(UW.GT.2.0)GO TO 1                                                SCO00620
      C2=0.15                                                             SCO00630
      IF(UW.GT.1.5)GO TO 1                                                SCO00640
      C2=0.18                                                             SCO00650
      IF(UW.GT.1.0)GO TO 1                                                SCO00660
      C2=0.25                                                             SCO00670
      IF(UW.GT.0.5)GO TO 1                                                SCO00680
      C2=0.39                                                             SCO00690
    1 CONTINUE                                                            SCO00700
```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 9/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL



```
  120   TB=TBRN                                                                SCO00710
        IF (ITYPE.EQ.1.AND.TBRN.GT.1.) TB=1.                                   SCO00720
        IF (NPWP.NE.0) TB=TB/PCNT                                              SCO00730
C***    IF NON-EXOTHERMIC, RETURN.                                             SCO00740
        IF (ITYPE.EQ.4) RETURN                                                 SCO00750
        EM=800.0                                                               SCO00760
        IF(ITYPE.EQ.3)EM=500.0                                                 SCO00770
        IF(ITYPE.EQ.5)EM=660.                                                  SCO00780
        F=0.00001*3.59*453.59*EM*FW*XN*(EFF/100.)/TB                           SCO00790
C***    CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT FOR UNSTABLE ATM.        SCO00800
C***    CONDITIONS (A,B,C)                                                     SCO00810
        C3=1.6/UW*(F**.3333)                                                   SCO00820
        IF(ITYPE.EQ.3) GO TO 200                                               SCO00830
        IF(ICAT.EQ.4) GO TO 131                                                SCO00840
        IF(ICAT.GT.4) GO TO 141                                                SCO00850
        TLIM=0.0                                                               SCO00860
C***    DETERMINE TIME AND TOTAL CLOUD HEIGHT AT TERMINATION OF HEAT RISE.     SCO00870
  123   TLIM=TLIM+2.                                                           SCO00880
        X=UW*TLIM                                                              SCO00890
        Y=(9.1+C1*X)/2.                                                        SCO00900
        V=.75*(2.*3.14159)**1.5*(Y/2.15)**2*(2.73+C3*X)/4.3                    SCO00910
        C=W/(V*TB)                                                             SCO00920
        IF(C.GT.0.11) GO TO 123                                                SCO00930
        HLIM=C3*X**.667                                                        SCO00940
        RETURN                                                                 SCO00950
C***    CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENTS FOR NEUTRAL ATM.        SCO00960
C***    CONDITIONS (D).                                                        SCO00970
  131   XS=10.0*F**0.4                                                         SCO00980
        CNEUT=1.6*(F**.3333)*XS**0.667/UW                                      SCO00990
        RETURN                                                                 SCO01000
  141   CONTINUE                                                               SCO01010
C***    CALCULATION OF CLOUD HEIGHT (EXO) COEFFICIENT AND TOTAL CLOUD          SCO01020
C***    HEIGHT AT TERMINATION OF HEAT RISE FOR STABLE ATM. CONDITIONS(E,F)SCO01030
        SBAR=9.8/(AIRT+273.0)*(TGRAD+0.0098)                                   SCO01040
        HLIM=1.4*(F/(UW*SBAR))**.333                                           SCO01050
        RETURN                                                                 SCO01060
C ***    CALCULATE PARAMETERS FOR HC 10 SEC. RISE TIME.                        SCO01070
  200   TLIM=10.                                                               SCO01080
        XLIM=TLIM*UW                                                           SCO01090
        YLIM=(9.1+C1*XLIM)/2.                                                  SCO01100
        ZLIM=2.73+C2*XLIM                                                      SCO01110
        IF (ICAT.EQ.4) GOTO 231                                                SCO01120
        HLIM=C3*XLIM**.667                                                     SCO01130
        ZLIM=ZLIM+HLIM                                                         SCO01140
        IF (ICAT.LT.5) RETURN                                                  SCO01150
        SBAR=9.8/(AIRT+273.)*(TGRAD+0.0098)                                    SCO01160
        HTEST=1.4*(F/(UW*SBAR))**.333                                          SCO01170
        IF (HLIM.LE.HTEST) RETURN                                              SCO01180
C***    IF UNSTABLE ATMOSPHERE REACHES MAX. BEFORE 10 SEC. COMPUTE             SCO01190
C***    TLIM.                                                                  SCO01200
        XLIM=(HTEST/C3)**1.5                                                   SCO01210
        TLIM=XLIM/UW                                                           SCO01220
        YLIM=(9.1+C1*XLIM)/2.                                                  SCO01230
        HLIM=HTEST                                                             SCO01240
        ZLIM=2.73+C2*XLIM+HLIM                                                 SCO01250
        RETURN                                                                 SCO01260
  231   XS=10.0*F**0.4                                                         SCO01270
        CNEUT=C3*(XS**.667)                                                    SCO01280
        HLIM=CNEUT*(0.4+0.64*(XLIM/XS)+2.2*(XLIM/XS)**2)/                      SCO01290
       *(1.+0.8*XLIM/XS)**2                                                    SCO01300
        ZLIM=ZLIM+HLIM                                                         SCO01310
        RETURN                                                                 SCO01320
        END                                                                    SCO01330
```

```
      SUBROUTINE SMASSP(XN,EFF,FW,RELHUM,W,ITYPE,YF)                    SMA00010
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSMA00020
C                                                                       SMA00030
C     SMOKE MODEL                                                       SMA00040
C     THIS SUBROUTINE CALCULATES THE SMOKE MASS PRODUCED BY XN          SMA00050
C     MUNITIONS AT THE SAME POINT AND SAME TIME.                        SMA00060
C     INPUTS  ITYPE SMOKE TYPE                                          SMA00070
C                   1 WHITE PHOSPHOROUS                                 SMA00080
C                   2 PLASTICIZED WHITE PHOSPHORUS OR WP WICK WEDGE     SMA00090
C                   3 HC                                                SMA00100
C                   4 FOG OIL                                           SMA00110
C                   5 RED PHOSPHORUS                                    SMA00120
C             XN    NUMBER OF MUNITIONS AT THIS POINT AND TIME.         SMA00130
C             RELHUM RELATIVE HUMIDITY (PERCENT)                        SMA00140
C             EFF   MUNITION EFFICIENCY (PERCENT)                       SMA00150
C             FW    FILL WEIGHT  (LBS FOR TYPES 1-3, GAL/HR FOR TYPE 4) SMA00160
C             W     SMOKE MASS PRODUCED FOR XN MUNITIONS.               SMA00170
C                     G FOR TYPES 1-3 AND 5, G/S FOR TYPE 4             SMA00180
C             YF    OPTIONAL USER SUPPLIED YIELD FACTOR                 SMA00190
C                                                                       SMA00200
      W=0.0                                                             SMA00210
      IF(ITYPE.LT.1.OR.ITYPE.GT.5) GO TO 90                             SMA00220
      GO TO (101,101,102,103,101),ITYPE                                 SMA00230
   90 WRITE(IOOUT,50)ITYPE                                              SMA00240
   50 FORMAT(//,28H ERROR IN SMASSP:SMOKE TYPE ,I4,10H UNDEFINED)       SMA00250
      RETURN                                                            SMA00260
  101 Y=.028*RELHUM+3.4                                                 SMA00270
      CONVER=453.592                                                    SMA00280
      GO TO 110                                                         SMA00290
  102 Y=1.17 +.014*RELHUM                                               SMA00300
      CONVER=453.592                                                    SMA00310
      GO TO 110                                                         SMA00320
C***  FOR FOG OIL, W IS A RATE, WHERE FW IS IN GAL/HR AND .93           SMA00330
C***  CONVERTS TO G/S.                                                  SMA00340
  103 Y=1.0                                                             SMA00350
      CONVER=0.93                                                       SMA00360
  110 IF (YF.NE.0.) Y=YF                                                SMA00370
      YF=Y                                                              SMA00380
      W=XN*Y*FW*CONVER*(EFF/100.0)                                      SMA00390
      RETURN                                                            SMA00400
      END                                                               SMA00410
```

```
      SUBROUTINE STRANS(CL,SMTRAN,ITYPE,EXTC,ICALL)                     STR00010
C                                                                       STR00020
C                                                                       STR00030
C     SMOKE MODEL     CLOUD TRANSMISSION                                STR00040
C     THIS SUBROUTINE CALCULATES TRANSMISSION IN 7 SPECTRAL REGIONS     STR00050
C    (0.4-0.7,0.7-1.2,1.06,3.0-5.0,8.0-12.0,10.6 MICROMETERS AND 94.GHZ) STR00060
C     FOR A GIVEN SMOKE TYPE AND CONCENTRATION LENGTH                   STR00070
C     INPUTS    ITYPE   SMOKE TYPE                                      STR00080
C                         1   WHITE PHOSPHOROUS                        STR00090
C                         2   PLASTICIZED WHITE PHOSPHORUS, WP WICK WEDGESTR00100
C                         3   HC                                        STR00110
C                         4   FOG OIL                                   STR00120
C                         5   RED PHOSPHORUS                           STR00130
C               CL      COMPUTED CL IN (G/M**2)                       STR00140
C               ICALL     = 0 SETS UP EXTC ARRAY USED FOR COMPUTATIONS ANDSTR00150
C                           ALLOWS USER TO OVER-RIDE DEFAULT ALPHAS.    STR00160
C                         = 1 EXECUTES TRANSMISSION CALCULATION.        STR00170
C                                                                       STR00180
C     OUTPUTS   SMTRAN TRANSMISSION THROUGH SMOKE (DECIMAL)             STR00190
C               EXTC   ARRAY OF EXTINCTION COEFF. ACTUALLY USED         STR00200
C                      IN TRANSMISSION CALCULATION. EXTC(8) IS          STR00210
C                      USED AS A FLAG FOR ICALL=0 REPLACEMENT.          STR00220
C                                                                       STR00230
C               IF EXTC(8)=ITYPE, NO CHANGES ARE MADE IN EXTC ARRAY IE., STR00240
C                         ALPHA VALUES DO NOT REPLACE EXTC VALUES.      STR00250
C               IF EXTC(8)=0.    , ONLY THOSE VALUES IN EXTC WHICH ARE  STR00260
C                         ZERO ARE REPLACED BY THE STORED VALUES IN     STR00270
C                         ALPHA (ITYPE COLUMN).                         STR00280
C               IF EXTC(8) IS NOT 0. OR ITYPE, THEN ALL EXTC VALUES     STR00290
C                         ARE REPLACED BY CORRESPONDING ALPHA VALUES ANDSTR00300
C                         EXTC(8) IS SET TO ITYPE.                      STR00310
C                                                                       STR00320
      DIMENSION ALPHA(7,5),SMTRAN(7),EXTC(8)                            STR00330
      DATA ALPHA          /4.304,2.166,1.541,0.350,0.338,0.364,0.001,   STR00340
     *                     4.304,2.166,1.541,0.350,0.338,0.364,0.001,   STR00350
     *                     4.579,2.186,2.040,0.190,0.052,0.051,0.001,   STR00360
     *                     6.851,4.592,3.497,0.245,0.020,0.018,0.001,   STR00370
     *                     4.304,2.166,1.541,0.350,0.338,0.364,0.001/   STR00380
C***  TRANSMISSION CALCULATED BY BEER'S LAW APPROXIMATION               STR00390
C***    IF ICALL=0, EXTC ARRAY IS FORMED OR MODIFIED...                 STR00400
      IF (ICALL.NE.0) GOTO 20                                           STR00410
      IF (EXTC(8).EQ.FLOAT(ITYPE)) GOTO 18                              STR00420
      IF (EXTC(8).EQ.0.) GOTO 15                                        STR00430
      DO 13 J=1,7                                                       STR00440
   13 EXTC(J)=0.                                                        STR00450
   15 EXTC(8)=FLOAT(ITYPE)                                              STR00460
      DO 17 J=1,7                                                       STR00470
      IF (EXTC(J).EQ.0.) EXTC(J)=ALPHA(J,ITYPE)                         STR00480
   17 CONTINUE                                                          STR00490
   18 RETURN                                                            STR00500
C***  FOR ICALL NON-ZERO, COMPUTE TRANSMISSION USING EXTC VALUES.       STR00510
   20 DO 30 I=1,7                                                       STR00520
      SMTRAN(I)=EXP(-EXTC(I)*CL)                                        STR00530
   30 CONTINUE                                                          STR00540
      RETURN                                                            STR00550
      END                                                               STR00560
```

```
      SUBROUTINE WGGEOM(ICALL,CLGAUS,ITYPE,XPP1,YPP1,ZPP1,XPP2,YPP2,     WGG00010
     1ZPP2,C1,C2,C3,TO,UW,ICAT,HLIM,TLIM,CNEUT,XS,PATHL,X,Y,Z,XLIM,     WGG00020
     2YLIM,ZLIM)                                                        WGG00030
C                                                                       WGG00040
C  NOTE: THE FOLLOWING COMMON BLOCK ALLOWS MUNITION BURN DURATION       WGG00050
C        AND OBSCURATION PERIODS UP TO 16.0 MINUTES (960 SEC)          WGG00060
C                                                                       WGG00070
      COMMON /MO5/ SMAS(960),PVOL(960),CLTOT(960),SMTRAN(7),R1(9),      WGG00080
     *EXTC(8),ZL(2),XL(2),YL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)        WGG00090
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     WGG00100
      DATA MAXS /960/                                                   WGG00110
C                                                                       WGG00120
C   NOTE: TO CHANGE LENGTH OF BURN OR OBSCURATION, RESET                WGG00130
C       MAXS AND REDIMENSION SMAS, PVOL AND CLTOT ARRAYS                WGG00140
C                                                                       WGG00150
C    SMOKE MODEL   GEOMETRY                                             WGG00160
C      INPUT:                                                           WGG00170
C      ITYPE = SMOKE TYPE CODE                                          WGG00180
C      XPP1,YPP1,ZPP1 = OBSERVER IN MUNITION CENTERED COORDINATES       WGG00190
C      YPP1,YPP2,ZPP2 = TARGET IN MUNITION CENTERED COORD. (X AXIS ALONG WGG00200
C                       WIND VECTOR.)                                   WGG00210
C      C1,C2,C3 = CLOUD GROWTH PARAMETERS (SEE PROGRAM SCONST)          WGG00220
C      TO = TIME SINCE DETONATION (SECONDS)                             WGG00230
C      UW = WIND SPEED (METERS/SECOND)                                  WGG00240
C      ICAT=PASQUILL CATEGORY                                           WGG00250
C      HLIM,TLIM,CNEUT,XS= EXOTHERMIC RISE PARAMETERS                   WGG00260
C      ICALL = FLAG SET BY USER TO 0 FOR 1ST CALL. RESET BY PROGRAM     WGG00270
C              TO 1 THEREAFTER.                                         WGG00280
C     OUTPUT:                                                           WGG00290
C     X = CLOUD LENGTH (METERS)                                         WGG00300
C     Y = CLOUD HALF-WIDTH (METERS)                                     WGG00310
C     Z = CLOUD HEIGHT (METERS)                                         WGG00320
C     PATHL = PATHLENGTH OF LOS THROUGH SMOKE CLOUD                     WGG00330
C                                                                       WGG00340
C  SUBROUTINES CALLED... DIRECTLY: XYZINT, GPUFF, INDIRECTLY: QROOT     WGG00350
C                                                                       WGG00360
C *** TRANSLATION TO PLACE MUNITION AT (0,0,0)                          WGG00370
      IF (ICALL.NE.0) GOTO 10                                           WGG00380
      ICALL=1                                                           WGG00390
      KCALL=0                                                           WGG00400
C***  CALCULATE LEADING EDGE LOCATION AT TO.                            WGG00410
10    A=UW*TO                                                           WGG00420
      B=(9.1/2.0)+A*C1/2.                                               WGG00430
      IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) GOTO 33                             WGG00440
C***  WP/PWP/RP COMPUTATION.                                            WGG00450
      GO TO (11,11,11,21,31,31),ICAT                                    WGG00460
 11   CONTINUE                                                          WGG00470
      C=2.73+C2*A+C3*A**.667                                            WGG00480
      CLIM=2.73+C2*A+HLIM                                               WGG00490
      IF(TO.GT.TLIM.AND.C.GT.CLIM)C=CLIM                                WGG00500
      GO TO 41                                                          WGG00510
 21   CONTINUE                                                          WGG00520
      C=2.73+C2*A                                                       WGG00530
      IF (A.LE.XS) C=C+C3*A**.667                                       WGG00540
      IF (A.GT.XS) C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/           WGG00550
     1(1.0+0.8*(A/XS))**2                                               WGG00560
      GO TO 41                                                          WGG00570
 31   CONTINUE                                                          WGG00580
      C=2.73+C2*A+C3*A**0.667                                           WGG00590
      CLIM=2.73+C2*A+HLIM                                               WGG00600
      IF(C.GT.CLIM)C=CLIM                                               WGG00610
      GOTO 41                                                           WGG00620
C*** HC, FOG OIL COMPUTATION.                                           WGG00630
   33 C=2.73+C2*A                                                       WGG00640
      IF (ITYPE.EQ.4) GOTO 41                                           WGG00650
      IF (TO.LT.TLIM) GOTO 35                                           WGG00660
      C=C+HLIM                                                          WGG00670
      GOTO 41                                                           WGG00680
   35 IF (ICAT.NE.4.OR.A.LE.XS)C=C+C3*A**0.667                          WGG00690
      IF (ICAT.EQ.4 .AND. A.GT.XS)                                      WGG00700
```

```
     1 C=C+CNEUT*(0.4+0.64*(A/XS)+2.2*(A/XS)**2)/                       WGG00710
     1(1.0+0.8*(A/XS))**2                                              WGG00720
   41 CONTINUE                                                         WGG00730
C*** NOW COMPUTE PATH LENGTHS -- SELECT ELLIPSOID FOR PHOSPHORUS       WGG00740
C*** OR ELLIPTICAL CONES FOR HC, FOG OIL.                              WGG00750
C***  FIRST GAUSSIAN PUFF AND IMAGE PUFF CONTRIBUTION, THEN FULLY MIXED WGG00760
      NCODE=2                                                          WGG00770
      IF (ITYPE.EQ.3.OR.ITYPE.EQ.4) NCODE=1                            WGG00780
      IF (ITYPE.NE.4)                                                  WGG00790
     *CALL GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1,ZPP2)  WGG00800
      X=A                                                              WGG00810
      Y=B                                                              WGG00820
      Z=C                                                              WGG00830
      IF (ITYPE.EQ.3) GOTO 200                                         WGG00840
100   X0=A                                                             WGG00850
      Y0=B                                                             WGG00860
      Z0=C                                                             WGG00870
150   XL(1)=XPP1                                                       WGG00880
      XL(2)=XPP2                                                       WGG00890
      YL(1)=YPP1                                                       WGG00900
      YL(2)=YPP2                                                       WGG00910
      ZL(1)=ZPP1                                                       WGG00920
      ZL(2)=ZPP2                                                       WGG00930
      CALL XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)        WGG00940
      PATHL=SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+            WGG00950
     *(ZINT(1)-ZINT(2))**2)                                            WGG00960
      IF (ITYPE.EQ.3) GOTO 220                                         WGG00970
      RETURN                                                           WGG00980
200   ICONE=1                                                          WGG00990
      IF (TO.LE.TLIM) GOTO 100                                         WGG01000
      ICONE=0                                                          WGG01010
C*** FOR HC, BREAK PATH UP INTO TWO PARTS. FIRST RISE PORTION CONE,    WGG01020
C     THEN POST-RISE FRUSTRUM.                                         WGG01030
      X0=XLIM                                                          WGG01040
      Y0=YLIM                                                          WGG01050
      Z0=ZLIM                                                          WGG01060
      GOTO 150                                                         WGG01070
220   IF (ICONE.EQ.1) RETURN                                           WGG01080
      ICONE=1                                                          WGG01090
C*** NOW POST RISE PORTION,                                            WGG01100
      IF (PATHL.EQ.0.) GOTO 100                                        WGG01110
      IF (IFLAG(1).NE.3.AND.IFLAG(2).NE.3) RETURN                      WGG01120
      IF (IFLAG(1).EQ.3.AND.IFLAG(2).EQ.3) RETURN                      WGG01130
      K=1                                                              WGG01140
      IF (IFLAG(2).EQ.3) K=2                                           WGG01150
      J=1                                                              WGG01160
      IF (XL(2).LT.XL(1)) J=2                                          WGG01170
      XL(J)=XINT(K)                                                    WGG01180
      YL(J)=YINT(K)                                                    WGG01190
      ZL(J)=ZINT(K)                                                    WGG01200
      X0=A                                                             WGG01210
      Y0=B                                                             WGG01220
      Z0=C                                                             WGG01230
      CALL YYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)        WGG01240
      PATHL=PATHL+SQRT((XINT(1)-XINT(2))**2+(YINT(1)-YINT(2))**2+      WGG01250
     *(ZINT(1)-ZINT(2))**2)                                            WGG01260
      RETURN                                                           WGG01270
      END                                                              WGG01280
```

```
      FUNCTION QROOT(ISIGN,A,B,C)                                       QR000010
C*****PURPOSE:                                                          QR000020
C         TO FIND ROOTS OF A QUADRATIC EQUATION.                        QR000030
      IF (A.EQ.0.) GOTO 2                                               QR000040
      XX=1.0*ISIGN                                                      QR000050
      TEST= B*B - 4.0*A*C                                               QR000060
      IF(TEST.LT.0.0)GO TO 1                                            QR000070
      QROOT=(-1.0*B + XX*SQRT(TEST))/(2.0*A)                            QR000080
      GO TO 100                                                         QR000090
    1 QROOT=0.0                                                         QR000100
      GOTO 100                                                          QR000110
    2 IF (B.EQ.0.) GOTO 1                                               QR000120
      QROOT=-C/B                                                        QR000130
  100 RETURN                                                            QR000140
      END                                                               QR000150
```

```
      SUBROUTINE XYZINT(NCODE,XL,YL,ZL,X0,Y0,Z0,XINT,YINT,ZINT,IFLAG)   XYZ00010
      DIMENSION XL(2),YL(2),ZL(2),XINT(2),YINT(2),ZINT(2),IFLAG(2)      XYZ00020
      DIMENSION DIST(2),TEST(2)                                         XYZ00030
C*****SUBROUTINE XYZINT*****                                            XYZ00040
C*****PURPOSE:                                                          XYZ00050
C        TO FIND THE X,Y,Z INTERCEPTS OF A TARGET-OBSERVER LINE OF SIGHTXYZ00060
C        WITH A SMOKE CLOUD DESCRIBED BY:                               XYZ00070
C          NCODE = 1 A HALF ELLIPTIC CONE WITH APEX AT THE ORIGIN       XYZ00080
C                    AND LEADING EDGE TRUNCATION BY THE X=X0 PLANE      XYZ00090
C                    AND BOTTOM EDGE TRUNCATION BY THE Z=0 PLANE.       XYZ00100
C          NCODE = 2 A QUARTER ELLIPSOID  WITH APEX                     XYZ00110
C                    AT THE ORIGIN AND WITH LEADING EDGE TRUNCATED BY   XYZ00120
C                    THE X=X0 AND BOTTOM EDGE TRUNCATED BY THE Z=0 PLANE.XYZ00130
C*****INPUT:                                                            XYZ00140
C        XL(2),YL(2),ZL(2) = X,Y,Z COORDINATES OF TWO POINTS THROUGH    XYZ00150
C        WHICH THE LOS PASSES(IE. TARGET AND OBSERVER COORDINATES).     XYZ00160
C        X0,Y0,Z0 = LENGTH OF SEMI-AXES OF ELLIPSOID.                   XYZ00170
C*****OUTPUT:                                                           XYZ00180
C        XINT(2),YINT(2),ZINT(2) = X,Y,Z COORDINATES OF THE INTERCEPTS  XYZ00190
C        OF THE LOS WITH THE ELLIPSOID.                                 XYZ00200
C        IFLAG(2) - INTERCEPT TYPE FOR EACH INTERCEPT COORD:            XYZ00210
C                 = 0  NO INTERCEPT                                     XYZ00220
C                 = 1  INTERIOR TO VOLUME                               XYZ00230
C                 = 2  ON CONICAL OR ELLIPTICAL SURFACE                 XYZ00240
C                 = 3  ON LEADING EDGE OF SURFACE                       XYZ00250
C*****MATHEMATICAL APPROACH:                                            XYZ00260
C        THE EQUATION OF THE ELLIPSOID CAN BE WRITTEN AS:               XYZ00270
C        ((X-X0)/X0)**2 + (Y/Y0)**2 +(Z/Z0)**2 = 1                      XYZ00280
C        AND THE EQUATION OF THE LOS CAN BE WRITTEN AS:                 XYZ00290
C        (X-XL1)/(XL2-XL1) = (Y-YL1)/(YL2-YL1) = (Z-ZL1)/(ZL2-ZL1)      XYZ00300
C        THE TWO EQUATIONS ARE COMBINED TO FORM A QUADRATIC EQUATION    XYZ00310
C        WHICH IS SOLVED TO GIVE THE INTERCEPTS.                        XYZ00320
C        SIMILARLY FOR THE LOS EQUATIONS AND ELLIPTIC CONE :            XYZ00330
C         (Z/Z0)**2 + (Y/Y0)**2 - (X/X0)**2 = 0                         XYZ00340
C*****SPECIAL NOTES                                                     XYZ00350
C    (1) WHEN TWO OR MORE COORDINATES ARE THE SAME, SPECIAL CASES ARE   XYZ00360
C        FORMED WHICH MUST BE DEALT WITH SEPARATELY BECAUSE OF          XYZ00370
C        SINGULARITIES IN THE LOS EQUATION.                             XYZ00380
C    (2) WHEN TARGET AND/OR OBSERVER ARE INSIDE THE CLOUD INTERCEPTS    XYZ00390
C        ARE TAKEN AS THE TARGET AND/OR OBSERVER COORDINATES.           XYZ00400
C    (3) PROPER ACCOUNT IS TAKEN FOR A LOS INTERCEPTING THE CLOUD       XYZ00410
C        LEADING EDGE BUT.....                                          XYZ00420
C    (4) ALL COORDINATES MUST BE ABOVE THE Z=0 PLANE (IE. ABOVE THE     XYZ00430
C        SURFACE.)                                                      XYZ00440
C                                                                       XYZ00450
C  SUBROUTINE CALLED... QROOT                                           XYZ00460
C*****INITIALIZE PROGRAM VARIABLES                                      XYZ00470
      I1=0                                                              XYZ00480
      I2=0                                                              XYZ00490
      INT=0                                                             XYZ00500
      LEAD=0                                                            XYZ00510
      ISURF=0                                                           XYZ00520
      K1=0                                                              XYZ00530
      K2=0                                                              XYZ00540
      K3=0                                                              XYZ00550
      K4=0                                                              XYZ00560
      TEST(1)=0.0                                                       XYZ00570
      TEST(2)=0.0                                                       XYZ00580
      TEST3=0.0                                                         XYZ00590
      TEST4=0.0                                                         XYZ00600
      DIST(1)=0.0                                                       XYZ00610
      DIST(2)=0.0                                                       XYZ00620
      DELX=XL(2)-XL(1)                                                  XYZ00630
      DELY=YL(2)-YL(1)                                                  XYZ00640
      DELZ=ZL(2)-ZL(1)                                                  XYZ00650
      IFLAG(1)=0                                                        XYZ00660
      IFLAG(2)=0                                                        XYZ00670
C*** REJECT IMMEDIATELY IF BOTH TGT/OBS BELOW Z=0.                      XYZ00680
      IF (ZL(1).LT.0..AND.ZL(2).LT.0.) GOTO 800                         XYZ00690
C*****DETERMINE SPECIAL CASES FOR LOS                                   XYZ00700
```

```
      ICASE=1                                                           XYZ00710
      IF (ABS(DELX).GT.(.01*ABS(DELY)).AND.                             XYZ00720
     *    ABS(DELX).GT.ABS(.01*ABS(DELZ))) GOTO 12                     XYZ00730
      ICASE=2                                                           XYZ00740
      IF (ABS(DELY).GT.(.01*ABS(DELZ))) GOTO 12                         XYZ00750
      ICASE=3                                                           XYZ00760
      IF (DELZ.GT.1.E-2.OR.DELZ.LT.-1.E-2) GOTO 12                      XYZ00770
      ICASE=4                                                           XYZ00780
   12 CONTINUE                                                          XYZ00790
C*****DEFAULT SPECIAL CASE OF OBS-TAR COINCIDENT                        XYZ00800
      IF(ICASE.NE.4)GO TO 14                                            XYZ00810
      GO TO 800                                                         XYZ00820
   14 CONTINUE                                                          XYZ00830
C*****SET UP TEST TO DETERMINE IF TARGET AND/OR OBSERVER ARE IN THE     XYZ00840
C     INTERIOR OF THE CLOUD                                             XYZ00850
      DO 1 I=1,2                                                        XYZ00860
      IF (NCODE.EQ.2) TEST(I)= ((XL(I)-X0)/X0)**2 + (YL(I)/Y0)**2 +     XYZ00870
     *(ZL(I)/Z0)**2-1.                                                  XYZ00880
      IF (NCODE.EQ.1) TEST(I)=(YL(I)/Y0)**2 + (ZL(I)/Z0)**2 -           XYZ00890
     *(XL(I)/X0)**2                                                     XYZ00900
    1 CONTINUE                                                          XYZ00910
      IF(TEST(1).GT.0.0)GO TO 2                                         XYZ00920
      IF (XL(1).LT.0..OR.ZL(1).LT.0.) GOTO 2                            XYZ00930
      IF (XL(1).GT.X0) GOTO 2                                           XYZ00940
      IF (XL(1).EQ.X0) K1=1                                             XYZ00950
      IF (TEST(1).EQ.0..OR.ZL(1).EQ.0.) K3=1                            XYZ00960
      I1=1                                                              XYZ00970
    2 CONTINUE                                                          XYZ00980
      IF(TEST(2).GT.0.0)GO TO 3                                         XYZ00990
      IF (XL(2).LT.0..OR.ZL(2).LT.0.) GOTO 3                            XYZ01000
      IF (XL(2).GT.X0) GOTO 3                                           XYZ01010
      IF (XL(2).EQ.X0) K2=1                                             XYZ01020
      IF (TEST(2).EQ.0..OR.ZL(2).EQ.0.) K4=1                            XYZ01030
      I2=1                                                              XYZ01040
C*****IF BOTH POINTS ARE IN THE CLOUD SET INTERCEPTS EQUAL TO THE       XYZ01050
C     TARGET-OBSERVER COORDINATES AND RETURN--OTHERWISE CONTINUE        XYZ01060
    3 IF(I1.EQ.0)GO TO 4                                                XYZ01070
      IF(I2.EQ.0)GO TO 4                                                XYZ01080
      DO 5 I=1,2                                                        XYZ01090
      IFLAG(I)=1                                                        XYZ01100
      XINT(I)=XL(I)                                                     XYZ01110
      YINT(I)=YL(I)                                                     XYZ01120
    5 ZINT(I)=ZL(I)                                                     XYZ01130
      IF (K3.EQ.1) IFLAG(1)=2                                           XYZ01140
      IF (K4.EQ.1) IFLAG(2)=2                                           XYZ01150
      IF (K1.EQ.1) IFLAG(1)=3                                           XYZ01160
      IF (K2.EQ.1) IFLAG(2)=3                                           XYZ01170
      GO TO 999                                                         XYZ01180
C*****IF ONLY ONE POINT IS IN CLOUD KEEP TRACK OF IT FOR LATER          XYZ01190
    4 CONTINUE                                                          XYZ01200
      IF(I1.EQ.0)GO TO 6                                                XYZ01210
      INT=1                                                             XYZ01220
    6 IF(I2.EQ.0)GO TO 7                                                XYZ01230
      INT=2                                                             XYZ01240
    7 CONTINUE                                                          XYZ01250
      IF (K1.EQ.1) LEAD=1                                               XYZ01260
      IF (K2.EQ.1) LEAD=2                                               XYZ01270
      IF (K3.EQ.1) ISURF=1                                              XYZ01280
      IF (K4.EQ.1) ISURF=2                                              XYZ01290
C*****SET UP LOS EQUATION DEPENDING UPON CASE                           XYZ01300
      GO TO (10,20,30),ICASE                                            XYZ01310
C*****CASE 1                                                           XYZ01320
   10 SX=DELX/DELX                                                      XYZ01330
      SY=DELY/DELX                                                      XYZ01340
      SZ=DELZ/DELX                                                      XYZ01350
      XI=XL(1)-SX*XL(1)                                                 XYZ01360
      YI=YL(1)-SY*XL(1)                                                 XYZ01370
      ZI=ZL(1)-SZ*XL(1)                                                 XYZ01380
      GO TO 101                                                         XYZ01390
C*****CASE 2                                                           XYZ01400
```

```
   20 SX=DELX/DELY                                                                  XYZ01410
      SY=DELY/DELY                                                                  XYZ01420
      SZ=DELZ/DELY                                                                  XYZ01430
      XI=XL(1)-SX*YL(1)                                                             XYZ01440
      YI=YL(1)-SY*YL(1)                                                             XYZ01450
      ZI=ZL(1)-SZ*YL(1)                                                             XYZ01460
      GO TO 101                                                                     XYZ01470
C*****CASE 3                                                                        XYZ01480
   30 SX=DELX/DELZ                                                                  XYZ01490
      SY=DELY/DELZ                                                                  XYZ01500
      SZ=DELZ/DELZ                                                                  XYZ01510
      XI=XL(1)-SX*ZL(1)                                                             XYZ01520
      YI=YL(1)-SY*ZL(1)                                                             XYZ01530
      ZI=ZL(1)-SZ*ZL(1)                                                             XYZ01540
  101 CONTINUE                                                                      XYZ01550
C*****SET UP QUADRATIC COEFICIENTS                                                  XYZ01560
      IF (NCODE.EQ.1) GOTO 60                                                       XYZ01570
      A=(SX/X0)**2 + (SY/Y0)**2 + (SZ/Z0)**2                                        XYZ01580
      B=2.0*((XI/X0)*(SX/X0) + (YI/Y0)*(SY/Y0) + (ZI/Z0)*(SZ/Z0)                    XYZ01590
     *-(SX/X0))                                                                     XYZ01600
      C=(XI/X0)**2 + (YI/Y0)**2 +(ZI/Z0)**2 - 2.0*XI/X0                             XYZ01610
      GOTO 61                                                                       XYZ01620
   60 A=(SY/Y0)**2 + (SZ/Z0)**2 -(SX/X0)**2                                         XYZ01630
      B=2.0*((SY/Y0)*(YI/Y0) + (SZ/Z0)*(ZI/Z0) + (SX/X0)*(XI/X0))                   XYZ01640
      C=(YI/Y0)**2 + (ZI/Z0)**2 -(XI/X0)**2                                         XYZ01650
   61 CONTINUE                                                                      XYZ01660
C*****DEFAULT ALL INTERCEPTS IF ROOTS ARE COMPLEX                                   XYZ01670
      TEST0=B*B-4.0*A*C                                                             XYZ01680
      IF(TEST0.GE.0.0)GO TO 888                                                     XYZ01690
  800 DO 13 I=1,2                                                                   XYZ01700
      IFLAG(I)=0                                                                    XYZ01710
      XINT(I)=0.0                                                                   XYZ01720
      YINT(I)=0.0                                                                   XYZ01730
   13 ZINT(I)=0.0                                                                   XYZ01740
      GO TO 999                                                                     XYZ01750
C*****SOLVE QUADRATIC FOR X,Y OR Z DEPENDING ON CASE                                XYZ01760
  888 GO TO (100,200,300),ICASE                                                     XYZ01770
  100 XINT(1)=QROOT(+1,A,B,C)                                                       XYZ01780
      XINT(2)=QROOT(-1,A,B,C)                                                       XYZ01790
      DO 11 I=1,2                                                                   XYZ01800
      YINT(I)=YI+SY*XINT(I)                                                         XYZ01810
   11 ZINT(I)=ZI+SZ*XINT(I)                                                         XYZ01820
      GOTO 400                                                                      XYZ01830
  200 YINT(1)=QROOT(+1,A,B,C)                                                       XYZ01840
      YINT(2)=QROOT(-1,A,B,C)                                                       XYZ01850
      DO 21 I=1,2                                                                   XYZ01860
      XINT(I)=XI+SX*YINT(I)                                                         XYZ01870
   21 ZINT(I)=ZI+SZ*YINT(I)                                                         XYZ01880
      GO TO 400                                                                     XYZ01890
  300 ZINT(1)=QROOT(+1,A,B,C)                                                       XYZ01900
      ZINT(2)=QROOT(-1,A,B,C)                                                       XYZ01910
      DO 31 I=1,2                                                                   XYZ01920
      XINT(I)=XI+SX*ZINT(I)                                                         XYZ01930
   31 YINT(I)=YI+SY*ZINT(I)                                                         XYZ01940
C*** TEST FOR VALID INTERCEPTS                                                      XYZ01950
400   I1=0                                                                          XYZ01960
      I2=0                                                                          XYZ01970
      IFLAG(1)=2                                                                    XYZ01980
      IFLAG(2)=2                                                                    XYZ01990
      IF (ZINT(1).GE.0..AND.XINT(1).GE.0..AND.XINT(1).LE.X0) I1=1                   XYZ02000
      IF (ZINT(2).GE.0..AND.XINT(2).GE.0..AND.XINT(2).LE.X0) I2=1                   XYZ02010
      IF (I1.EQ.0.OR.I2.EQ.0) GOTO 450                                              XYZ02020
      IF (XINT(1).EQ.XINT(2).AND.YINT(1).EQ.YINT(2).AND.ZINT(1).EQ.                 XYZ02030
     *    ZINT(2)) I2=0                                                            XYZ02040
      IF (I1.EQ.1.AND.I2.EQ.1) GOTO 600                                             XYZ02050
C*** AT LEAST ONE INTERCEPT INVALID. FIRST COMPUTE POSSIBLE Z=0. INTCPT             XYZ02060
450   GOTO (460,470,480),ICASE                                                     XYZ02070
460   Z=0.                                                                          XYZ02080
      IF (SZ.EQ.0.) GOTO 500                                                        XYZ02090
      X=-ZI/SZ                                                                      XYZ02100
```

```
      Y=YI+SY*X                                                          XYZ02110
      GOTO 490                                                           XYZ02120
470   Z=0.                                                               XYZ02130
      IF (SZ.EQ.0.) GOTO 500                                             XYZ02140
      Y=-ZI/SZ                                                           XYZ02150
      X=XI+SX*Y                                                          XYZ02160
      GOTO 490                                                           XYZ02170
480   Z=0.                                                               XYZ02180
      X=XI                                                               XYZ02190
      Y=YI                                                               XYZ02200
C***  TEST IF Z=0, INTERCEPT VALID.                                      XYZ02210
490   IF (X.GT.X0.OR.X.LT.0.) GOTO 500                                   XYZ02220
      IF (NCODE.EQ.1) TEST3=(Y/Y0)**2-(X/X0)**2                          XYZ02230
      IF (NCODE.EQ.2) TEST3=((X-X0)/X0)**2 + (Y/Y0)**2 - 1.              XYZ02240
      IF (TEST3.GT.0.) GOTO 500                                          XYZ02250
      IF (I1.EQ.1) GOTO 495                                              XYZ02260
      XINT(1)=X                                                          XYZ02270
      YINT(1)=Y                                                          XYZ02280
      ZINT(1)=Z                                                          XYZ02290
      I1=1                                                               XYZ02300
      IF (I2.EQ.1) GOTO 600                                              XYZ02310
      GOTO 500                                                           XYZ02320
495   XINT(2)=X                                                          XYZ02330
      YINT(2)=Y                                                          XYZ02340
      ZINT(2)=Z                                                          XYZ02350
      I2=1                                                               XYZ02360
      IF (I1.EQ.1) GOTO 600                                              XYZ02370
C***  NEXT COMPUTE INTERCEPT ON X=X0                                     XYZ02380
500   GOTO (510,505,507),ICASE                                           XYZ02390
505   X=X0                                                               XYZ02400
      Y=(X-XI)/SX                                                        XYZ02410
      Z=ZI+SZ*Y                                                          XYZ02420
      GOTO 515                                                           XYZ02430
507   X=X0                                                               XYZ02440
      Z=(X-XI)/SX                                                        XYZ02450
      Y=YI+SY*Z                                                          XYZ02460
      GOTO 515                                                           XYZ02470
510   X=X0                                                               XYZ02480
      Y=YI+SY*X                                                          XYZ02490
      Z=ZI+SZ*X                                                          XYZ02500
515   IF (Z.LT.0.) GOTO 800                                              XYZ02510
      TEST4=(Y/Y0)**2 + (Z/Z0)**2 - 1.                                   XYZ02520
      IF (TEST4.GT.0.) GOTO 800                                          XYZ02530
      IF (I1.EQ.1) GOTO 520                                              XYZ02540
      XINT(1)=X                                                          XYZ02550
      YINT(1)=Y                                                          XYZ02560
      ZINT(1)=Z                                                          XYZ02570
      I1=1                                                               XYZ02580
      IF (I2.EQ.0) GOTO 800                                              XYZ02590
      GOTO 600                                                           XYZ02600
520   XINT(2)=X                                                          XYZ02610
      YINT(2)=Y                                                          XYZ02620
      ZINT(2)=Z                                                          XYZ02630
      I2=1                                                               XYZ02640
600   DO 620 I=1,2                                                       XYZ02650
      X1=(XL(1)-XINT(I))**2 + (YL(1)-YINT(I))**2 + (ZL(1)-ZINT(I))**2    XYZ02660
      X2=(XL(2)-XINT(I))**2 + (YL(2)-YINT(I))**2 + (ZL(2)-ZINT(I))**2    XYZ02670
      DIST(I)=SQRT(X1)+SQRT(X2)                                          XYZ02680
620   CONTINUE                                                           XYZ02690
      TEST5=1.0001*SQRT(DELX**2+DELY**2+DELZ**2)                         XYZ02700
      IF(DIST(1).GT.TEST5.AND.DIST(2).GT.TEST5) GOTO 800                 XYZ02710
      IF(INT.EQ.0) GOTO 700                                              XYZ02720
C***  REPLACE ONE INTERCEPT WITH INTERIOR OBS/TGT.                       XYZ02730
      DO 610 I=1,2                                                       XYZ02740
      IF (XL(INT).GT.(XINT(I)+.001).OR.XL(INT).LT.(XINT(I)-.001))        XYZ02750
     * GOTO 610                                                          XYZ02760
      IF (YL(INT).GT.(YINT(I)+.001).OR.YL(INT).LT.(YINT(I)-.001))        XYZ02770
     * GOTO 610                                                          XYZ02780
      IF (ZL(INT).GT.(ZINT(I)+.001).OR.ZL(INT).LT.(ZINT(I)-.001))        XYZ02790
     *   GOTO 610                                                        XYZ02800
```

```
      IFLAG(I)=1                                                        XYZ02810
      IF (LEAD.EQ.INT) IFLAG(I)=3                                       XYZ02820
      IF (ISURF.EQ.INT) IFLAG(I)=2                                      XYZ02830
      GOTO 700                                                          XYZ02840
610   CONTINUE                                                          XYZ02850
C***  REPLACE CLOSEST INTERCEPT OUTSIDE LOS RANGE                       XYZ02860
      K1=1                                                              XYZ02870
      IF (DIST(1).LT.DIST(2)) K1=2                                      XYZ02880
      XINT(K1)=XL(INT)                                                  XYZ02890
      YINT(K1)=YL(INT)                                                  XYZ02900
      ZINT(K1)=ZL(INT)                                                  XYZ02910
      IFLAG(K1)=1                                                       XYZ02920
700   IF (XINT(1  .E.(X0-.001).AND.XINT(1).LE.(X0+.001)) IFLAG(1)=3     XYZ02930
      IF (XINT(2  .E.(X0-.001).AND.XINT(2).LE.(X0+.001)) IFLAG(2)=3     XYZ02940
 999  RETURN                                                            XYZ02950
      END                                                               XYZ02960
```

```
      SUBROUTINE BRATE(IERR,MUNRD,TYPM,XN,FW,TBURN,ITYPE,EFF,YF,          BRA00010
     *BRAT1,BRAT2,BRAT3,BRAT4,BRAT5)                                      BRA00020
C                                                                         BRA00030
C     THIS ROUTINE PROVIDES DEFAULT MUNITION CHARACTERISTIC VALUES TO     BRA00040
C     SMOKE.  SMOKE MUNITION TYPES (TYPM) ARE GIVEN IN THE COMMENTS       BRA00050
C     OF THE MAIN ROUTINE IN SMOKE                                        BRA00060
C                                                                         BRA00070
      DIMENSION B1(21),B2(21),B3(21),B4(21),B5(21)                        BRA00080
      DIMENSION F(21),T(21),E(21),IT(21)                                  BRA00090
      DATA B1 / .537,.631,.2218,.537,.2218,1.,1.,1.,1.,1.,1.,0.,          BRA00100
     *      0.,.521,1.631,1.808,.1204,.653,1.731,0.,1. /                  BRA00110
      DATA B2 /.476,-.4985,3.915,.476,3.915,0.,0.,0.,0.,0.,0.,  0.,  0.,BRA00120
     *         2.106,.678,-2.556,3.1012,-3.136,-2.852,3.6832,0./          BRA00130
      DATA B3 / 4.779,6.745,-1.7368,4.779,-1.7368,0.,0.,0.,0.,0.,0.,      BRA00140
     * 0.,  0.,   -1.11,-5.907,2.883,-2.2104,15.309,4.341,-5.3472,0./     BRA00150
      DATA B4 / -5.472,-6.52,-2.3995,-5.472,-2.3995,0.,0.,0.,0.,0.,0.,0., BRA00160
     * 0.,  0.,   -.748,4.012,-2.008,.206,-12.872,-3.108,3.8348,0./       BRA00170
      DATA B5 / 11*0.,2*1.,8*0./                                          BRA00180
      DATA IT /3,3,3,3,3,3,3,1,1,1,1,1,2,2,2,2,2,2,5,5,5,4 /              BRA00190
      DATA  T /100.,  70.,  120.,  100.,  120.,  900.,  1.,  1.,  1.,  1.,  1.,  600.,  BRA00200
     *     600.,240.,470.,390.,721.,260.,380.,750.,900. /                 BRA00210
C                                                                         BRA00220
      DATA  F /5.46,2.69,1.65,17.19,7.50,30.,0.76,1.75,8.14,15.6,         BRA00230
     *          3.83,8.14,13.52,.463,.139,.234,19.98,.128,.0243,          BRA00240
     *          19.4,40.0 /                                               BRA00250
      DATA E / 40.,40.,40.,40.,40.,24.,100.,100.,100.,100.,100.,          BRA00260
     *         60.,60.,66.,71.,67.,77.,53.,55.,51.,100. /                 BRA00270
      MAXS=21                                                             BRA00280
      ITP=IFIX(TYPM+.0001)                                                BRA00290
      IF (ITP.EQ.0) RETURN                                                BRA00300
      IF (ITP.LE.MAXS) GOTO 10                                            BRA00310
      IERR=1                                                              BRA00320
      RETURN                                                              BRA00330
10    BRAT1=B1(ITP)                                                       BRA00340
      BRAT2=B2(ITP)                                                       BRA00350
      BRAT3=B3(ITP)                                                       BRA00360
      BRAT4=B4(ITP)                                                       BRA00370
      BRAT5=B5(ITP)                                                       BRA00380
      ITYPE=IT(ITP)                                                       BRA00390
      IF (MUNRD.NE.0) GOTO 20                                             BRA00400
      XN=1.                                                               BRA00410
      YF=0.                                                               BRA00420
      TBURN=T(ITP)                                                        BRA00430
      EFF=E(ITP)                                                          BRA00440
      FW=F(ITP)                                                           BRA00450
      RETURN                                                              BRA00460
20    IF (XN.EQ.0.) XN=1.                                                 BRA00470
      IF (TBURN.EQ.0.) TBURN=T(ITP)                                       BRA00480
      IF (EFF.EQ.0.) EFF=E(ITP)                                           BRA00490
      IF (FW.EQ.0.) FW=F(ITP)                                             BRA00500
      RETURN                                                              BRA00510
      END                                                                 BRA00520
```

```
      SUBROUTINE GPUFF(KCALL,CLGAUS,A,B,C,C2,XPP1,XPP2,YPP1,YPP2,ZPP1,   GPU00010
     *ZPP2)                                                             GPU00020
C                                                                       GPU00030
C   THIS ROUTINE COMPUTES THE CL CONTRIBUTION FROM AN EXOTHERMIC,       GPU00040
C     BUOYANTLY RISING SMOKE CLOUD OF GAUSSIAN DISTRIBUTION AND         GPU00050
C     UNIT CONCENTRATION.  IT IS CENTERED ON THE LEADING EDGE AT        GPU00060
C     COORDINATES  (A,B,C-2*SIGZ) WHERE                                 GPU00070
C  INPUTS    A    = CLOUD DOWNWIND DISTANCE                           GPU00080
C            B    = CLOUD BASE HALF-WIDTH (AT LEADING EDGE)           GPU00090
C            C    = CLOUD HEIGHT (AT LEADING EDGE)                    GPU00100
C            C2   = MOMENTUM RISE COEFFICIENT. (USUALLY BRIGGS OR LIMITED GPU00110
C                     RISE BRIGGS)                                     GPU00120
C            XPP1,...ZPP2 COORDINATES OF TARGET AND OBSERVER IN MUNITION GPU00130
C                     CENTERED COORDINATE SYSTEM WITH WIND VECTOR X-AXIS GPU00140
C  OUTPUT    CLGAUS = CL VALUE FOR UNIT MASS (METER**-2)               GPU00150
C            KCALL  = SET TO 1 AFTER EVERY CHANGE IN TGT/OBS COORD.    GPU00160
      DIMENSION AV(2),AP(2),BP(2),CP(2)                                 GPU00170
      CLGAUS=0.                                                         GPU00180
      IF (KCALL.NE.0) GOTO 100                                          GPU00190
      KCALL=1                                                           GPU00200
      ICASE=0                                                           GPU00210
C*** COMPUTE LOS GENERALIZED COORDINATES.                               GPU00220
      IF (ZPP1.LT.0. .AND. ZPP2.LT.0.) RETURN                           GPU00230
      DELX=XPP2-XPP1                                                    GPU00240
      DELY=YPP2-YPP1                                                    GPU00250
      DELZ=ZPP2-ZPP1                                                    GPU00260
      IF (ABS(DELX).LE..01*ABS(DELY) .OR. ABS(DELX).LE..01*ABS(DELZ))   GPU00270
     1    GOTO 10                                                       GPU00280
      ICASE=1                                                           GPU00290
      SX=DELX/DELX                                                      GPU00300
      SY=DELY/DELX                                                      GPU00310
      SZ=DELZ/DELX                                                      GPU00320
      XI=XPP1-SX*XPP1                                                   GPU00330
      YI=YPP1-SY*XPP1                                                   GPU00340
      ZI=ZPP1-SZ*XPP1                                                   GPU00350
      AV(1)=XPP1                                                        GPU00360
      AV(2)=XPP2                                                        GPU00370
      GOTO 80                                                           GPU00380
10    IF (ABS(DELY).LE..01*ABS(DELZ)) GOTO 20                           GPU00390
      ICASE=2                                                           GPU00400
      SX=DELX/DELY                                                      GPU00410
      SY=DELY/DELY                                                      GPU00420
      SZ=DELZ/DELY                                                      GPU00430
      XI=XPP1-SX*YPP1                                                   GPU00440
      YI=YPP1-SY*YPP1                                                   GPU00450
      ZI=ZPP1-SZ*YPP1                                                   GPU00460
      AV(1)=YPP1                                                        GPU00470
      AV(2)=YPP2                                                        GPU00480
      GOTO 80                                                           GPU00490
20    IF (ABS(DELZ).LT..001) RETURN                                     GPU00500
      ICASE=3                                                           GPU00510
      SX=DELX/DELZ                                                      GPU00520
      SY=DELY/DELZ                                                      GPU00530
      SZ=DELZ/DELZ                                                      GPU00540
      XI=XPP1-SX*ZPP1                                                   GPU00550
      YI=YPP1-SY*ZPP1                                                   GPU00560
      ZI=ZPP1-SZ*ZPP1                                                   GPU00570
      AV(1)=ZPP1                                                        GPU00580
      AV(2)=ZPP2                                                        GPU00590
80    IF (ZPP1.LT.0.) AV(1)=-ZI/SZ                                      GPU00600
      IF (ZPP2.LT.0.) AV(2)=-ZI/SZ                                      GPU00610
      SMUL=SQRT(SX*SX+SY*SY+SZ*SZ)                                      GPU00620
100   IF (ICASE.EQ.0) RETURN                                            GPU00630
C*** COMPUTE GAUSSIAN PARAMETERS, REAL AND REFLECTED IMAGE CLOUD TO     GPU00640
C     ACCOUNT FOR GROUND REFLECTED SMOKE                                GPU00650
      SIGZ=(2.73+C2*A)/2.15                                             GPU00660
      ZB=C-(2.73+C2*A)                                                  GPU00670
      IF (ZB.GE.C) RETURN                                               GPU00680
      SIGY=B*SQRT(1.-(ZB/C)**2)/2.15                                    GPU00690
      IF (ZB.LT.0.) SIGY=B/2.15                                         GPU00700
```

```
      SIGX=SIGY                                                          GPU00710
      ASIG=(SX/SIGX)**2 + (SY/SIGY)**2 + (SZ/SIGZ)**2                    GPU00720
      BMEAN=2.*((SX*(XI-A)/SIGX**2) + (SY*YI/SIGY**2))                   GPU00730
      BP(1)=BMEAN+2.*SZ*(ZI-ZB)/SIGZ**2                                  GPU00740
      BP(2)=BMEAN+2.*SZ*(ZI+ZB)/SIGZ**2                                  GPU00750
      CTOT=((XI-A)/SIGX)**2 + (YI/SIGY)**2                               GPU00760
      CP(1)=CTOT + ((ZI-ZB)/SIGZ)**2                                     GPU00770
      CP(2)=CTOT + ((ZI+ZB)/SIGZ)**2                                     GPU00780
C***  CALCULATE FOR LOS INTEGRAL                                         GPU00790
      CMUL=SMUL/(2.*3.14159*SIGX*SIGY*SIGZ*SQRT(ASIG))                   GPU00800
      DO 220 I=1,2                                                       GPU00810
      CEXU=.5*(CP(I)-(BP(I)**2)/(4.*ASIG))                               GPU00820
      IF (CEXU.GT.20.) GOTO 220                                          GPU00830
C***  INFINITE PATH LOS                                                  GPU00840
      CLU=EXP(-CEXU)                                                     GPU00850
C***  CORRECTION FOR FINITE PATH                                         GPU00860
      DO 210 J=1,2                                                       GPU00870
      AP1=(AV(J)+BP(I)/(2.*ASIG))*SQRT(ASIG/2.)                          GPU00880
      P1=ABS(AP1)                                                        GPU00890
      CP1=0.                                                             GPU00900
      IF (P1.LE.5.) CP1=0.5/(1.+P1*(.0705230784+P1*(.042282013+P1*(      GPU00910
     *.0092705272+P1*(.0001520134+P1*(.0002765672+P1*.0000430638))))))   GPU00920
     ***16                                                               GPU00930
      IF (AP1.GE.0.) CP1=1.-CP1                                          GPU00940
      AP(J)=CP1                                                          GPU00950
210   CONTINUE                                                           GPU00960
      CLGAUS=CLGAUS+CLU*ABS(AP(2)-AP(1))                                 GPU00970
220   CONTINUE                                                           GPU00980
      CLGAUS=CMUL*CLGAUS                                                 GPU00990
      RETURN                                                             GPU01000
      END                                                                GPU01010
```

```
      SUBROUTINE LZTRAN(WAVE1,ICLMAT,LAZTRN,IERR)                       LZT00010
C                                                                       LZT00020
C   THIS SUBROUTINE CALCULATES MOLECULAR ABSORPTION COEFFICIENTS AT     LZT00030
C   LASER FREQUENCIES. PH2O AND T ARE THE PARTIAL PRESSURE OF WATER     LZT00040
C   VAPOR AND TEMPERATURE, IN TORR AND DEGREES K RESPECTIVELY.          LZT00050
C   LID IS THE LASER LINE IDENTIFICATION AS DESCRIBED BELOW.            LZT00060
C   ABCOEF IS THE ABSORPTION COEFFICIENT RETURNED IN KM-1.              LZT00070
C                                                                       LZT00080
C*****************************************************************    LZT00090
C         INPUT                                                         LZT00100
C   CARD 1    LNAME1, LNAME2, PH2O, T, LZPATH FORMAT (2A4,3F10.3)       LZT00110
C                                                                       LZT00120
C      LNAME1  FIRST 4 CHARS OF LASER LINE   (A4)                       LZT00130
C      LNAME2  SECOND 4 CHARS OF LASER LINE   (A4)                      LZT00140
C           ***IF LNAME NOT ENTERED WILL USE WAVELENGTH READ IN MAIN     LZT00150
C                                                                       LZT00160
C      PH2O     WATER VAPOR  PRESSURE  0. TO .35   (MB)  (F10.3)       LZT00170
C                                                                       LZT00180
C      T        AMBIENT AIR TEMPERATURE 260. TO .320  (C)  (F10.3)     LZT00190
C                                                                       LZT00200
C      LZPATH        PATHLENGTH IN KM F(F10.3)                         LZT00210
C           ***PH2O, AND T NOT REQUIRED WHEN ICLIMATE(IN MAIN)=1 OR 2   LZT00220
C*****************************************************************    LZT00230
C         OUTPUT                                                        LZT00240
C      LZTRAN    TRANSMISSION                                          LZT00250
C                                                                       LZT00260
C      NOTES                                                            LZT00270
C     ABCOEF  RETURNS THE ABSORPTION COEFFICIENT (KM-1)                 LZT00280
C                                                                       LZT00290
C     LNAME1  FIRST 4 CHARS OF LASER LINE ON NORMAL RETURN              LZT00300
C                                  BLANK ON ERROR RETURN                LZT00310
C                                                                       LZT00320
C     LNAME2  SECOND 4 CHARS OF LASER LINE ON NORMAL RETURN             LZT00330
C                                    BLANK ON ERROR RETURN              LZT00340
C                                                                       LZT00350
C      ++ CALLED PROGRAMS ++                                            LZT00360
C                                                                       LZT00370
C     LZIDNM                                                            LZT00380
C                                                                       LZT00390
C*****************************************************************    LZT00400
C  LASER LINE IDENTIFICATION                                            LZT00410
C   LID=1   ND:YAG LASER, 1.06 MICRONS                                  LZT00420
C   LID=2 CO2 LASER LINE P(20)  ,  10.591 MICRONS                       LZT00430
C   LID=101 TO 127 DF LASER, 3.521 TO 4.089 MICRONS                     LZT00440
C    101 P3(12) * 107 P3(8)  * 113 P3(5)  * 119 P2(5)  * 125 P1(5)      LZT00450
C    102 P3(11) * 108 P2(11) * 114 P2(8)  * 120 P1(8)  * 126 P1(4)      LZT00460
C    103 P3(10) * 109 P3(7)  * 115 P2(7)  * 121 P2(4)  * 127 P1(3)      LZT00470
C    104 P2(13) * 110 P2(10) * 116 P1(10) * 122 P1(7)  *                LZT00480
C    105 P3(9)  * 111 P3(6)  * 117 P2(6)  * 123 P2(3)  *                LZT00490
C    106 P2(12) * 112 P2(9)  * 118 P1(9)  * 124 P1(6)  *                LZT00500
C   LID=201 TO 219 CO LASER, 4.908 TO 5.088 MICRONS                     LZT00510
C    201 P6(12) * 205 P6(8)  * 209 P5(12)  * 213 P5(8)  * 217 P4(9)     LZT00520
C    202 P6(11) * 206 P5(15) * 210 P5(11)  * 214 P5(7)  * 218 P4(8)     LZT00530
C    203 P6(10) * 207 P5(14) * 211 P5(10)  * 215 P4(11) * 219 P4(7)     LZT00540
C    204 P6(9)  * 208 P5(13) * 212 P5(9)   * 216 P4(10) *               LZT00550
C   LID=301 TO 305  GA AS LASER, (GA.85 TO GA.950) LASER LINE NAMES     LZT00560
C    301 0.850 MICROMETERS      *  304  0.925 MICROMETERS               LZT00570
C    302 0.875 MICROMETERS      *  305  0.950 MICROMETERS               LZT00580
C    303 0.900 MICROMETERS      *                                       LZT00590
      INTEGER LNAME1,LNAME2,BLANK,LNAME3,LNAME4                         LZT00600
      REAL LAZTRN,LZPATH                                                LZT00610
      DIMENSION ADF0(30),ADF1(30),ADF2(30),ADF3(30),ADF4(30)            LZT00620
     1         ,ADF5(30)                                                LZT00630
      DIMENSION ACO0(20),ACO1(20),ACO2(20),ACO3(20),ACO4(20)            LZT00640
     1         ,ACO5(20)                                                LZT00650
      DIMENSION AGA0(5),AGA1(5),AGA2(5),AGA3(5),AGA4(5),                LZT00660
     1          AGA5(5),AGA6(5),AGA7(5),AGA8(5)                         LZT00670
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     LZT00680
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,             LZT00690
     1       FOGPRB,WNDVEL,WNDDIR,IPASCT                                LZT00700
```

```
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTULZT00710
      COMMON /GEOMET/PTS(15),IGEOSW                                     LZT00720
C     THE POLYNOMIAL COEFFICIENTS ARE SELECTED BY THE LID.  THE         LZT00730
C   INDEX FOR THE COEFFICIENT ARRAYS FOR THE DF LASER IS                LZT00740
C   I = LID - 100.  NOW I IS IN THE RANGE 1..27 SINCE THERE ARE         LZT00750
C   27 DF LASER LINES.  WHEN THE POLYNOMIAL IS EVALUATED I IS           LZT00760
C   USED TO INDEX THE ARRAYS ADF/0..5/ THUS SELECTING THE               LZT00770
C   CORRECT COEFFICIENTS FOR THE LASER LINE SELECTED.                   LZT00780
C   COEFFICIENTS FOR THE OTHER LASER POLYNOMIALS ARE SELECTED           LZT00790
C   IN THE SAME FASHION.                                                LZT00800
C   POLYNOMIAL COEFFICIENTS  FOR DF LASER LINES, (1..27)                LZT00810
      DATA ADF0/.1019,.08352,.04083,.03675,.02042,.01833,               LZT00820
     1      .04738,.03134,.07870,.05844,.1096E-2,.8353E-               LZT00830
     2      2,.2537E-2,.3254E-2,-.1103E-2,-.6471E-3,-.1423E-           LZT00840
     3      3,-.4664E-2,-.1221E-4,-.1698E-3,.1172E-3,.6195E-           LZT00850
     4      2,.1272E-2,.5485E-2,.1651E-2,.6913E-2,-.4498E-             LZT00860
     5      2,3*0./                                                    LZT00870
      DATA ADF1/-.9718E-4,-.1160E-3,-.4892E-4,-.4230E-4,-               LZT00880
     1      .1750E-4,-.1524E-4,-.5589E-4,-.4642E-4,-.1218E-            LZT00890
     2      3,-.3683E-4,.1672E-4,-.1346E-4,-.4765E-6,-.8548E-          LZT00900
     3      6,.6089E-5,.8897E-5,.1507E-5,.4448E-4,.4540E-              LZT00910
     4      7,.5569E-6,.1327E-6,-.1462E-4,-.7524E-6,-.1398E-           LZT00920
     5      5,-.7025E-6,-.1044E-4,.1816E-4,3*0./                       LZT00930
      DATA ADF2/.9666E-2,.7252E-2,.7050E-2,.7142E-2,.6320E-             LZT00940
     1      2,.6191E-2,.5400E-2,.5344E-2,.4064E-2,.4682E-              LZT00950
     2      2,.3734E-2,.5839E-2,.5075E-2,.2484E-2,.8190E-              LZT00960
     3      2,.6920E-2,.8779E-2,.7914E-2,.7094E-2,.5327E-              LZT00970
     4      3,.6692E-2,.9452E-2,.01025,.01367,.01279,.7844E-           LZT00980
     5      2,.01201,3*0./                                             LZT00990
      DATA ADF3/-.2655E-4,-.1805E-4,-.1879E-4,-.1937E-4,-               LZT01000
     1      .1704E-4,-.1669E-4,-.1412E-4,-.1417E-4,-.9076E-            LZT01010
     2      5,-.1165E-4,-.7371E-5,-.9113E-5,-.1290E-4,-                LZT01020
     3      .1070E-5,-.6746E-5,-.1573E-4,-.1683E-4,-.1883E-            LZT01030
     4      4,-.1774E-4,.3002E-4,-.1025E-4,-.2488E-4,-.2596E-          LZT01040
     5      4,-.2848E-4,-.2858E-4,-.4991E-5,-.2608E-4,3*0./            LZT01050
      DATA ADF4/.7847E-4,.5729E-4,.5585E-4,.5606E-4,.4847E-             LZT01060
     1      4,.4668E-4,.4145E-4,.4218E-4,.3314E-4,.3642E-              LZT01070
     2      4,.3170E-4,.3682E-4,.3835E-4,.4798E-4,.3537E-              LZT01080
     3      4,.4635E-4,.4941E-4,.5601E-4,.5494E-4,-.1120E-             LZT01090
     4      4,.6567E-4,.9104E-4,.7398E-4,.8509E-4,.8746E-              LZT01100
     5      4,.1050E-3,.1060E-3,3*0./                                  LZT01110
      DATA ADF5/-.2056E-6,-.1374E-6,-.1408E-6,-.1432E-6,-               LZT01120
     1      .1222E-6,-.1172E-6,-.1011E-6,-.1054E-6,-.7169E-            LZT01130
     2      7,-.8600E-7,-.6588E-7,-.8998E-7,-.9081E-7,-                LZT01140
     3      .1320E-6,-.1570E-7,-.9751E-7,-.8159E-7,-.1234E-            LZT01150
     4      6,-.1313E-6,.1836E-6,-.1469E-6,-.2540E-6,-.1796E-          LZT01160
     5      6,-.1691E-6,-.1886E-6,-.2635E-6,-.2855E-6,3*0./            LZT01170
C   POLYNOMIAL COEFFICIENTS FOR CO LASER LINES, (1..19)                 LZT01180
      DATA AC00/-1.813E-3,-9.289E-4,1.153E-3,-1.985E-3,-                LZT01190
     1      4.523E-3,-1.205E-3,-2.225E-4,-4.061E-3,-4.522E-            LZT01200
     2      2,-2.267E-6,-5.917E-3,-1.423E-3,-3.640E-3,1.096E-          LZT01210
     3      3,6.455E-4,-3.922E-3,-1.873E-5,-1.055E-4,1.489E-           LZT01220
     4      2,0./                                                      LZT01230
      DATA AC01/3.426E-5,3.658E-6,-1.372E-6,7.229E-6,1.641E-            LZT01240
     1      5,6.428E-6,1.042E-7,1.435E-5,1.593E-4,-5.334E-             LZT01250
     2      7,1.498E-5,5.284E-6,1.806E-5,-3.651E-6,-1.303E-            LZT01260
     3      6,1.835E-5,4.755E-6,2.330E-6,6.196E-6,0./                  LZT01270
      DATA AC02/8.813E-2,-1.020E-1,4.881E-2,6.872E-2,-                  LZT01280
     1      6.244E-1,4.474E-2,1.226E-2,-1.462E-2,1.490E-               LZT01290
     2      1,1.428E-2,1.934,9.034E-3,-1.091E-1,1.284E-2,-             LZT01300
     3      2.131E-2,6.543E-3,2.824E-2,9.463E-3,-9.885E-2,0./          LZT01310
      DATA AC03/8.384E-4,1.211E-3,-4.687E-6,-7.765E-                    LZT01320
     1      5,2.641E-3,-3.135E-5,1.620E-4,6.707E-4,2.581E-             LZT01330
     2      3,4.018E-4,-3.113E-3,9.692E-5,7.907E-4,1.070E-             LZT01340
     3      4,3.026E-4,1.216E-4,-1.859E-5,4.274E-5,6.241E-             LZT01350
     4      4,0./                                                      LZT01360
      DATA AC04/2.850E-4,-4.934E-5,-2.176E-5,1.253E-                    LZT01370
     1      4,1.158E-3,1.926E-5,9.823E-5,-2.395E-4,-6.255E-            LZT01380
     2      5,-6.630E-5,4.851E-3,-5.183E-5,-6.993E-4,-8.196E-          LZT01390
     3      5,-2.239E-4,-1.120E-4,5.415E-5,-8.393E-5,-5.893E-          LZT01400
```

```
     4      4,0./                                                          LZT01410
      DATA ACO5/4.209E-6,3.426E-6,5.383E-7,6.252E-8,-4.149E-              LZT01420
     1      6,2.374E-7,-8.637E-9,3.454E-6,1.526E-5,2.140E-6,-             LZT01430
     2      4.282E-6,5.433E-7,4.231E-6,7.778E-7,1.695E-                   LZT01440
     3      6,8.830E-7,-1.548E-8,4.629E-7,3.241E-6,0./                    LZT01450
C  POLYNOMIAL COEFFICIENTS FOR CO2 LASER                                  LZT01460
      DATA ACO20,ACO21,ACO22,ACO23,ACO24,ACO25,ACO26,ACO27,              LZT01470
     1      ACO28/.4488,-4.1864E-3,5.7903E-2,-3.6770E-                    LZT01480
     2      4,3.8521E-3,-4.7330E-6,1.0000E-5,6.0131E-7,-                  LZT01490
     3      1.7441E-8/                                                    LZT01500
C  POLYNOMIAL COEFFICIENTS FOR GA AS LASER, (1..5)                        LZT01510
      DATA AGA0/7.947E-3,7.590E-3,1.010,0.6094,4.271/                    LZT01520
      DATA AGA1/-3.543E-5,-3.544E-5,-8.012E-3,-4.777E-3,-                LZT01530
     1      3.371E-2/                                                     LZT01540
      DATA AGA2/1.740E-4,1.203E-4,8.736E-2,4.527E-2,.3364/               LZT01550
      DATA AGA3/-5.855E-7,-5.093E-7,-2.135E-4,-1.104E-4,-                LZT01560
     1      8.425E-4/                                                     LZT01570
      DATA AGA4/-1.282E-5,-9.236E-6,-3.964E-3,-2.154E-3,-                LZT01580
     1      1.615E-2/                                                     LZT01590
      DATA AGA5/5.586E-8,4.720E-8,1.596E-5,8.512E-6,6.350E-              LZT01600
     1      5/                                                            LZT01610
      DATA AGA6/5.124E-8,5.235E-8,1.600E-5,9.392E-6,6.705E-              LZT01620
     1      5/                                                            LZT01630
      DATA AGA7/2.706E-10,4.379E-10,-1.396E-7,-6.982E-8,-                LZT01640
     1      4.721E-7/                                                     LZT01650
      DATA AGA8/-4.963E-11,-5.728E-11,-1.125E-8,-5.657E-9,-              LZT01660
     1      4.113E-8/                                                     LZT01670
      DATA BLANK/1H /                                                    LZT01680
C     CHANGE ACCURACY TO 3 DECIMAL PTS (PGM DATA LIMIT)                   LZT01690
      WAVEL=FLOAT(IFIX(1000.*WAVE1))/1000.                               LZT01700
      READ (IOIN,1100) LNAME1,LNAME3,LNAME2,LNAME4,PH2O,T,LZPATH         LZT01710
      IF(IGEOSW.NE.1)GO TO 99                                            LZT01720
      LZPATH=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+                 LZT01730
     +(PTS(6)-PTS(3))**2)                                                LZT01740
   99 CONTINUE                                                           LZT01750
C     CHANGE UNITS - MB TO TORR; C TO K                                   LZT01760
      PH2O=PH2O/TORRMB                                                   LZT01770
      T=T+CDEGK                                                          LZT01780
      IF (ICLMAT.EQ.1) T=TEMP+CDEGK                                      LZT01790
      IF (ICLMAT.EQ.1) PH2O=6.11*10.**                                   LZT01800
     1  (7.5*TEMP/(TEMP+237.3))*RH/(100.*TORRMB)                        LZT01810
C  PRINT HEADER WHEN THE WAVELENGTH CHANGES                              LZT01820
      IF (OLDWAV.NE.WAVEL) WRITE (IOOUT,1000)                            LZT01830
      OLDWAV=WAVEL                                                       LZT01840
      ABCOEF=0.                                                          LZT01850
      IF (WAVEL.EQ.0.0) GO TO 100                                        LZT01860
      IF (WAVEL.LT.0.8.OR.WAVEL.GT.11.0) GO TO 900                       LZT01870
  100 CALL LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID)                 LZT01880
      IF (LID.EQ.0) RETURN                                               LZT01890
      P2=PH2O*PH2O                                                       LZT01900
      IF (T.GE.260.AND.T.LE.320.AND.PH2O.GE.0.AND.                       LZT01910
     1    PH2O.LE.35) GO TO 200                                          LZT01920
C  PRINT WARNING THAT TEMP OR PRESSURE IS OUT OF RANGE FOR               LZT01930
C  ACCURATE CALCULATIONS AND CONTINUE.                                   LZT01940
      WRITE (IOOUT,1300)                                                 LZT01950
  200 IF (LID.GT.100) GO TO 500                                          LZT01960
      IF (LID.LT.1.OR.LID.GT.2) GO TO 900                                LZT01970
      IF (LID-2) 300,400,900                                             LZT01980
C    ND:YAG LASER. NO MOLECULAR ABSORPTION AT 1.06 MICRONS.               LZT01990
  300 GO TO 800                                                          LZT02000
C   CO2 LASER LINE P(20)                                                 LZT02010
  400 T2=T*T                                                             LZT02020
      ABCOEF=ACO20+ACO21*T+ACO22*PH2O+ACO23*T*PH2O+ACO24*P2+             LZT02030
     1       ACO25*T*P2+ACO26*T2+ACO27*T2*PH2O+ACO28*T2*P2               LZT02040
      GO TO 800                                                          LZT02050
  500 IF (LID.GT.200) GO TO 600                                          LZT02060
C   DF LASER.  I IS THE LASER LINE INDEX                                 LZT02070
      I=LID-100                                                          LZT02080
      IF (I.GT.27) GO TO 900                                             LZT02090
      ABCOEF=ADF0(I)+ADF1(I)*T+ADF2(I)*PH2O+ADF3(I)*T*PH2O+              LZT02100
```

```
     1        ADF4(I)*P2+ADF5(I)*T*P2                                    LZT02110
      GO TO 800                                                          LZT02120
  600 IF (LID.GT.300) GO TO 700                                          LZT02130
C  CO LASER.  I IS THE LASER LINE INDEX.                                 LZT02140
      I=LID-200                                                          LZT02150
      IF (I.GT.19) GO TO 900                                             LZT02160
      ABCOEF=ACO0(I)+ACO1(I)*T+ACO2(I)*PH2O+ACO3(I)*T*PH2O+              LZT02170
     1        ACO4(I)*P2+ACO5(I)*T*P2                                    LZT02180
      GO TO 800                                                          LZT02190
  700 IF (LID.GT.400) GO TO 900                                          LZT02200
C  GA AS LASER.  I IS WAVELENGTH INDEX.                                  LZT02210
      I=LID-300                                                          LZT02220
      IF (I.GT.5) GO TO 900                                              LZT02230
      T2=T*T                                                             LZT02240
      ABCOEF=AGA0(I)+AGA1(I)*T+AGA2(I)*PH2O+AGA3(I)*T*PH2O+              LZT02250
     1        AGA4(I)*P2+AGA5(I)*T*P2+AGA6(I)*T2+AGA7(I)*T2*             LZT02260
     2        PH2O+AGA8(I)*T2*P2                                         LZT02270
      GO TO 800                                                          LZT02280
C  NORMAL RETURN                                                        LZT02290
  800 IF (ABCOEF.LT.0.) ABCOEF=0.                                        LZT02300
C     COMPUTE TRANSMISSION                                              LZT02310
      LAZTRN=EXP(-LZPATH*ABCOEF)                                         LZT02320
      WRITE (IOOUT,1200) WAVEL,PH2O,T,ABCOEF,LNAME1,LNAME3,              LZT02330
     +LNAME2,LNAME4,LZPATH,LAZTRN                                        LZT02340
      RETURN                                                             LZT02350
C  ERROR RETURN                                                         LZT02360
  900 WRITE (IOOUT,1400)                                                 LZT02370
      LNAME1=BLANK                                                       LZT02380
      LNAME3=BLANK                                                       LZT02390
      LNAME2=BLANK                                                       LZT02400
      LNAME4=BLANK                                                       LZT02410
      LAZTRN=1.                                                          LZT02420
      IERR=1                                                             LZT02430
      RETURN                                                             LZT02440
C                                                                        LZT02450
 1000 FORMAT (/,69X,10HABSORPTION,/,23X,11H WAVELENGTH,4X,               LZT02460
     1        12HH2O PRESSURE,4X,11HTEMPERATURE,4X,                     LZT02470
     2        11HCOEFFICIENT,6X,4HLINE,9X,10HPATHLENGTH,4X,             LZT02480
     3        12HTRANSMISSION,/,24X,9H(MICRONS),8X,6H(TORR)             LZT02490
     4        ,11X,5H(ABS),10X,6H(KM-1),24X,4H(KM),/)                   LZT02500
 1100 FORMAT (4(A2),3F10.3)                                              LZT02510
 1200 FORMAT (1H ,22X,F09.3,F15.3,F16.2,E16.3,7X,4(A2),5X,               LZT02520
     1        E10.4,5X,E10.4)                                           LZT02530
 1300 FORMAT (39H *** WARNING VALUE OF T OR PH2O OUT OF ,                LZT02540
     1        10HRANGE *** /28H     T RANGE = 260 TO 320 K   ,          LZT02550
     2        25HPH2O RANGE = 0 TO 35 TORR)                             LZT02560
 1400 FORMAT (40H *** ERROR WAVELENGTH OUT OF ACCEPTABLE                 LZT02570
     1        7HRANGE: ,26H .8 TO 11.0 MICRONS ***    ,/,               LZT02580
     2        37H CONTROL RETURNED TO MAIN FROM LZTRAN)                 LZT02590
      END
```

```
      SUBROUTINE LZIDNM(WAVEL,LNAME1,LNAME3,LNAME2,LNAME4,LID)          LZI00010
C*************************************************                      LZI00020
C                                                                       LZI00030
C     THIS SUBROUTINE CONVERTS THE WAVELENGTH IN MICRONS OR THE LASER   LZI00040
C   LINE NAME, IF WAVEL = 0, TO AN INTEGER LASER ID NUMBER WHICH IS     LZI00050
C   USED BY LZTRAN.                                                     LZI00060
C      ++ WAVEL TO LID, WHEN WAVEL NOT = 0  ++                          LZI00070
C   THE SUBROUTINE DOES A BINARY SEARCH OF THE ARRAY AWAVEL TO FIND     LZI00080
C   A MATCH, WAVEL = AWAVEL(K).  WHEN A MATCH IS FOUND THE LASER ID     LZI00090
C   IS LOADED FROM THE ID ARRAY, LID = AID(K).  THE LASER LINE NAME     LZI00100
C   IS ALSO LOADED INTO TWO VARIABLES, LNAME1 = INAME1(K) LNAME2 =      LZI00110
C   INAME2(K).  IF AN EXACT MATCH IS NOT FOUND THE CLOSEST STANDARD     LZI00120
C   TO THE PARAMETER WAVEL IS USED.  A WARNING IS PRINTED AND K IS      LZI00130
C   SET TO THE PROPER VALUE SO IT CAN BE USED TO INDEX THE ID AND       LZI00140
C   LINE ARRAYS.                                                        LZI00150
C      ++ LASER LINE TO LID, WHEN WAVEL = 0  ++                         LZI00160
C   WHEN THE WAVELENGTH PARAMETER IS ZERO THE CONVERSION IS DONE        LZI00170
C   FROM LASER LINE NAME TO LID.  A SEQUENTIAL SEARCH OF THE LINE       LZI00180
C   NAME ARRAY IS PERFORMED.  WHEN A MATCH IS FOUND K IS SET AND        LZI00190
C   WAVEL AND LID ARE LOADED FROM THE APPROPRIATE ARRAYS.  WHEN         LZI00200
C   NO MATCH IS FOUND AN ERROR MESSAGE IS PRINTED AND LID IS SET        LZI00210
C   TO ZERO.  LID IS USED TO NOTIFY LZTRAN THAT AN ERROR HAS            LZI00220
C   OCCURED AND NO CALCULATIONS SHOULD BE PERFORMED.                    LZI00230
C                                                                       LZI00240
C      ++   PARAMETERS ++                                               LZI00250
C                                                                       LZI00260
C     WAVEL     LASER WAVELENGTH (MICRONS)                            LZI00270
C                                                                       LZI00280
C        *** INPUTS IF WAVEL = 0.0 ***                                  LZI00290
C     LNAME1   FIRST 4 CHARS OF LASER LINE                              LZI00300
C                                                                       LZI00310
C     LNAME2   SECOND 4 CHARS OF LASER LINE                             LZI00320
C                                                                       LZI00330
C      ++ RESULTS ++                                                    LZI00340
C                                                                       LZI00350
C     LID       LASER LINE IDENTIFIER                                   LZI00360
C                                                                       LZI00370
C     LNAME1   FIRST 4 CHARS OF LASER LINE                              LZI00380
C                                                                       LZI00390
C     LNAME2   SECOND 4 CHARS OF LASER LINE                             LZI00400
C                                                                       LZI00410
C*************************************************                      LZI00420
      INTEGER AID(53)                                                   LZI00430
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTULZI00440
      DIMENSION AWAVEL(53),INAME1(106),INAME2(53)                       LZI00450
      DATA AWAVEL/.85,.875,.9,.                                         LZI00460
     1      925,.95,1.06,3.521,3.55,3.581,3.612,3.636,3.645,           LZI00470
     2      3.666,3.679,3.698,3.715,3.731,3.752,3.765,3.8,             LZI00480
     3      3.82,3.837,3.854,3.875,3.89,3.915,3.927,3.956,             LZI00490
     4      3.965,3.999,4.005,4.046,4.089,4.908,4.918,4.928,           LZI00500
     5      4.938,4.948,4.972,4.982,4.992,5.002,5.012,5.022,           LZI00510
     6      5.032,5.043,5.047,5.054,5.057,5.067,5.078,5.088,           LZI00520
     7      10.591/                                                    LZI00530
      DATA AID/301,302,303,304,305,1,127,126,125,124,123,               LZI00540
     1      122,121,120,119,118,117,116,115,114,113,112,111,           LZI00550
     2      110,109,108,107,106,105,104,103,102,101,219,218,           LZI00560
     3      217,216,215,214,213,212,211,210,209,208,207,205,           LZI00570
     4      206,204,203,202,201,2/                                     LZI00580
      DATA INAME1/2HGA,2H.8,2HGA,2H.8,2HGA,2H.9,2HGA,2H.9,2HGA,2H.9,
     1 2HRU,2HBY,2HP1,2H(3,2HP1,2H(4,2HP1,2H(5,2HP1,2H(6,2HP2,2H(3,
     2 2HP1,2H(7,2HP2,2H(4,2HP1,2H(8,2HP2,2H(5,2HP1,2H(9,2HP2,2H(6,
     3 2HP1,2H(1,2HP2,2H(7,2HP2,2H(8,2HP3,2H(5,2HP2,2H(9,2HP3,2H(6,
     4 2HP2,2H(1,2HP3,2H(7,2HP2,2H(1,2HP3,2H(8,2HP2,2H(1,2HP3,2H(9,
     5 2HP2,2H(1,2HP3,2H(1,2HP3,2H(1,2HP3,2H(1,2HP4,2H(7,2HP4,2H(8,
     6 2HP4,2H(9,2HP4,2H(1,2HP4,2H(1,2HP5,2H(7,2HP5,2H(8,2HP5,2H(9,
     7 2HP5,2H(1,2HP5,2H(1,2HP5,2H(1,2HP5,2H(1,2HP5,2H(1,2HP6,2H(8,
     8 2HP5,2H(1,2HP6,2H(9,2HP6,2H(1,2HP6,2H(1,2HP6,2H(1,2HP(,2H20/
      DATA INAME2/1H5,2H75,1H ,2H25,1H5,1H ,11*1H),2H0),5*              LZI00660
     1      1H),2H0),1H),2H1),1H),2H2),1H),2H3),2H0),2H1),             LZI00670
     2      2H2),3*1H),2H0),2H1),3*1H),2H0),2H1),2H2),2H3),            LZI00680
```

```
     3     2H4),1H),2H5),1H),2H0),2H1),2H2),1H)/                          LZI00690
      DATA IFIRST,ILAST/1,53/                                              LZI00700
      I=IFIRST                                                             LZI00710
      J=ILAST                                                              LZI00720
      TWAVEL=WAVEL                                                         LZI00730
C  CHECK FOR WAVELENGTH OR LASER LINE PASSED AS INPUT.  IF                 LZI00740
C  WAVEL = 0 THEN DO A SEQUENTIAL SEARCH ON LINE NAME,                     LZI00750
C  LNAME1,LNAME2.                                                          LZI00760
      IF (WAVEL.NE.0.0) GO TO 200                                          LZI00770
C  INPUT = LASER LINE NAME                                                 LZI00780
C  SEQUENTIAL SEARCH LOOP                                                  LZI00790
      DO 100 K=IFIRST,ILAST                                                LZI00800
      KK=2*K-1
      IF((LNAME1.EQ.INAME1(KK)).AND.(LNAME3.EQ.INAME1(KK+1))) GO TO 10
      GO TO 100
   10 IF (LNAME2.EQ.INAME2(K)) GO TO 600                                   LZI00820
  100 CONTINUE                                                             LZI00830
C  ERROR, NO MATCH ON LASER NAME                                           LZI00840
      LID=0                                                                LZI00850
C  PRINT ERROR MESSAGE                                                     LZI00860
      WRITE (IOOUT,900) LNAME1,LNAME3,LNAME2,LNAME4                        LZI00870
      GO TO 700                                                            LZI00880
C  INPUT = WAVELENGTH                                                      LZI00890
C  BINARY SEARCH LOOP                                                      LZI00900
  200 K=(I+J)/2                                                            LZI00910
      IF (WAVEL.LE.AWAVEL(K)) J=K-1                                        LZI00920
      IF (WAVEL.GE.AWAVEL(K)) I=K+1                                        LZI00930
      IF (I.LE.J) GO TO 200                                                LZI00940
C  DID WAVELENGTH MATCH A STANDARD IN AWAVEL(*) ?                          LZI00950
      IF (I-1.GT.J) GO TO 600                                              LZI00960
C  WAVELENGTH NOT EXACTLY EQUAL TO ONE OF THE STANDARDS IN ARRAY           LZI00970
C  AWAVEL.  CHANGE WAVELENGTH TO EQUAL THE STANDARD IT IS CLOSEST TO.      LZI00980
C  THEN PRINT WARNING OF CHANGE.                                           LZI00990
      IF (WAVEL.GT.AWAVEL(K).AND.K.EQ.ILAST) GO TO 500                     LZI01000
      IF (WAVEL.LT.AWAVEL(K).AND.K.EQ.IFIRST) GO TO 500                    LZI01010
      IF (WAVEL-AWAVEL(K)) 300,600,400                                     LZI01020
C  WAVEL LT AWAVEL(K)                                                      LZI01030
C  CHECK IF CLOSER TO AWAVEL(K) OR AWAVEL(K-1)                             LZI01040
  300 DELTA1=WAVEL-AWAVEL(K-1)                                             LZI01050
      DELTA2=AWAVEL(K)-WAVEL                                               LZI01060
      IF (DELTA1.LT.DELTA2) K=K-1                                          LZI01070
      GO TO 500                                                            LZI01080
C  WAVEL GT AWAVEL(K)                                                      LZI01090
C  CHECK IF WAVEL CLOSER TO AWAVEL(K) OR AWAVEL(K+1)                       LZI01100
  400 DELTA1=WAVEL-AWAVEL(K)                                               LZI01110
      DELTA2=AWAVEL(K+1)-WAVEL                                             LZI01120
      IF (DELTA2.LT.DELTA1) K=K+1                                          LZI01130
C  PRINT WARNING                                                           LZI01140
  500 WAVEL=AWAVEL(K)                                                      LZI01150
      WRITE (IOOUT,800) TWAVEL,WAVEL                                       LZI01160
C  LOAD LASER ID NUMBER                                                    LZI01170
  600 LID=AID(K)                                                           LZI01180
C  LOAD LINE NAME                                                          LZI01190
      KK=2*K-1
      LNAME1=INAME1(KK)                                                    LZI01200
      LNAME3=INAME1(KK+1)
      LNAME2=INAME2(K)                                                     LZI01210
C  LOAD WAVELENGTH                                                         LZI01220
      WAVEL=AWAVEL(K)                                                      LZI01230
  700 RETURN                                                               LZI01240
C                                                                          LZI01250
  800 FORMAT (29H *** WARNING INPUT WAVELENGTH,F7.3,                       LZI01260
     1        11H CHANGED TO,F7.3,18H NEAREST STANDARD ,                   LZI01270
     2        14HWAVELENGTH ***)                                           LZI01280
  900 FORMAT (24H  *** ERROR LASER LINE #,4(A2),6H# NOT ,                  LZI01290
     1        9HVALID ***,/,26H CONTROL RETURNED TO MAIN ,                 LZI01300
     2        12HFROM LZTRAN.)                                             LZI01310
      END                                                                  LZI01320
```

```
      SUBROUTINE DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)                       DRT00010
C                                                                       DRT00020
C*********************************************************************  DRT00030
C                                                                       DRT00040
C                                                                       DRT00050
C  PURPOSE                                                              DRT00060
C     DIRTRAN-2 EXPLOSION PRODUCED AND VEHICLE GENERATED DUST MODEL     DRT00070
C                                                                       DRT00080
C  INPUT,OUTPUT AND CALLING PROGRAM                                     DRT00090
C                                                                       DRT00100
C  INPUTS                                                               DRT00110
C                                                                       DRT00120
C     VALUES IN ARGUMENT LIST                                           DRT00130
C                                                                       DRT00140
C  ICLMAT  INTEGER VALUE USED TO INDICATE HOW METEORLOGICAL DATA IS TO   DRT00150
C       BE MADE AVAILABLE  IF ICLMAT IS                                 DRT00160
C          0 - MET1 IDENTIFIER WITH THE APPROPRIATE PARAMETERS ARE TO    DRT00170
C              BE READ IN                                               
C          1 - NECESSARY METEORLOGICAL DATA IS PASSED IN COMMON/CLYMAT/  DRT00180
C              AND MET1 IS NOT TO BE READ IN                            DRT00190
C                                                                       DRT00200
C     WAVE1       WAVELENGTH IN MICROMETERS.  USED TO DETERMINE NWL,    DRT00210
C                 WHERE NWL IS AN INTEGER INDEX FOR WAVELENGTH DETERMINEDDRT00220
C                 WITHIN THE CODE                                       DRT00230
C                     NWL          WAVE1                                DRT00240
C                      1   FOR   0.4 - 0.7 MICROMETER  (VISIBLE)        DRT00250
C                      2   FOR   0.8 - 1.1 MICROMETER                   DRT00260
C                      3   FOR   3.5 - 4.0 MICROMETER                   DRT00270
C                      4   FOR   8.5 - 12.0 MICROMETER                  DRT00280
C                      5   FOR   2100 - 3200 MICROMETER                 DRT00290
C                                                                       DRT00300
C  INPUTS TO BE READ                                                    DRT00310
C                                                                       DRT00320
C     EACH INPUT RECORD BEGINS WITH A 4 LETTER IDENTIFIER IN COLUMNS    DRT00330
C  1-4 FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, UP TO 9, 8 COLUMNS  DRT00340
C  PER FIELD BEGINNING IN COLUMN 9.                                     DRT00350
C     THE INPUT FILE MAY CONTAIN SEVERAL SEQUENCES OF THE FOLLOWING     DRT00360
C  RECORDS.  EACH SEQUENCE SEPERATED BY A GO CARD.  ONCE THE INITIAL     DRT00370
C  SEQUENCE HAS BEEN READ IN AND THE MINIMUM REQUIREMENTS FOR EXECUTION DRT00380
C  OF THE DESIRED OPTION HAS BEEN SATISFIED,ANY FOLLOWING SEQUENCE MAY   DRT00390
C  CONTAIN A SUBSET OF THE INITIAL RECORDS REDEFINING INPUT VARIABLES    DRT00400
C  AS DESIRED OR MAY CONTAIN A COMPLETELY NEW SET OF RECORDS.            DRT00410
C                                                                       DRT00420
C  ** EACH SET OF INPUTS MUST END WITH A DONE CARD**                    DRT00430
C                                                                       DRT00440
C RECORD 1                                                              DRT00450
C  MET1                                                                 DRT00460
C                                                                       DRT00470
C     NATMOS      INTEGER WITH VALUES 1 TO 6 CORRESPONDING TO PASQUILL  DRT00480
C                 CATEGORIES A TO F.                                    DRT00490
C                                                                       DRT00500
C     ZTMP        THE HEIGHT AT WHICH A TEMPERATURE MEASUREMENT IS      DRT00510
C                 AVAILABLE.   VALID RANGE  0.5 - 100.0 M.              DRT00520
C                                                                       DRT00530
C     TMPMES      THE TEMPERATURE MEASURED IN DEGREES KELVIN TAKEN AT   DRT00540
C                 HEIGHT ZTMP.   VALID RANGE 270.0 - 315.0.             DRT00550
C                                                                       DRT00560
C     ZWND       THE HEIGHT AT WHICH A WIND SPEED MEASUREMENT IS        DRT00570
C                AVAILABLE.  VALID RANGE  0.5 - 100.0 M.                DRT00580
C                                                                       DRT00590
C     WNDMES      THE WIND SPEED IN METERS/SECOND MEASURED AT ZWND      DRT00600
C                 VALID RANGE  .1 - 20.0 M/S                            DRT00610
C                                                                       DRT00620
C     THWND       THE ANGLE THAT THE WIND VELOCITY VECTOR MAKES         DRT00630
C                 WITH THE USER'S POSITIVE X AXIS MEASURED IN DEGREES   DRT00640
C                 COUNTERCLOCKWISE. WHERE THE USERS POSITIVE X-AXIS     DRT00650
C                 POINTS EAST. THUS THWND IS THE ANGLE THAT THE WIND    DRT00660
C                 VELOCITY VECTOR MAKES WITH THE EAST.                  DRT00670
C                 VALID RANGE:  -360.0 - 360.0 DEGREES.                 DRT00680
C                 NOTE: THWND IS NOT NEEDED FOR OPTION 3                DRT00690
```

```
C                                                                       DRT00700
C RECORD 2                                                              DRT00710
C  MET2                                                                 DRT00720
C                                                                       DRT00730
C     ID          A FLAG TO INDICATE WHETHER THE INVERSION LAYER HEIGHT  DRT00740
C                 IS GROWING OR NOT.  IF ID IS                          DRT00750
C                   0.  THE INVERSION LAYER HEIGHT IS RELATIVELY CONSTANTDRT00760
C                   1.  THE INVERSION LAYER HEIGHT IS GROWING           DRT00770
C                                                                       DRT00780
C     PHI         THE LATITUDE OF THE DETONATION SITE.                  DRT00790
C                 VALID RANGE 1.0 - 90. DEGREES.  THAT IS THE NORTHERN  DRT00800
C                 HEMISPHERE.                                           DRT00810
C                                                                       DRT00820
C RECORD 3                                                              DRT00830
C  SOIL                                                                 DRT00840
C                                                                       DRT00850
C     NSOIL       INTEGER INDEX OF SOIL TYPE.  NSOIL IS                 DRT00860
C                   1.  FOR SOIL-1, (DATA GRAF-II) EXPLOSIONS ONLY.     DRT00870
C                   2.  FOR SOIL-2, (DATA DIRT-I) EXPLOSIONS ONLY.      DRT00880
C                   3.  FOR SOIL-3, (DATA SMOKEWEEK-II) VEHICLES ONLY.  DRT00890
C                                                                       DRT00900
C     DSOD        DEPTH OF SOD IN METERS                                DRT00910
C                 VALID RANGE: 0.0 - 1.0 M.                             DRT00920
C                 NOTE: FOR VEHICLE MODEL IF DSOD>0.0 NO DUST IS        DRT00930
C                 GENERATED                                             DRT00940
C                                                                       DRT00950
C     SILT        SILT CONTENT OF SOIL (PARTICLE DIAMETERS < 75 MICRONS) DRT00960
C                 I.E.   SILT=.15 INDICATES A SILT CONTENT OF 15%       DRT00970
C                 NOTE: THIS INPUT NEEDED ONLY FOR VEHICLE MODEL (IOPT=5)DRT00980
C                                                                       DRT00990
C RECORD 4                                                              DRT01000
C  CHAR                                                                 DRT01010
C                                                                       DRT01020
C     NCHRG       CHARGE TYPE INDEX WITH FOLLOWING VALUES               DRT01030
C                     1.    SURFACE - LIVE FIRE OR 30 DEGREE TILTED     DRT01040
C                         STATIC, TIP ON GROUND                         DRT01050
C                     2.    BARE CHARGE ON SURFACE                      DRT01060
C                     3.    30 DEGREE TILTED TIP AT 0.3 METER DEPTH     DRT01070
C                     4.    30 DEGREE TILTED TIP AT 0.6 METER DEPTH     DRT01080
C                     5.    HORIZONTAL PROJECTILE ON SURFACE            DRT01090
C                  DEFAULT VALUE IS 1 IF NCHRG IS NOT BETWEEN 1 AND 5.  DRT01100
C                                                                       DRT01110
C     CHWT        THE WEIGHT OF THE CHARGE IN KG-TNT.                   DRT01120
C                 VALID RANGE:  0.1 - 100.0 KG-TNT.                     DRT01130
C                                                                       DRT01140
C     DETDEP      THE DEPTH OF DETONATION IN METERS.                    DRT01150
C                 VALID RANGE:  0.0 - 2.0 M.                            DRT01160
C                                                                       DRT01170
C RECORD 5                                                              DRT01180
C  EXPL                                                                 DRT01190
C                                                                       DRT01200
C     NARY        TYPE OF CHARGE DISTRIBUTION (USED FOR PROPER INPUT AND DRT01210
C                 OUTPUT FORMATS) IF THE VALUE OF NARY AND IOPT OF THE GODRT01220
C                 CARD ARE NOT COMPATIBLE CATASTROPHE COULD RESULT!     DRT01230
C                 NOTE: WHEN NARY IS                                    DRT01240
C                       1.    IOPT MUST ALSO BE 1.                      DRT01250
C                       2.    IOPT MUST ALSO BE 2.                      DRT01260
C                       3.    IOPT MUST BE    4.                        DRT01270
C               1.-SIMULTANEOUS BURST,UNIFORMLY DISTRIBUTED CHARGES     DRT01280
C                 IN A PARALLELOGRAM.                                   DRT01290
C                 (SPECIAL CASES ARE ,SINGLE CHARGE ,RECTANGLE AND      DRT01300
C                  ZIG ZAG PATTERN)                                     DRT01310
C                                                                       DRT01320
C               2.-SIMULTANEOUS BURST,RANDOMLY DISTRIBUTED CHARGES.     DRT01330
C                                                                       DRT01340
C               3.-SEQUENTIAL IN TIME AND RANDOM IN SPACE DISTRIBUTION OFDRT01350
C                 CHARGES.                                              DRT01360
C                                                                       DRT01370
C                 NOTE:                                                 DRT01380
C                   WHEN NARY=2. EACH CHARGE LOCATION MUST BE SPECIFIED DRT01390
```

```
C                  WHEN NARY=3, EACH CHARGE LOCATION AND DETONATION TIMEDRT01400
C                  MUST BE SPECIFIED                                    DRT01410
C                  THE CHARGE LOCATIONS ARE INPUT DIRECTLY FOLLOWING    DRT01420
C                  THIS INPUT RECORD WITH ONE LOCATION (AND DETONATION  DRT01430
C                  TIME IF APPROPRIATE) PER RECORD.                     DRT01440
C                                                                       DRT01450
C     NCHS         SINGLY DIMENSIONED ARRAY SPECIFING NUMBER OF CHARGES DRT01460
C                  WITH THE MAXIMUM TOTAL OF CHARGES 200.               DRT01470
C                  WHEN NARY=1, NCHS(1) IS THE NUMBER OF CHARGES IN THE DRT01480
C                  DIRECTION OF SIDE1 AND NCHS(2) IS THE NUMBER OF      DRT01490
C                  CHARGES IN THE DIRECTION OF SIDE2. FOR A SINGLE CHARGEDRT01500
C                  SET NCHS(1)=NCHS(2)=1..                              DRT01510
C                  WHEN NARY=2, OR 3, NCHS(1) IS THE TOTAL NUMBER OF    DRT01520
C                  CHARGES AND SET NCHS(2)=1..                          DRT01530
C                                                                       DRT01540
C     SRCBAS       SINGLY DIMENSIONED ARRAY CONTAINING THE COORDINATES OFDRT01550
C                  A CORNER POINT OF THE BOUNDING PARALLELOGAM WHEN     DRT01560
C                  NARY=1.  AND IS ALSO USED AS THE REFERENCE CHARGE    DRT01570
C                  BY THE OBSERVER. THAT IS, SRCBAS IS THE ORIGIN OF THEDRT01580
C                  OBSERVERS COORDINATE SYSTEM.  WHEN NARY=2, OR 3,     DRT01590
C                  SRCBAS SHOULD NOT APPEAR ON THE INPUT FILE AS COOR(I,1)DRT01600
C                  IS USED AS THE REFERENCE CHARGE, WHERE COOR(I,1) IS  DRT01610
C                  THE FIRST CHARGE LOCATION SPECIFIED.                 DRT01620
C                  VALID RANGE -10000.0 - 10000.0                       DRT01630
C                                                                       DRT01640
C     SIDE1        SINGLY DIMENSIONED ARRAY NECESSARY ONLY WHEN NARY=1.,DRT01650
C                  SPECIFING ONE SIDE OF THE BOUNDING PARALLELOGRAM     DRT01660
C                  FROM THE POINT SRCBAS(I). THAT IS, SIDE1 IS A VECTOR DRT01670
C                  TO THE NEXT CHARGE ALONG ONE SIDE OF THE PARALLELOGRAM.DRT01680
C                  WHEN NARY=2, OR 3. THIS VARIABLE SHOULD NOT APPEAR ONDRT01690
C                  THE INPUT FILE.                                      DRT01700
C                                                                       DRT01710
C     SIDE2        SINGLY DIMENSIONED ARRAY NECESSARY ONLY WHEN NARY=1.  DRT01720
C                  SPECIFING A SECOND SIDE OF THE BOUNDING PARALLELOGRAM DRT01730
C                  FROM THE POINT SRCBAS(I).THAT IS SIDE2 IS A VECTOR TO DRT01740
C                  TO THE NEXT CHARGE ALONG THE SECOND SIDE OF THE      DRT01750
C                  PARALLELOGRAM.                                       DRT01760
C                  WHEN NARY=2, OR 3. THIS VARIABLE SHOULD NOT APPEAR ON DRT01770
C                  THE INPUT FILE.                                      DRT01780
C                                                                       DRT01790
C    ** THE FOLLOWING RECORD MUST APPEAR THE APPROPRIATE NUMBER OF TIMES DRT01800
C    ** IMEDIATELY FOLLOWING THE ABOVE RECORD IF NARY IS 2, OR 3. ON THE DRT01810
C    ** ABOVE RECORD.  THAT IS IT MUST APPEAR THE SAME NUMBER OF TIMES  DRT01820
C    ** AS THER ARE CHARGES AS SPECIFIED ON RECORD EXPL. (I.E. IF NARY=2.DRT01830
C    ** AND NCHS=5. THE THIS RECORD MUST APPEAR 5 TIMES.                DRT01840
C                                                                       DRT01850
C RECORD 6                                                              DRT01860
C  LOCA                                                                 DRT01870
C                                                                       DRT01880
C     COOR         DOUBLY DIMENSIONED ARRAY CONTAINING THE DETONATION   DRT01890
C                  COORDINATES FOR EACH CHARGE WHEN NARY=2, OR 3.. WHEN DRT01900
C                  NARY=1. THIS VARIABLE NEED NOT BE SPECIFIED AS THE   DRT01910
C                  CHARGE LOCATIONS ARE CALCULATED IN THE CODE FROM     DRT01920
C                  NCHS,SIDE1,SIDE2.                                    DRT01930
C                                                                       DRT01940
C     TSTAG        SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF      DRT01950
C                  DETONATION OF EACH CHARGE. THIS IS ONLY SPECIFIED WHEN DRT01960
C                  NARY=3..                                             DRT01970
C                                                                       DRT01980
C RECORD 7                                                              DRT01990
C  VEHC                                                                 DRT02000
C                                                                       DRT02010
C     V0           DOUBLY DIMENSIONED ARRAY CONTAINING THE INITIAL      DRT02020
C                  POSITION OF THE VEHICLE. V0(1)=X-COORDINATE          DRT02030
C                  V0(2)=Y-COORDINATE.                                  DRT02040
C                  VALID RANGE: -10000.0 - 10000.0                      DRT02050
C                                                                       DRT02060
C     VEHDIR       VEHICLE DIRECTION  THE ANGLE THAT THE VEHICLE VELOCITY DRT02070
C                  VECTOR MAKES WITH THE USER'S POSITIVE X-AXIS MEASURED DRT02080
C                  IN DEGREES COUNTERCLOCKWISE. WHERE THE USER'S POSITIVE DRT02090
```

```
C                X-AXIS POINTS EAST. THUS VEHDIR IS THE ANGLE THE        DRT02100
C                VELOCITY VECTOR MAKES WITH THE EAST.                    DRT02110
C                VALID RANGE:  -360.0 - 360.0                            DRT02120
C                                                                        DRT02130
C     VEHSPD     VEHICLE SPEED IN  M/S                                   DRT02140
C                                                                        DRT02150
C     VEHWID     VEHICLE WIDTH IN METERS                                 DRT02160
C                                                                        DRT02170
C     VEHWHT     VEHICLE WEIGHT IN KGS.                                  DRT02180
C                                                                        DRT02190
C     VEHTYP     TRACTION MECHANISM                                      DRT02200
C                                                                        DRT02210
C               =0.  VEHICLE HAS TIRES                                   DRT02220
C               =1.  VEHICLE IS TRACKED                                  DRT02230
C                                                                        DRT02240
C RECORD 8                                                               DRT02250
C  TRNC                                                                  DRT02260
C                                                                        DRT02270
C     TRNCOR     A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE         DRT02280
C                COORDINATES OF THE TRANSMITTER.  THE COORDINATE SYSTEM  DRT02290
C                MUST BE IN METERS.  THE THIRD COORDINATE IS RESTRICTED  DRT02300
C                TO BE BETWEEN .5 AND 10000.0 METERS (HEIGHT).           DRT02310
C                VALID RANGE OF THE FIRST TWO COORDINATES:               DRT02320
C                -10000.0  -  10000.0 M.                                 DRT02330
C
C  ** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE
C  ** ARRAY TRNCOR NEED NOT BE SPECIFIED.
C                                                                        DRT02340
C     TRNMIN     VALUE SUCH THAT A TRANSMITTANCE BELOW THIS VALUE CAN    DRT02350
C                BE CONSIDERED ZERO.  DEFAULT IS 1.E-05                  DRT02360
C                VALID RANGE: 1.0 - 1.E-05                               DRT02370
C                                                                        DRT02380
C RECORD 9                                                               DRT02390
C  RECC                                                                  DRT02400
C                                                                        DRT02410
C     RECCOR     A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE         DRT02420
C                COORDINATES OF THE RECEIVER.  (METERS)                  DRT02430
C                THE THIRD COORDINATE IS RESTRICTED TO BE BETWEEN        DRT02440
C                .5 AND 10000.0 METERS.  VALID RANGE OF THE FIRST TWO    DRT02450
C                COORDINATES IS:  -10000.0  -  10000.0 M.                DRT02460
C
C  ** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE
C  ** ARRAY RECCOR NEED NOT BE  SPECIFIED.
C                                                                        DRT02470
C RECORD 10                                                              DRT02480
C  OBSC                                                                  DRT02490
C                                                                        DRT02500
C     OBSCOR     A SINGLY DIMENSIONED ARRAY CONTAINING THE X AND Y       DRT02510
C                COORDINATES, RESP., OF THE OBSERVER. (METERS)           DRT02520
C                VALID RANGE:   -10000.0  -  10000.0                     DRT02530
C                                                                        DRT02540
C     SPCHT      A SPECIFIED HEIGHT IN METERS AT WHICH THE WIDTH OF      DRT02550
C                THE CLOUD AS VIEWED FROM POSITION OBSCOR IS DESIRED.    DRT02560
C                MUST BE BETWEEN 1. AND 5. METERS.                       DRT02570
C
C  ** IF THE COORDINATES ARE PASSED THROUGH THE GEOMET OPTION, THEN THE
C  ** ARRAY OBSCOR AND VARIABLE SPCHT NEED NOT BE SPECIFIED.
C                                                                        DRT02580
C RECORD 11                                                              DRT02590
C  TIMS                                                                  DRT02600
C                                                                        DRT02610
C     TSTART     TIME AFTER DETONATION TO START TRANSMITTANCE AND/OR     DRT02620
C                CLOUD DIMENSION CALCULAIONS                             DRT02630
C                VALID RANGE: .5 - 1000.0 SEC.                           DRT02640
C                                                                        DRT02650
C     TEND       TIME AFTER DETONATION TO TERMINATE TRANSMITTANCE        DRT02660
C                AND/OR CLOUD DIMENSIONS.                                DRT02670
C                VALID RANGE: .5-1000.0 SEC. (TEND MUST BE .GE. TSTART)  DRT02680
C                                                                        DRT02690
C     TINC       TIME INCREMENT BETWEEN CALCULATIONS                     DRT02700
```

```
C
C RECORD 12                                                                    DRT02710
C  GO      ** THIS CARD INDICATES THAT THIS SEQUENCE OF INPUTS ARE             DRT02720
C             COMPLETE AND CALCULATIONS ARE TO BEGIN.                          DRT02730
C                                                                              DRT02740
C                                                                              DRT02750
C     IOPT    OPTION TO BE USED                                                DRT02760
C                                                                              DRT02770
C             1. SIMULTANEOUS BURST ,UNIFORMLY DISTRIBUTED CHARGES IN A        DRT02780
C                PARALLELOGRAM                                                 DRT02790
C                                                                              DRT02800
C             2. SIMUTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES               DRT02810
C                                                                              DRT02820
C             3. THE CODE IS TO PRECOMPUTE A SINGLE CLOUD AND STORE ON         DRT02830
C                AN EXTERNAL FILE FOR USE LATER                                DRT02840
C                                                                              DRT02850
C             4. THE CODE IS TO USE A CLOUD THAT HAS BEEN PRECOMPUTED          DRT02860
C                (NO CLOUD DIMENSIONS ARE COMPUTED FOR THIS OPTION)            DRT02870
C                                                                              DRT02880
C             5. VEHICLE DUST MODEL                                            DRT02890
C                                                                              DRT02900
C     IFILE   FORTRAN LOGICAL UNIT TO WHICH THE CODE IS TO WRITE FOR           DRT02910
C             OPTION 3 OTHERWISE IT NEED NOT BE SPECIFIED                      DRT02920
C                                                                              DRT02930
C RECORD 13                                                                    DRT02940
C  DONE    ** THIS RECORD INDICATES THAT THE USER HAS COMPLETED HIS            DRT02950
C             DESIRED SEQUENCE OF INPUTS AND ALL CALCULATIONS ARE              DRT02960
C             TERMINATED                                                       DRT02970
C                                                                              DRT02980
C                                                                              DRT02990
C  OUTPUTS                                                                     DRT03000
C                                                                              DRT03010
C     ZINV        THE ESTIMATED INVERSION HEIGHT.                              DRT03020
C                                                                              DRT03030
C     TRNLOS      THE TRANSMITTANCE ALONG THE LINE OF SIGHT BETWEEN            DRT03040
C                 THE TRANSMITTER AND THE RECEIVER.                            DRT03050
C                                                                              DRT03060
C     IERR        INTEGER ERROR CODE WHICH EQUALS 1 IF A FATAL ERROR           DRT03070
C                 OCCURS AND 0 OTHERWISE                                       DRT03080
C                                                                              DRT03090
C     NERR         INTEGER ERROR CODE WITH THE VALUES                          DRT03100
C                    0     NO ERRORS                                           DRT03110
C                    4     NO TRANSMITTER AND RECEIVER OR OBSERVER             DRT03120
C                          COORDINATES WERE SPECIFIED SO NO RESULTS WERE       DRT03130
C                          CALCULATED.                                         DRT03140
C                    7     THE CALCULATION OF ATMOSPHERIC PARAMETERS DID       DRT03150
C                          NOT CONVERGE.                                       DRT03160
C                                                                              DRT03170
C                                                                              DRT03180
C     CNTRD       A SINGLY DIMENSIONED ARRAY CONTAINING THE HORIZONTAL         DRT03190
C                 COORDINATE AND THE VERTICAL COORDINATE OF THE                DRT03200
C                 CENTROID OF THE CLOUD.                                       DRT03210
C                                                                              DRT03220
C     HEIGHT      THE HEIGHT OF THE CLOUD IN METERS.                           DRT03230
C                                                                              DRT03240
C     CENWTH      THE WIDTH OF THE CLOUD IN METERS AT THE CENTROID             DRT03250
C                 HEIGHT                                                       DRT03260
C                                                                              DRT03270
C     SPCWTH      THE WIDTH OF THE CLOUD IN METERS AT THE SPECIFIED            DRT03280
C                 HEIGHT                                                       DRT03290
C                                                                              DRT03300
C     NCPTS       THE NUMBER OF POINTS DETERMINED ON THE EDGE OF THE           DRT03310
C                 CLOUD.                                                       DRT03320
C                                                                              DRT03330
C     CPTS        A DOUBLY DIMENSIONED ARRAY CONTAINING THE COORDINATES        DRT03340
C                 OF POINTS ON THE EDGE OF THE CLOUD.  CPTS(1,J)               DRT03350
C                 IS THE HORIZONTAL COORDINATE OF THE J-TH POINT               DRT03360
C                 AND CPTS(2,J)IS THE VERTICAL COORDINATE OF THE               DRT03370
C                 J-TH POINT.  THE FIRST INDEX MUST BE DIMENSIONED             DRT03380
C                 TO 2.                                                        DRT03390
C                                                                              DRT03400
```

```
C      SUBROUTINES CALLED                                                      DRT03410
C                                                                              DRT03420
C      DUSTCL   CONTROLING ROUTINE FOR THE CALCULATION OF CLOUD DIMENSIONSDRT03430
C               AND TRANSMITTANCES THROUGH DUST CLOUDS FOR OPTIONS 1 AND 2DRT03440
C               GIVEN METEORLOGICAL DATA, SOIL AND EXPLOSIVE             DRT03450
C               CHARACTERISTICS,AND WAVELENGTH.                          DRT03460
C                                                                        DRT03470
C      COMPCL   CONTROLING ROUTINE  FOR PRECOMPUTING A SINGLE EXPLOSION  DRT03480
C               (OPTION 3) GIVEN METEOROLOGICAL DATA, SOIL AND EXPLOSION DRT03490
C               CHARACTERISTICS.  ALSO USES THIS PRECOMPUTED CLOUD AT SOMEDRT03500
C               LATER RUNNING OF THE CODE (OPTION 4) TO ESTIMATE A       DRT03510
C               TRANSMITTANCE GIVEN TRANSMITTER AND RECEIVER COODINATES. DRT03520
C                                                                        DRT03530
C      VEHCL    CONTROLING ROUTINE FOR THE CALCULATION OF A TRANSMITTANCE DRT03540
C               THROUGH A VEHICLE GENERATED DUST CLOUD (OPTION 5) GIVEN  DRT03550
C               METEORLOGICAL DATA, SOIL CHARACTERISTICS, VEHICLE        DRT03560
C               CHARACTERISTICS, AND WAVELENGTH.                         DRT03570
C                                                                        DRT03580
C***********************************************************************DRT03590
      LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM,      CLMRED,DHDT,ONCE    DRT03600
      LOGICAL TEST,NEWVEH,NEWCOR                                         DRT03610
      LOGICAL M1,M2,SL,CH,EX,TC,RC,OC,TM,VH                              DRT03620
      INTEGER VEHTYP                                                     DRT03630
      REAL M,N
      DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2),TRNCOR(3)            DRT03640
      DIMENSION SRCBAS(2),SIDE1(2),SIDE2(2),NCHS(2)                      DRT03650
     1   ,RECCOR(3),CPTS(2,6),CNTRD(2),OBSCOR(2)                         DRT03660
      DIMENSION RDIN(10),RKEY(12),V0(2),PAS(6)                           DRT03670
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUDRT03680
      COMMON /CLYMAT/ TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,             DRT03690
     1 FOGPRB,WNDVEL,WINDIR,IPASCT                                       DRT03700
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,           DRT03710
     +           COOR(2,200),TSTAG(200),DMMY(401)                        DRT03720
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV                            DRT03730
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL                                DRT03740
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)                DRT03750
      COMMON /GEOMET/PTS(15),IGEOSW                                      DRT03760
      COMMON/OPTION/IOPT,IFILE                                           DRT03770
      DATA RKEY/4HMET1,4HMET2,4HSOIL,4HCHAR,4HEXPL,4HVEHC,4HTRNC,        DRT03780
     1          4HRECC,4HOBSC,4HTIMS,4HGO   ,4HDONE/                     DRT03790
      DATA PAS/4HA   ,4HB   ,4HC   ,4HD   ,4HE   ,4HF   /               DRT03800
      DATA M1,M2,SL,CH,EX,TC,RC,OC,TM,VH/.FALSE.,.FALSE.,.FALSE.,        DRT03810
     1.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE./          DRT03820
      DATA NEWATM,NEWSRC,NEWVEH,LOSTRN,EDGE,NEWTIM/.FALSE.,.FALSE.,      DRT03830
     1.FALSE.,.FALSE.,.FALSE.,.FALSE./                                   DRT03840
      DATA NEWCOR/.FALSE./                                               DRT03850
      DATA VISEXT,RTPI/.1,1.772454/                                      DRT03860
      IERR=0                                                             DRT03870
      CLMRED=.FALSE.                                                     DRT03880
      DHDT=.FALSE.                                                       DRT03890
      ONCE=.FALSE.                                                       DRT03900
      TEST=.FALSE.                                                       DRT03910
      WRITE(IOOUT,800)                                                   DRT03920
  800 FORMAT(1H0,36X,42HDIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION,      DRT03930
     1 15H    CALCULATION,//,36X,60H*** NOTE -- ALL UNITS ARE MKS UNLESSDRT03940
     2 OTHERWISE SPECIFIED ***,//)                                       DRT03950
      DO 5 K=1,200                                                       DRT03960
      TSTAG(K)=0.0                                                       DRT03970
    5 CONTINUE                                                           DRT03980
C                                                                        DRT03990
C  DETERMINE INTEGER INDEX FOR WAVELENGTH                                DRT04000
C                                                                        DRT04010
   10 IF(WAVE1.LT.0.4)GO TO 29                                           DRT04020
      IF(WAVE1.GT.0.7)GO TO 21                                           DRT04030
      NWL=1                                                              DRT04040
      GO TO 30                                                           DRT04050
   21 IF(WAVE1.LT.0.8)GO TO 29                                           DRT04060
      IF(WAVE1.GT.1.1)GO TO 22                                           DRT04070
      NWL=2                                                              DRT04080
      GO TO 30                                                           DRT04090
```

```
   22 IF(WAVE1.LT.3.5)GO TO 29                                          DRT04100
      IF(WAVE1.GT.4.0)GO TO 23                                          DRT04110
      NWL=3                                                             DRT04120
      GO TO 30                                                          DRT04130
   23 IF(WAVE1.LT.8.5)GO TO 29                                          DRT04140
      IF(WAVE1.GT.12.0)GO TO 24                                         DRT04150
      NWL=4                                                             DRT04160
      GO TO 30                                                          DRT04170
   24 IF(WAVE1.LT.2100.)GO TO 29                                        DRT04180
      IF(WAVE1.GT.3200.)GO TO 29                                        DRT04190
      NWL=5                                                             DRT04200
      GO TO 30                                                          DRT04210
   29 WRITE(IOOUT,802)                                                  DRT04220
  802 FORMAT(37X,38H*** DIRTRAN ERROR - WAVE1 OUT OF RANGE)             DRT04230
      IERR=1                                                            DRT04240
      GO TO 999                                                         DRT04250
   30 CONTINUE                                                          DRT04260
C                                                                       DRT04270
CC  READ DATA AND STORE APPROPRIATELY                                   DRT04280
C                                                                       DRT04290
      DO 300 II=1,15                                                    DRT04300
      IF(II.EQ.15)GO TO 900                                             DRT04310
      READ(IOIN,700)(RDIN(J),J=1,10)                                    DRT04320
  700 FORMAT(A4,4X,9F8.2)                                               DRT04330
      IF(RDIN(1).EQ.RKEY(1))GO TO 50                                    DRT04340
      IF(RDIN(1).EQ.RKEY(2))GO TO 70                                    DRT04350
      IF(RDIN(1).EQ.RKEY(3))GO TO 90                                    DRT04360
      IF(RDIN(1).EQ.RKEY(4))GO TO 110                                   DRT04370
      IF(RDIN(1).EQ.RKEY(5))GO TO 130                                   DRT04380
      IF(RDIN(1).EQ.RKEY(6))GO TO 150                                   DRT04390
      IF(RDIN(1).EQ.RKEY(7))GO TO 170                                   DRT04400
      IF(RDIN(1).EQ.RKEY(8))GO TO 190                                   DRT04410
      IF(RDIN(1).EQ.RKEY(9))GO TO 210                                   DRT04420
      IF(RDIN(1).EQ.RKEY(10))GO TO 230                                  DRT04430
      IF(RDIN(1).EQ.RKEY(11))GO TO 310                                  DRT04440
      IF(RDIN(1).EQ.RKEY(12))GO TO 999                                  DRT04450
      WRITE(IOOUT,804)                                                  DRT04460
  804 FORMAT(33X,52H***DIRTRAN-2 ERROR, INPUT DOES NOT CONFORM TO PROPERDRT04470
     1       14H CONVENTION***)                                         DRT04480
      WRITE(IOOUT,806)(RDIN(J),J=1,9)                                   DRT04490
  806 FORMAT(26X,A4,4X,9F8.2)                                           DRT04500
      GO TO 999                                                         DRT04510
C                                                                       DRT04520
CC  STORE AND PRINT OUT ATMOSPHERIC CONDITIONS                          DRT04530
C                                                                       DRT04540
   50 CONTINUE                                                          DRT04550
      M1=.TRUE.                                                         DRT04560
      NIO=1                                                             DRT04570
      IF(ICLMAT.EQ.1)GO TO 55                                           DRT04580
      NATMOS=IFIX(RDIN(2))                                              DRT04590
      ZTMP(1)=RDIN(3)                                                   DRT04600
      TMPMES(1)=RDIN(4)                                                 DRT04610
      ZWND(1)=RDIN(5)                                                   DRT04620
      WNDMES(1)=RDIN(6)                                                 DRT04630
      THWND=RDIN(7)                                                     DRT04640
      GO TO 60                                                          DRT04650
C                                                                       DRT04660
CC  IPSCAT    PASQUILL CATEGORY                                         DRT04670
CC  WNDVEL    WIND VELOCITY IN M/S MEASURED AT 2 M. ABOVE GROUND        DRT04680
CC  WINDIR    WIND DIRECTION IN DEGREES   CLOCKWISE FROM TRUE NORTH     DRT04690
CC  TEMP      TEMPERATURE IN DEGREES C MEASURED AT 2 M. ABOVE GROUND     DRT04700
C                                                                       DRT04710
   55 NATMOS=IPASCT                                                     DRT04720
      ZTMP(1)=2.                                                        DRT04730
      ZWND(1)=2.                                                        DRT04740
      WNDMES(1)=WNDVEL                                                  DRT04750
      TMPMES(1)=TEMP+273.0                                              DRT04760
      THWND=270.0-WINDIR                                                DRT04770
   60 CONTINUE                                                          DRT04780
      WRITE(IOOUT,808)                                                  DRT04790
```

```
 808  FORMAT(1X)                                                          DRT04800
      WRITE(IOOUT,810)PAS(NATMOS)                                         DRT04810
  810 FORMAT(50X,28HPASQUILL CATEGORY              ,A4)                   DRT04820
      WRITE(IOOUT,812)(ZTMP(I),TMPMES(I),ZWND(I),WNDMES(I),I=1,NIO)       DRT04830
  812 FORMAT(36X,4H HT ,F8.2,7H  TEMP ,F8.2,7H     HT,F8.2,7H  WIND ,    DRT04840
     1 F8.2)                                                              DRT04850
      WRITE(IOOUT,814)THWND                                               DRT04860
  814 FORMAT(51X,22H  WIND DIRECTION       ,F8.2)                         DRT04870
      GO TO 300                                                           DRT04880
   70 CONTINUE                                                            DRT04890
      M2=.TRUE.                                                           DRT04900
      ID=IFIX(RDIN(2))                                                    DRT04910
      IF(ID.NE.0)DHDT=.TRUE.                                              DRT04920
      PHI=RDIN(3)                                                         DRT04930
      WRITE(IOOUT,819)PHI                                                 DRT04940
      IF(DHDT)WRITE(IOOUT,816)                                            DRT04950
  816 FORMAT(47X,37HTHE INVERSION LAYER HEIGHT IS GROWING)                DRT04960
      IF(.NOT.DHDT)WRITE(IOOUT,818)                                       DRT04970
  818 FORMAT(47X,38HTHE INVERSION LAYER HEIGHT IS CONSTANT)               DRT04980
  819 FORMAT(/,52X,20HLATITUDE              ,F8.2)                        DRT04990
      GO TO 300                                                           DRT05000
C                                                                         DRT05010
CC   STORE AND WRITE SOIL CHARACTERISTICS                                 DRT05020
C                                                                         DRT05030
   90 CONTINUE                                                            DRT05040
      SL=.TRUE.                                                           DRT05050
      NSOIL=IFIX(RDIN(2))                                                 DRT05060
      DSOD=RDIN(3)                                                        DRT05070
      SILT=RDIN(4)                                                        DRT05080
      IF(NSOIL.EQ.1)WRITE(IOOUT,821)                                      DRT05090
  820 FORMAT(56X,15HSILT CONTENT    ,F5.2)                                DRT05100
  821 FORMAT(/,63X,6HSOIL-1)                                              DRT05110
      IF(NSOIL.EQ.2)WRITE(IOOUT,822)                                      DRT05120
  822 FORMAT(/,63X,6HSOIL-2)                                              DRT05130
      IF (NSOIL.EQ.3) WRITE (IOOUT,710)                                   DRT05140
710   FORMAT(/,23X,8H  SOIL-3)                                            DRT05150
      IF(SILT.GT.1.E-06)WRITE(IOOUT,820)SILT                              DRT05160
      WRITE(IOOUT,823)DSOD                                                DRT05170
  823 FORMAT(53X,21H  DEPTH OF SOD        ,F5.2)                          DRT05180
      IF (NSOIL.LT.1.OR.NSOIL.GT.2) NSOIL=2                               DRT05190
      GO TO 300                                                           DRT05200
C                                                                         DRT05210
CC   STORE AND WRITE EXPLOSIVE CHARGE CHARACTERISTICS                     DRT05220
C                                                                         DRT05230
  110 CONTINUE                                                            DRT05240
      CH=.TRUE.                                                           DRT05250
      NCHRG=IFIX(RDIN(2))                                                 DRT05260
      IF(NCHRG.LT.1.OR.NCHRG.GT.5)NCHRG=1                                 DRT05270
      CHWT=RDIN(3)                                                        DRT05280
      DETDEP=RDIN(4)                                                      DRT05290
      IF(NCHRG.EQ.1)WRITE(IOOUT,824)                                      DRT05300
  824 FORMAT(/,35X,47HSURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC,,    DRT05310
     1       14H TIP ON GROUND)                                           DRT05320
      IF(NCHRG.EQ.2)WRITE(IOOUT,825)                                      DRT05330
  825 FORMAT(/,55X,22HBARE CHARGE ON SURFACE)                             DRT05340
      IF(NCHRG.EQ.3)WRITE(IOOUT,826)                                      DRT05350
  826 FORMAT(/,46X,39H30 DEGREE TILTED TIP AT 0.3 METER DEPTH)            DRT05360
      IF(NCHRG.EQ.4)WRITE(IOOUT,827)                                      DRT05370
  827 FORMAT(/,46X,39H30 DEGREE TILTED TIP AT 0.6 METER DEPTH)            DRT05380
      IF(NCHRG.EQ.5)WRITE(IOOUT,828)                                      DRT05390
  828 FORMAT(/,50X,32HHORIZONTAL PROJECTILE ON SURFACE)                   DRT05400
      WRITE(IOOUT,829)CHWT                                                DRT05410
  829 FORMAT(45X,30HWEIGHT OF CHARGE              ,F8.2,4H KG.)           DRT05420
      WRITE(IOOUT,830)DETDEP                                              DRT05430
  830 FORMAT(47X,30HDETONATION DEPTH               ,F8.2)                 DRT05440
      GO TO 300                                                           DRT05450
C                                                                         DRT05460
CC   STORE AND WRITE OUT INFORMATION ABOUT THE DETONATION LOCATIONS       DRT05470
C                                                                         DRT05480
  130 CONTINUE                                                            DRT05490
```

```
      EX=.TRUE.                                                          DRT05500
      NARY=IFIX(RDIN(2))                                                 DRT05510
      NCHS(1)=IFIX(RDIN(3))                                              DRT05520
      NCHS(2)=IFIX(RDIN(4))                                              DRT05530
      SRCBAS(1)=RDIN(5)                                                  DRT05540
      SRCBAS(2)=RDIN(6)                                                  DRT05550
      SIDE1(1)=RDIN(7)                                                   DRT05560
      SIDE1(2)=RDIN(8)                                                   DRT05570
      SIDE2(1)=RDIN(9)                                                   DRT05580
      SIDE2(2)=RDIN(10)                                                  DRT05590
      IF(NARY.EQ.2)GO TO 133                                             DRT05600
      IF(NARY.EQ.3)GO TO 136                                             DRT05610
C                                                                        DRT05620
C  CHARGE DISTRIBUTION TYPE 1                                            DRT05630
C                                                                        DRT05640
      WRITE(IOOUT,831)                                                   DRT05650
  831 FORMAT(/,31X,42HSIMULTANEOUS BURST , UNIFORMLY DISTRIBUTED,        DRT05660
     +27H CHARGES IN A PARALLELOGRAM)                                    DRT05670
      NCH=NCHS(1)*NCHS(2)                                                DRT05680
      WRITE(IOOUT,832)NCH,(SRCBAS(I),I=1,2)                              DRT05690
  832 FORMAT(27X,28HTOTAL NUMBER OF CHARGES IS ,1X,I3,1X,                DRT05700
     +27H WITH REFERENCE CHARGE AT (,F8.2,1H,,F8.2,1H))                  DRT05710
      WRITE(IOOUT,834)NCHS(1),(SIDE1(I),I=1,2)                           DRT05720
  834 FORMAT(32X,I3,1X,45HCHARGES WITH DIRECTION AND SPACING GIVEN BY (,DRT05730
     +F8.2,1H,,F8.2,1H))                                                 DRT05740
      WRITE(IOOUT,834)NCHS(2),(SIDE2(I),I=1,2)                           DRT05750
      GO TO 300                                                          DRT05760
C                                                                        DRT05770
C  CHARGE DISTRIBUTION TYPE 2                                            DRT05780
C                                                                        DRT05790
  133 NCH=NCHS(1)                                                        DRT05800
      DO 134 J=1,NCH                                                     DRT05810
      READ(IOIN,701)(COOR(K,J),K=1,2)                                    DRT05820
  701 FORMAT(8X,2F8.2)                                                   DRT05830
  134 CONTINUE                                                           DRT05840
      WRITE(IOOUT,836)                                                   DRT05850
  836 FORMAT(/,42X,48HSIMULTANEOUS BURST, RANDOMLY DISTRIBUTED CHARGES)  DRT05860
      WRITE(IOOUT,838)NCHS(1)                                            DRT05870
  838 FORMAT(51X,26HTOTAL NUMBER OF CHARGES IS  ,1X,I3 )                 DRT05880
      WRITE(IOOUT,840)                                                   DRT05890
  840 FORMAT(55X,22HDETONATION COORDINATES)                              DRT05900
      DO 135 J=1,NCH                                                     DRT05910
      WRITE(IOOUT,842)(COOR(I,J),I=1,2)                                  DRT05920
  842 FORMAT(53X,2(3X,F8.2))                                             DRT05930
  135 CONTINUE                                                           DRT05940
      GO TO 300                                                          DRT05950
C                                                                        DRT05960
C  CHARGE DISTRIBUTION TYPE 3                                            DRT05970
C                                                                        DRT05980
  136 NCH=NCHS(1)                                                        DRT05990
      DO 137 J=1,NCH                                                     DRT06000
      READ(IOIN,702)(COOR(K,J),K=1,2),TSTAG(J)                           DRT06010
  702 FORMAT(8X,3F8.2)                                                   DRT06020
  137 CONTINUE                                                           DRT06030
      WRITE(IOOUT,844)                                                   DRT06040
  844 FORMAT(/,30X,38HSEQUENTIAL IN TIME AND RANDOM IN SPACE,            DRT06050
     +24H DISTRIBUTION OF CHARGES)                                       DRT06060
      WRITE(IOOUT,838)NCH                                                DRT06070
      WRITE(IOOUT,846)                                                   DRT06080
  846 FORMAT(45X,25H DETONATION COORDINATES  ,7X,10HBLAST TIME)          DRT06090
      DO 138 J=1,NCH                                                     DRT06100
      WRITE(IOOUT,848)(COOR(I,J),I=1,2),TSTAG(J)                         DRT06110
  848 FORMAT(46X,F8.2,3X,F8.2,12X,F8.2)                                  DRT06120
  138 CONTINUE                                                           DRT06130
      GO TO 300                                                          DRT06140
C                                                                        DRT06150
C  STORE AND PRINT OUT INFORMATION ABOUT VEHICLE                         DRT06160
C                                                                        DRT06170
  150 CONTINUE                                                           DRT06180
      VH=.TRUE.                                                          DRT06190
```

```
      V0(1)=RDIN(2)                                                          DRT06200
      V0(2)=RDIN(3)                                                          DRT06210
      VEHDIR=RDIN(4)                                                         DRT06220
      VEHSPD=RDIN(5)                                                         DRT06230
      VEHWID=RDIN(6)                                                         DRT06240
      VEHWHT=RDIN(7)                                                         DRT06250
      VEHTYP=IFIX(RDIN(8))                                                   DRT06260
      WRITE(IOOUT,850)V0(1),V0(2)                                            DRT06270
  850 FORMAT(//,44X,26HINITIAL VEHICLE POSITION (,F8.2,1H,,F8.2,1H))         DRT06280
      WRITE(IOOUT,852)VEHDIR                                                 DRT06290
  852 FORMAT(44X,19HVEHICLE DIRECTION  ,F8.2,17H  (CCW FROM EAST))           DRT06300
      WRITE(IOOUT,854)VEHSPD                                                 DRT06310
  854 FORMAT(50X,20HVEHICLE SPEED       ,F8.2,4H M/S)                        DRT06320
      WRITE(IOOUT,856)VEHWID                                                 DRT06330
  856 FORMAT(52X,20HVEHICLE WIDTH       ,F8.2)                               DRT06340
      WRITE(IOOUT,858)VEHWHT                                                 DRT06350
  858 FORMAT(52X,20HVEHICLE WEIGHT      ,F8.2)                               DRT06360
      IF(VEHTYP.EQ.0)WRITE(IOOUT,891)                                        DRT06370
  891 FORMAT(58X,15HWHEELED VEHICLE)                                         DRT06380
      IF(VEHTYP.EQ.1)WRITE(IOOUT,892)                                        DRT06390
  892 FORMAT(58X,15HTRACKED VEHICLE)                                         DRT06400
      GO TO 300                                                              DRT06410
C                                                                            DRT06420
CC   STORE TRANSMITTER COORDINATES AND TRANSMISSION TRHESHOLD                DRT06430
C                                                                            DRT06440
  170 CONTINUE                                                               DRT06450
      TC=.TRUE.                                                              DRT06460
      NEWCOR=.TRUE.                                                          DRT06470
      TRNCOR(1)=RDIN(2)                                                      DRT06480
      TRNCOR(2)=RDIN(3)                                                      DRT06490
      TRNCOR(3)=RDIN(4)                                                      DR106500
      TRNMIN=RDIN(5)                                                         DRT06510
      IF(TRNMIN.LT.1.E-05)TRNMIN=1.E-05                                      DRT06520
      THRESH=-ALOG(TRNMIN)                                                   DRT06530
      GO TO 300                                                              DRT06540
C                                                                            DRT06550
CC   STORE RECEIVER COORDINATES                                              DRT06560
C                                                                            DRT06570
  190 CONTINUE                                                               DRT06580
      RC=.TRUE.                                                              DRT06590
      RECCOR(1)=RDIN(2)                                                      DRT06600
      RECCOR(2)=RDIN(3)                                                      DRT06610
      RECCOR(3)=RDIN(4)                                                      DRT06620
      GO TO 300                                                              DRT06630
C                                                                            DRT06640
CC   STORE OBSERVER COORDINATES                                              DRT06650
C                                                                            DRT06660
  210 CONTINUE                                                               DRT06670
      OC=.TRUE.                                                              DRT06680
      OBSCOR(1)=RDIN(2)                                                      DRT06690
      OBSCOR(2)=RDIN(3)                                                      DRT06700
      SPCHT=RDIN(4)                                                          DRT06710
      GO TO 300                                                              DRT06720
C                                                                            DRT06730
CC   STORE TIME INTERVAL FOR CALCULATIONS                                    DRT06740
C                                                                            DRT06750
  230 CONTINUE                                                               DRT06760
      TM=.TRUE.                                                              DRT06770
      TSTART=RDIN(2)                                                         DRT06780
      TEND=RDIN(3)                                                           DRT06790
      TINC=RDIN(4)                                                           DRT06800
      IF (TINC.LE.0.0) TINC=1.                                               DRT06810
      IF(TEND.LT.TSTART)GO TO 903                                            DRT06820
      LIM=IFIX((TEND-TSTART)/TINC)+1                                         DRT06830
  300 CONTINUE                                                               DRT06840
  310 CONTINUE                                                               DRT06850
      IF(IGEOSW.NE.1) GO TO 333                                              DRT06860
      TRNCOR(1)=PTS(1)*1000.                                                 DRT06870
      TRNCOR(2)=PTS(2)*1000.                                                 DRT06880
      TRNCOR(3)=PTS(3)*1000.
```

```
      RECCOR(1)=PTS(4)*1000.                                                    DRT06890
      RECCOR(2)=PTS(5)*1000.                                                    DRT06900
      RECCOR(3)=PTS(6)*1000.                                                    DRT06910
      OBSCOR(1)=PTS(10)*1000.                                                   DRT06920
      OBSCOR(2)=PTS(11)*1000.                                                   DRT06930
      SPCHT=PTS(12)*1000.                                                       DRT06940
  333 CONTINUE                                                                  DRT06950
      IOPT=IFIX(RDIN(2))                                                        DRT06970
      IFILE=IFIX(RDIN(3))                                                       DRT06980
      IF(IOPT.NE.3) GO TO 305                                                   DRT06990
      WRITE(IOOUT,301)                                                          DRT07000
      WRITE(IOOUT,302)                                                          DRT07010
      WRITE(IOOUT,303) IFILE                                                    DRT07020
      WRITE(IOOUT,304)                                                          DRT07030
      WRITE(IOOUT,301)                                                          DRT07040
  301 FORMAT(1H0,130(1H*),/)                                                    DRT07050
  302 FORMAT(1H0,58X,15HDIRTRAN WARNING,/)                                      DRT07060
  303 FORMAT(1H0,38X,13HLOGICAL UNIT ,I2,27H IS ASSIGNED TO A TEMPORARY,DRT07070
     + 13H STORAGE FILE)                                                        DRT07080
  304 FORMAT(1H0,28X,46HCARE MUST BE TAKEN TO INSURE THAT THIS UNIT IS,         DRT07090
     +27H NOT IN USE BY ANOTHER FILE,/)                                         DRT07100
  305 CONTINUE                                                                  DRT07110
      IF((IOPT.EQ.1.OR.IOPT.EQ.2).AND.(EX.AND.(.NOT.CH)))GO TO 909              DRT07120
      IF((IOPT.EQ.1.OR.IOPT.EQ.2).AND.(CH.AND.(.NOT.EX)))GO TO 909              DRT07130
      IF((IOPT.EQ.3).AND.OC)OC=.FALSE.                                          DRT07140
      IF((IOPT.EQ.4).AND.OC)OC=.FALSE.                                          DRT07150
      IF((IOPT.EQ.5).AND.OC)OC=.FALSE.                                          DRT07160
      IF((TC.AND.(.NOT.RC)).OR.(RC.AND.(.NOT.TC)))GO TO 913                     DRT07170
      IF(M1. OR.M2)NEWATM=.TRUE.                                                DRT07180
      IF(IOPT.EQ.3.AND.CH)NEWSRC=.TRUE.                                         DRT07190
      IF(EX.AND.CH)NEWSRC=.TRUE.                                                DRT07200
      IF(TC.AND.RC)LOSTRN=.TRUE.                                                DRT07210
      IF(OC)EDGE=.TRUE.                                                         DRT07220
      IF(VH)NEWVEH=.TRUE.                                                       DRT07230
      IF(IOPT.EQ.3)LIM=1                                                        DRT07240
      DO 400 J=1,LIM                                                            DRT07250
      TIME=TSTART+TINC*FLOAT(J-1)                                               DRT07260
      NEWTIM=.TRUE.                                                             DRT07270
      NERR=0                                                                    DRT07280
      IF(IOPT.EQ.1.OR.IOPT.EQ.2)GO TO 320                                       DRT07290
      IF(IOPT.EQ.3.OR.IOPT.EQ.4)GO TO 325                                       DRT07300
      IF((IOPT.EQ.1).AND.(NARY.NE.1))GO TO 915                                  DRT07310
      IF((IOPT.EQ.2).AND.(NARY.NE.2))GO TO 915                                  DRT07320
      IF((IOPT.EQ.4).AND.(NARY.NE.3))GO TO 915                                  DRT07330
C                                                                               DRT07340
C  COMPUTE FOR VEHICLE SOURCE                                                   DRT07350
C                                                                               DRT07360
C  CHECK TO SEE IF WE HAVE THE MINIMUM INPUT REQUIREMENTS                       DRT07370
C                                                                               DRT07380
      IF(DSOD.GT.0.0)GO TO 315                                                  DRT07390
      IF(.NOT.(M1.AND.M2.AND.SL.AND.VH.AND.TC.AND.RC.AND.TM))GO TO 911          DRT07400
      CALL VEHCL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,PHI,NSOIL,                DRT07410
     1           SILT,NWL,TRNCOR,RECCOR,TIME,DHDT,V0,VEHDIR,                    DRT07420
     2           VEHSPD,VEHWID,VEHWHT,VEHTYP,NEWATM,NEWVEH,TRNLOS,NERR)         DRT07430
      NEWVEH=.FALSE.                                                            DRT07440
      NEWATM=.FALSE.                                                            DRT07450
      GO TO 330                                                                 DRT07460
  315 TRNLOS=1.0                                                                DRT07470
      GO TO 335                                                                 DRT07480
  325 CONTINUE                                                                  DRT07490
      IF(IOPT.EQ.4.AND.(.NOT.(EX.AND.TC.AND.RC.AND.TM)))GO TO 911               DRT07500
      IF(IOPT.EQ.3.AND.(.NOT.(M1.AND.M2.AND.SL.AND.CH)))GO TO 911               DRT07510
      CALL COMPCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,                  DRT07520
     1            PHI,NEWSRC,CHWT,NCHRG,NCHS,DETDEP,NSOIL,DSOD,NWL,             DRT07530
     2            TRNCOR,RECCOR,TIME,DHDT,TRNLOS,NERR)                          DRT07540
      NEWSRC=.FALSE.                                                            DRT07550
      NEWATM=.FALSE.                                                            DRT07560
      IF(IOPT.EQ.3)GO TO 410                                                    DRT07570
      GO TO 330                                                                 DRT07580
  320 CONTINUE                                                                  DRT07590
```

```
C                                                                       DRT07600
C    CHECK TO SEE IF MINIMUM INPUTS ARE AVAILABLE                       DRT07610
C                                                                       DRT07620
      IF(.NOT.(M1.AND.M2.AND.SL.AND.CH.AND.EX.AND.(TC.AND.RC.OR.OC).AND.DRT07630
     1TM))GO TO 911                                                     DRT07640
      CALL DUSTCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,            DRT07650
     1            THWND,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,           DRT07660
     2            LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,       DRT07670
     3            TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS,   DRT07680
     4            NERR,NCHS,SRCBAS,SIDE1,SIDE2,DHDT)                   DRT07690
      NEWSRC=.FALSE.                                                    DRT07700
      NEWATM=.FALSE.                                                    DRT07710
  330 IF(NERR.EQ.0)GO TO 335                                            DRT07720
      WRITE(IOOUT,857)NERR                                              DRT07730
  857 FORMAT(55X,30H ***** DIRTRAN ERROR NUMBER     ,I2)                DRT07740
      GO TO 400                                                         DRT07750
  335 IF(ONCE.AND.(.NOT.NEWCOR))GO TO 340                               DRT07760
      NEWCOR=.FALSE.                                                    DRT07770
      IZINV=IFIX(ZINV)                                                  DRT07780
      WRITE(IOOUT,859)IZINV                                             DRT07790
  859 FORMAT(//,47X,30HESTIMATED INVERSION HEIGHT     ,I7)              DRT07800
C                                                                       DRT07810
C   IF OBSERVER IS SPECIFIED, OUTPUT IS LABELED FOR EACH TIME.          DRT07820
C   IF ONLY TRANSMITTER AND RECEIVER ARE INPUT, OUTPUT IS TABULAR       DRT07830
C                                                                       DRT07840
      IF(TC.AND.RC.AND.OC)GO TO 350                                     DRT07850
      IF(OC)GO TO 350                                                   DRT07860
      IF(.NOT.(TC.AND.RC))GO TO 905                                     DRT07870
      WRITE(IOOUT,860)WAVE1                                             DRT07880
  860 FORMAT(/,47X,18HWAVELENGTH          ,F7.2,12H MICROMETERS)        DRT07890
      WRITE(IOOUT,862)(TRNCOR(I),I=1,3)                                 DRT07900
      WRITE(IOOUT,864)(RECCOR(I),I=1,3)                                 DRT07910
  862 FORMAT(37X,28HTRANSMITTER COORDINATES      ,3F10.2)               DRT07920
  864 FORMAT(37X,28HRECEIVER    COORDINATES      ,3F10.2)               DRT07930
      WRITE(IOOUT,866)                                                  DRT07940
  866 FORMAT(/52X,4HTIME,10X,13HTRANSMITTANCE)                          DRT07950
  340 CONTINUE                                                          DRT07960
      WRITE(IOOUT,868)TIME,TRNLOS                                       DRT07970
  868 FORMAT(52X,F8.2,10X,E10.5)                                        DRT07980
      ONCE=.TRUE.                                                       DRT07990
      GO TO 400                                                         DRT08000
  350 WRITE(IOOUT,923)TIME                                              DRT08010
  923 FORMAT(//,48X,28HTIME AFTER BLAST              ,F7.2)             DRT08020
      IF(.NOT.(TC.AND.RC))GO TO 360                                     DRT08030
      WRITE(IOOUT,808)                                                  DRT08040
      WRITE(IOOUT,860)WAVE1                                             DRT08050
      WRITE(IOOUT,862)(TRNCOR(I),I=1,3)                                 DRT08060
      WRITE(IOOUT,864)(RECCOR(I),I=1,3)                                 DRT08070
      WRITE(IOOUT,870)TRNLOS                                            DRT08080
  870 FORMAT(42X,38HTRANSMITTANCE ALONG THE LINE OF SIGHT  ,E10.3)      DRT08090
  360 WRITE(IOOUT,808)                                                  DRT08100
      WRITE(IOOUT,872)                                                  DRT08110
  872 FORMAT(57X,28HAERODYNAMIC CLOUD DIMENSIONS)                       DRT08120
      WRITE(IOOUT,808)                                                  DRT08130
      WRITE(IOOUT,874)(OBSCOR(I),I=1,2)                                 DRT08140
  874 FORMAT(41X,28HOBSERVER    COORDINATES      ,2F10.2)               DRT08150
      WRITE(IOOUT,876)HEIGHT                                            DRT08160
  876 FORMAT(39X,26HTHE HEIGHT OF THE CLOUD IS,10X,F10.2,7H METERS)     DRT08170
      WRITE(IOOUT,878)(CNTRD(I0),I0=1,2)                                DRT08180
  878 FORMAT(38X,28HTHE CENTROID COORDINATES ARE,8X,2F10.2)             DRT08190
      WRITE(IOOUT,880)CENWTH                                            DRT08200
  880 FORMAT(38X,28HTHE WIDTH AT THE CENTROID IS,8X,F10.2, 7H METERS)   DRT08210
      WRITE(IOOUT,882)SPCHT,SPCWTH                                      DRT08220
  882 FORMAT(39X,12HTHE WIDTH AT,F8.2,11H METERS IS ,5X,F10.2,7H METERS)DRT08230
      WRITE(IOOUT,884)NCPTS                                             DRT08240
  884 FORMAT(46X,I3,37H  CONTOUR POINTS HAVE BEEN DETERMINED )          DRT08250
      WRITE(IOOUT,886)((CPTS(I0,IPT),I0=1,2),IPT=1,NCPTS)               DRT08260
  886 FORMAT((60X,2(F10.3,2X)))                                         DRT08270
  400 CONTINUE                                                          DRT08280
      GO TO 10                                                          DRT08290
```

```
410 WRITE(IOOUT,888)                                                    DRT08300
888 FORMAT(/,38X,48H** THE CLOUD HAS BEEN PRECOMPUTED AND STORED ON,    DRT08310
    1 5H FILE)                                                          DRT08320
    GO TO 10                                                            DRT08330
900 WRITE(IOOUT,901)                                                    DRT08340
901 FORMAT(/,24X,48H*** DIRTRAN ERROR - MORE THAN 15 RECORDS OF DATA,   DRT08350
    1   35H HAVE BEEN INPUT WITHOUT A GO CARD.)                         DRT08360
    IERR=1                                                              DRT08370
    GO TO 999                                                           DRT08380
903 WRITE(IOOUT,904)                                                    DRT08390
904 FORMAT(/,39X,47H*** DIRTRAN ERROR - TIMES ARE NOT IN INCREASING,    DRT08400
    + 6H ORDER)                                                         DRT08410
    IERR=1                                                              DRT08420
    GO TO 999                                                           DRT08430
905 WRITE(IOOUT,906)                                                    DRT08440
906 FORMAT(/18X,46H*** DIRTRAN ERROR -NO TRANSMITTER AND RECEIVER,      DRT08450
    + 49H AND/OR OBSERVER COORDINATES HAVE BEEN SPECIFIED.)             DRT08460
    IERR=1                                                              DRT08470
    GO TO 999                                                           DRT08480
909 WRITE(IOOUT,910)                                                    DRT08490
910 FORMAT(/,25X,44H*** DIRTRAN ERROR - ONLY ONE DATA RECORD FOR,       DRT08500
    1 38H CHARGE INFORMATION HAS BEEN SPECIFIED)                        DRT08510
    IERR=1                                                              DRT08520
    GO TO 999                                                           DRT08530
911 WRITE(IOOUT,912)                                                    DRT08540
912 FORMAT(/,16X,49H*** DIRTRAN ERROR - MINIMUM AMOUNT OF INFORMATION,DRT08550
    1 26H REQUIRED IS NOT AVAILABLE,/,10X,14H  CHECK INPUTS)            DRT08560
    IERR=1                                                              DRT08570
    GO TO 999                                                           DRT08580
913 WRITE(IOOUT,914)                                                    DRT08590
914 FORMAT(/20X,49H*** DIRTRAN ERROR - BOTH TRANSMITTER AND RECEIVER,   DRT08600
    1 43H LOCATIONS MUST BE SPECIFIED,  CHECK INPUTS)                   DRT08610
    IERR=1                                                              DRT08620
    GO TO 999                                                           DRT08630
915 WRITE(IOOUT,916)                                                    DRT08640
916 FORMAT(/30X,50H IOPT AND NARY DO NOT AGREE SEE THE ABOVE COMMENTS,DRT08650
    1 21H FOR CORRECT MATCHING)                                         DRT08660
    IERR=1                                                              DRT08670
999 RETURN                                                              DRT08680
    END                                                                 DRT08690
```

```
      SUBROUTINE AMOUNT(VOLSPH,WAKAL,SPHAL)                              AMOU0250
C                                                                        AMOU0010
C  SUBROUTINE TO DETERMINE LOADING FOR THE SPHERE AND WAKE               AMOU0020
C                                                                        AMOU0030
C  INPUTS                                                                AMOU0040
C                                                                        AMOU0050
C     VOLSPH - VOLUME OF THE BUOYANTLY RISING SPHERE                     AMOU0060
C                                                                        AMOU0070
C     ALL OTHER NEEDED INFORMATION IS PASSED VIA COMMON BLOCKS           AMOU0080
C                                                                        AMOU0090
C  OUTPUTS                                                               AMOU0100
C                                                                        AMOU0110
C     WAKAL  - AMOUNT OF INITIAL LOADING OF SPHERE THAT HAS BEEN         AMOU0120
C              DEPOSIITED IN THE WAKE                                    AMOU0130
C                                                                        AMOU0140
C     SPHAL  - AMOUNT OF INITIAL LOADIING OF SPHERE THAT IS LEFT         AMOU0150
C              IN THE SPHERE                                             AMOU0160
C                                                                        AMOU0170
C                                                                        AMOU0180
C  FUNCTIONS AND SUBROUTINES NEEDED                                      AMOU0190
C                                                                        AMOU0200
C     NONE                                                               AMOU0210
C                                                                        AMOU0220
C*************************************************************************AMOU0230
C                                                                        AMOU0240
      COMMON/NTAL/TNOT,VOLNOT,TNO,CBLEED                                 AMOU0260
      COMMON/BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM   AMOU0270
      TSPH=TNO+DELT                                                      AMOU0280
      STUFF=CBLEED*(VOLSPH/TSPH-VOLNOT/TNOT)                             AMOU0290
      WAKAL=AMIN1(SPHNS(1),STUFF)                                        AMOU0300
      SPHAL=SPHNS(1)-WAKAL                                               AMOU0310
      RETURN                                                             AMOU0320
      END                                                                AMOU0330
```

```
      SUBROUTINE ATMCAL(NATM,ZT,TMES,ZU,UMES,PHI,BETA,DHDT,ERR)          ATMC0010
      REAL M,N,K,KM                                                      ATMC0020
      LOGICAL ERR,DHDT                                                   ATMC0030
      DIMENSION ZT(2),TMES(2),ZU(2),UMES(2),ZL0(6)                       ATMC0040
      COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,M,N,ZINV                          ATMC0050
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                                 ATMC0060
      COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT                              ATMC0070
      COMMON/STARS/USTAR,TSTAR,ZSTAR                                     ATMC0080
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUATMC0090
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                      ATMC0100
      DATA ZL0/-2.5,-4.5,-13.5,10000.,55.,20./                           ATMC0110
      DATA OMEGA,K /7.2722E-05,.4/                                       ATMC0120
C*************************************************************          ATMC0130
C                                                                        ATMC0140
C                                                                        ATMC0150
C     PURPOSE                                                            ATMC0160
C                                                                        ATMC0170
C          TO FIT THE BEST POWER-LAW PROFILES OF WINDSPEED AND           ATMC0180
C          DIFFUSIVITY CONSISTENT WITH KNOWN RELATIONS GOVERNING         ATMC0190
C          THE CONSTANT SHEAR STRESS LAYER TO GIVEN MEASUREMENTS         ATMC0200
C          AT ONE OR TWO HEIGHTS.  ALSO TO CALCULATE PARAMETERS          ATMC0210
C          NEEDED FOR VERTICAL VARIATION IN WIND DIRECTION LAYER,        ATMC0220
C          FOR WIND AND TEMPERATURE PROFILES.                            ATMC0230
C                                                                        ATMC0240
C                                                                        ATMC0250
C     INPUTS                                                             ATMC0260
C                                                                        ATMC0270
C          NATM       INTEGER WHICH IS 0 IF WINDSPEED AND TEMPERATURE    ATMC0280
C                     ARE AVAILABLE AT TWO HEIGHTS AND EQUAL TO THE      ATMC0290
C                     PASQUILL CATEGORY OTHERWISE.                       ATMC0300
C                                                                        ATMC0310
C          ZT         SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS    ATMC0320
C                     (IN METERS) AT WHICH TEMPERATURES WILL BE GIVEN.   ATMC0330
C                     MUST BE IN ASCENDING ORDER.                        ATMC0340
C                                                                        ATMC0350
C          TMES       SINGLY DIMENSIONED ARRAY CONTAINING THE TWO        ATMC0360
C                     TEMPERATURE MEASUREMENTS IN DEGREES KELVIN         ATMC0370
C                     AT HEIGHTS ZT.                                     ATMC0380
C                                                                        ATMC0390
C          ZU         SINGLY DIMENSIONED ARRAY CONTAINING ONE OR TWO     ATMC0400
C                     HEIGHTS (METERS) AT WHICH WIND SPEEDS WILL BE      ATMC0410
C                     GIVEN.  MUST BE IN ASCENDING ORDER.                ATMC0420
C                                                                        ATMC0430
C          UMES       SINGLY DIMENSIONED ARRAY CONTAINING THE ONE OR     ATMC0440
C                     TWO WIND SPEED MEASUREMENTS (M/S) AT HEIGHTS UMES. ATMC0450
C                                                                        ATMC0460
C                                                                        ATMC0470
C          PHI        LATITUDE OF DETONATION SITE.                       ATMC0480
C                                                                        ATMC0490
C          BETA       ANGLE OF WIND VELOCITY VECTOR MEASURED COUNTER-    ATMC0500
C                     CLOCKWISE FROM EAST.                               ATMC0510
C                                                                        ATMC0520
C          DHDT       A LOGICAL VARIABLE WHICH IS .FALSE. IF THE         ATMC0530
C                     INVERSION LAYER HEIGHT IS RELATIVELY CONSTANT      ATMC0540
C                     AND .TRUE. IF THE LAYER HEIGHT IS INCREASING.      ATMC0550
C                                                                        ATMC0560
C                                                                        ATMC0570
C     OUTPUTS                                                            ATMC0580
C                                                                        ATMC0590
C          ERR         A LOGICAL WHICH IS TRUE IF AN ERROR IS INCURRED   ATMC0600
C                      DURING THE CALCULATION.                           ATMC0610
C                                                                        ATMC0620
C          DXZ0        THE RATIO OF THE DIFFUSIVITY IN THE X DIRECTION   ATMC0630
C                      TO THE DIFFUSIVITY IN THE Z DIRECTION. RETURNED   ATMC0640
C                      IN COMMON /WNDPRM/.                               ATMC0650
C                                                                        ATMC0660
C          DYX0        THE RATIO OF THE DIFFUSIVITY IN THE Y DIRECTION   ATMC0670
C                      TO THE DIFFUSIVITY IN THE X DIRECTION. RETURNED   ATMC0680
C                      IN COMMON /WNDPRM/.                               ATMC0690
C                                                                        ATMC0700
```

```
C          DZ0          THE COEFFICIENT OF Z**N IN THE VERTICAL PROFILE    ATMC0710
C                       OF VERTICAL DIFFUSIVITY.  RETURNED IN COMMON       ATMC0720
C                       /WNDPRM/.                                          ATMC0730
C                                                                          ATMC0740
C          U0           THE COEFFICIENT OF Z**M IN THE VERTICAL PROFILE    ATMC0750
C                       OF HORIZONTAL WIND SPEED.  RETURNED IN COMMON      ATMC0760
C                       /WNDPRM/.                                          ATMC0770
C                                                                          ATMC0780
C          M            THE EXPONENT OF Z IN THE HORIZONTAL WIND SPEED     ATMC0790
C                       PROFILE.  RETURNED IN COMMON /WNDPRM/.             ATMC0800
C                                                                          ATMC0810
C          N            THE EXPONENT OF Z IN THE VERTICAL DIFFUSIVITY      ATMC0820
C                       PROFILE.  RETURNED IN COMMON /WNDPRM/.             ATMC0830
C                                                                          ATMC0840
C          ZINV       ESTIMATED INVERSION HEIGHT. RETURNED IN /WNDPRM/.   ATMC0850
C                                                                          ATMC0860
C          USTAR      VELOCITY PROFILE SCALE RETURNED IN COMMON           ATMC0870
C                     /STARS/.                                            ATMC0880
C                                                                          ATMC0890
C          TSTAR      TEMPERATURE PROFILE SCALE. RETURNED IN COMMON       ATMC0900
C                     /STARS/.                                            ATMC0910
C                                                                          ATMC0920
C          ZSTAR      HEIGHT AT WHICH THE VERTICAL VARIATION IN WIND      ATMC0930
C                     DIRECTION PROFILES FOR WIND AND TEMPERATURE         ATMC0940
C                     TAKE EFFECT.                                        ATMC0950
C                                                                         ATMC0960
C                   OUTPUT RETURNED IN COMMON /EKWIND/ ARE THE            ATMC0970
C                   PARAMETERS NEEDED FOR THE WIND PROFILE ABOVE ZSTAR.   ATMC0980
C                                                                         ATMC0990
C                   OUTPUT RETURNED IN COMMON /EKTEMP/ ARE THE            ATMC1000
C                   PARAMETERS NEEDED FOR THE TEMPERATURE PROFILES ABOVE  ATMC1010
C                   ZSTAR.                                                ATMC1020
C                                                                         ATMC1030
C     CALLED FROM DUSTCL                                                  ATMC1040
C                                                                         ATMC1050
C     NEEDED FUNCTIONS AND SUBROUTINES                                    ATMC1060
C                                                                         ATMC1070
C           TMPCAL     CALCULATES SCALED TEMPERATURE PROFILES             ATMC1080
C                                                                         ATMC1090
C           WNDCAL     CALCULATES SCALED WIND SPEED PROFILES              ATMC1100
C                                                                         ATMC1110
C           DIFFUS     FUNCTION TO CALCULATE THE DIFFUSIVITY AT A GIVEN   ATMC1120
C                      HEIGHT.                                            ATMC1130
C                                                                         ATMC1140
C           TEMP       CALCULATES THE POTENTIAL TEMPERATURE AND GRADIENT  ATMC1150
C                      AT A GIVEN HEIGHT.                                 ATMC1160
C                                                                         ATMC1170
C*************************************************************************ATMC1180
C                                                                         ATMC1190
      ERR=.FALSE.                                                         ATMC1200
C                                                                         ATMC1210
C     DELTH IS THE DIFFERENCE IN POTENTIAL TEMPERATURE BETWEEN THE        ATMC1220
C     TWO HEIGHTS WHERE TEMPERATURE IS GIVEN.                             ATMC1230
C                                                                         ATMC1240
      Z0=0.01                                                             ATMC1250
      T0=TMES(1)                                                          ATMC1260
      IF(NATM.EQ.0)GO TO 100                                              ATMC1270
C                                                                         ATMC1280
C     ASSIGN ATMOSPHERIC PROFILE ACCORDING TO SPECIFIED PASQUILL          ATMC1290
C     CATEGORY                                                            ATMC1300
C                                                                         ATMC1310
C     Z0      FRICTION HEIGHT                                             ATMC1320
C     ZL      MONIN-OBUKOV LENGTH                                         ATMC1330
C     USTAR   THE FRICTION VELOCITY                                       ATMC1340
C     TSTAR   THE SCALING TEMPERATURE                                     ATMC1350
C                                                                         ATMC1360
      ZL=ZL0(NATM)                                                        ATMC1370
      IF(NATM.GE.5)Z0=1.E-04*ABS(ZL)                                      ATMC1380
      IF(NATM.LE.3)Z0=1.E-03*ABS(ZL)                                      ATMC1390
      NP=IFIX(SIGN(1.,ZL))                                                ATMC1400
```

```
      USTAR=UMES(1)/WNDCAL(Z0,ZL,ZU(1))                                 ATMC1410
      TSTAR=TMES(1)*USTAR**2/1.568/ZL                                   ATMC1420
      IF(NATM-4)200,300,210                                             ATMC1430
  100 CONTINUE                                                          ATMC1440
C                                                                       ATMC1450
CC    USE ITERATIVE PROCEDURE TO CONVERGE ON BEST ATMOSPHERIC PROFILE   ATMC1460
CC    TO MATCH DATA AT TWO HEIGHTS                                      ATMC1470
C                                                                       ATMC1480
      DELTH=TMES(2)-TMES(1)+.0098*(ZT(2)-ZT(1))                         ATMC1490
      NP=SIGN(1.,DELTH)                                                 ATMC1500
      DELU=UMES(2)-UMES(1)                                              ATMC1510
      ZULOG=ALOG(ZU(2)/ZU(1))                                           ATMC1520
      ZTLOG=ALOG(ZT(2)/ZT(1))                                           ATMC1530
      USTAR=(UMES(2)-UMES(1))/ZULOG                                     ATMC1540
      TSTAR=DELTH/ZTLOG                                                 ATMC1550
      ZL=.638*TMES(1)*USTAR**2/TSTAR                                    ATMC1560
      IF(ABS(ZL).GE.1000.)GO TO 300                                     ATMC1570
      DO 110 ITER=1,100                                                 ATMC1580
      USTAR=DELU/(WNDCAL(Z0,ZL,ZU(2))-WNDCAL(Z0,ZL,ZU(1)))              ATMC1590
      TSTAR=DELTH/(TMPCAL(Z0,ZL,ZT(2))-TMPCAL(Z0,ZL,ZT(1)))             ATMC1600
      ZLP=ZL                                                            ATMC1610
      ZL=.638*TMES(1)*USTAR**2/TSTAR                                    ATMC1620
      IF(ABS((ZL-ZLP)/ZLP).LT..01)GO TO 120                             ATMC1630
  110 CONTINUE                                                          ATMC1640
      ERR=.TRUE.                                                        ATMC1650
      GO TO 999                                                         ATMC1660
  120 CONTINUE                                                          ATMC1670
      IF(ZL.GT.0.)Z0=1.E-04*ABS(ZL)                                     ATMC1680
      IF(ZL.LE.0.)Z0=1.E-03*ABS(ZL)                                     ATMC1690
      IF(NP)200,300,210                                                 ATMC1700
  200 CONTINUE                                                          ATMC1710
C                                                                       ATMC1720
CC    UNSTABLE ATMOSPHERE                                               ATMC1730
C                                                                       ATMC1740
      DXZ0=2.6                                                          ATMC1750
      M=.079943                                                         ATMC1760
      N=4./3.                                                           ATMC1770
      DZ0=.7609*USTAR*ABS(ZL)**(1.-N)                                   ATMC1780
      U0=USTAR*14.2478/ABS(ZL)**M                                       ATMC1790
      GO TO 430                                                         ATMC1800
  210 CONTINUE                                                          ATMC1810
C                                                                       ATMC1820
CC    STABLE ATMOSPHERE                                                 ATMC1830
C                                                                       ATMC1840
      DXZ0=3.3                                                          ATMC1850
      N=.45644                                                          ATMC1860
      M=.28414                                                          ATMC1870
      DZ0=.059517*USTAR*ABS(ZL)**(1.-N)                                 ATMC1880
      U0=USTAR*36.6642/ABS(ZL)**M                                       ATMC1890
      GO TO 430                                                         ATMC1900
  300 CONTINUE                                                          ATMC1910
C                                                                       ATMC1920
CC    NEUTRAL ATMOSPHERE                                                ATMC1930
C                                                                       ATMC1940
      DZ0=.4*USTAR                                                      ATMC1950
      DXZ0=2.8                                                          ATMC1960
      NP=0                                                              ATMC1970
      N=1.                                                              ATMC1980
      M=1./7.                                                           ATMC1990
      U0=45.92*USTAR/ABS(ZL)**M                                         ATMC2000
  430 CONTINUE                                                          ATMC2010
C                                                                       ATMC2020
CC    COMMON CALCULATION TO UNSTABLE, NEUTRAL, AND STABLE ATMOSPHERES   ATMC2030
C                                                                       ATMC2040
      DYX0=1.                                                           ATMC2050
      IF(NATM.EQ.0)U0=(U0+UMES(2)/ZU(2)**M)/2.                          ATMC2060
C                                                                       ATMC2070
CC    ESTIMATE THE INVERSION HEIGHT AND COMPUTE THE NECESSARY           ATMC2080
CC    PARAMETERS FOR THE WIND AND TEMPERATURE PROFILES BETWEEN          ATMC2090
C     ZSTAR AND ZINV WHEN DHDT IS .FALSE..                              ATMC2100
```

```
C                                                                       ATMC2110
      APHI=PHI*PIRAD                                                    ATMC2120
      FREQ=2.*OMEGA*SIN(APHI)                                           ATMC2130
      HC=K*USTAR/FREQ                                                   ATMC2140
      IF(ZL.GT.0.0.AND.ZL.LT.1.E3)GO TO 500                             ATMC2150
      ZINV=HC                                                           ATMC2160
      GO TO 501                                                         ATMC2170
  500 ZINV=.26*HC                                                       ATMC2180
  501 ZSTAR=.13*ZINV                                                    ATMC2190
      KM=DIFFUS(Z0,ZL,ZSTAR)                                            ATMC2200
      SP=USTAR*WNDCAL(Z0,ZL,ZSTAR)                                      ATMC2210
      ALP=SQRT(FREQ/(2.*KM))                                            ATMC2220
      IF(DHDT)GO TO 813                                                 ATMC2230
      ARG=ALP*ZSTAR                                                     ATMC2240
      ARG1=BETA+ALP*ZINV                                                ATMC2250
      ARG2=ALP*(ZINV-ZSTAR)                                             ATMC2260
      C=SP*EXP(ARG)*SIN(ARG1)/SIN(ARG2)                                 ATMC2270
      PYF=C*EXP(-ARG)*COS(ARG)-SP*COS(BETA)                             ATMC2280
      PXF=C*EXP(-ARG)*SIN(ARG)+SP*SIN(BETA)                             ATMC2290
      UE=C*EXP(-ARG)*COS(ARG)-PYF                                       ATMC2300
      VE=-C*EXP(-ARG)*SIN(ARG)+PXF                                      ATMC2310
      UHAT=UE/SQRT(UE*UE+VE*VE)                                         ATMC2320
      VHAT=VE/SQRT(UE*UE+VE*VE)                                         ATMC2330
      CALL TEMP(ZSTAR,TA,DTADZ)                                         ATMC2340
      DTADH=0.0                                                         ATMC2350
      TC3=(DTADH-DTADZ)/(2.*(ZINV-ZSTAR))                               ATMC2360
      TC2=DTADZ-2.*TC3*ZSTAR                                            ATMC2370
      TC1=TA-TC2*ZSTAR-TC3*ZSTAR**2                                     ATMC2380
      GO TO 999                                                         ATMC2390
  813 ZSTAR=1.E4                                                        ATMC2400
  999 RETURN                                                            ATMC2410
      END                                                               ATMC2420
```

```
      SUBROUTINE AVRG(ZX,TIME,QTOT,XBAVG,SIG2X,SIG2Y)                   AVRG0230
C                                                                       AVRG0010
C  ROUTINE FOR FINDING AVERAGES OF THE MOMENTS OF THE DISCS             AVRG0020
C                                                                       AVRG0030
C                                                                       AVRG0040
C  INPUTS                                                               AVRG0050
C                                                                       AVRG0060
C     ZX    - HEIGHT AT WHICH THE AVERAGES ARE DESIRED                  AVRG0070
C                                                                       AVRG0080
C     TIME  - PRESENT TIME                                              AVRG0090
C                                                                       AVRG0100
C     QTOT  - SUM OF T00*QDSCS                                          AVRG0110
C                                                                       AVRG0120
C     XBAVG- AVERAGE OF THE FIRST ORDER MOMENTS (ALONGWIND DISPLACEMENT AVRG0130
C            IN THE WIND DIRECTION)                                     AVRG0140
C                                                                       AVRG0150
C     SIG2X- AVERAGE OF THE SQUARE OF ONE OF THE SECOND ORDER MOMENTS   AVRG0160
C            (ALONGWIND SPREAD)                                         AVRG0170
C                                                                       AVRG0180
C     SIG2Y- AVERAGE OF THE SQUARE OF ANOTHER SECOND ORDER MOMENT       AVRG0190
C            (CROSSWIND SPREAD)                                         AVRG0200
C                                                                       AVRG0210
C*********************************************************************  AVRG0220
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)AVRG0240
      COMMON/PRTINF/R0,VGRAV(3),NPRTS                                   AVRG0250
      QTOT=0.0                                                          AVRG0260
      QSIG2X=0.0                                                        AVRG0270
      QSIG2Y=0.0                                                        AVRG0280
      QXBAR=0.0                                                         AVRG0290
      Z=ZX                                                              AVRG0300
      DO 10 I=1,NDSCS                                                   AVRG0310
      H=ZDSC(I)                                                         AVRG0320
      ROH2=R2DSC(I)                                                     AVRG0330
      TOF=TIME-TDSC(I)                                                  AVRG0340
      CALL MOMENT(VGRAV(1),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)                  AVRG0350
      QT=QDSC(I,1)*Q                                                    AVRG0360
      QTOT=QTOT+QT                                                      AVRG0370
      QSIG2X=QSIG2X+(SIGW2+ROH2/2.)*QT                                  AVRG0380
      QSIG2Y=QSIG2Y+(SIGP2+ROH2/2.)*QT                                  AVRG0390
      QXBAR=QXBAR+(XBAR+XDSC(I))*QT                                     AVRG0400
   10 CONTINUE                                                          AVRG0410
      XBAVG=QXBAR/QTOT                                                  AVRG0420
      SIG2X=QSIG2X/QTOT                                                 AVRG0430
      SIG2Y=QSIG2Y/QTOT                                                 AVRG0440
  999 RETURN                                                            AVRG0450
      END                                                               AVRG0460
```

```
      SUBROUTINE CLIMB(FUNCT,GFUN,P1,FP1,NSERCH,NOCONT)                 CLI00520
C                                                                       CLI00010
C       THIS MODULE IS A SUBROUTINE THAT FINDS A POINT ON A CONTOUR     CLI00020
C  BY FINDING THE GRADIENT VECTOR AT THAT POINT AND MARCHING ALONG      CLI00030
C  IT UNTIL IT FINDS ITSELF IN A REGION GREATER THAN THE CONTOUR LEVEL. CLI00040
C  AT WHICH POINT IT MARCHES HORIZONTALLY, HALVING THE STEP SIZE        CLI00050
C  UNTIL THE CONTOUR IS REACHED WITHIN SPECIFIED RESOLUTION.            CLI00060
C      IN ADDITION IT WILL DETERMINE IF A CONTOUR EXISTS.               CLI00070
C                                                                       CLI00080
C                                                                       CLI00090
C     ARGUMENTS PASSED.                                                 CLI00100
C                                                                       CLI00110
C     INPUT                                                             CLI00120
C         FUNCT-THE FUNCTION(X,Y) ALSO GIVEN IN EXTERNAL.               CLI00130
C         P1-THE STARTING POINT.                                        CLI00140
C                                                                       CLI00150
C                                                                       CLI00160
C    OUTPUT                                                             CLI00170
C                                                                       CLI00180
C         P1     - THE POINT ON THE CONTOUR OR THE POINT AT WHICH       CLI00190
C                  THE FUNCTION REACHES A MAXIMUM BELOW THE CONTOUR     CLI00200
C                  LEVEL                                                CLI00210
C         FP1    - THE VALUE OF THE FUNCTION AT P                       CLI00220
C         NOCONT-THE ERROR FLAG.                                        CLI00230
C                  F-NO PROBLEM                                         CLI00240
C                  T-NO CONTOUR FOUND.                                  CLI00250
C         ERR-ERROR FLAG RETURNED BY 'NTRSCT'                           CLI00260
C             F-NO ERROR                                                CLI00270
C             T-ITERATION DIVERGED OR MAXIMUM SEARCH AREA EXCEDED       CLI00280
C                                                                       CLI00290
C     IN ADDITION,IN COMMON ARE..,                                      CLI00300
C                                                                       CLI00310
C         YMIN-THE LOWER LIMIT ON Y.                                    CLI00320
C         DELTA- THE STEP SIZE,MODIFIED IN THIS SUBROUTINE.             CLI00330
C         CON-THE CONTOUR LEVEL.                                        CLI00340
C         RES-THE RESOLUTION LENGTH                                     CLI00350
C                                                                       CLI00360
C     OTHER VARIABLES INCLUDE                                           CLI00370
C         GRAD-THE GRADIENT VECTOR                                      CLI00380
C         P0-THE CURRENT POINT ON THE GRADIENT.                         CLI00390
C         P1-THE POINT ON THE GRADIENT BEING TESTED                     CLI00400
C            TO SEE ABOUT CONTOUR EXISTENCE.                            CLI00410
C         FP0,FP1-THE FUNCTION VALUES OF P0 AND P1.                     CLI00420
C                                                                       CLI00430
C     CALLED SUBROUTINES                                                CLI00440
C                                                                       CLI00450
C         GRAD2-FINDS THE GRADIENT VECTOR OF A FUNCTION AT              CLI00460
C               A POINT AND THE SLOPE THERE.                            CLI00470
C         UNIT-CALCULATES THE NORM AND MAGNITUDE OF A 2 VECTOR.         CLI00480
C         VSUM-VECTOR SUM OF THE FORM C=A+SB WHERE S IS SCALAR          CLI00490
C              MULTIPLIER OF B.                                         CLI00500
C                                                                       CLI00510
      EXTERNAL FUNCT                                                    CLI00530
      LOGICAL NOCONT                                                    CLI00540
      DIMENSION GRAD(2),P0(2),P1(2)                                     CLI00550
      COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON        CLI00560
      COMMON/LIMIT/YMIN,FMIN                                            CLI00570
      NOCONT=.FALSE.                                                    CLI00580
      ONEM=-1.0                                                         CLI00590
      IF (NSERCH.EQ.0)GO TO 7                                           CLI00600
      DELTA=SIGN(DELTA,FLOAT(NSERCH))                                   CLI00610
      FP1=FUNCT(P1(1),P1(2))                                            CLI00620
      IF(FP1.LT.CON)GO TO 25                                            CLI00630
      GO TO 22                                                          CLI00640
    3 CONTINUE                                                          CLI00650
      P0(1)=P1(1)                                                       CLI00660
      P0(2)=P1(2)                                                       CLI00670
      FP0=FP1                                                           CLI00680
C  ** FINDING THE UNIT GRADIENT AND THE NEXT POINT ALONG IT.            CLI00690
    4 CALL GRAD2(P0,FUNCT,RES,GRAD,DFDS)                                CLI00700
```

```
    5 CALL VSUM(P0,GRAD,DELTA,P1)                                      CLI00710
C ** IS THE POINT HEADING BELOW YMIN **                                CLI00720
      IF(P1(2).GE.YMIN)GO TO 7                                         CLI00730
      P1(2)=YMIN                                                       CLI00740
      CALL VSUM(P1,P0,ONEM,GRAD)                                       CLI00750
      CALL UNIT(GRAD,GRAD,DELTA)                                       CLI00760
      IF(ABS(DELTA).LT.RES)GO TO 25                                    CLI00770
    7 FP1=FUNCT(P1(1),P1(2))                                           CLI00780
C ** HAS THE CONTOUR BEEN CROSSED **                                   CLI00790
    8 IF(FP1.GE.CON)GO TO 22                                           CLI00800
      IF(FP1.GT.FP0)GO TO 3                                            CLI00810
      DELTA=DELTA/2.                                                   CLI00820
      IF(ABS(DELTA).LT.RES)GO TO 25                                    CLI00830
      GO TO 5                                                          CLI00840
   25 NOCONT=.TRUE.                                                    CLI00850
      GO TO 99                                                         CLI00860
   22 CONTINUE                                                         CLI00870
C    BEGIN HORIZONTAL SEARCH                                           CLI00880
      P0(2)=P1(2)                                                      CLI00890
   31 P0(1)=P1(1)                                                      CLI00900
      FP0=FP1                                                          CLI00910
   40 P1(1)=P0(1)+DELTA                                                CLI00920
      FP1=FUNCT(P1(1),P1(2))                                           CLI00930
      IF(ABS(DELTA).LT.RES/2.)GO TO 99                                 CLI00940
      IF(FP1.GE.CON)GO TO 31                                           CLI00950
      DELTA=DELTA/2.                                                   CLI00960
      GO TO 40                                                         CLI00970
   99 CONTINUE                                                         CLI00980
      RETURN                                                           CLI00990
      END                                                              CLI01000
```

```
      SUBROUTINE CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS5,     CLD00010
     1 ERR)                                                              CLD00020
C*********************************************************************  CLD00030
C                                                                        CLD00040
C                                                                        CLD00050
C     PURPOSE                                                            CLD00060
C                                                                        CLD00070
C     CLDIM CALCULATES FIVE CONTOUR POINTS AND CLOUD DIMENSIONS AS       CLD00080
C     SEEN FROM THE SPECIFIED OBSERVER POSITION.  CLDIM REQUIRES CLOUD   CLD00090
C     PARAMETERS FROM THE BUOYANT RISE STAGE OF CLOUD DEVELOPMENT WHICH  CLD00100
C     ARE SUPPLIED IN COMMON STORAGE /BUOYCL/ AND /PRTINF/ AS WELL AS    CLD00110
C     VIEWING GEOMETRY WHICH IS SUPPLIED IN COMMON /GEOM/.  SPCHT IS     CLD00120
C     REQUIRED INPUT IN THE ARGUMENTS.  ALL OUTPUTS ARE ARGUMENTS.       CLD00130
C                                                                        CLD00140
C                                                                        CLD00150
C     INPUT                                                              CLD00160
C                                                                        CLD00170
C     SPCHT      THE SPECIFIED HEIGHT AT WHICH THE WIDTH OF THE CLOUD    CLD00180
C                IS DESIRED. (METERS)                                    CLD00190
C                                                                        CLD00200
C                                                                        CLD00210
C                                                                        CLD00220
C     OUTPUT                                                             CLD00230
C                                                                        CLD00240
C     CNTRD      A SINGLY DIMENSIONED ARRAY OF LENGTH 2 WHICH CONTAINS   CLD00250
C                THE HORIZONTAL AND VERTICAL COORDINATES, RESP., OF THE  CLD00260
C                CLOUD CENTROID.  (METERS)                               CLD00270
C     HEIGHT     THE HEIGHT OF THE CLOUD IN METERS                       CLD00280
C     CENWTH     THE WIDTH OF THE CLOUD AT THE CENTROID HEIGHT IN METERS CLD00290
C     SPCWTH     THE WIDTH OF THE CLOUD AT THE SPECIFIED HEIGHT (METERS) CLD00300
C     NCPTS      THE NUMBER OF CONTOUR POINTS (=6)                       CLD00310
C     CPTS       A DOUBLY DIMENSIONED ARRAY OF SIZE (2,N),N.GE.5, WHICH  CLD00320
C                CONTAINS THE HORIZONTAL AND VERTICAL COORDINATES OF     CLD00330
C                THE FIVE CONTOUR POINTS. (METERS)                       CLD00340
C                                                                        CLD00350
C                                                                        CLD00360
C     REQUIRED SUBROUTINES                                               CLD00370
C                                                                        CLD00380
C                                                                        CLD00390
C       CLIMB      DETERMINES IF THE CONTOUR EXISTS, IF SO FINDS A POINT CLD00400
C                  ON THE CONTOUR.                                       CLD00410
C                                                                        CLD00420
C     CALLED BY DUSTCL                                                   CLD00430
C                                                                        CLD00440
C*********************************************************************  CLD00450
      DIMENSION CNTRD(2),CPTS5(2,6),TOP(2)                               CLD00460
      LOGICAL HORIZ,NOCONT,SWITCH,CHANGE,ERR                             CLD00470
      REAL KZ,KX                                                         CLD00480
      COMMON /BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),TIM     CLD00490
      COMMON /PRTINF/R0,VGRAV(3),NPRTS                                   CLD00500
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)               CLD00510
      COMMON /MODE/ HORIZ                                                CLD00520
      COMMON /CLOCK/ T,TWIND                                             CLD00530
      COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF                      CLD00540
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV                          CLD00550
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE           CLD00560
      COMMON/SIG/SIG02,SIGC                                              CLD00570
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCLD00580
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),           CLD00590
     1 QDSC(20,3)                                                        CLD00600
      COMMON /SPECS/ RES,STEP,TANT,CON                                   CLD00610
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                      CLD00620
      EXTERNAL FUNCT,GFUN                                                CLD00630
      DATA RES,TANT /.4,.1/                                              CLD00640
      HORIZ=.TRUE.                                                       CLD00650
      ERR=.FALSE.                                                        CLD00660
      CON=ALOG(VISEXT)                                                   CLD00670
      CPTS5(2,1)=SPCHT                                                   CLD00680
      CPTS5(2,6)=SPCHT                                                   CLD00690
      U=U0*CPTS5(2,1)**UM                                                CLD00700
```

```
      CPTS5(1,1)=T*U*SINTH                                                  CLD00710
      CPTS5(1,6)=CPTS5(1,1)                                                 CLD00720
      NSERCH=-1                                                             CLD00730
      STEP=20.                                                              CLD00740
C                                                                           CLD00750
C  CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN ONE DIRECTION AT       CLD00760
C  THE HEIGHT OF THE OBSERVER.                                              CLD00770
C                                                                           CLD00780
      CALL CLIMB(FUNCT,GFUN,CPTS5,FP1,NSERCH,NOCONT)                        CLD00790
      NSERCH=1                                                              CLD00800
      STEP=20.                                                              CLD00810
C                                                                           CLD00820
C  CALL CLIMB TO SEARCH FOR THE EDGE OF THE CLOUD IN THE OPPOSITE           CLD00830
C  DIRECTION AT THE HEIGHT OF THE OBSERVER.                                 CLD00840
C                                                                           CLD00850
      CALL CLIMB(FUNCT,GFUN,CPTS5(1,6),FP1,NSERCH,NOCONT)                   CLD00860
      SPCWTH=CPTS5(1,6)-CPTS5(1,1)                                          CLD00870
      NCPTS=6                                                               CLD00880
      IF(T.LE.TWIND)GO TO 50                                                CLD00890
      CNTRD(1)=(XTR+VTR*(T-TTR))*SINTH+CENDIF                               CLD00900
      CNTRD(2)=ZTR                                                          CLD00910
      SIGX2=SIG02+2.*KX*(T-TTR)                                             CLD00920
      SIGZ2=SIG02+2.*KZ*(T-TTR)                                             CLD00930
      SIGX=SQRT(SIGX2)                                                      CLD00940
      SIGZ=SQRT(SIGZ2)                                                      CLD00950
      BOT=1./(2.*VISEXT)                                                    CLD00960
      ARG=BOT*QPUFF(1)/PI/SIGX/SIGZ                                         CLD00970
      IF(ARG.LT.1.0)GO TO 998                                               CLD00980
      TOP(1)=CNTRD(1)                                                       CLD00990
      TOP(2)=ZTR+SIGZ*SQRT(2.*ALOG(ARG))                                    CLD01000
      RAD=SIGX*SQRT(2.*ALOG(ARG))                                           CLD01010
      CENWTH=2.*(RAD+PRJARY)                                                CLD01020
      HEIGHT=TOP(2)                                                         CLD01030
      GO TO 100                                                             CLD01040
   50 CNTRD(1)=XCM*SCRN(1)+YCM*SCRN(2)+CENDIF                               CLD01050
      CNTRD(2)=ZCM                                                          CLD01060
      TOP(1)=XTOP*SCRN(1)+YTOP*SCRN(2)+CENDIF                               CLD01070
      TOP(2)=ZCM+RSPH                                                       CLD01080
      IF(TOP(2).GT.ZINV)TOP(2)=ZINV                                         CLD01090
      HEIGHT=TOP(2)                                                         CLD01100
      CENWTH=2.*(RSPH+PRJARY)                                               CLD01110
  100  CPTS5(1,2)=CNTRD(1)-CENWTH/2.                                        CLD01120
      CPTS5(2,2)=CNTRD(2)                                                   CLD01130
      CPTS5(1,3)=TOP(1)-PRJARY                                              CLD01140
      CPTS5(2,3)=TOP(2)                                                     CLD01150
      CPTS5(1,4)=TOP(1)+PRJARY                                              CLD01160
      CPTS5(2,4)=TOP(2)                                                     CLD01170
       CPTS5(1,5)=CNTRD(1)+CENWTH/2.                                        CLD01180
      CPTS5(2,5)=CNTRD(2)                                                   CLD01190
       NCPTS=6                                                              CLD01200
      GO TO 999                                                             CLD01210
  998 WRITE(IOOUT,1000)                                                     CLD01220
 1000 FORMAT(50H *** UPPER PART OF CLOUD HAS DISSIPATED ***         )       CLD01230
  999 RETURN                                                                CLD01240
      END                                                                   CLD01250
```

```
      SUBROUTINE COMPCL(NEWATM,NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,    COM00160
     1                  PHI,NEWSRC,CHWT,NCHRG,NCHS,DETDEP,NSOIL,DSOD,   COM00170
     2                  NWL,TRNCOR,RECCOR,TIME,DHDT,TRNLOS,NERR)        COM00180
C                                                                       COM00010
C  CONTROLING ROUTINE FOR PRECOMPUTING A SINGLE , EXPLOSION PRODUCED    COM00020
C  DUST CLOUD AND STORING IT ON AN EXTERNAL FILE TO BE USED AT A        COM00030
C  LATER RUNNING OF THE CODE FOR A RANDOM DISTRIBUTION IN SPACE AND TIMECOM00040
C  OF CHARGES                                                           COM00050
C                                                                       COM00060
C  INPUTS                                                               COM00070
C     FOR DETAILS SEE DRTRAN                                            COM00080
C                                                                       COM00090
C                                                                       COM00100
C  OUTPUTS                                                              COM00110
C                                                                       COM00120
C     TRNLOS -CALCULATE TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT COM00130
C                                                                       COM00140
C*********************************************************************COM00150
      DIMENSION ZTEMP(2),TMPMES(2),ZWND(2),WNDMES(2),TRNCOR(3),         COM00190
     1RECCOR(3),TRNFRM(2,2),SIDE1(2),SIDE2(2),NCHS(2),ORIG(2),TRN(3),   COM00200
     2          REC(3),SRCBAS(2),PAS(6)                                 COM00210
      LOGICAL DHDT,NEWATM,NEWSRC,ERR,FLAG                               COM00220
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)               COM00230
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCOM00240
      COMMON/VL/VLOAD                                                   COM00250
      COMMON/OPTION/IOPT,IFILE                                          COM00260
      COMMON/CLOCK/FTIME,TWIND                                          COM00270
      COMMON/BUOYCL/Y(8),SPHNS(3),RTIM                                  COM00280
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)COM00290
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,          COM00300
     +           COOR(2,200),TSTAG(200),                                COM00310
     +             ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),       COM00320
     1             RB(3,25),Z2(3,25)                                    COM00330
      COMMON/WNDPRM/DXZ0,DYZ0,DZ0,U0,UM,DN,ZINV                         COM00340
      COMMON/CARB/RCARB1,RCARB2                                         COM00350
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     COM00360
      DATA FLAG/.TRUE./                                                 COM00370
      DATA PAS/4HA   ,4HB   ,4HC   ,4HD   ,4HE   ,4HF   /               COM00380
      IF(IOPT.EQ.4)GO TO 500                                            COM00390
C                                                                       COM00400
C  PRECOMPUTE INFORMATION AND STORE ON FORTRAN UNIT IFILE               COM00410
C                                                                       COM00420
      THETAX=THWND*PI/180.                                              COM00430
      TWIND=1.E5                                                        COM00440
      TTR=1.E5                                                          COM00450
      IF(.NOT.NEWATM)GO TO 10                                           COM00460
      CALL ATMCAL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,PHI,THETAX,DHDT,ERR)  COM00470
      IF(.NOT.ERR)GO TO 10                                              COM00480
      NERR=7                                                            COM00490
      GO TO 999                                                         COM00500
   10 CONTINUE                                                          COM00510
      IF(.NOT.NEWSRC)GO TO 20                                           COM00520
      CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD)                         COM00530
   20 CONTINUE                                                          COM00540
C                                                                       COM00550
C  COMPUTE INITIAL LOADING                                              COM00560
C                                                                       COM00570
      SUM=0.0                                                           COM00580
      DO 25 I=1,NDSCS                                                   COM00590
      SUM=SUM+QDSC(I,1)                                                 COM00600
   25 CONTINUE                                                          COM00610
      VLOAD=SPHNS(1)+SUM                                                COM00620
C                                                                       COM00630
C  CALL PRECL TO COMPUTE AND STORE THE QUADRATIC FITS NECESSARY         COM00640
C  FOR THE CONE                                                         COM0065
C                                                                       COM00660
      CALL PRECL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,PHI,DHDT,        COM00670
     1           CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT)                     COM00680
      GO TO 999                                                         COM00690
  500 CONTINUE
```

```
      IF(.NOT.FLAG) GO TO 35                                                    COM00700
      READ(IFILE)NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)                    COM00710
      READ(IFILE)DHDT,PHI,CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT,ZINV                COM00720
      READ(IFILE)YLOAD,RCARB1,RCARB2                                            COM00730
      READ(IFILE)ICOUNT                                                         COM00740
      DO 30 J=1,ICOUNT                                                          COM00750
      READ(IFILE) TIMES(J),(RT(I,J),RB(I,J),Z2(I,J),XC0(I,J),                   COM00760
     1            XC1(I,J),I=1,3)                                               COM00770
   30 CONTINUE                                                                  COM00780
      WRITE(IOOUT,800)                                                          COM00790
  800 FORMAT(/,5X,85H ATMOSPHERIC, CHARGE, AND SOIL CHARACTERISTICS USEDCOM00800
     1 WHEN THE CLOUD WAS PRECOMPUTED.       )                                  COM00810
C                                                                               COM00820
CC   WRITE OUT ATMOSPHERIC INFORMATION                                          COM00830
C                                                                               COM00840
      WRITE(IOOUT,808)                                                          COM00850
  808 FORMAT(1X)                                                                COM00860
      WRITE(IOOUT,810)PAS(NATMOS)                                               COM00870
  810 FORMAT(30H  PASQUILL CATEGORY                ,A4)                         COM00880
      NIO=1                                                                     COM00890
      WRITE(IOOUT,812)(ZTEMP(I),TMPMES(I),ZWND(I),WNDMES(I),I=1,NIO)            COM00900
  812 FORMAT(8H     HT ,F8.2,7H  TEMP ,F8.2,7H     HT,F8.2,7H  WIND ,         COM00910
     1 F8.2)                                                                    COM00920
      WRITE(IOOUT,814)THWND                                                     COM00930
  814 FORMAT(22H  WIND DIRECTION          ,F8.2)                                COM00940
      WRITE(IOOUT,819)PHI                                                       COM00950
      IF(DHDT)WRITE(IOOUT,816)                                                  COM00960
  816 FORMAT(40H  THE INVERSION LAYER HEIGHT IS GROWING       )                 COM00970
      IF(.NOT.DHDT)WRITE(IOOUT,818)                                             COM00980
  818 FORMAT(42H  THE INVERSION LAYER HEIGHT IS CONSTANT      )                 COM00990
  819 FORMAT(/22H  LATITUDE                       ,F8.2)                        COM01000
C                                                                               COM01010
CC    WRITE SOIL CHARACTERISTICS                                                COM01020
C                                                                               COM01030
      IF(NSOIL.EQ.1)WRITE(IOOUT,821)                                            COM01040
  820 FORMAT(15H  SILT CONTENT       ,F5.2)                                     COM01050
  821 FORMAT(/,15H  SOIL-1           )                                          COM01060
      IF(NSOIL.EQ.2)WRITE(IOOUT,822)                                            COM01070
  822 FORMAT(/,15H  SOIL-2           )                                          COM01080
      WRITE(IOOUT,820)SILT                                                      COM01090
      WRITE(IOOUT,823)DSOD                                                      COM01100
  823 FORMAT(21H  DEPTH OF SOD            ,F5.2)                                COM01110
C                                                                               COM01120
CC    WRITE EXPLOSIVE CHARGE CHARACTERISTICS                                    COM01130
C                                                                               COM01140
      IF(NCHRG.EQ.1)WRITE(IOOUT,824)                                            COM01150
  824 FORMAT(/,65H  SURFACE - LIVE FIRE OR 30 DEGREE TILTED STATIC, TIP COM01160
     1 ON GROUND          )                                                     COM01170
      IF(NCHRG.EQ.2)WRITE(IOOUT,825)                                            COM01180
  825 FORMAT(/,25H  BARE CHARGE ON SURFACE      )                               COM01190
      IF(NCHRG.EQ.3)WRITE(IOOUT,826)                                            COM01200
  826 FORMAT(/,45H  30 DEGREE TILTED TIP AT 0.3 METER DEPTH        )            COM01210
      IF(NCHRG.EQ.4)WRITE(IOOUT,827)                                            COM01220
  827 FORMAT(/,45H  30 DEGREE TILTED TIP AT 0.6 METER DEPTH        )            COM01230
      IF(NCHRG.EQ.5)WRITE(IOOUT,828)                                            COM01240
  828 FORMAT(/,40H  HORIZONTAL PROJECTILE ON SURFACE                )           COM01250
      WRITE(IOOUT,829)CHWT                                                      COM01260
  829 FORMAT(30H  WEIGHT OF CHARGE              ,F8.2,5H KG. )                  COM01270
      WRITE(IOOUT,830)DETDEP                                                    COM01280
  830 FORMAT(30H  DETONATION DEPTH               ,F8.2)                       COM01290
      FLAG=.FALSE.                                                              COM01300
C                                                                               COM01310
C   COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER              COM01320
C   DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN               COM01330
C   THE WIND DIRECTION.                                                         COM01340
C                                                                               COM01350
   35 CONTINUE
      IF (.NOT.NEWATM) GO TO 45
      THETAX=THWND*PI/180.                                                      COM01360
      TRNFRM(1,1)=COS(THETAX)                                                   COM01370
```

```
      TRNFRM(2,2)=TRNFRM(1,1)                                          COM01380
      TRNFRM(1,2)=SIN(THETAX)                                          COM01390
      TRNFRM(2,1)=-TRNFRM(1,2)                                         COM01400
C                                                                      COM01410
CCC   MAKE THE ORIGIN OF THE LOCAL COORDINAT SYSTEM THE FIRST          COM01420
CCC   CHARGE LOCATION THAT WAS INPUT BY THE USER                       COM01430
C                                                                      COM01440
      DO 40 I=1,2                                                      COM01450
      SRCBAS(I)=COOR(I,1)                                              COM01460
      ORIG(I)=SRCBAS(I)                                                COM01470
   40 CONTINUE                                                         COM01480
C                                                                      COM01490
CCC   CALL SETUP TO COMPUTE THE ARRAY OF DIFFERENCE VECTORS            COM01500
C                                                                      COM01510
      CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)                       COM01520
C                                                                      COM01530
CCC   COMPUTE COORDINATES OF THE TRANSMITTER AND RECEIVER IN THE LOCAL COM01540
CCC   COORDINATE SYSTEM                                                COM01550
C                                                                      COM01560
      TRN(3)=TRNCOR(3)                                                 COM01570
      REC(3)=RECCOR(3)                                                 COM01580
      DO 60 I=1,2                                                      COM01590
      TRN(I)=0.0                                                       COM01600
      REC(I)=0.0                                                       COM01610
      DO 50 J=1,2                                                      COM01620
      TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))                    COM01630
      REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))                    COM01640
   50 CONTINUE                                                         COM01650
   60 CONTINUE                                                         COM01660
45    CONTINUE                                                         COM01670
C                                                                      COM01680
CCC   CALL PRETRN TO COMPUTE THE TRANSMITTANCE ALONG THE SPECIFIED LINE OF COM01690
CCC   SIGHT                                                            COM01700
C                                                                      COM01710
      CALL PRETRN(TRN,REC,TIME,TRNLOS)                                 COM01720
  999 RETURN                                                           COM01730
      END
```

```
      SUBROUTINE CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,      CNL00380
     1XNORM,PLEN)                                                        CNL00390
C                                                                        CNL00010
C  ROUTINE TO FIND THE LENGTH OF A NON-HORIZONTAL LINE THAT INTERSECTS   CNL00020
C  A CONE                                                                CNL00030
C                                                                        CNL00040
C  INPUTS                                                                CNL00050
C                                                                        CNL00060
C     U     - UNIT VECTOR ALONG LINE CONNECTING THE TRANSMITTER AND      CNL00070
C             RECEIVER.                                                  CNL00080
C                                                                        CNL00090
C     TR    - TRANSMITTER COORDINATES                                    CNL00100
C                                                                        CNL00110
C     HTTOP- HEIGHT OF THE TOP OF THE CONE SHAPED PORTION OF THE CLOUD   CNL00120
C                                                                        CNL00130
C     HTBOT- HEIGHT OF THE BOTTOM OF THE CONE SHAPED PORTION OF THE      CNL00140
C             CLOUD                                                      CNL00150
C                                                                        CNL00160
C     XCEN  - X POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP   CNL00170
C                                                                        CNL00180
C     YCEN  - Y POSITION OF THE CENTER OF THE CONE SHAPED CLOUD AT TOP   CNL00190
C                                                                        CNL00200
C     RTOP  - RADIUS OF THE CONE AT THE TOP                              CNL00210
C                                                                        CNL00220
C     RBOT  - RADIUS OF THE CONE AT THE BOTTOM                           CNL00230
C                                                                        CNL00240
C     XB    - X POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD          CNL00250
C                                                                        CNL00260
C     YB    - Y POSITION OF THE BOTTOM OF THE CONE SHAPED CLOUD          CNL00270
C                                                                        CNL00280
C  OUTPUT                                                                CNL00290
C                                                                        CNL00300
C     PLEN - LENGTH OF THE INTERSECTION OF CONE AND THE LINE OF SIGHT    CNL00310
C                                                                        CNL00320
C  FUNTIONS AND SUBROUTINES                                              CNL00330
C                                                                        CNL00340
C     NONE                                                               CNL00350
C                                                                        CNL00360
C**********************************************************************  CNL00370
      DIMENSION U(3),TR(3)                                               CNL00400
      IF(U(3).LT.0.0)GO TO 40                                            CNL00410
C                                                                        CNL00420
C  SET UP BOUNDS SO INTERSECTION OF LINE IS SUCH THAT HTBOT < Z < HTTOP  CNL00430
C                                                                        CNL00440
      PMIN=(HTBOT-TR(3))/U(3)                                            CNL00450
      PMAX=(HTTOP-TR(3))/U(3)                                            CNL00460
      GO TO 50                                                           CNL00470
   40 PMIN=(HTTOP-TR(3))/U(3)                                            CNL00480
      PMAX=(HTBOT-TR(3))/U(3)                                            CNL00490
   50 P1=(HTTOP-HTBOT)/U(3)                                              CNL00500
      P0=(HTBOT-TR(3))/U(3)                                              CNL00510
C                                                                        CNL00520
C  SET UP QUADRATIC TO BE SOLVED                                         CNL00530
C                                                                        CNL00540
      DX1=U(1)*P1-XCEN+XB                                                CNL00550
      DY1=U(2)*P1-YCEN+YB                                                CNL00560
      DR=RTOP-RBOT                                                       CNL00570
      A=DX1**2+DY1**2-DR**2                                              CNL00580
      DX0=TR(1)+U(1)*P0-XB                                               CNL00590
      DY0=TR(2)+U(2)*P0-YB                                               CNL00600
      B=2.*(DX1*DX0+DY1*DY0-DR*RBOT)                                     CNL00610
      C=(DX0**2+DY0**2-RBOT**2)                                          CNL00620
      RADIC1=(DX1*RBOT-DX0*DR)**2                                        CNL00630
      RADIC2=(DY1*RBOT-DY0*DR)**2                                        CNL00640
      RADIC3=(DX1*DY0-DY1*DX0)**2                                        CNL00650
C                                                                        CNL00660
C  DETERMINE PATH LENGTH IF THE LINE INTERSECTS THE CONE                 CNL00670
C                                                                        CNL00680
      IF(ABS(A).LT.1.E-20)GO TO 60                                       CNL00690
      RADIC=RADIC1+RADIC2-RADIC3                                         CNL00700
```

```
      IF(RADIC.LT.0.0)GO TO 68                                          CNL00710
      ROOT1=-B/A/2.+SQRT(RADIC)/A                                       CNL00720
      ROOT2=-B/A/2.-SQRT(RADIC)/A                                       CNL00730
C                                                                       CNL00740
CC  DETERMINE WHICH POINT IS CLOSEST AND WHICH IS FARTHEST FROM THE     CNL00750
C   TRANSMITTER                                                         CNL00760
C                                                                       CNL00770
      TEMP1=P1*ROOT1+P0                                                 CNL00780
      TEMP2=P1*ROOT2+P0                                                 CNL00790
      IF (A.LT.0.0) GO TO 57                                            CNL00800
      IF(TEMP1.GT.TEMP2)GO TO 55                                        CNL00810
      PFAR=TEMP2                                                        CNL00820
      PNEAR=TEMP1                                                       CNL00830
      GO TO 67                                                          CNL00840
   55 PFAR=TEMP1                                                        CNL00850
      PNEAR=TEMP2                                                       CNL00860
      GO TO 67                                                          CNL00870
57    IF (U(3).GT.0.0) GO TO 58                                         CNL00880
      PNEAR=-1.E20                                                      CNL00890
      PFAR=AMIN1(TEMP1,TEMP2)                                           CNL00900
      GO TO 67                                                          CNL00910
58    PNEAR=AMAX1(TEMP1,TEMP2)                                          CNL00920
      PFAR=1.E20                                                        CNL00930
      GO TO 67                                                          CNL00940
C                                                                       CNL00950
CC  CONSIDER DEGENERATE CASE                                            CNL00960
C                                                                       CNL00970
   60 IF(B.LT.1.E-20)GO TO 68                                           CNL00980
      IF(P1/B.LT.0.0)GO TO 65                                           CNL00990
      PFAR=P1*(-C/B)+P0                                                 CNL01000
      PNEAR=0.0                                                         CNL01010
      GO TO 67                                                          CNL01020
   65 PFAR=XNORM                                                        CNL01030
      PNEAR=P1*(-C/B)+P0                                                CNL01040
   67 PLEN=AMIN1(PFAR,XNORM,PMAX)-AMAX1(PNEAR,0.0,PMIN)                 CNL01050
      IF(PLEN.LT.0.0)PLEN=0.0                                           CNL01060
      GO TO 999                                                         CNL01070
   68 PLEN=0.0                                                          CNL01080
  999 RETURN                                                            CNL01090
      END                                                               CNL01100
```

```
      SUBROUTINE CONVRT(T)                                                  COV00010
      LOGICAL SWITCH,CHANGE                                                 COV00020
      REAL KZ,KX,N,MDIF                                                     COV00030
      COMMON/PRTINF/R0,VGRAV(3),NPRTS                                       COV00040
      COMMON/BUOYCL/RSPH,DELT,VZSPH,XCMSPH,YCMSPH,ZCMSPH,XTOP,YTOP,         COV00050
     1SPHNS(3),RISTIM                                                       COV00060
      COMMON/CLOCK/TIME,TWIND                                               COV00070
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE              COV00080
      COMMON/SIG/SIG02,SIGC                                                 COV00090
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCOV00100
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)                   COV00110
      COMMON/LOAD/WAKAL,SPHAL                                               COV00120
      COMMON/WNDPRM/DXZ0,DYZ0,DZ0,U0,M,NDIF,ZINV                            COV00130
      COMMON/STARS/USTAR,TSTAR,ZSTAR                                        COV00140
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                                    COV00150
      COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAVRG              COV00160
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                        COV00170
C ***************************************************************         COV00180
C                                                                           COV00190
C  PURPOSE                                                                  COV00200
C                                                                           COV00210
C     TO CONVERT THE CURRENT BUOYANT CLOUD INTO A THREE DIMENSIONAL         COV00220
C     GAUSSIAN PUFF TO BE USED BY THE WIND DISPERSION MODEL.                COV00230
C                                                                           COV00240
C  INPUT                                                                    COV00250
C                                                                           COV00260
C    T   TIME IN SECONDS AFTER DETONATION                                   COV00270
C                                                                           COV00280
C                                                                           COV00290
C  CALLED BY RISE                                                           COV00300
C                                                                           COV00310
C     FUNCTIONS AND SUBROUTINES NEEDED                                      COV00320
C                                                                           COV00330
C       WNDCAL                                                              COV00340
C       AMOUNT                                                              COV00350
C       AVRG                                                                COV00360
C                                                                           COV00370
C***************************************************************          COV00380
      CHANGE=.TRUE.                                                         COV00390
      TWIND=T                                                               COV00400
      VTR=USTAR*WNDCAL(Z0,ZL,ZCMSPH)                                        COV00410
      KZ=DIFFUS(Z0,ZL,ZCMSPH)                                               COV00420
      KX=KZ*DXZ0                                                            
      TTR=T                                                                 COV00430
      XTR=XCMSPH                                                            COV00440
      ZTR=ZCMSPH                                                            COV00450
      DO 10 IPRTS=1,NPRTS                                                   COV00460
      QPUFF(IPRTS)=SPHNS(IPRTS)                                             COV00470
   10 CONTINUE                                                              COV00480
      VOLSPH=(4./3.)*PI*RSPH**3                                             COV00490
      CALL AMOUNT(VOLSPH,WAKAL,SPHAL)                                       COV00500
      QPUFF(1)=SPHAL                                                        COV00510
   60 SIG02=((2./3.)*RSPH)**2                                               COV00520
      SIG0=SQRT(SIG02)                                                      COV00530
      ZX=5.0                                                                COV00540
   61 CALL AVRG(ZX,T,QTOT,XBAVRG SIG2X,SIG2Y)                               COV00550
      SIGX=SQRT(SIG2X)                                                      COV00560
      SIGY=SQRT(SIG2Y)                                                      COV00570
C                                                                           COV00580
C  COMPUTE PARAMETERS NEEDED FOR LOCAL AND COLUMN DENISTY FOR THE WAKE      COV00590
C  SOLUTION                                                                 COV00600
C                                                                           COV00610
      SIGC=(SQRT(SIGX*SIGY)+SQRT(SIG02))/2.                                 COV00620
      XDIF=XCMSPH-XBAVRG                                                    COV00630
      YDIF=YCMSPH                                                           COV00640
      ZDIF=ZCMSPH-5.0                                                       COV00650
      TDIF=SIGC**2/KX/2.                                                    COV00660
      TDX=2.*KX                                                             COV00670
      TDZ=2.*KZ                                                             COV00680
      QLOC=WAKAL/4./PI/SQRT(2.)                                             COV00690
```

```
      QCOL=WAKAL/4./SQRT(PI)                                        COV00700
  998 RETURN                                                        COV00710
      END                                                           COV00720
```

```
      FUNCTION CSPHER(X,Y,Z,T)                                          CSP00230
C                                                                       CSP00010
C  COMPUTE EITHER THE COLUMN DENSITY FOR A GIVEN LINE OF SIGHT OR       CSP00020
C  COMPUTE THE LOCAL CONCENTRATION AT X,Y,Z FOR THE SPHERE              CSP00030
C                                                                       CSP00040
C  INPUT                                                                CSP00050
C                                                                       CSP00060
C      X,Y,Z     COORDINATES IN METERS.  IF LINE INTEGRAL IS DESIRED,   CSP00070
C                Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.         CSP00080
C                                                                       CSP00090
C      T         THE TIME IN SECONDS AFTER DETONATON                   CSP00100
C                                                                       CSP00110
C  OUTPUT                                                               CSP00120
C                                                                       CSP00130
C     RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND      CSP00140
C     THE LINE INTEGRAL OF CONCENTRATION (COLUMN DENSITY) IF HORIZ      CSP00150
C     IS .TRUE.                                                         CSP00160
C                                                                       CSP00170
C  FUNTIONS AND SUBROUTINES CALLED                                      CSP00180
C                                                                       CSP00190
C      NONE                                                             CSP00200
C                                                                       CSP00210
C***********************************************************************CSP00220
      REAL M,N,KZ,KX                                                    CSP00240
      LOGICAL HORIZ,SWITCH,CHANGE,TEST                                  CSP00250
      COMMON/PRTINF/ R0,VGRAV(3),NPRTS                                  CSP00260
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)              CSP00270
      COMMON /MODE/ HORIZ                                               CSP00280
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV                          CSP00290
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),          CSP00300
     1 QDSC(20,3)                                                       CSP00310
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,          CSP00320
     +           COOR(2,200),TSTAG(200),DMMY(401)                       CSP00330
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE          CSP00340
      COMMON/SIG/SIG02,SIGC                                             CSP00350
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCSP00360
      COMMON/LOAD/WAKAL,SPHAL                                           CSP00370
      COMMON/ACL/CWINDS,CWINDC,CWINDW                                   CSP00380
      COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAVRG          CSP00390
      COMMON/LOS/TR(3),RE(3),U(3)                                       CSP00400
      COMMON/CHARGE/NCHG                                                CSP00410
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL                               CSP00420
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     CSP00430
      CSPHER=0.0                                                        CSP00440
      CWNDSC=0.0                                                        CSP00450
      CWINDC=0.0                                                        CSP00460
      IF(NARY.EQ.3)GO TO 999                                            CSP00470
      IF((T-TTR).LT.1.E-20)GO TO 999                                    CSP00480
C                                                                       CSP00490
C  COMPUTE CONTRIBUTION FROM BUOYANT CLOUD AFTER SWITCHING TO THE       CSP00500
C  WIND MODEL USING A THREE DIMENSIONAL GAUSSIAN PUFF                   CSP00510
C                                                                       CSP00520
      SIGX2=SIG02+2.*KX*(T-TTR)                                         CSP00530
      SIGZ2=SIG02+2.*KZ*(T-TTR)                                         CSP00540
      SIGX=SQRT(SIGX2)                                                  CSP00550
      SIGZ=SQRT(SIGZ2)                                                  CSP00560
      ARG2=(Z-ZTR)**2/(2.*SIGZ2)                                        CSP00570
      IF(ABS(ARG2).GT.30.)GO TO 25                                      CSP00580
      TERM2=EXP(-ARG2)                                                  CSP00590
      GO TO 26                                                          CSP00600
   25 TERM2=0.0                                                         CSP00610
   26 ARG3=(Z+ZTR)**2/(2.*SIGZ2)                                        CSP00620
      IF(ABS(ARG3).GT.30.)GO TO 27                                      CSP00630
      TERM3=EXP(-ARG3)                                                  CSP00640
      GO TO 28                                                          CSP00650
   27 TERM3=0.0                                                         CSP00660
   28 IF(HORIZ)GO TO 50                                                 CSP00670
C                                                                       CSP00680
C  COMPUTE CONCENTRATION AT X,Y,Z                                       CSP00690
C                                                                       CSP00700
```

```
      XC=XTR+VTR*(T-TTR)+DIFF(1,NCHG)                                   CSP00710
      YC=DIFF(2,NCHG)                                                   CSP00720
      ARG1=((X-XC)**2+(Y-YC)**2)/2./SIGX2                               CSP00730
      IF(ABS(ARG1).GT.30.)GO TO 999                                     CSP00740
      TERM=SPHAL/((2.*PI)**(3./2.))/SIGZ/SIGX2                          CSP00750
      CWINDC=TERM*EXP(-ARG1)*(TERM2+TERM3)                              CSP00760
      GO TO 100                                                         CSP00770
C                                                                       CSP00780
C  COMPUTE COLUMN DENSITY                                               CSP00790
C                                                                       CSP00800
   50 TERM=SPHAL/2./PI/SIGX/SIGZ                                        CSP00810
      DO 90 I=1,NCHTOT                                                  CSP00820
      XC=XTR+VTR*(T-TTR)+DIFF(1,I)                                      CSP00830
      YC=DIFF(2,I)                                                      CSP00840
      XY=XC*SCRN(1)+YC*SCRN(2)                                          CSP00850
      ARG1=(X-XY)**2/2./SIGX2                                           CSP00860
      IF(ABS(ARG1).GT.30.)GO TO 90                                      CSP00870
      CWNDSC=TERM*EXP(-ARG1)*(TERM2+TERM3)                              CSP00880
      CWINDC=CWINDC+CWNDSC                                              CSP00890
      CALL TRNCHK(CWINDS,CWINDW,CWINDC)                                 CSP00900
      IF(TEST)GO TO 999                                                 CSP00910
   90 CONTINUE                                                          CSP00920
  100  CSPHER=CWINDC                                                    CSP00930
  999 RETURN                                                            CSP00940
      END                                                               CSP00950
```

```
      FUNCTION CWAKE(X,Y,Z,T)                                                  CWA00290
C                                                                              CWA00010
CCC   FUNCTION TO COMPUTE THE LOCAL CONCENTRATION OR COLUMN DENSITY AT         CWA00020
CCC   X,Y,Z AND TIME T AFTER THE BLAST FOR THE WAKE.                           CWA00030
CCC                                                                            CWA00040
CCC                                                                            CWA00050
CCC   INPUT                                                                    CWA00060
CCC                                                                            CWA00070
CCC      X,Y,Z     COORDINATES IN METERS. IF THE LINE INTEGRAL IS DESIRED     CWA00080
CCC                THESE ARE NOT USED AND THE LINE IS SPECIFIED BY THE        CWA00090
CCC                TRANSMITTER AND RECEIVER COORDINATES AND INFORMATION       CWA00100
CCC                CALCULATED AT THE TIME THE BUOYANT FIREBALL CONVERTED      CWA00110
CCC                TO THE WIND MODEL.                                         CWA00120
CCC                                                                           CWA00130
CCC      T         THE TIME IN SECONDS AFTER DETONATION                       CWA00140
CCC                                                                           CWA00150
CCC                                                                           CWA00160
CCC   OUTPUT                                                                  CWA00170
CCC                                                                           CWA00180
CCC      RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND         CWA00190
CCC      THE LINE INTEGRAL OF CONCENTRATION (COLUMN DENSITY) IF               CWA00200
CCC      HORIZ IS .TRUE.                                                      CWA00210
CCC                                                                           CWA00220
CCC                                                                           CWA00230
CCC   FUNCTIONS AND SUBROUTINES NEEDED                                        CWA00240
CCC                                                                           CWA00250
CCC      ERF     COMPUTE THE ERROR FUNCTION                                   CWA00260
CCC                                                                           CWA00270
C*************************************************************************** CWA00280
      REAL M,N,KZ,KX                                                          CWA00300
      LOGICAL HORIZ,SWITCH,CHANGE,TEST                                        CWA00310
      COMMON/PRTINF/ R0,VGRAV(3),NPRTS                                        CWA00320
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)                    CWA00330
      COMMON /MODE/ HORIZ                                                     CWA00340
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV                                CWA00350
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),                CWA00360
     1 QDSC(20,3)                                                             CWA00370
      COMMON/MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,                CWA00380
     +           COOR(2,200),TSTAG(200),DMMY(401)                             CWA00390
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE                CWA00400
      COMMON/SIG/SIG02,SIGC                                                   CWA00410
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCWA00420
      COMMON/LOAD/WAKAL,SPHAL                                                 CWA00430
      COMMON/ACL/CWINDS,CWINDC,CWINDW                                         CWA00440
      COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAVRG                CWA00450
      COMMON/LOS/TR(3),RE(3),U(3)                                             CWA00460
      COMMON/CHARGE/NCHG                                                      CWA00470
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL                                     CWA00480
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                           CWA00490
      IF(NARY.EQ.3)GO TO 999                                                  CWA00500
      IF((T-TTR).LT.1.E-20)GO TO 999                                          CWA00510
C                                                                             CWA00520
CCC   COMPUTE CONTRIBUTION FROM THE WAKE AFTER SWITCHING TO THE WIND MODEL    CWA00530
CCC   FOR A SINGLE CHARGE                                                     CWA00540
C                                                                             CWA00550
      CWAKE=0.0                                                               CWA00560
      CWINDW=0.0                                                              CWA00570
      CWNDSW=0.0                                                              CWA00580
      SIGX2=TDX*(TDIF+(T-TTR))                                                CWA00590
      SIGZ2=TDZ*(T-TTR)                                                       CWA00600
      IF(HORIZ)GO TO 210                                                      CWA00610
      XB=XBAVRG+DIFF(1,NCHG)                                                  CWA00620
      YB=DIFF(2,NCHG)                                                         CWA00630
C                                                                             CWA00640
CCC   COMPUTE THE LOCAL CONCENTRATION                                         CWA00650
C                                                                             CWA00660
      A=-(XDIF**2+YDIF**2)/2./SIGX2-(ZDIF**2/2./SIGZ2)                        CWA00670
      XX=X-VTR*(T-TTR)                                                        CWA00680
      B0=(XDIF*(XX-XB)+YDIF*(Y-YB))/SIGX2                                     CWA00690
      B1=B0+(ZDIF*(Z-5.)/SIGZ2)                                               CWA00700
```

```
      B2=B0-(ZDIF*(Z+5.)/SIGZ2)                                          CWA00710
      C0=-((XX-XB)**2+(Y-YB)**2)/2./SIGX2                                CWA00720
      C1=C0-((Z-5.)**2/2./SIGZ2)                                         CWA00730
      C2=C0-((Z+5.)**2/2./SIGZ2)                                         CWA00740
      RTMA=SQRT(-A)                                                      CWA00750
      ARG1=C1-(B1**2/4./A)                                               CWA00760
      ARG2=-B1/2./RTMA                                                   CWA00770
      ARG3=RTMA+ARG2                                                     CWA00780
      ARG4=C2-(B2**2/4./A)                                               CWA00790
      ARG5=-B2/2./RTMA                                                   CWA00800
      ARG6=RTMA+ARG5                                                     CWA00810
      IF(ARG1.LT.-30.)GO TO 221                                          CWA00820
      TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))                              CWA00830
      GO TO 222                                                          CWA00840
  221 TERM1=0.0                                                          CWA00850
  222 CONTINUE                                                           CWA00860
      IF(ARG4.LT.-30.)GO TO 223                                          CWA00870
      TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))                              CWA00880
      GO TO 224                                                          CWA00890
  223 TERM2=0.0                                                          CWA00900
  224 CWAKE=QLOC/SIGX2/SQRT(SIGZ2)/RTMA*(TERM1+TERM2)                    CWA00910
      GO TO 999                                                          CWA00920
C                                                                        CWA00930
C  COMPUTE COLUMN DENSITY                                                CWA00940
C                                                                        CWA00950
  210 DO 245 J=1,NCHTOT                                                  CWA00960
      XB=XBAVRG+DIFF(1,J)                                                CWA00970
      YB=DIFF(2,J)                                                       CWA00980
  230 A=-(XDIF*U(2)-YDIF*U(1))**2/2./SIGX2-(ZDIF)**2/2./SIGZ2            CWA00990
      TRR=TR(1)-VTR*(T-TTR)                                              CWA01000
      B0=(YDIF*U(1)-XDIF*U(2))*((TR(2)-YB)*U(1)-(TRR-XB)*U(2))/SIGX2     CWA01010
      B1=B0+ZDIF*(TR(3)-5.)/SIGZ2                                        CWA01020
      B2=B0-ZDIF*(TR(3)+5.)/SIGZ2                                        CWA01030
      C0=-((TR(2)-YB)*U(1)-(TRR-XB)*U(2))**2/2./SIGX2                    CWA01040
      C1=C0-(TR(3)-5.)**2/2./SIGZ2                                       CWA01050
      C2=C0-(TR(3)+5.)**2/2./SIGZ2                                       CWA01060
      RTMA=SQRT(-A)                                                      CWA01070
      ARG1=C1-B1**2/4./A                                                 CWA01080
      ARG2=-B1/2./RTMA                                                   CWA01090
      ARG3=RTMA+ARG2                                                     CWA01100
      ARG4=C2-B2**2/4./A                                                 CWA01110
      ARG5=-B2/2./RTMA                                                   CWA01120
      ARG6=RTMA+ARG5                                                     CWA01130
      IF(ARG1.GT.30.)ARG1=30.                                            CWA01140
      IF(ARG1.LT.-30.)GO TO 231                                          CWA01150
      TERM1=EXP(ARG1)*(ERF(ARG3)-ERF(ARG2))                              CWA01160
      GO TO 232                                                          CWA01170
  231 TERM1=0.0                                                          CWA01180
  232 CONTINUE                                                           CWA01190
      IF(ARG4.LT.-30.)GO TO 233                                          CWA01200
      TERM2=EXP(ARG4)*(ERF(ARG6)-ERF(ARG5))                              CWA01210
      GO TO 234                                                          CWA01220
  233 TERM2=0.0                                                          CWA01230
  234 ARG=SIGX2*SIGZ2                                                    CWA01240
      CWNDSW=(QCOL/SQRT(ARG)/RTMA)*(TERM1+TERM2)                         CWA01250
  240 CONTINUE                                                           CWA01260
      CWINDW=CWINDW+CWNDSW                                               CWA01270
      CALLTRNCHK(CWINDS,CWINDW,CWINDC)                                   CWA01280
      IF(TEST)GO TO 999                                                  CWA01290
  245 CONTINUE                                                           CWA01300
      CWAKE=CWINDW                                                       CWA01310
  999 RETURN                                                             CWA01320
      END                                                                CWA01330
```

```
      FUNCTION CWIND(X,Y,Z,T)                                                CWI00270
C                                                                            CWI00010
C     PURPOSE                                                                CWI00020
C                                                                            CWI00030
C       TO COMPUTE THE CONCENTRATION AT A POINT OR INTEGRATED ALONG          CWI00040
C       A HORIZONTAL LINE  (SKIRT PORTION OF DUST CLOUD)                     CWI00050
C                                                                            CWI00060
C     INPUT                                                                  CWI00070
C                                                                            CWI00080
C       X,Y,Z     COORDINATES IN METERS.  IF LINE INTEGRAL IS DESIRED,       CWI00090
C                 Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.             CWI00100
C                                                                            CWI00110
C       T         THE TIME IN SECONDS AFTER DETONATION                       CWI00120
C                                                                            CWI00130
C     OUTPUT                                                                 CWI00140
C                                                                            CWI00150
C       RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND         CWI00160
C       THE LINE INTEGRAL OF CONCENTRATION IF HORIZ IS .TRUE.                CWI00170
C                                                                            CWI00180
C     SUBROUTINES CALLED                                                     CWI00190
C                                                                            CWI00200
C       MOMENT    COMPUTES ZERO ORDER MOMENT AND INTERPOLATES FROM           CWI00210
C                 TABLE OF HIGHER ORDER MOMENTS.                             CWI00220
C                                                                            CWI00230
C       CALLED BY FUNCT,TRNCAL                                               CWI00240
C                                                                            CWI00250
C*************************************************************************** CWI00260
      REAL M,N,KZ,KX                                                         CWI00280
      LOGICAL HORIZ,SWITCH,CHANGE,TEST,SKIP                                  CWI00290
      DIMENSION REF(2),REF0(2)                                               CWI00300
      COMMON/PRTINF/ R0,VGRAV(3),NPRTS                                       CWI00310
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)                   CWI00320
      COMMON /MODE/ HORIZ                                                    CWI00330
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV                               CWI00340
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),               CWI00350
     1 QDSC(20,3)                                                            CWI00360
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,               CWI00370
     +           COOR(2,200),TSTAG(200),DMMY(401)                            CWI00380
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE               CWI00390
      COMMON/SIG/SIG02,SIGC                                                  CWI00400
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCWI00410
      COMMON/LOAD/WAKAL,SPHAL                                                CWI00420
      COMMON/ACL/CWINDS,CWINDC,CWINDW                                        CWI00430
      COMMON/WAKE/XDIF,YDIF,ZDIF,TDIF,TDX,TDZ,QLOC,QCOL,XBAVRG               CWI00440
      COMMON/LOS/TR(3),RE(3),U(3)                                            CWI00450
      COMMON/CHARGE/NCHG                                                     CWI00460
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL                                    CWI00470
      COMMON/SKIPIT/SKIP                                                     CWI00480
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                          CWI00490
C                                                                            CWI00500
C     COMMON /PRTINF/                                                        CWI00510
C                                                                            CWI00520
C       R0        INITIAL RADIUS OF THE CLOUD IN METERS                      CWI00530
C       VGRAV     SINGLY DIMENSIONED ARRAY.  VGRAV(I) IS THE OPTICALLY       CWI00540
C                 WEIGHTED AVERAGE SETTLING VELOCITY FOR PARTICLES IN THE    CWI00550
C                 I SIZE RANGE                                               CWI00560
C       NPRTS     THE NUMBER OF PARTICLE SIZE RANGES                         CWI00570
C                                                                            CWI00580
C     COMMON /DISCS/                                                         CWI00590
C                                                                            CWI00600
C       NDSCS     THE NUMBER OF DISC SOURCES                                 CWI00610
C       TDSC      SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF RELEASE    CWI00620
C                 OF THE DISC SOURCES                                        CWI00630
C       XDSC      SINGLY DIMENSIONED ARRAY CONTAINING THE X COORDINATE       CWI00640
C                 OF THE CENTER OF THE DISC SOURCES                          CWI00650
C       ZDSC      SINGLY DIMENSIONED ARRAY CONTAINING THE Z COORDINATE       CWI00660
C                 OF THE CENTER OF THE DISC SOURCES                          CWI00670
C       R2DSC     SINGLY DIMENSIONED ARRAY CONTAINING THE SQUARE OF THE      CWI00680
C                 RADII OF THE DISC SOURCES                                  CWI00690
C       QDSC      DOUBLY DIMENSIONED ARRAY.   QDSC(I,J) IS THE OPTICALLY     CWI00700
```

```
C              WEIGHTED MASS OF PARTICLES OF THE J SIZE RANGE IN THE      CWI00710
CC             I DISC.                                                   CWI00720
C                                                                        CWI00730
C                                                                        CWI00740
C      SUM THE CONTRIBUTIONS OF THE DISC SOURCES TO THE                  CWI00750
C      OPTICALLY WEIGHTED CONCENTRATION AT (X,Y,Z,T)                     CWI00760
C                                                                        CWI00770
       CWIND=0.                                                          CWI00780
       CWNDSC=0.0                                                        CWI00790
       CWINDS=0.0                                                        CWI00800
       CWINDC=0.0                                                        CWI00810
       CWINDW=0.0                                                        CWI00820
       IF(HORIZ)GO TO 110                                                CWI00830
C                                                                        CWI00840
CC  COMPUTE CONCENTRATION AT X,Y,Z  (FOR SIMULTANEOUS BURST)             CWI00850
C                                                                        CWI00860
       DO 100 I=1,NDSCS                                                  CWI00870
       REF0(1)=XDSC(I)                                                   CWI00880
       REF0(2)=0.0                                                       CWI00890
       ROH2=R2DSC(I)                                                     CWI00900
       H=ZDSC(I)                                                         CWI00910
       TOF=T-TDSC(I)                                                     CWI00920
       DO 90 J=1,NPRTS                                                   CWI00930
C                                                                        CWI00940
CC  DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z                       CWI00950
C                                                                        CWI00960
       CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)                  CWI00970
       IF(Q.GT.1.E-10)GO TO 50                                           CWI00980
       CWNDSC=0.0                                                        CWI00990
       GO TO 100                                                         CWI01000
    50 CONTINUE                                                          CWI01010
       RX2=ROH2+2.*SIGW2                                                 CWI01020
       RY2=ROH2+2.*SIGP2                                                 CWI01030
       DO 114 NA=1,2                                                     CWI01040
       REF(NA)=REF0(NA)+DIFF(NA,NCHG)                                    CWI01050
   114 CONTINUE                                                          CWI01060
       ARG=-(X-REF(1)-XBAR)**2/RX2                                       CWI01070
       IF(ABS(ARG).GT.30.)GO TO 100                                      CWI01080
       CWNDSC=(Q/RTPI/SQRT(RX2))*EXP(ARG)                                CWI01090
       ARG=-(Y-REF(2))**2/RY2                                            CWI01100
       IF(ABS(ARG).GT.30.)GO TO 100                                      CWI01110
       CY=EXP(ARG          )/RTPI/SQRT(RY2)                              CWI01120
       CWNDSC=QDSC(I,J)*CWNDSC*CY                                        CWI01130
       CWINDS=CWINDS+CWNDSC                                              CWI01140
    90 CONTINUE                                                          CWI01150
   100 CONTINUE                                                          CWI01160
       CWIND=CWINDS                                                      CWI01170
       GO TO 999                                                         CWI01180
   110 DO 220 ICHG=1,NTOT                                                CWI01190
       IF(T.LT.TSTAG(ICHG))GO TO 220                                     CWI01200
       DO 211 I=1,NDSCS                                                  CWI01210
       TOF=T-TDSC(I)-TSTAG(ICHG)                                         CWI01220
       REF0(1)=XDSC(I)                                                   CWI01230
       REF0(2)=0.                                                        CWI01240
       ROH2=R2DSC(I)                                                     CWI01250
       H=ZDSC(I)                                                         CWI01260
       IF(HORIZ) REF0(1)=REF0(1)*SINTH                                   CWI01270
       DO 210 J=1,NPRTS                                                  CWI01280
       CWNDSC=0.0                                                        CWI01290
C                                                                        CWI01300
CC     DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z                    CWI01310
C                                                                        CWI01320
       CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)                  CWI01330
       IF(Q.GT.1.E-10) GO TO 113                                         CWI01340
C                                                                        CWI01350
CC     IF Q IS TOO SMALL, ITS CONTRIBUTION IS IGNORED                    CWI01360
C                                                                        CWI01370
       CWNDSC=0.                                                         CWI01380
       GO TO 210                                                         CWI01390
   113 CONTINUE                                                          CWI01400
```

```
      DO 200 MA=1,NCHTOT                                                CWI01410
      INDX=MA                                                           CWI01420
      IF(NARY.EQ.3)INDX=ICHG                                            CWI01430
      RX2=R0H2+2.*SIGW2                                                 CWI01440
      RY2=R0H2+2.*SIGP2                                                 CWI01450
C                                                                       CWI01460
C     COMPUTE CONCENTRATION ALONG LINE OF SIGHT SPECIFIED BY X,Z        CWI01470
C                                                                       CWI01480
  120 CONTINUE                                                          CWI01490
      REF(1)=REF0(1)+PRSEP(INDX)                                        CWI01500
      REFF2=RX2*SINTH2+RY2*COSTH2                                       CWI01510
      ARG=-(X-REF(1)-XBAR*SINTH)**2/REFF2                               CWI01520
      IF(ABS(ARG).GT.30.)GO TO 150                                      CWI01530
      CWNDSC=EXP(ARG)                /SQRT(REFF2)/RTPI                  CWI01540
      CWNDSC=CWNDSC*Q*QDSC(I,J)                                         CWI01550
  150 CONTINUE                                                          CWI01560
      CWINDS=CWINDS+CWNDSC                                              CWI01570
      IF(SKIP)GO TO 190                                                 CWI01580
      CALL TRNCHK(CWINDS,CWINDW,CWINDC)                                 CWI01590
      IF(TEST)GO TO 999                                                 CWI01600
  190 CONTINUE                                                          CWI01610
  200 CONTINUE                                                          CWI01620
  210 CONTINUE                                                          CWI01630
  211 CONTINUE                                                          CWI01640
  220 CONTINUE                                                          CWI01650
      CWIND=CWINDS                                                      CWI01660
  999 RETURN                                                            CWI01670
      END                                                               CWI01680
```

```
      SUBROUTINE DIFEQ(N,T,Y,YP)                                        DFQ00010
      REAL KM,KZ,KX                                                     DFQ00020
      LOGICAL SWITCH,CHANGE                                             DFQ00030
      DIMENSION Y(N),YP(N)                                              DFQ00040
      COMMON/PRTINF/ROCL,VGRAV(3),NPRTS                                 DFQ00050
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV                         DFQ00060
      COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF                     DFQ00070
      COMMON/MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,          DFQ00080
     +           DMM(600),DMMY(401)                                     DFQ00090
      COMMON /BURST/ ACCEL,TBURST                                       DFQ00100
      COMMON/STARS/USTAR,TSTAR,ZSTAR                                    DFQ00110
      COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT                             DFQ00120
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                                DFQ00130
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE          DFQ00140
      COMMON/SIG/SIG02,SIGC                                             DFQ00150
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUDFQ00160
      DATA ALPHAK/.25/                                                  DFQ00170
C*********************************************************************  DFQ00180
C                                                                       DFQ00190
C                                                                       DFQ00200
C     PURPOSE                                                           DFQ00210
C                                                                       DFQ00220
C     DIFEQ CONTAINS THE PARTIAL DIFFERENTIAL EQUATIONS FOR THE         DFQ00230
C     RISE OF A BUOYANT CLOUD WHICH ARE USED BY SUBROUTINE RKM.         DFQ00240
C                                                                       DFQ00250
C                                                                       DFQ00260
C     INPUT                                                             DFQ00270
C                                                                       DFQ00280
C     N             THE NUMBER OF DEPENDENT VARIABLES                   DFQ00290
C     T             THE INDEPENDENT VARIABLE, I.E. TIME                 DFQ00300
C    Y(1)           RADIUS OF CLOUD                                     DFQ00310
C    Y(2)           CLOUD TEMPERATURE MINUS SURROUNDING TEMPERATURE     DFQ00320
C    Y(3)           VERTICAL VELOCITY OF CLOUD                          DFQ00330
C    Y(4)           X-COORDINATE OF CENTER OF MASS FOR THE CLOUD        DFQ00340
C    Y(5)           Y-COORDINATE OF CENTER OF MASS FOR THE CLOUD        DFQ00350
C    Y(6)           THE HEIGHT OF THE CLOUD C.O.M.                      DFQ00360
C    Y(7)           X-COORDINATE OF TOP OF CLOUD                        DFQ00370
C    Y(8)           Y-COORDINATE OF TOP OF CLOUD                        DFQ00380
C                                                                       DFQ00390
C                                                                       DFQ00400
C     OUTPUT                                                            DFQ00410
C                                                                       DFQ00420
C     YP    AN ARRAY CONTAINING COMPUTED DERIVATIVES OF THE DEPENDENT   DFQ00430
C           VARIABLES WITH RESPECT TO THE INDEPENDENT VARIABLE.         DFQ00440
C                                                                       DFQ00450
C                                                                       DFQ00460
C                                                                       DFQ00470
C     REQUIRED FUNCTIONS                                                DFQ00480
C                                                                       DFQ00490
C       TEMP     CALCULATES AMBIENT ATMOSPHERIC TEMPERATURE AND THE     DFQ00500
C                TEMPERATURE GRADIENT AT CLOUD HEIGHT.                  DFQ00510
C                                                                       DFQ00520
C       WIN      CALCULATES THE WIND SPEED IN THE X AND Y DIRECTION AT  DFQ00530
C                CLOUD HEIGHT.                                          DFQ00540
C                                                                       DFQ00550
C       DIFFUS   COMPUTES THE DIFFUSIVITY AT A SPECIFIED HEIGHT.        DFQ00560
C                                                                       DFQ00570
C     CALLED BY RKM                                                     DFQ00580
C                                                                       DFQ00590
C*********************************************************************  DFQ00600
      IF(T.LT.TBURST)GO TO 200                                          DFQ00610
C                                                                       DFQ00620
      IF(Y(6).GT.ZSTAR)GO TO 5                                          DFQ00630
      CALL TEMP(Y(6),TA,DTADZ)                                          DFQ00640
      GO TO 6                                                           DFQ00650
    5 TA=TC1+TC2*Y(6)+TC3*Y(6)**2                                       DFQ00660
      DTADZ=TC2+2.*TC3*Y(6)                                             DFQ00670
    6 CALL WIN(Y(6),XWCM,YWCM)                                          DFQ00680
      TOP=Y(6)+Y(1)                                                     DFQ00690
      CALL WIN(TOP,XWTOP,YWTOP)                                         DFQ00700
```

```
      TDIF=TA-T0                                                                DFQ00710
C                                                                               DFQ00720
CC    TA          THE AMBIENT ATMOSPHERIC TEMPERATURE AT CLOUD HEIGHT           DFQ00730
CC    DTADZ       THE TEMPERATURE GRADIENT AT CLOUD HEIGHT                      DFQ00740
CC    XWCM        THE WIND SPEED IN THE X DIRECTION AT CLOUD C.O.M.             DFQ00750
CC    YXCM        THE WIND SPEED IN THE Y DIRECTION AT CLOUD C.O.M.             DFQ00760
CC    XWTOP       THE WIND SPEED IN THE X DIRECTION AT THE TOP OF THE CLOUDDFQ00770
CC    YWTOP       THE WIND SPEED IN THE Y DIRECTION AT THE TOP OF THE CLOUDDFQ00780
CC    TR          THE RATIO OF CLOUD TEMPERATURE TO AMBIENT TEMPERATURE         DFQ00790
C                                                                               DFQ00800
      TR=Y(2)/TA                                                                DFQ00810
C                                                                               DFQ00820
CC    CALCULATE ARVOL, THE SURFACE AREA TO VOLUME RATIO                         DFQ00830
C                                                                               DFQ00840
      ARVOL=3./Y(1)                                                             DFQ00850
C                                                                               DFQ00860
CC  DEFINITION OF DIFFERENTIAL EQUATIONS                                        DFQ00870
C                                                                               DFQ00880
      YP(1)=ALPHAK*ABS(Y(3))                                                    DFQ00890
      ZZ1=Y(6)                                                                  DFQ00900
      KM=DIFFUS(Z0,ZL,ZZ1)                                                      DFQ00910
      GROWTH=KM/Y(1)                                                            DFQ00920
      IF(YP(1).LT.GROWTH)YP(1)=GROWTH                                           DFQ00930
      YP(2)=-(1.+TR)*ARVOL*Y(2)*YP(1)-Y(3)*(DTADZ)                              DFQ00940
      YP(3)=9.8*TR-1.4*ARVOL*Y(3)*YP(1)                                         DFQ00950
      IF(Y(1)+Y(6).GT.ZINV)YP(3)=0.0                                            DFQ00960
      YP(4)=XWCM                                                                DFQ00970
      YP(5)=YWCM                                                                DFQ00980
      YP(6)=Y(3)                                                                DFQ00990
      IF(Y(1)+Y(6).GT.ZINV)YP(6)=0.0                                            DFQ01000
      YP(7)=XWTOP                                                               DFQ01010
      YP(8)=YWTOP                                                               DFQ01020
      GO TO 999                                                                 DFQ01030
  200 CONTINUE                                                                  DFQ01040
      DO 210 I=1,N                                                              DFQ01050
      YP(I)=0.                                                                  DFQ01060
  210 CONTINUE                                                                  DFQ01070
      YP(3)=ACCEL                                                               DFQ01080
      YP(6)=Y(3)                                                                DFQ01090
  999 RETURN                                                                    DFQ01100
      END                                                                       DFQ01110
```

```
      FUNCTION DIFFUS(Z0,ZL,Z)                                          DIF00010
      COMMON/STARS/USTAR,TSTAR,ZSTAR                                    DIF00020
      COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT                               DIF00030
C                                                                       DIF00040
C*****************************************************************    DIF00050
C PURPOSE                                                             DIF00060
C                                                                     DIF00070
C    TO CALCULATE THE DIFFUSIVITY AT A GIVEN HEIGHT                   DIF00080
C                                                                     DIF00090
C  INPUTS                                                             DIF00100
C                                                                     DIF00110
C    Z0  FRICTION HEIGHT IN METERS.                                   DIF00120
C                                                                     DIF00130
C    ZL  MONIN OBUKHOV LENGTH IN METERS.                              DIF00140
C                                                                     DIF00150
C    Z   HEIGHT AT WHICH DIFFUSIVITY IS DESIRED.                      DIF00160
C                                                                     DIF00170
C    CALLED BY ATMCAL, RISE AND DIFEQ                                 DIF00180
C                                                                     DIF00190
C    SUBROUTINES AND FUNCTIONS NEEDED                                 DIF00200
C                                                                     DIF00210
C      NONE                                                           DIF00220
C                                                                     DIF00230
C*****************************************************************    DIF00240
      ZZ=Z                                                            DIF00250
      IF(Z.GT.ZSTAR)Z=ZSTAR                                           DIF00260
C                                                                     DIF00270
C     NEUTRAL CASE                                                    DIF00280
C                                                                     DIF00290
      IF(ABS(ZL).LT.1.E3)GO TO 100                                    DIF00300
      DIFFUS=.4*USTAR*Z                                               DIF00310
      GO TO 999                                                       DIF00320
  100 IF(ZL.GT.0.0)GO TO 200                                          DIF00330
C                                                                     DIF00340
C     UNSTABLE CASE                                                   DIF00350
C                                                                     DIF00360
      S=Z/ZL                                                          DIF00370
      IF(S.LT.-2.)GO TO 110                                           DIF00380
      DIFFUS=.4*ABS(ZL*USTAR*S*(1.-16.*S)**(1./4.))                   DIF00390
      GO TO 999                                                       DIF00400
  110 DIFFUS=.4*ABS(ZL*(3./AW)*(-1.*S)**(4./3.))*USTAR                DIF00410
      GO TO 999                                                       DIF00420
C                                                                     DIF00430
C     STABLE CASE                                                     DIF00440
C                                                                     DIF00450
  200 S=Z/ZL                                                          DIF00460
      S0=Z0/ZL                                                        DIF00470
      IF(S.GT.1.5)GO TO 210                                           DIF00480
      DIFFUS=.4*ZL*USTAR*ABS(1./(1./(S0+S)+7.))                       DIF00490
      GO TO 999                                                       DIF00500
  210 DIFFUS=.4*ZL*USTAR*ABS(1./BW)                                   DIF00510
      Z=ZZ                                                            DIF00520
  999 RETURN                                                          DIF00530
      END                                                             DIF00540
```

```
      FUNCTION DOTPRD(A,B)                                              DOT00010
      DIMENSION A(2),B(2)                                               DOT00020
C                                                                       DOT00030
CC    DOTPRD IS THE SCALER PRODUCT OF A AND B                           DOT00040
C                                                                       DOT00050
      DOTPRD=A(1)*B(1)+A(2)*B(2)                                        DOT00060
      RETURN                                                            DOT00070
      END                                                               DOT00080
```

```
      FUNCTION DTERPI(NDIM,XI,XVAL,VAL,VMIN,WORK)                       DTI00010
C*************************************************************        DTI00020
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCDTI00030
CC                                                                      DTI00040
CC     PURPOSE                                                          DTI00050
CC                                                                      DTI00060
CC   PERFORMS AN N-DIMENSIONAL LINEAR INTERPOLATION                     DTI00070
CC                                                                      DTI00080
CC                                                                      DTI00090
CC     INPUT                                                            DTI00100
CC                                                                      DTI00110
CC    NDIM - THE NUMBER OF DIMENSIONS.  (- DONT RECALCUALTE WEIGHTS)    DTI00120
CC      XI - THE POINT IN THE HYPERSPACE AT WHICH THE INTERPOLATED      DTI00130
CC           VALUE IS DESIRED.  XI MUST BE A VECTOR OF ATLEAST NDIM     DTI00140
CC           IN LENGTH.                                                 DTI00150
CC    XVAL - THE COORDINATE VALUES AT THE CORNERS OF THE HYPERCUBE.     DTI00160
CC           THE VECTOR MUST BE SET UP LIKE A TWO-DIMENSIONAL ARRAY     DTI00170
CC           (2 X NDIM), WHERE THE FIRST SUBSCRIPT REFERS TO THE        DTI00180
CC           HYPERCUBE COORDINATES IN THE SECOND SUBSCRIPTS             DTI00190
CC           DIRECTION.                                                 DTI00200
CC     VAL - THE FUNCTIONAL VALUES AT THE CORNERS OF THE HYPERCUBE      DTI00210
CC           SURROUNDING XI.  THIS VECTOR MUST BE FILLED EQUIVALENT     DTI00220
CC           TO AN NDIM ARRAY WITH EACH DIMENSION AS 2.  THE SIZE       DTI00230
CC           OF VAL SHOULD BE ATLEAST 2**NDIM.                          DTI00240
CC    VMIN - A MINIMUM VALUE OF VAL FOR WHICH THE INTERPOLATION         DTI00250
CC           WILL USE A CORNER VALUE.                                   DTI00260
CC    WORK - A WORK VECTOR OF ATLEAST NDIM*2.  USE TO STORE COOR-       DTI00270
CC           DINATE WEIGHTS.                                            DTI00280
CC                                                                      DTI00290
CC                                                                      DTI00300
CC     OUTPUT                                                           DTI00310
CC                                                                      DTI00320
CC       RETURNS INTERPOLATED VALUE OF VAL AT XI                        DTI00330
CC                                                                      DTI00340
CC     CALLED BY MOMENT                                                 DTI00350
CC                                                                      DTI00360
C*************************************************************        DTI00370
      DIMENSION XI(4),XVAL(8),VAL(16),WORK(8)                           DTI00380
C                                                                       DTI00390
CC   SET UP THE COORDINATE WEIGHTS                                      DTI00400
C                                                                       DTI00410
      NDI=IABS(NDIM)                                                    DTI00420
      IF(NDIM .LT. 0) GO TO 1                                           DTI00430
      DO 100 I=1,NDI                                                    DTI00440
      I2=I*2                                                            DTI00450
      I1=I2-1                                                           DTI00460
      WORK(I2)=(XI(I)-XVAL(I1))/(XVAL(I2)-XVAL(I1))                     DTI00470
      WORK(I1)=1. - WORK(I2)                                            DTI00480
100   CONTINUE                                                          DTI00490
C                                                                       DTI00500
CC  INTERPOLATE  -  USE BINARY COUNTER FOR COORDINATE LOCATION          DTI00510
C                                                                       DTI00520
1     DTERPI=0.                                                         DTI00530
      SUM=0.                                                            DTI00540
      ND=2**NDI                                                         DTI00550
      DO 201 I=1,ND                                                     DTI00560
      IF(VAL(I) .LT. VMIN) GO TO 201                                    DTI00570
      L=I-1                                                             DTI00580
      WEIGHT=1.                                                         DTI00590
      DO 200 J=1,NDI                                                    DTI00600
      N=MOD(L,2) + J*2 - 1                                              DTI00610
      WEIGHT=WEIGHT*WORK(N)                                             DTI00620
      L=L/2                                                             DTI00630
200   CONTINUE                                                          DTI00640
      SUM=SUM + WEIGHT                                                  DTI00650
      DTERPI=DTERPI + WEIGHT*VAL(I)                                     DTI00660
201   CONTINUE                                                          DTI00670
      IF(SUM .EQ. 0.) GO TO 202                                         DTI00680
      DTERPI=DTERPI/SUM                                                 DTI00690
      RETURN                                                            DTI00700
```

```
C
202     STOP                                                    DTI00710
        END                                                     DTI00720
                                                                DTI00730
```

```
      SUBROUTINE DTERPS(II,X,VAL,NZ)                                    DTS00010
      DIMENSION X(81,4,3),VAL(16),II(4)                                 DTS00020
C ***************************************************************     DTS00030
C                                                                       DTS00040
C                                                                       DTS00050
C     PURPOSE                                                           DTS00060
C                                                                       DTS00070
C       TO SET UP A ONE DIMENSIONAL ARRAY OF THE VALUES CORRESPONDING   DTS00080
C       TO THE CORNERS OF THE CUBE WITHIN A TABULATED ARRAY WITH        DTS00090
C       LOWEST CORNER INDICES GIVEN                                     DTS00100
C                                                                       DTS00110
C     INPUT                                                             DTS00120
C                                                                       DTS00130
C       II      SINGLY DIMENSIONED ARRAY CONTAINING THE INDICES OF THE  DTS00140
C               LOWEST CORNER OF THE CUBE                               DTS00150
C       X       A TRIPLY DIMENSIONED ARRAY CONTAINING THE TABULATED     DTS00160
C               VALUES TO BE SET UP.  THE FIRST INDEX IS THE COLLAPSED  DTS00170
C               INDEX FOR THE FIRST TWO INDICES OF A FOUR-DIMENSIONAL   DTS00180
C               ARRAY                                                   DTS00190
C       NZ      THE RANGE OF THE FIRST INDEX OF THE FOUR-DIMENSIONAL    DTS00200
C               ARRAY                                                   DTS00210
C                                                                       DTS00220
C                                                                       DTS00230
C     OUTPUT                                                            DTS00240
C                                                                       DTS00250
C       VAL     SINGLY DIMENSIONED ARRAY CONTAINING THE VALUES OF X     DTS00260
C               FOR THE 16 CORNER POINTS OF THE CUBE                    DTS00270
C                                                                       DTS00280
C     CALLED BY MOMENT                                                  DTS00290
C                                                                       DTS00300
C ***************************************************************     DTS00310
      M=0                                                               DTS00320
      DO 104 L=1,2                                                      DTS00330
      LX=L + II(4) - 1                                                  DTS00340
      DO 103 K=1,2                                                      DTS00350
      KX=K + II(3) - 1                                                  DTS00360
      DO 102 JI=1,2                                                     DTS00370
      JIX=(JI + II(2) - 2)*NZ                                           DTS00380
      DO 101 IJ=1,2                                                     DTS00390
      IJX=JIX + IJ + II(1) -1                                           DTS00400
      M=M+1                                                             DTS00410
      VAL(M)=X(IJX,KX,LX)                                               DTS00420
101   CONTINUE                                                          DTS00430
102   CONTINUE                                                          DTS00440
103   CONTINUE                                                          DTS00450
104   CONTINUE                                                          DTS00460
      RETURN                                                            DTS00470
      END                                                               DTS00480
```

```
      SUBROUTINE DUSTCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,       DUS00010
     1  THWND,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,                       DUS00020
     2  LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,                   DUS00030
     3  TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS,NERR,          DUS00040
     4  NCHS,SRCBAS,SIDE1,SIDE2,DHDT)                                    DUS00050
      LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM,HORIZ,ERR                DUS00060
      LOGICAL SWITCH,CHANGE,DHDT                                        DUS00070
      DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2),TRNCOR(3)           DUS00080
     1  ,RECCOR(3),CPTS(2,6),ORIG(2),TRNFRM(2,2),TRN(3),REC(3)          DUS00090
     2                      ,CNTRD(2),OBSCOR(2),DIR(2)                  DUS00100
     3  ,SRCBAS(2),SIDE1(2),SIDE2(2),TEMP(2),NCHS(2)                    DUS00110
      REAL KZ,KX                                                        DUS00120
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)              DUS00130
      COMMON /MODE/ HORIZ                                               DUS00140
      COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,UM,DN,ZINV                       DUS00150
      COMMON /CLOCK/ FTIME,TWIND                                        DUS00160
      COMMON/MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,          DUS00170
     +           COOR(2,200),TSTAG(200),DMMY(401)                       DUS00180
      COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF                     DUS00190
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUDUS00200
      COMMON/CARB/RCARB1,RCARB2                                         DUS00210
      COMMON/ACL/CWINDS,CWINDC,CWINDW                                   DUS00220
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE          DUS00230
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     DUS00240
      DATA ONEM/-1./                                                    DUS00250
C ********************************************************************* DUS00260
C                                                                       DUS00270
C                                                                       DUS00280
C     PURPOSE                                                           DUS00290
C                                                                       DUS00300
C          DUSTCL CALCULATES DUST CLOUD DIMENSIONS AND TRANSMITTANCES   DUS00310
C     THROUGH DUST CLOUDS FOR GIVEN METEOROLOGICAL DATA, SOIL TYPE,     DUS00320
C     EXPLOSIVE CHARACTERISTICS, AND WAVELENGTH.                        DUS00330
C                                                                       DUS00340
C     SEE COMMENTS IN DRTRAN FOR DETAILS.                               DUS00350
C                                                                       DUS00360
C     SUBROUTINES CALLED                                                DUS00370
C                                                                       DUS00380
C     ATMCAL      ACCEPTS METEOROLOGICAL DATA AS ARGUMENTS AND COMPUTES DUS00390
C                 NECESSARY PARAMETERS IN COMMON /WNDPRM/, /EKWIND/     DUS00400
C                 /EKTEMP, /STARS/                                      DUS00410
C                                                                       DUS00420
C     SOURCE      ACCEPTS SOIL, CHARGE, AND WAVELENGTH SPECIFICATIONS   DUS00430
C                 AS INPUT AND COMPUTES NECESSARY PARAMETERS AND INITIAL DUS00440
C                 VALUES IN COMMON /PRTINF/ , /BUOYCL/ AND /CARB/       DUS00450
C                                                                       DUS00460
C     SETUP       ACCEPTS THE USER DEFINED COORDINATES OF THE CHARGES   DUS00470
C                 AND CONVERTS THEM TO THE INTERNAL (LOCAL) CORRDINATE  DUS00480
C                 SYSTEM. ALSO COMPUTES THE AREA AND PERIMETER OF THE   DUS00490
C                 BOUNDING PARALLELOGRAM AND OVERLAP DISTANCE OF THE    DUS00500
C                 CHARGES AND RETURNS THEM IN COMMON /ARRAY/ AND        DUS00510
C                 /SEPRTN/.                                             DUS00520
C                                                                       DUS00530
C     RISE        GIVEN CLOUD DIMENSIONS DURING BUOYANT RISE DEVELOPMENT DUS00540
C                 OF CLOUD, RISE CALCULATES THE DIMENSIONS AT A LATER   DUS00550
C                 TIME                                                  DUS00560
C                                                                       DUS00570
C    CLDIM        DETERMINES THE OUTPUT VARIABLES DESCRIBING THE CLOUD  DUS00580
C                 DIMENSIONS.                                           DUS00590
C                                                                       DUS00600
C     TRNCAL      CONTROLING ROUTINE FO THE CALCULATION OF TRANMITTANCES.DUS00610
C                                                                       DUS00620
C ********************************************************************* DUS00630
      IF(LOSTRN.OR.EDGE)GO TO 101                                       DUS00640
      NERR=4                                                            DUS00650
      GO TO 999                                                         DUS00660
  101 IF(.NOT.NEWATM) GO TO 200                                         DUS00670
      THETAX=THWND*PIRAD                                                DUS00680
      CALL ATMCAL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,THETAX,DHDT,ERR)   DUS00690
99999 IF(.NOT.ERR)GO TO 155                                             DUS00700
```

```
      NERR=7                                                            DUS00710
      GO TO 999                                                         DUS00720
  155 CONTINUE                                                          DUS00730
C                                                                       DUS00740
CCC   COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT USER        DUS00750
CCC   DEFINED COORDINATES INTO LOCAL COORDINATES WITH X AXIS IN         DUS00760
CCC   THE WIND DIRECTION.                                               DUS00770
C                                                                       DUS00780
      THETAX=THWND*PIRAD                                                DUS00790
      TRNFRM(1,1)=COS(THETAX)                                           DUS00800
      TRNFRM(2,2)=TRNFRM(1,1)                                           DUS00810
      TRNFRM(1,2)=SIN(THETAX)                                           DUS00820
      TRNFRM(2,1)=-TRNFRM(1,2)                                          DUS00830
  200 CONTINUE                                                          DUS00840
      IF(.NOT.NEWSRC) GO TO 300                                         DUS00850
      TWIND=1.E5                                                        DUS00860
      TTR=1.E5                                                          DUS00870
      TPRES=0.                                                          DUS00880
      DEL=.001                                                          DUS00890
      DO 250 I=1,2                                                      DUS00900
      IF(NARY.GT.1)SRCBAS(I)=COOR(I,1)                                  DUS00910
      ORIG(I)=SRCBAS(I)                                                 DUS00920
  250 CONTINUE                                                          DUS00930
      CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD)                         DUS00940
      CALL SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)                        DUS00950
  300 CONTINUE                                                          DUS00960
      IF(.NOT.LOSTRN) GO TO 400                                         DUS00970
C                                                                       DUS00980
CCC   CONVERT TRNCOR AND RECCOR TO LOCAL COORDINATES WITH ORIGIN AT     DUS00990
CCC   SRCBAS AND X AXIS IN WIND DIRECTION.                              DUS01000
C                                                                       DUS01010
      TRN(3)=TRNCOR(3)                                                  DUS01020
      REC(3)=RECCOR(3)                                                  DUS01030
      DO 320 I=1,2                                                      DUS01040
      TRN(I)=0.                                                         DUS01050
      REC(I)=0.                                                         DUS01060
      DO 310 J=1,2                                                      DUS01070
      TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))                     DUS01080
      REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))                     DUS01090
  310 CONTINUE                                                          DUS01100
  320 CONTINUE                                                          DUS01110
  400 CONTINUE                                                          DUS01120
      IF(.NOT.EDGE) GO TO 500                                           DUS01130
C                                                                       DUS01140
CCC   COMPUTE A UNIT VECTOR IN THE DIRECTION OF THE OBSERVERS LINE      DUS01150
CCC   OF SIGHT AND A UNIT VECTOR PERPENDICULAR TO THE LINE OF SIGHT     DUS01160
C                                                                       DUS01170
      CALL VSUM(ORIG,OBSCOR,ONEM,DIR)                                   DUS01180
      CALL UNIT(DIR,DIR,RANGE)                                          DUS01190
      COSTH=0.                                                          DUS01200
      SINTH=0.                                                          DUS01210
      DO 410 J=1,2                                                      DUS01220
      COSTH=COSTH+TRNFRM(1,J)*DIR(J)                                    DUS01230
      SINTH=SINTH+TRNFRM(2,J)*DIR(J)                                    DUS01240
  410 CONTINUE                                                          DUS01250
      SINTH2=SINTH*SINTH                                                DUS01260
      COSTH2=COSTH**2                                                   DUS01270
      SCRN(1)=SINTH                                                     DUS01280
      SCRN(2)=-COSTH                                                    DUS01290
C                                                                       DUS01300
CCC    COMPUTE THE PROJECTION OF EACH DIFFERENCE VECTOR DIFF ONTO THE   DUS01310
CCC    VECTOR PERPENDICULAR TO THE LINE OF SIGHT,(DIFF(1,J),DIFF(2,J))  DUS01320
CCC    IS THE VECTOR FROM THE REFERENCE CHARGE TO THE JTH CHARGE        DUS01330
CCC    LOCATION IN THE INTERNAL COORDINATE SYSTEM.                      DUS01340
C                                                                       DUS01350
      PARY1=0.0                                                         DUS01360
      PARY2=0.0                                                         DUS01370
      DO 420 J=1,NCHTOT                                                 DUS01380
      DO 415 I=1,2                                                      DUS01390
      TEMP(I)=DIFF(I,J)                                                 DUS01400
```

```
  415 CONTINUE                                                                  DUS01410
      PRSEP(J)=DOTPRD(TEMP,SCRN)                                                DUS01420
      X=PRSEP(J)                                                                DUS01430
      IF(X.LT.0.0)GO TO 416                                                     DUS01440
      IF(X.GT.PARY1)PARY1=X                                                     DUS01450
      GO TO 420                                                                 DUS01460
  416 IF(X.LT.PARY2)PARY2=X                                                     DUS01470
  420 CONTINUE                                                                  DUS01480
      PRJARY=(PARY1-PARY2)/2.                                                   DUS01490
      CENDIF=(PARY1+PARY2)/2.                                                   DUS01500
  500 CONTINUE                                                                  DUS01510
      IF(NARY.EQ.3)GO TO 600                                                    DUS01520
      IF(NEWTIM) CALL RISE(TPRES,TIME,DEL)                                      DUS01530
  600 IF(.NOT.EDGE) GO TO 650                                                   DUS01540
      FTIME=TIME                                                                DUS01550
      CALL CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS,ERR)               DUS01560
      IF(.NOT.ERR)GO TO 650                                                     DUS01570
      NERR=6                                                                    DUS01580
      GO TO 999                                                                 DUS01590
  650 CONTINUE                                                                  DUS01600
      IF(.NOT.LOSTRN)GO TO 999                                                  DUS01610
C                                                                               DUS01620
      CALL TRNCAL(TRN,REC,TIME,TRNLOS)                                          DUS01630
  999 RETURN                                                                    DUS01640
      END                                                                       DUS01650
```

```
      FUNCTION ERF(X)                                                   ERF00120
C     CALCULATES THE ERROR FUNCTION                                     ERF00010
C                                                                       ERF00020
C  INPUT                                                                ERF00030
C                                                                       ERF00040
C     X    VALUE AT WHICH ERROR FUNCTION IS DESIRED                     ERF00050
C                                                                       ERF00060
C  FUNCTIONS AND SUBROUTINES NEEDED                                     ERF00070
C                                                                       ERF00080
C     NONE                                                              ERF00090
C                                                                       ERF00100
C***********************************************************************ERF00110
      DIMENSION P(3,5),Q(3,5)                                           ERF00130
      DATA RPI/.5641896/                                                ERF00140
      DATA P/2.138533E+01,7.373888E+00,-4.257996E-02,                   ERF00150
     +       1.722276E+00,6.865018E+00,-1.960690E-01,                   ERF00160
     +       3.166529E-01,3.031799E+00,-5.168823E-02,                   ERF00170
     +       0.          ,5.631696E-01,0.           ,                   ERF00180
     +       0.          ,4.318779E-05,0.           /                   ERF00190
      DATA Q/1.895226E+01,7.373961E+00,1.509421E-01,                    ERF00200
     +       7.843746E+00,1.518491E+01,9.214524E-01,                    ERF00210
     +       1.000000E+00,1.279553E+01,1.000000E+00,                    ERF00220
     +       0.          ,5.354217E+00,0.           ,                   ERF00230
     +       0.          ,1.000000E+00,0.           /                   ERF00240
      AX=ABS(X)                                                         ERF00250
      ERFC=0.0                                                          ERF00260
      IF(AX.GT.11.0) GO TO 300                                          ERF00270
      X2=AX*AX                                                          ERF00280
      I=2                                                               ERF00290
      IF(AX.LT.0.5) I=1                                                 ERF00300
      IF(AX.GT.4.) I=3                                                  ERF00310
      IF(I-2) 10,20,30                                                  ERF00320
   10 N=3                                                               ERF00330
      Z=X2                                                              ERF00340
      GO TO 40                                                          ERF00350
   20 N=5                                                               ERF00360
      Z=AX                                                              ERF00370
      GO TO 40                                                          ERF00380
   30 N=3                                                               ERF00390
      Z=1./X2                                                           ERF00400
   40 SP=P(I,N)                                                         ERF00410
      SQ=Q(I,N)                                                         ERF00420
      N1=N-1                                                            ERF00430
      DO 50 K=1,N1                                                      ERF00440
      J=N-K                                                             ERF00450
      SP=SP*Z+P(I,J)                                                    ERF00460
   50 SQ=SQ*Z+Q(I,J)                                                    ERF00470
      IF(I-2) 60,70,80                                                  ERF00480
   60 ERFC=1.0-X*SP/SQ                                                  ERF00490
      ERF=1.-ERFC                                                       ERF00500
      RETURN                                                            ERF00510
   70 ERFC=EXP(-X2)*SP/SQ                                               ERF00520
      GO TO 300                                                         ERF00530
   80 ERFC=EXP(-X2)/AX*(RPI+SP/(SQ*X2))                                 ERF00540
  300 IF(X.LT.0.0) ERFC=2.0-ERFC                                        ERF00550
      ERF=1.-ERFC                                                       ERF00560
      RETURN                                                            ERF00570
      END                                                               ERF00580
```

```
      SUBROUTINE FIT(X,F,A,B,C)                                          FIT00190
C                                                                        FIT00010
C  QUADRATIC FIT TO THREE POINTS USING NEWTON'S FUNDAMENTAL FORMULA      FIT00020
C                                                                        FIT00030
C  INPUTS                                                                FIT00040
C                                                                        FIT00050
C     X     - 3 VALUES OF THE INDEPENDENT VARIIABLE                      FIT00060
C                                                                        FIT00070
C     F     - 3 FUNCTION VALUES CORRESPONDIING TO THE X VALUES           FIT00080
C                                                                        FIT00090
C  OUTPUTS                                                               FIT00100
C                                                                        FIT00110
C     A     - COEFFICIENT OF THE  X**2  TERM                             FIT00120
C                                                                        FIT00130
C     B     - COEFFICIENT OF THE  X  TERM                                FIT00140
C                                                                        FIT00150
C     C     - CONSTANT TERM                                              FIT00160
C                                                                        FIT00170
C*************************************************************           FIT00180
      DIMENSION X(3),F(3)                                                FIT00200
      H=X(2)-X(1)                                                        FIT00210
      DF1=(F(2)-F(1))/H                                                  FIT00220
      DF2=(F(3)-2.*F(2)+F(1))/(2.*H**2)                                  FIT00230
      A=DF2                                                              FIT00240
      B=DF1-DF2*(X(2)+X(1))                                              FIT00250
      C=F(1)+X(1)*(X(2)*DF2-DF1)                                         FIT00260
      RETURN                                                             FIT00270
      END                                                                FIT00280
```

```
      FUNCTION FUNCT(X,Z)                                                   FUC00010
      LOGICAL HORIZ,SKIP                                                    FUC00020
      COMMON /CLOCK/ TIME,TWIND                                             FUC00030
      COMMON/MODE/HORIZ                                                     FUC00040
      COMMON/SKIPIT/SKIP                                                    FUC00050
C *****************************************************                   FUC00060
C                                                                           FUC00070
C                                                                           FUC00080
C     PURPOSE                                                               FUC00090
C                                                                           FUC00100
C       TO SUPPLY A TRANSMITTANCE FUNCTION FOR THE CONTOUR TRACING          FUC00110
C       ROUTINE IN ORDER TO DETERMINE THE CLOUD EDGE.                       FUC00120
C                                                                           FUC00130
C     INPUT                                                                 FUC00140
C                                                                           FUC00150
C       X    THE HORIZONTAL COORDINATE IN METERS                            FUC00160
C       Z    THE VERTICAL COORDINATE IN METERS                              FUC00170
C                                                                           FUC00180
C     OUTPUT                                                                FUC00190
C                                                                           FUC00200
C       RETURNS THE LOG OF THE OPTICALLY WEIGHTED CL VALUE (AT              FUC00210
C       VISIBLE WAVELENGTHS) FOR THE LINE OF SIGHT SPECIFIED BY X,Z         FUC00220
C                                                                           FUC00230
C     FUNCTIONS CALLED                                                      FUC00240
C                                                                           FUC00250
C       CWIND                                                               FUC00260
C                                                                           FUC00270
C     CALLED BY GFUN, CLIMB, GRAD2                                          FUC00280
C                                                                           FUC00290
C *************************************************************           FUC00300
      HORIZ=.TRUE.                                                          FUC00310
      SKIP=.TRUE.                                                           FUC00320
      Y=0.                                                                  FUC00330
      EXT1=0.0                                                              FUC00340
      EXT2=0.0                                                              FUC00350
      IF(Z.LE.0.)GO TO 100                                                  FUC00360
      EXT1=CWIND(X,Y,Z,TIME)                                                FUC00370
      IF(TIME.LE.TWIND)GO TO 10                                             FUC00380
      EXT2=CSPHER(X,Y,Z,TIME)                                               FUC00390
   10 EXT=EXT1+EXT2                                                         FUC00400
      IF(EXT.LE.1.E-30)GO TO 100                                            FUC00410
      FUNCT=ALOG(EXT)                                                       FUC00420
      GO TO 999                                                             FUC00430
  100 FUNCT=-30.                                                            FUC00440
  999 CONTINUE                                                              FUC00450
      RETURN                                                                FUC00460
      END                                                                   FUC00470
```

```
      SUBROUTINE GAMMA(XX,GX,IER)                                          GAM00310
C     SUBROUTINE GAMMA                                                     GAM00010
C                                                                          GAM00020
C     PURPOSE                                                              GAM00030
C        COMPUTES THE GAMMA FUNCTION FOR A GIVEN ARGUMENT                  GAM00040
C                                                                          GAM00050
C     USAGE                                                                GAM00060
C        CALL GAMMA(XX,GX,IER)                                             GAM00070
C                                                                          GAM00080
C     DESCRIPTION OF PARAMETERS                                            GAM00090
C        XX  -THE ARGUMENT FOR THE GAMMA FUNCTION                          GAM00100
C        GX  -THE RESULTANT GAMMA FUNCTION                                 GAM00110
C        IER -THE RESULTANT ERROR CODE WHERE                               GAM00120
C             IER=0  NO ERROR                                              GAM00130
C             IER=1  XX IS WITHIN .000001 OF BEING A NEGATIVE INTEGER      GAM00140
C             IER=2  XX GT 57, OVERFLOW, GX SET TO 1.E32                   GAM00150
C                                                                          GAM00160
C     COMMENTS                                                             GAM00170
C         NONE                                                             GAM00180
C                                                                          GAM00190
C     SUBROUTINES AND FUNCTIONS                                            GAM00200
C         NONE                                                             GAM00210
C                                                                          GAM00220
C     METHOD                                                               GAM00230
C        THE RECURSION RLATION AND POLYNOMIAL APPROXIMATION                GAM00240
C        BY C. HASTINGS,JR., 'APPROXIMATIONS FOR DIGITAL COMPUTERS',       GAM00250
C        PRINCETON UNIVERSITY PRESS, 1955                                  GAM00260
C                                                                          GAM00270
C     ..................................................................   GAM00280
C                                                                          GAM00290
      IF(XX-57.) 6,6,4                                                     GAM00300
    4 IER=2                                                                GAM00320
      GX=1.E32                                                             GAM00330
      RETURN                                                               GAM00340
    6 X=XX                                                                 GAM00350
      ERR=1.0E-6                                                           GAM00360
      IER=0                                                                GAM00370
      GX=1.0                                                               GAM00380
      IF(X-2.0) 50,50,15                                                   GAM00390
   10 IF(X-2.0) 110,110,15                                                 GAM00400
   15 X=X-1.0                                                              GAM00410
      GX=GX*X                                                              GAM00420
      GO TO 10                                                             GAM00430
   50 IF(X-1.0) 60,120,110                                                 GAM00440
C                                                                          GAM00450
C     SEE IF X IS NEAR NEGATIVE INTEGER OR ZERO                            GAM00460
C                                                                          GAM00470
   60 IF(X-ERR) 62,62,80                                                   GAM00480
   62 Y=FLOAT(INT(X))-X                                                    GAM00490
      IF(ABS(Y)-ERR) 130,130,70                                            GAM00500
C                                                                          GAM00510
C     X NOT NEAR A NEGATIVE INTEGER OR ZERO                                GAM00520
C                                                                          GAM00530
   70 IF(X-1.0)80,80,110                                                   GAM00540
   80 GX=GX/X                                                              GAM00550
      X=X+1.0                                                              GAM00560
      GO TO 70                                                             GAM00570
  110 Y=X-1.0                                                              GAM00580
      GY=1.0+Y*(-0.5771017+Y*(0.9858540+Y*(-0.8764218+Y*(0.8328212+        GAM00590
     1Y*(-0.5684729+Y*(0.2548205+Y*(-0.0514993)))))))                      GAM00600
      GX=GX*GY                                                             GAM00610
  120 RETURN                                                               GAM00620
  130 IER=1                                                                GAM00630
      RETURN                                                               GAM00640
      END                                                                  GAM00650
                                                                           GAM00660
```

```
      FUNCTION GFUN(S)                                                      GFU00010
C                                                                           GFU00020
CC      GFUN IS THE RESTRICTION OF THE TWO DIMENSIONAL FUNCTION, F, TO      GFU00030
C             A LINE. I.E. FORM G(S)=F(X,Y), WHERE (X,Y)=BASE+S*DIR.        GFU00040
C                                                                           GFU00050
      EXTERNAL FUNCT                                                        GFU00060
      DIMENSION                P(2)                                         GFU00070
      COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON            GFU00080
      CALL VSUM(BASE,DIR,S,P)                                               GFU00090
      GFUN=FUNCT(P(1),P(2))                                                 GFU00100
      RETURN                                                                GFU00110
      END                                                                   GFU00120
```

```
      SUBROUTINE GRAD2(PT,FUNCT,RES,GRAD,SLOPE)                          GRA00240
C                                                                        GRA00010
C   THIS SUBROUTINE CALCULATES THE UNIT GRADIENT VECTOR                  GRA00020
C   OF A FUNCTION AT THE GIVEN POINT USING THE FORMULA                   GRA00030
C   PARTIAL DF/DX = (F(X+R,Y)-F(X,Y))/R WHERE R IS                       GRA00040
C   SMALL, SIMILARLY FOR PARTIAL DF/DY. IT THEN                          GRA00050
C   NORMALIZES THE RESULTANT VECTOR FOR THE UNIT GRADIENT                GRA00060
C   AND FINDS THE SLOPE,WHICH IS THE MAGNITUDE OF                        GRA00070
C   OF THE REGULAR GRADIENT.                                             GRA00080
C                                                                        GRA00090
C   ARGUMENTS PASSED                                                     GRA00100
C      PT-THE POINT AT WHICH THE UNIT GRADIENT IS FOUND.                 GRA00110
C      FUNCT-THE FUNCTION(X,Y)                                           GRA00120
C      RES-R                                                             GRA00130
C      GRAD-THE UNIT GRADIENT                                            GRA00140
C      SLOPE-SLOPE AT PT                                                 GRA00150
C                                                                        GRA00160
C   OTHER VARIABLES                                                      GRA00170
C      C00,C10,C01-THE FUNCTION AT THE POINTS F(X,Y),F(X+R,Y),           GRA00180
C          AND F(X,Y+R) RESPECTIVELY.                                    GRA00190
C                                                                        GRA00200
C   SUBROUTINES CALLED                                                   GRA00210
C      UNIT-NORMALIZES A VECTOR AND FINDS ITS MAGNITUDE.                 GRA00220
C                                                                        GRA00230
      DIMENSION PT(2),GRAD(2)                                            GRA00250
      C00=FUNCT(PT(1),PT(2))                                             GRA00260
      C10=FUNCT(PT(1)+RES,PT(2))                                         GRA00270
      C01=FUNCT(PT(1),PT(2)+RES)                                         GRA00280
      GRAD(1)=(C10-C00)/RES                                              GRA00290
      GRAD(2)=(C01-C00)/RES                                              GRA00300
      CALL UNIT(GRAD,GRAD,SLOPE)                                         GRA00310
      RETURN                                                             GRA00320
      END                                                                GRA00330
```

```
      SUBROUTINE GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)                GRD00320
C                                                                       GRD00010
C   ROUTINE TO EVALUTE THE INTEGRAND FOR THE TRAPEZOIDAL INTEGRATION    GRD00020
C   FOR FINDING THE OPTICALLY WEIGHTED CONCENTRATION ALONG THE LINE OF  GRD00030
C   SIGHT WHERE THE DUST IS GENERATED BY A VEHICLE.                     GRD00040
C                                                                       GRD00050
C   INPUTS                                                              GRD00060
C                                                                       GRD00070
C      U      - A UNIT VECTOR ALONG THE LINE OF SIGHT                   GRD00080
C                                                                       GRD00090
C      TR     - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATEGRD00100
C               SYSTEM                                                  GRD00110
C                                                                       GRD00120
C     XNORM - DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER             GRD00130
C                                                                       GRD00140
C     TIME - PRESENT TIME AT WHICH A TRANSMITTANCE IS WANTED            GRD00150
C                                                                       GRD00160
C     TIVEN- TIME THAT THE VEHICLE HAS TRAVELED                         GRD00170
C                                                                       GRD00180
C     VDIR - VECTOR CONTAINING DESCRIBING THE VEHICLE DIRECTION AND     GRD00190
C            SPEED                                                      GRD00200
C                                                                       GRD00210
C   OUPUT                                                               GRD00220
C                                                                       GRD00230
C     VALUE - VALUE OF THE INTEGRAND                                    GRD00240
C                                                                       GRD00250
C   FUNCTIONS AND SUBROUTINES NEEDED                                    GRD00260
C                                                                       GRD00270
C      CONLEN - TO FIND THE LENGTH OF THE INTERSECTION OF THE LINE OF   GRD00280
C               SIGHT AND THE TILTED CYLINDER                           GRD00290
C                                                                       GRD00300
C***********************************************************************GRD00310
      DIMENSION U(3),TR(3),VDIR(2),VP(2)                                GRD00330
      COMMON/MO5/DMMMY(604),DMM(600),                                   GRD00340
     +           ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),         GRD00350
     +           RB(3,25),Z2(3,25)                                      GRD00360
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUGRD00370
      COMMON/VL/VLOAD                                                   GRD00380
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     GRD00390
C                                                                       GRD00400
C   FIND THE VEHICLE POSITION AT TIME TIVEH                             GRD00410
C                                                                       GRD00420
      VP(1)=TIVEH*VDIR(1)                                               GRD00430
      VP(2)=TIVEH*VDIR(2)                                               GRD00440
      TOF=TIME-TIVEH                                                    GRD00450
      DO 10 I=1,ICOUNT                                                  GRD00460
      IND=I                                                             GRD00470
      IF(TOF.LT.TIMES(I))GO TO 20                                       GRD00480
   10 CONTINUE                                                          GRD00490
C                                                                       GRD00500
C   IF TOF (TIME OF FLIGHT) IS GREATER THAN TABULATED VALUES IT IS      GRD00510
C   ASSUMED THE THE CLOUD HAS DISSIPATED                                GRD00520
C   XC=X1*Z+X0  IS THE LINE THRIUGH THE CENTER OF THE CYLINDER          GRD00530
C                                                                       GRD00540
      GO TO 50                                                          GRD00550
   20 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)                     GRD00560
      X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)                     GRD00570
      RAD=TOF*(RT(1,IND)*TOF+RT(2,IND))+RT(3,IND)                       GRD00580
      HTTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)                     GRD00590
      HTBOT=0.0                                                         GRD00600
      RTOP=RAD                                                          GRD00610
      RBOT=RAD                                                          GRD00620
      XCEN=VP(1)+(X1*HTTOP+X0)                                          GRD00630
      YCEN=VP(2)                                                        GRD00640
      XB=VP(1)+(X1*HTBOT+X0)                                            GRD00650
      YB=VP(2)                                                          GRD00660
      IF(U(3).LT.1.E-06)GO TO 30                                        GRD00670
C                                                                       GRD00680
C   COMPUTE INTERSECTION LENGTH FOR NON HORIZONTAL LINES OF SIGHT       GRD00690
C                                                                       GRD00700
```

```
      CALL CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,XNORM,PLEN)GRD00710
      GO TO 40                                                          GRD00720
C                                                                       GRD00730
C  DETERMINE LENGTH OF INTERSECTION FOR A HORIZONTAL LINE OF SIGHT      GRD00740
C                                                                       GRD00750
   30 IF(HTTOP.LT.TR(3))GO TO 35                                        GRD00760
      A=U(1)**2+U(2)**2                                                 GRD00770
      B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)                             GRD00780
      C=(TR(1)-XCEN)**2+(TR(2)-YCEN)**2-RAD**2                          GRD00790
      X=B**2-A*C                                                        GRD00800
      IF(X.LT.0.0)GO TO 35                                              GRD00810
      P1=(-B+SQRT(X))/A                                                 GRD00820
      P2=(-B-SQRT(X))/A                                                 GRD00830
      IF(P1.GT.XNORM.AND.P2.GT.XNORM)GO TO 35                           GRD00840
      IF(P2.LT.0.0.AND.P1.LT.0.0)GO TO 35                               GRD00850
      PLEN=AMIN1(P1,XNORM)-AMAX1(P2,0.0)                                GRD00860
      GO TO 40                                                          GRD00870
   35 PLEN=0.0                                                          GRD00880
   40 VOL=PI*HTTOP*(RAD**2)                                             GRD00890
      VALUE=VLOAD*PLEN/VOL                                              GRD00900
      GO TO 999                                                         GRD00910
   50 VALUE=0.0                                                         GRD00920
  999 RETURN                                                            GRD00930
      END                                                               GRD00940
```

```
      SUBROUTINE GREEN(Z,Z1,T,ALPHA,T0,IER)                              GRE00010
C*************************************************************          GRE00020
C                                                                        GRE00030
C                                                                        GRE00040
C     PURPOSE                                                            GRE00050
C                                                                        GRE00060
C       TO COMPUTE THE GENERALIZED GREENS FUNCTION                       GRE00070
C                                                                        GRE00080
C      USES GREEN1                                                       GRE00090
C                                                                        GRE00100
C      SEE GREEN1 FOR ARGUMENT LIST                                      GRE00110
C                                                                        GRE00120
C                                                                        GRE00130
C*************************************************************          GRE00140
      REAL N,M                                                           GRE00150
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV                           GRE00160
C                                                                        GRE00170
      IF(N .EQ. 1.) GO TO 2                                              GRE00180
      X2=2.-N                                                            GRE00190
      AT=ALPHA*T                                                         GRE00200
      T0=0.                                                              GRE00210
      IF(AT .GE. Z1) RETURN                                              GRE00220
      CALL GREEN1((Z+AT)**X2,Z1**X2,X2*X2*T,(N-1.)/X2,T1,IER)            GRE00230
      T1=T1*X2*Z1**(1.-N)                                                GRE00240
      U=1.                                                               GRE00250
      T2=0.                                                              GRE00260
      IF(ABS(ALPHA) .LT. 1.E-4) GO TO 1                                  GRE00270
      ZMZ=Z-Z1+AT                                                        GRE00280
      X2=N+1.                                                            GRE00290
      AN1=ALPHA*X2                                                       GRE00300
      ZMZN=Z1**X2 - (Z1-AT)**X2                                          GRE00310
      ARG=(-AN1*ZMZ*ZMZ)/(4.*ZMZN)                                       GRE00320
      IF(ARG .LT. -70.) GO TO 3                                          GRE00330
      T2=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)                          GRE00340
    3 IF(T1.LT.1.E-30 .AND. T2.LT.1.E-30) RETURN                         GRE00350
C                                                                        GRE00360
C     CALCULATION OF MIXING RATIO, U, BY N=1 ANALOGY                     GRE00370
C                                                                        GRE00380
      CALL GREEN1(Z+AT,Z1,T,0.,T1U,IER)                                  GRE00390
      X2=2.                                                              GRE00400
      AN1=ALPHA*X2                                                       GRE00410
      ZMZN=Z1**X2 - (Z1-AT)**X2                                          GRE00420
      T2U=0.                                                             GRE00430
      ARG=(-AN1*ZMZ*ZMZ)/(4.*ZMZN)                                       GRE00440
      IF(ARG .LT. -70.) GO TO 4                                          GRE00450
      T2U=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)                         GRE00460
    4 IF(T1U.LT.1.E-30 .AND. T2U.LT.1.E-30) GO TO 1                      GRE00470
      CALL GREEN1(Z,Z1,T,ALPHA,G,IER)                                    GRE00480
      U=(G-T2U)/(T1U-T2U)                                                GRE00490
    1 IF(U .LT. 0.) U=0.                                                 GRE00500
      IF(U .GT. 1.) U=1.                                                 GRE00510
C                                                                        GRE00520
C     COMBINE LIMITING SOLUTIONS WITH DETERMINED MIXING RATIO            GRE00530
C                                                                        GRE00540
      T0=U*T1 + (1.-U)*T2                                                GRE00550
      RETURN                                                             GRE00560
    2 CALL GREEN1(Z,Z1,T,ALPHA,T0,IER)                                   GRE00570
      RETURN                                                             GRE00580
      END                                                                GRE00590
```

```
      SUBROUTINE GREEN1(Z,Z1,T,NU,BI,IER)                                GRV00490
C                                                                        GRV00010
CCC     SUBROUTINE GREEN1                                                GRV00020
CCC                                                                      GRV00030
CCC     PURPOSE                                                          GRV00040
CCC        COMPUTE THE I BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER  GRV00050
CCC              AND MULTIPLY BY AN APPROPRIATE POWER OF THE ARGUMENT    GRV00060
CCC              AND AN EXPONENTIAL IN ORDER TO CALCULATE THE GREENS     GRV00070
CCC              FUNCTION FOR THE WIND DIFFUSION EQUATION                GRV00080
CCC                                                                      GRV00090
CCC     USAGE                                                            GRV00100
CCC        CALL GREEN1(Z,Z1,T,NU,BI,IER)                                 GRV00110
CCC                                                                      GRV00120
CCC     DESCRIPTION OF PARAMETERS                                        GRV00130
CCC        Z,Z1,T   -THE ARGUMENTS OF THE FUNCTION DESIRED               GRV00140
CCC        NU  -THE ORDER OF THE I BESSEL FUNCTION                       GRV00150
CCC        BI  -THE RESULTANT BESSEL FUNCTION                            GRV00160
CCC        IER -RESULTANT ERROR CODE WHERE                               GRV00170
CCC           IER=-1  EXPONENTIAL UNDERFLOW (NON-FATAL),  BI SET TO 0.0  GRV00180
CCC           IER=0 NO ERROR                                             GRV00190
CCC           IER=1 NU NEAR NEGATIVE INTEGER                             GRV00200
CCC           IER=2 OVERFLOW IN GAMMA                                    GRV00210
CCC           IER=3 UNDERFLOW, BI .LT. 1.E-32, BI SET TO 0.0             GRV00220
CCC           IER=4 OVERFLOW, X .GT. 90 WHERE X .GT. N                   GRV00230
CCC           IER=5 X IS NEGATIVE                                        GRV00240
CCC                                                                      GRV00250
CCC     REMARKS                                                          GRV00260
CCC        NU IS A  REAL NUMBER                                          GRV00270
CCC        N AND X MUST BE .GE. ZERO                                     GRV00280
CCC        THIS SUBROUTINE IS A MODIFICATION OF BESI WHICH COMPUTES THE  GRV00290
CCC        I BESSEL FUNCTION FOR INTEGER ORDERS.  THE CHANGE REQUIRES    GRV00300
CCC        USE OF THE GAMMA FUNCTION FOR COMPUTING THE FIRST TERM OF THE GRV00310
CCC        SERIES.  THE SUCCESSIVE TERMS ARE CALCULATED WITH THE SAME    GRV00320
CCC        RECURSION FORMULA AND THE ASYMPTOTIC APPROXIMATION IS ALSO    GRV00330
CCC        UNCHANGED.  BESI IS IN THE IBM SYSTEM/360 SCIENTIFIC          GRV00340
CCC        SUBROUTINE PACKAGE.  MODIFICATIONS MADE BY D. DVORE, AERODYNE GRV00350
CCC        RESEARCH  INC. JANUARY 15,1979.                               GRV00360
CCC                                                                      GRV00370
CCC     SUBROUTINES AND FUNCTIONS REQUIRED                               GRV00380
CCC        GAMMA  WHICH COMPUTES THE GAMMA FUNCTION                      GRV00390
CCC                                                                      GRV00400
CCC     METHOD                                                           GRV00410
CCC        COMPUTES I BESSEL FUNCTION USING SERIES OR ASYMPTOTIC         GRV00420
CCC        APPROXIMATION DEPENDING ON THE RANGE OF THE ARGUMENT.         GRV00430
CCC                                                                      GRV00440
CCC     CALLED BY MOMENT                                                 GRV00450
CCC                                                                      GRV00460
CCC     .....................................................            GRV00470
C                                                                        GRV00480
      REAL NU                                                            GRV00500
      X=2.*SQRT(Z*Z1)/T                                                  GRV00510
C                                                                        GRV00520
CCC   CHECK FOR ERRORS IN NU AND X AND EXIT IF ANY ARE PRESENT           GRV00530
C                                                                        GRV00540
      IER=0                                                              GRV00550
      BI=1.0                                                             GRV00560
      IF(NU)10,15,10                                                     GRV00570
   10 IF(X)160,20,20                                                     GRV00580
   15 IF(X)160,17,20                                                     GRV00590
   17 ARG=-(Z+Z1)/T                                                      GRV00600
      IF(ARG .LT. -80.) GO TO 170                                        GRV00610
      BI=EXP(ARG)/T                                                      GRV00620
      RETURN                                                             GRV00630
C                                                                        GRV00640
CCC   DEFINE TOLERANCE                                                   GRV00650
C                                                                        GRV00660
   20 TOL=1 E-3                                                          GRV00670
C                                                                        GRV00680
CCC   IF ARGUMENT GT 12 AND GT NU, USE ASYMPTOTIC FORM                   GRV00690
C                                                                        GRV00700
```

```
      IF(X-12.)40,40,30                                                 GRV00710
   30 IF(X-ABS(NU))40,40,110                                            GRV00720
C                                                                       GRV00730
CC    COMPUTE FIRST TERM OF SERIES AND SET INITIAL VALUE OF THE SUM     GRV00740
C                                                                       GRV00750
   40 XX=X/2.                                                           GRV00760
      N=INT(NU)                                                         GRV00770
      FN=N                                                              GRV00780
      R=NU-FN                                                           GRV00790
      CALL GAMMA(1.+NU,GR,IER)                                          GRV00800
      IF(IER .EQ. 0) GO TO 60                                           GRV00810
   50 BI=0.0                                                            GRV00820
      RETURN                                                            GRV00830
   60 TERM=1./GR                                                        GRV00840
   70 BI=TERM                                                           GRV00850
      XX=XX*XX                                                          GRV00860
C                                                                       GRV00870
CC    COMPUTE TERMS, STOPPING WHEN ABS(TERM) LE ABS(SUM OF TERMS)*TOLERAGRV00880
C                                                                       GRV00890
      DO 90 K=1,1000                                                    GRV00900
      IF(ABS(TERM)-ABS(BI*TOL))95,95,80                                 GRV00910
   80 FK=K                                                              GRV00920
      FK=FK*(NU+FK)                                                     GRV00930
      TERM=TERM*(XX/FK)                                                 GRV00940
   90 BI=BI+TERM                                                        GRV00950
   95 ARG=-(Z+Z1)/T                                                     GRV00960
      IF(ARG .LT. -80.) GO TO 170                                       GRV00970
      BI=BI*(Z1/T)**NU*EXP(ARG)/T                                       GRV00980
C                                                                       GRV00990
CC    RETURN BI AS ANSWER                                               GRV01000
C                                                                       GRV01010
  100 RETURN                                                            GRV01020
C                                                                       GRV01030
CC    X GT 12 AND X GT NU, SO USE ASYMPTOTIC APPROXIMATION              GRV01040
C                                                                       GRV01050
  110 FN=4.*NU*NU                                                       GRV01060
  115 XX=1./(8.*X)                                                      GRV01070
      TERM=1.                                                           GRV01080
      BI=1.                                                             GRV01090
      DO 130 K=1,30                                                     GRV01100
      IF(ABS(TERM)-ABS(BI*TOL)) 140,140,120                             GRV01110
  120 FK=(2*K-1)**2                                                     GRV01120
      TERM=TERM*XX*(FK-FN)/FLOAT(K)                                     GRV01130
  130 BI=BI+TERM                                                        GRV01140
C                                                                       GRV01150
CC    SIGNIFICANCE LOST AFTER 30 TERMS, TRY SERIES                      GRV01160
C                                                                       GRV01170
      GO TO 40                                                          GRV01180
  140 PI=3.141592653                                                    GRV01190
      ARG=X-(Z+Z1)/T                                                    GRV01200
      IF(ARG .LT. -80.) GO TO 170                                       GRV01210
      BI=BI*(Z1/Z)**(NU/2.)*EXP(ARG)/SQRT(2.*PI*X)/T                    GRV01220
      GO TO 100                                                         GRV01230
  160 IER=5                                                             GRV01240
      GO TO 100                                                         GRV01250
  170 BI=0.0                                                            GRV01260
      GO TO 50                                                          GRV01270
      END                                                               GRV01280
```

```
      SUBROUTINE MOMENT(VGRAV,ZIN,H,TIN,Q,XBAR,SIGW2,SIGP2)                  MOM00010
      REAL M,N,NM                                                            MOM00020
      DIMENSION AL(9),Z(9),T(9),XB(81,4,3),SW(81,4,3),SP(81,4,3),NM(9)       MOM00030
      DIMENSION VAL(16),XVAL(8),W(8),XI(4),IB(4),NTC(4),II(4),X(9,4)         MOM00040
      LOGICAL FIRST                                                          MOM00050
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV                               MOM00060
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUMOM00070
      EQUIVALENCE (Z(1),X(1,1)),(T(1),X(1,2)),(AL(1),X(1,3))                 MOM00080
      EQUIVALENCE (NM(1),X(1,4))                                             MOM00090
      DATA FIRST/.TRUE./,IB/4,4,1,2/,ITC/11/,HREF/1./                        MOM00100
C *****************************************************************        MOM00110
C                                                                            MOM00120
C                                                                            MOM00130
C     PURPOSE                                                                MOM00140
C                                                                            MOM00150
C       TO CONVERT PARAMETERS TO NONDIMENSIONAL FORM AND THEN COMPUTE        MOM00160
C     THE ZERO ORDER MOMENT AND INTERPOLATE FROM TABULATED VALUES OF         MOM00170
C     THE HIGHER ORDER MOMENTS                                               MOM00180
C                                                                            MOM00190
C     INPUT                                                                  MOM00200
C                                                                            MOM00210
C       VGRAV    THE GRAVITATIONAL SETTLING VELOCITIES OF THE PARTICLE       MOM00220
C                IN METERS / SEC                                             MOM00230
C       ZIN        THE HEIGHT (METERS) AT WHICH THE MOMENTS ARE DESIRED      MOM00240
C       H        THE HEIGHT OF RELEASE OF THE PARTICLES IN METERS            MOM00250
C       TIN      THE TIME IN SECONDS AFTER RELEASE                           MOM00260
C                                                                            MOM00270
C     OUTPUT                                                                 MOM00280
C                                                                            MOM00290
C       Q        THE VERTICAL CONCENTRATION OF PARTICLES AT HEIGHT Z         MOM00300
C       XBAR     THE DISPLACEMENT (METERS) IN THE X (IE WIND) DIRECTION      MOM00310
C                OF THE CENTER OF MASS OF PARTICLES AT HEIGHT Z              MOM00320
C       SIGW2    THE SQUARE OF THE STANDARD DEVIATION OF THE WINDWARD        MOM00330
C                DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2      MOM00340
C       SIGP2    THE SQUARE OF THE STANDARD DEVIATION OF THE CROSS-WIND      MOM00350
C                DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2      MOM00360
C                                                                            MOM00370
C                                                                            MOM00380
C     SUBROUTINES CALLED                                                     MOM00390
C                                                                            MOM00400
C       DTERPS    PUTS THE NEEDED VALUES OF THE TABULATED MOMENTS            MOM00410
C                 INTO A ONE-DIMENSIONAL ARRAY                               MOM00420
C       DTERPI    A FUNCTION WHICH RETURNS THE INTERPOLATED VALUE            MOM00430
C                 FOR GIVEN ARGUMENTS AND ARRAYS                             MOM00440
C       GREEN     CALCULATES THE GREENS FUNCTION WHICH IS THE                MOM00450
C                 0-ORDER MOMENT                                             MOM00460
C                                                                            MOM00470
C     CALLED BY CWIND                                                        MOM00480
C                                                                            MOM00490
C *****************************************************************        MOM00500
      IF(.NOT.FIRST)GO TO 5                                                  MOM00510
C                                                                            MOM00520
C     READ IN THE TABLE OF MOMENTS ON THE FIRST CALL OF MOMENT               MOM00530
C                                                                            MOM00540
C      Z       LOG OF NON-DIMENSIONAL HEIGHTS AT WHICH MOMENTS ARE TABULAMOM00550
C      T       LOG OF NON-DIMENSIONAL TIMES   AT WHICH MOMENTS ARE TABULAMOM00560
C      AL      NON-DIMENSIONAL SETTLING VELOCITIES AT WHICH MOMENTS ARE  MOM00570
C                    TABULATED                                           MOM00580
C      NM      DIFFUSIVITY POWER LAW EXPONENTS AT WHICH MOMENTS ARE      MOM00590
C                     TABULATED                                          MOM00600
C                                                                        MOM00610
C      XB      TABULATED VALUES OF LOGS OF FIRST ORDER MOMENTS (RELATED  MOM00620
C                   TO MEAN HORIZONTAL DISPLACEMENT)                     MOM00630
C      SW      TABULATED VALUES OF LOGS OF WIND SHEAR COMPONENT OF SECONDMOM00640
C               ORDER MOMENT (CONTRIBUTES TO VARIANCE IN WIND DIRECTION) MOM00650
C      SP      TABULATED VALUES OF LOGS OF SECOND ORDER MOMENT COMMON TO MOM00660
C                  WIND AND CROSS-WIND VARIANCES                         MOM00670
C                                                                        MOM00680
      READ(NDIRTU,1) NZ,NT,NA,NM                                         MOM00690
    1 FORMAT(4I3)                                                        MOM00700
```

```
      NTC(1)=NZ-1                                                       MOM00710
      NTC(2)=NT-1                                                       MOM00720
      NTC(3)=NA-1                                                       MOM00730
      NTC(4)=NN-1                                                       MOM00740
      READ(NDIRTU,2) (Z(I),I=1,NZ)                                      MOM00750
    2 FORMAT(6E13.5)                                                    MOM00760
      READ(NDIRTU,2) (T(I),I=1,NT)                                      MOM00770
      READ(NDIRTU,2) (AL(I),I=1,NA)                                     MOM00780
      READ(NDIRTU,2) (NM(I),I=1,NN)                                     MOM00790
      NZT=NZ*NT                                                         MOM00800
      DO 3 L=1,NN                                                       MOM00810
      READ(NDIRTU,2) ((XB(IJ,K,L),IJ=1,NZT),K=1,NA)                     MOM00820
      READ(NDIRTU,2) ((SW(IJ,K,L),IJ=1,NZT),K=1,NA)                     MOM00830
      READ(NDIRTU,2) ((SP(IJ,K,L),IJ=1,NZT),K=1,NA)                     MOM00840
    3 CONTINUE                                                          MOM00850
      FIRST=.FALSE.                                                     MOM00860
      REWIND NDIRTU                                                     MOM00870
    5 CONTINUE                                                          MOM00880
C                                                                       MOM00890
CC    CONVERT INPUT PARAMETERS TO NONDIMENSIONAL FORM                   MOM00900
C                                                                       MOM00910
      SCLU=DZ0*H**(N-1.)                                                MOM00920
      XI(1)=ZIN/H                                                       MOM00930
      XI(2)=SCLU*TIN/H                                                  MOM00940
      XI(3)=VGRAV/SCLU                                                  MOM00950
      XI(4)=N                                                           MOM00960
      CALL GREEN(XI(1),HREF,XI(2),XI(3),Q,IER)                          MOM00970
      Q=Q/H                                                             MOM00980
      IF(Q .LE. 1.E-10) GO TO 999                                       MOM00990
C                                                                       MOM01000
CC    TAKE LOGS FOR LOGARITHMIC INTERPOLATION                           MOM01010
C                                                                       MOM01020
      XI(1)=ALOG(XI(1))                                                 MOM01030
      XI(2)=ALOG(XI(2))                                                 MOM01040
C                                                                       MOM01050
CC    DETERMINE INDICES OF LOWEST CORNER POINT OF THE CUBE TO           MOM01060
CC    BE USED IN INTERPOLATION MAKING SURE THAT ENOUGH CORNER POINTS    MOM01070
CC    OF THE CUBE HAVE TABULATED VALUES                                 MOM01080
C                                                                       MOM01090
      DO 100 I=1,4                                                      MOM01100
      II(I)=IB(I)                                                       MOM01110
  100 CONTINUE                                                          MOM01120
      DO 101 III=1,4                                                    MOM01130
      I=5-III                                                           MOM01140
    6 IA=II(I)                                                          MOM01150
      IF(XI(I) .GE. X(IA,I) .AND. XI(I) .LE. X(IA+1,I)) GO TO 101       MOM01160
      IF(XI(I) .LT. X(IA,I) .AND. IA .EQ. 1) GO TO 101                  MOM01170
      IF(XI(I) .GT. X(IA,I) .AND. IA .EQ. NTC(I)) GO TO 101             MOM01180
      ISAV=II(I)                                                        MOM01190
      II(I)=IA + IFIX(SIGN(1.,XI(I)-X(IA,I)))                           MOM01200
      IT=0                                                              MOM01210
      DO 102 JI=1,2                                                     MOM01220
      JIX=JI + II(1) - 1                                                MOM01230
      DO 102 IJ=1,2                                                     MOM01240
      IJX=JIX + (IJ + II(2) - 2)*NZ                                     MOM01250
      DO 102 K=1,2                                                      MOM01260
      KX=K-1 + II(3)                                                    MOM01270
      DO 102 L=1,2                                                      MOM01280
      LX=L-1 + II(4)                                                    MOM01290
      IF(XB(IJX,KX,LX) .GT. -100.) IT=IT+1                              MOM01300
  102 CONTINUE                                                          MOM01310
      IF(IT .GT. ITC) GO TO 6                                           MOM01320
      II(I)=ISAV                                                        MOM01330
  101 CONTINUE                                                          MOM01340
C                                                                       MOM01350
CC    PERFORM THE INTERPOLATION WITH DETERMINED CUBE OF POINTS          MOM01360
C                                                                       MOM01370
      DO 103 I=1,4                                                      MOM01380
      I2=I*2                                                            MOM01390
      I1=I2-1                                                           MOM01400
```

```
      IA=II(I)                                                          MOM01410
      XVAL(I1)=X(IA,I)                                                  MOM01420
      XVAL(I2)=X(IA+1,I)                                                MOM01430
103   CONTINUE                                                          MOM01440
      CALL DTERPS(II,XB,VAL,NZ)                                         MOM01450
      XBAR=DTERPI(4,XI,XVAL,VAL,-100.,W)                                MOM01460
      CALL DTERPS(II,SW,VAL,NZ)                                         MOM01470
      SIGW2=DTERPI(-4,XI,XVAL,VAL,-100.,W)                              MOM01480
      CALL DTERPS(II,SP,VAL,NZ)                                         MOM01490
      SIGP2=DTERPI(-4,XI,XVAL,VAL,-100.,W)                              MOM01500
C                                                                       MOM01510
C     CONVERT THE LOG OF THE NONDIMENSIONAL VALUES INTERPOLATED         MOM01520
C     TO THE USUAL DIMENSIONAL FORM                                     MOM01530
C                                                                       MOM01540
      SCL=U0*H**(M+1.)/SCLU                                             MOM01550
      XBAR=SCL*EXP(XBAR)                                                MOM01560
      SIGW2=SCL*SCL*EXP(SIGW2)                                          MOM01570
      SIGP2=2.*DX20*H*H*EXP(SIGP2)                                      MOM01580
      SIGW2=SIGW2+SIGP2                                                 MOM01590
      SIGP2=DYX0*SIGP2                                                  MOM01600
999   RETURN                                                            MOM01610
      END                                                               MOM01620
```

```
      SUBROUTINE PATH(T,U,XCEN,YCEN,RAD,PLEN)                          PATH0290
C                                                                      PATH0010
C  THIS SUBROUTINE COMPUTES THE PATH LENGTH THROUGH THE SPHERE OR      PATH0020
C  WAKE FOR A HORIZONTAL PATH.                                         PATH0030
C                                                                      PATH0040
C  INPUTS                                                              PATH0050
C                                                                      PATH0060
C     T      - TRANSMITTER COORDINATE IN THE LOCAL COORDINATE SYSTEM    PATH0070
C                                                                      PATH0080
C     U      - UNIT VECTOR ALONG THE LINE CONNECTING THE TRANSMIITTER   PATH0090
C              AND RECEIVER                                            PATH0100
C                                                                      PATH0110
C     XCEN   - X COORDINATE OF THE CENTER OF THE CIRCLE                PATH0120
C                                                                      PATH0130
C     YCEN   - Y COORDINATE OF THE CENTER OF THE CIRCLE                PATH0140
C                                                                      PATH0150
C     RAD    - RADIUS AT THE DESIRED HEIGHT                            PATH0160
C                                                                      PATH0170
C  OUTPUT                                                              PATH0180
C                                                                      PATH0190
C     PLEN   - LENGTH OF THE INTERSECTION OF THE CONE AT HEIGHT T(3)   PATH0200
C              AND THE LINE OF SIIGHT                                  PATH0210
C                                                                      PATH0220
C                                                                      PATH0230
C  FUNCTIONS AND SUBROUTINES NEEDED                                    PATH0240
C                                                                      PATH0250
C     NONE                                                             PATH0260
C                                                                      PATH0270
C*********************************************************************PATH0280
      DIMENSION T(3),U(3)                                              PATH0300
      A=U(1)**2+U(2)**2                                                PATH0310
      PLEN=0.
      X=RAD**2*A-(U(2)*(T(1)-XCEN)-U(1)*(T(2)-YCEN))**2
      IF (X.GT.0.) PLEN=2.*SQRT(X)/A
      RETURN
      END                                                              PATH0400
```

```
      SUBROUTINE PERP(A,B)                                              PER00010
      DIMENSION A(2),B(2)                                               PER00020
C *** B IS ROTATED 90 DEGREES COUNTERCLOCKWISE FROM A                   PER00030
      B(1)=-A(2)                                                        PER00040
      B(2)=A(1)                                                         PER00050
      RETURN                                                            PER00060
      END                                                               PER00070
```

```
      SUBROUTINE PRECL(NATMOS,ZTEMP,TMPMES,ZWND,WNDMES,THWND,PHI,DHDT,   PRE00350
     1           CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT)                     PRE00360
C                                                                       PRE00010
C  ROUTINE TO PRECOMPUTE EXPLOSION PRODUCED DUST CLOUD AND STORE ON     PRE00020
C  EXTERNAL FILE UNIT IFILE                                             PRE00030
C                                                                       PRE00040
C  OUTPUTS                                                              PRE00050
C                                                                       PRE00060
C     XC1  COEFFICIENTS OF QUADRATIC FIT TO SLOPE OF LINE DESCRIBING    PRE00070
C          X DISPLACEMENT OF CONE (ALONGWIND)                           PRE00080
C                                                                       PRE00090
C     XC0  COEFFICIENTS OF QUADRATIC FIT TO CONSTANT TERM OF LINE       PRE00100
C          DESCRIBING X DISPLACEMENT OF CONE (ALONGWIND)                PRE00110
C                                                                       PRE00120
C     Z2   COEFFICIENTS OF QUADRATIC FIT TO HEIGHT OF TOP OF CLOUD      PRE00130
C                                                                       PRE00140
C     RT   COEFFICIENTS OF QUADRATIC FIT TO THE RADIUS OF THE TOP OF THE PRE00150
C          CONE                                                         PRE00160
C                                                                       PRE00170
C     RB   COEFFICIENTS OF THE QUADRATIC FIT TO THE RADIUS OF THE CONE  PRE00180
C          AT A HEIGHT OF THE AVERAGE OF THE DISC SOURCES               PRE00190
C                                                                       PRE00200
C                                                                       PRE00210
C  THE ABOVE OUTPUT ARE THE COEFFICIENTS OF QUADRATIC FITS THROUGH      PRE00220
C  THREE CONSECUTIVE POINT IN TIME. THE QUADRATIC FITS ARE STORED IN    PRE00230
C  COMMON/  MO5 / WITH THE ARRAY TIMES CONTAINING THE LAST TIME         PRE00240
C  OF EACH QUADRATIC PIECE WITH THE FIRST STARTING AT 0.0               PRE00250
C  THE FITS ARE WRITTEN ONTO A FILE INDICATED BY IFILE USING            PRE00260
C  A BINARY WRITE                                                       PRE00270
C  THE FITS ARE STORED SUCH THAT                                        PRE00280
C                                                                       PRE00290
C      F(TIME)=VAR(1,J)*TIME**2 + VAR(2,J)*TIME + VAR(3,J)              PRE00300
C                                                                       PRE00310
C        AND       TIMES(J-1) < TIME < TIMES(J)                       PRE00320
C                                                                       PRE00330
C***********************************************************************PRE00340
      LOGICAL SWITCH,CHANGE,DHDT                                        PRE00370
      REAL KZ,KX                                                        PRE00380
      DIMENSION T(3),FRB(3),FRT(3),FXC1(3),FXC0(3),XB(3),OWF(5,2)       PRE00390
      DIMENSION FZ2(3),ZTEMP(2),TMPMES(2),ZWND(2),WNDMES(2),OWFC(5)     PRE00400
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUPRE00410
      COMMON/OPTION/IOPT,IFILE                                          PRE00420
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)PRE00430
      COMMON/BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),TIM     PRE00440
      COMMON/MO5/DMMMY(604),DMM(600),                                   PRE00450
     +           ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),         PRE00460
     +           RB(3,25),Z2(3,25)                                      PRE00470
      COMMON/VL/VLOAD                                                   PRE00480
      COMMON/SIG/SIG02,SIGC                                             PRE00490
      COMMON/CLOCK/FTIME,TWIND                                          PRE00500
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE          PRE00510
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV                         PRE00520
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)               PRE00530
      COMMON/CARB/RCARB1,RCARB2                                         PRE00540
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     PRE00550
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/                PRE00560
      DATA OWFC/1.,.95,.5,.2,1.E-03/                                    PRE00570
      TINC=1.0                                                          PRE00580
      TPRES=0.0                                                         PRE00590
      ICOUNT=0                                                          PRE00600
      DEL=.001                                                          PRE00610
      T(3)=0.0                                                          PRE00620
      SUM2=0.0                                                          PRE00630
C                                                                       PRE00640
C  FIND AVERAGE OF THE DISC RELEASE HEIGHTS AND THE AVERAGE OF THE      PRE00650
C  INITIAL SPREADS OF THE DISCS                                         PRE00660
C                                                                       PRE00670
      DO 5 J=1,NDSCS                                                    PRE00680
      SUM2=SUM2+R2DSC(J)                                                PRE00690
    5 CONTINUE                                                          PRE00700
```

```
      ZREF=2.0                                                          PRE00710
      ROH2=SUM2/NDSCS                                                   PRE00720
   10 ICOUNT=ICOUNT+1                                                   PRE00730
      DO 50 I=1,3                                                       PRE00740
      T(I)=T(3)+FLOAT(I-1)*TINC                                         PRE00750
C                                                                       PRE00760
CC  FIND THE AVERAGE OF THE MOMENTS AT HEIGHT ZREF                      PRE00770
C                                                                       PRE00780
      CALL AVRG(ZREF,T(I),QTOT,XBAVRG,SIG2X,SIG2Y)                      PRE00790
      IF(QTOT.LT.1.E-10)GO TO 15                                        PRE00800
      SIGX=SQRT(SIG2X+ROH2/2.)                                          PRE00810
      SIGY=SQRT(SIG2Y+ROH2/2.)                                          PRE00820
      FRB(I)=SQRT(SIGX*SIGY)*1.5                                        PRE00830
      XB(1)=XBAVRG                                                      PRE00840
      GO TO 20                                                          PRE00850
   15 CALL WIN(ZREF,UX,V)                                               PRE00860
      XB(1)=UX*T(I)                                                     PRE00870
      FRB(I)=0.0                                                        PRE00880
   20 IF(T(I).GT.TWIND)GO TO 30                                         PRE00890
      CALL RISE(TPRES,T(I),DEL)                                         PRE00900
      FZ2(I)=ZCM+(2./3.)*RSPH                                           PRE00910
      IF(FZ2(I).GT.ZINV)FZ2(I)=ZINV                                     PRE00920
      FRT(I)=RSPH                                                       PRE00930
      XB(2)=XCM                                                         PRE00940
      GO TO 40                                                          PRE00950
   30 XB(2)=XTR+VTR*(T(I)-TTR)                                          PRE00960
      SIGX2=SIG02+2.*KX*(T(I)-TTR)                                      PRE00970
      SIGZ2=SIG02+2.*KZ*(T(I)-TTR)                                      PRE00980
      SIGX=SQRT(SIGX2)                                                  PRE00990
      SIGZ=SQRT(SIGZ2)                                                  PRE01000
      SIG=SQRT(SIGX*SIGZ)                                               PRE01010
      FZ2(I)=ZTR+SIG                                                    PRE01020
      FRT(I)=1.5*SIG                                                    PRE01030
   40 FXC1(I)=(XB(2)-XB(1))/(FZ2(I)-ZREF)                               PRE01040
      FXC0(I)=XB(1)-FXC1(I)*ZREF                                        PRE01050
   50 CONTINUE                                                          PRE01060
C                                                                       PRE01070
CC  COMPUT AND STORE QUADRATIC FITS                                     PRE01080
C                                                                       PRE01090
      TIMES(ICOUNT)=T(3)                                                PRE01100
C                                                                       PRE01110
CC  FIT AND STORE RADIUS AT TOP                                         PRE01120
C                                                                       PRE01130
      CALL FIT(T,FRT,A,B,C)                                             PRE01140
      RT(1,ICOUNT)=A                                                    PRE01150
      RT(2,ICOUNT)=B                                                    PRE01160
      RT(3,ICOUNT)=C                                                    PRE01170
C                                                                       PRE01180
CC  FIT AND STORE RADIUS AT BOTTOM                                      PRE01190
C                                                                       PRE01200
      CALL FIT(T,FRB,A,B,C)                                             PRE01210
      RB(1,ICOUNT)=A                                                    PRE01220
      RB(2,ICOUNT)=B                                                    PRE01230
      RB(3,ICOUNT)=C                                                    PRE01240
C                                                                       PRE01250
CC  FIT AND STORE HEIGHT OF CLOUD                                       PRE01260
C                                                                       PRE01270
      CALL FIT(T,FZ2,A,B,C)                                             PRE01280
      Z2(1,ICOUNT)=A                                                    PRE01290
      Z2(2,ICOUNT)=B                                                    PRE01300
      Z2(3,ICOUNT)=C                                                    PRE01310
C                                                                       PRE01320
CC  FIT AND STORE XC1                                                   PRE01330
C                                                                       PRE01340
      CALL FIT(T,FXC1,A,B,C)                                            PRE01350
      XC1(1,ICOUNT)=A                                                   PRE01360
      XC1(2,ICOUNT)=B                                                   PRE01370
      XC1(3,ICOUNT)=C                                                   PRE01380
C                                                                       PRE01390
CC  FIT AND STORE XC0                                                   PRE01400
```

```
C                                                                       PRE01410
      CALL FIT(T,FXC0,A,B,C)                                            PRE01420
      XC0(1,ICOUNT)=A                                                   PRE01430
      XC0(2,ICOUNT)=B                                                   PRE01440
      XC0(3,ICOUNT)=C                                                   PRE01450
C                                                                       PRE01460
CC  CHECK TO SEE IF PRECOMPUTE CAN BE STOPPED                           PRE01470
C                                                                       PRE01480
      IF(FRT(3).GE.FRB(3))PLEN=2.*FRT(3)                                PRE01490
      IF(FRB(3).GE.FRT(3))PLEN=2.*FRB(3)                                PRE01500
      VOL=(PI/3.)*(FZ2(3)-ZREF)*(FRT(3)**2+FRT(3)*FRB(3)+FRB(3)**2)     PRE01510
      NWL=1                                                             PRE01520
      ACL=(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL))*VLOAD*PLEN/VOL       PRE01530
      TINC=1.2*TINC                                                     PRE01540
      IF(ICOUNT.LT.25.AND.ACL.GT..001)GO TO 10                          PRE01550
C                                                                       PRE01560
CC  WRITE PRECOMPUTED CLOUD INFORMATION ONTO FORTRAN UNIT IFILE WITH    PRE01570
C   A BINARY WRITE                                                      PRE01580
C                                                                       PRE01590
      WRITE(IFILE)NATMOS,ZTEMP(1),TMPMES(1),ZWND(1),WNDMES(1)           PRE01600
      WRITE(IFILE)DHDT,PHI,CHWT,NCHRG,DETDEP,NSOIL,DSOD,SILT,ZINV       PRE01610
      WRITE(IFILE)VLOAD,RCARB1,RCARB2                                   PRE01620
      WRITE(IFILE)ICOUNT                                                PRE01630
      DO 1000 J=1,ICOUNT                                                PRE01640
      WRITE(IFILE) TIMES(J),(RT(I,J),RB(I,J),Z2(I,J),XC0(I,J),          PRE01650
     1             XC1(I,J),I=1,3)                                      PRE01660
 1000 CONTINUE                                                          PRE01670
      REWIND IFILE                                                      PRE01680
  999 RETURN                                                            PRE01690
      END                                                               PRE01700
```

```
      SUBROUTINE PRETRN(TRN,REC,TIME,TRNLOS)                            PRT00220
C                                                                       PRT00010
CCC   COMPUTE THE TRANSMITTANCE FOR THE RANDOM IN SPACE AND TIME        PRT00020
CCC   DISTRIBUTION OF CHARGES                                           PRT00030
CCC                                                                     PRT00040
CCC   INPUTS                                                            PRT00050
CCC                                                                     PRT00060
CCC      TRN    -TRANSMITTER COORDINATES IN LOCAL COORDINATE SYSTEM     PRT00070
CCC                                                                     PRT00080
CCC      REC    -RECEIVER COORDINATES IN LOCAL COODINATE SYSTEM         PRT00090
CCC                                                                     PRT00100
CCC      TIME   -TIME AT WHICH TRANSMITTANCE IS DESIRED                 PRT00110
CCC                                                                     PRT00120
CCC      NWL    -INTEGER INDEX FOR WAVELENGTH                           PRT00130
CCC                                                                     PRT00140
CCC      NSOIL  -SOIL TYPE                                              PRT00150
CCC                                                                     PRT00160
CCC   OUTPUTS                                                           PRT00170
CCC                                                                     PRT00180
CCC      TRNLOS -TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT        PRT00190
CCC                                                                     PRT00200
C*********************************************************************  PRT00210
      LOGICAL TEST                                                      PRT00230
      DIMENSION TRN(3),REC(3),TR(3),RE(3),OWF(5,2),U(3),OWFC(5)         PRT00240
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,          PRT00250
     +           COOR(2,200),TSTAG(200),                                PRT00260
     +             ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),       PRT00270
     1            RB(3,25),Z2(3,25)                                     PRT00280
      COMMON/VL/VLOAD                                                   PRT00290
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL                               PRT00300
      COMMON/CARB/RCARB1,RCARB2                                         PRT00310
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     PRT00320
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/                PRT00330
      DATA OWFC/1.,.95,.5,.2,1.E-03/                                    PRT00340
C                                                                       PRT00350
CCC   PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER     PRT00360
C                                                                       PRT00370
      TEST=.FALSE.                                                      PRT00380
      XNORM=0.0                                                         PRT00390
      DO 10 I=1,3                                                       PRT00400
      RE(I)=REC(I)                                                      PRT00410
      TR(I)=TRN(I)                                                      PRT00420
      U(I)=RE(I)-TR(I)                                                  PRT00430
      XNORM=XNORM+U(I)**2                                               PRT00440
   10 CONTINUE                                                          PRT00450
      XNORM=SQRT(XNORM)                                                 PRT00460
      U(1)=U(1)/XNORM                                                   PRT00470
      U(2)=U(2)/XNORM                                                   PRT00480
      U(3)=U(3)/XNORM                                                   PRT00490
C                                                                       PRT00500
CCC   COMPUTE THE CONTRIBUTION FROM EACH CHARGE TO THE OPTICALLY        PRT00510
CCC   WEIGHTED CONCENTRATION ALONG THE LINE OF SIGHT                    PRT00520
C                                                                       PRT00530
      SUM=0.0                                                           PRT00540
      DO 100 I=1,ITOT                                                   PRT00550
      IF(TIME.LT.TSTAG(I))GO TO 100                                     PRT00560
      TOF=TIME-TSTAG(I)                                                 PRT00570
      IF(TOF.GT.TIMES(ICOUNT))GO TO 100                                 PRT00580
      DO 20 J=1,ICOUNT                                                  PRT00590
      IND=J                                                             PRT00600
      IF(TOF.LE.TIMES(J))GO TO 30                                       PRT00610
   20 CONTINUE                                                          PRT00620
C                                                                       PRT00630
CCC   DETERMINE NECESSARY PARAMETERS DESCRIBING THE CONICAL SHAPE SO THAT PRT00640
CCC   THE LENGTH OF INTERSECTION OF THE LINE OF SIGHT AND CONE CAN BE   PRT00650
CCC   DETERMINED                                                        PRT00660
C                                                                       PRT00670
   30 X0=TOF*(XC0(1,IND)*TOF+XC0(2,IND))+XC0(3,IND)                     PRT00680
      X1=TOF*(XC1(1,IND)*TOF+XC1(2,IND))+XC1(3,IND)                     PRT00690
      HTTOP=TOF*(Z2(1,IND)*TOF+Z2(2,IND))+Z2(3,IND)                     PRT00700
```

```
      RTOP=TOF*(RT(1,IND)*TOF+RT(2,IND))+RT(3,IND)                      PRT00710
      HTBOT=0.0                                                         PRT00720
      RBOT=TOF*(RB(1,IND)*TOF+RB(2,IND))+RB(3,IND)                      PRT00730
      XCEN=DIFF(1,I)+(X1*HTTOP+X0)                                      PRT00740
      YCEN=DIFF(2,I)                                                    PRT00750
      XB=DIFF(1,I)+(X1*HTBOT+X0)                                        PRT00760
      YB=DIFF(2,I)                                                      PRT00770
      IF(ABS(U(3)).LT.1.E-06)GO TO 40                                   PRT00780
C                                                                       PRT00790
CC    COMPUTE THE INTERSECTION LENGTH FOR A NON-HORIZONTAL LINE OF SIGHT PRT00800
C                                                                       PRT00810
      CALL CONLEN(U,TR,HTTOP,HTBOT,XCEN,YCEN,RTOP,RBOT,XB,YB,           PRT00820
     1            XNORM,PLEN)                                           PRT00830
      GO TO 50                                                          PRT00840
C                                                                       PRT00850
CC    COMPUTE THE INTERSECTION LENGTH FOR A HORIZONTAL LINE OF SIGHT    PRT00860
C                                                                       PRT00870
   40 IF(HTTOP.LT.TR(3))GO TO 45                                        PRT00880
      A=U(1)**2+U(2)**2                                                 PRT00890
      B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)                             PRT00900
C                                                                       PRT00910
CC    DETERMINE THE RADIUS,X,AND Y POSITIONS OF THE CONE AT THE         PRT00920
CC    TRANSMITTER HEIGHT                                                PRT00930
C                                                                       PRT00940
      ZETA=TR(3)/HTTOP                                                  PRT00950
      RAD=ZETA*RTOP+(1.-ZETA)*RBOT                                      PRT00960
      XCEN=ZETA*XCEN+(1.-ZETA)*XB                                       PRT00970
      YCEN=ZETA*YCEN+(1.-ZETA)*YB                                       PRT00980
      A=U(1)**2+U(2)**2                                                 PRT00990
      B=U(1)*(TR(1)-XCEN)+U(2)*(TR(2)-YCEN)                             PRT01000
      C=(TR(1)-XCEN)**2+(TR(2)-YCEN)**2-RAD**2                          PRT01010
      X=B**2-A*C                                                        PRT01020
      IF(X.LT.0.0)GO TO 45                                              PRT01030
      P1=(-B+SQRT(X))/A                                                 PRT01040
      P2=(-B-SQRT(X))/A                                                 PRT01050
      IF(P1.GT.XNORM.AND.P2.GT.XNORM)GO TO 45                           PRT01060
      IF(P2.LT.0.0.AND.P1.LT.0.0)GO TO 45                               PRT01070
      PLEN=AMIN1(P1,XNORM)-AMAX1(P2,0.0)                                PRT01080
      GO TO 50                                                          PRT01090
   45 PLEN=0.0                                                          PRT01100
   50 VOL=(PI/3.)*(HTTOP-HTBOT)*(RTOP**2+RTOP*RBOT+RBOT**2)             PRT01110
      CONT=VLOAD*PLEN/VOL                                               PRT01120
      SUM=SUM+CONT                                                      PRT01130
      ACLSKT=0.0                                                        PRT01140
      ACLSPH=0.0                                                        PRT01150
      CALL TRNCHK(ACLSKT,SUM,ACLSPH)                                    PRT01160
      IF(TEST)GO TO 998                                                 PRT01170
  100 CONTINUE                                                          PRT01180
      TRNLOS=EXP(-SUM*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL)))         PRT01190
      GO TO 999                                                         PRT01200
  998 TRNLOS=0.0                                                        PRT01210
  999 RETURN                                                            PRT01220
      END                                                               PRT01230
```

```
      SUBROUTINE PREVEH(NSOIL,NWL)                                          PRV00360
C                                                                           PRV00010
CCCCC  ROUTINE FOR PRECOMPUTING VEHICLE GENERATED DUST CLOUD                PRV00020
CCCCC                                                                       PRV00030
CCCCC                                                                       PRV00040
CCCCC  INPUTS                                                               PRV00050
CCCCC                                                                       PRV00060
CCCCC     NSOIL - SOIL TYPE  (SEE DRTRAN FOR DETAILS)                       PRV00070
CCCCC                                                                       PRV00080
CCCCC     NWL   - WAVELENGTH INDEX  (SEE DRTRAN)                            PRV00090
CCCCC                                                                       PRV00100
CCCCC  OUTPUTS                                                              PRV00110
CCCCC                                                                       PRV00120
CCCCC     RT    - COEFFICIENTS OF QUADRATIC FIT TO RADIUS OF THE CLOUD       PRV00130
CCCCC                                                                       PRV00140
CCCCC     Z2    - COEFFICIENTS OF QUADRATIC FIT TO HEIGHT OF THE CLOUD       PRV00150
CCCCC                                                                       PRV00160
CCCCC     XC1   - COEFFICIENTS OF QUADRATIC FIT TO SLOPE OF LINE DESCRIBING  PRV00170
CCCCC             X DISPLACEMENT OF CONE (ALONGWIND)                        PRV00180
CCCCC                                                                       PRV00190
CCCCC     XC0   - COEFFICIENTS OF QUADRATIC FIT TO CONSTANT TERM OF LINE    PRV00200
CCCCC             DESCRIBING X DISPLACEMENT OF THE CONE  (ALONGWIND)        PRV00210
CCCCC                                                                       PRV00220
CCCCC  THE ABOVE OUTPUT ARE THE COEFFICIENTS OF QUADRATIC FITS THROUGH      PRV00230
CCCCC  THREE CONSECUTIVE POINTS IN TIME.  THE QUADRATIC FITS ARE STORED IN  PRV00240
CCCCC  COMMON/  MO5 / WITH THE ARRAY TIMES CONTAINING THE LAST TIME OF THAT PRV00250
CCCCC  INTERVAL.  THE FITS ARE WRITTEN ONTO A FILE INDICATED BY IFILE USING PRV00260
CCCCC  A BINARY WRITE.                                                      PRV00270
CCCCC                                                                       PRV00280
CCCCC  THE FITS ARE STORED SUCH THAT                                        PRV00290
CCCCC                                                                       PRV00300
CCCCC          F(TIME)=VAR(1,J)*TIME**2 + VAR(2,J)*TIME + VAR(3,J)          PRV00310
CCCCC                                                                       PRV00320
CCCCC     AND       TIMES(J-1) < TIME < TIMES(J)                            PRV00330
CCCCC                                                                       PRV00340
C*************************************************************************  PRV00350
      LOGICAL HORIZ                                                         PRV00370
      DIMENSION                          OWF(5,2),T(3),FR(3),FZ2(3)         PRV00380
      DIMENSION FXC1(3),FXC0(3),XB(2)                                       PRV00390
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)    PRV00400
      COMMON/PRE/ZT0,RT2DZ                                                  PRV00410
      COMMON/MODE/HORIZ                                                     PRV00420
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,DMM(600),     PRV00430
     +                ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),       PRV00440
     +                RB(3,25),Z2(3,25)                                    PRV00450
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU    PRV00460
      COMMON/PRTINF/R0,VGRAV(3),NPRTS                                       PRV00470
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)                   PRV00480
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                         PRV00490
C                                                                           PRV00500
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/                    PRV00510
C                                                                           PRV00520
CC  SET UP CORRECT PARAMETERS SO THAT CWIND CAN BE USED                     PRV00530
C                                                                           PRV00540
      NARY=1                                                                PRV00550
      ITOT=1                                                                PRV00560
      NTOT=1                                                                PRV00570
      NCHTOT=1                                                              PRV00580
      HORIZ=.TRUE.                                                          PRV00590
      ONEM=-1.                                                              PRV00600
      COSTH=1.0                                                             PRV00610
      SINTH=0.0                                                             PRV00620
      COSTH2=COSTH**2                                                       PRV00630
      SINTH2=SINTH**2                                                       PRV00640
      SCRN(1)=SINTH                                                         PRV00650
      SCRN(2)=-COSTH                                                        PRV00660
      X=0.0                                                                 PRV00670
      Y=0.0                                                                 PRV00680
      TMAX=1000.                                                            PRV00690
C                                                                           PRV00700
```

```
C   COMPUTE R , Z2 , XC1 , XC0 AT THREE CONSECUTIVE TIMES WITH SPACING    PRV00710
C   TINC AND THEN CALL FIT , WHICH CALCULATES A QUADRATIC FIT TO THESE    PRV00720
C   POINTS AND STORE THEM IN COMMON /QUADFT/                              PRV00730
C                                                                         PRV00740
      TINC=1.0                                                            PRV00750
      ICOUNT=0                                                            PRV00760
      T(3)=0.0                                                            PRV00770
   10 ICOUNT=ICOUNT+1                                                     PRV00780
      DO 20 I=1,3                                                         PRV00790
      T(I)=T(3)+FLOAT(I-1)*TINC                                           PRV00800
      ZREF=ZT0+SQRT(T(I))*RT2DZ                                           PRV00810
      TOF=T(I)-TDSC(NDSCS)                                                PRV00820
      CALL MOMENT(VGRAV,ZREF,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)              PRV00830
C                                                                         PRV00840
C   COMPUTE R THE RADIUS OF THE CLOUD                                     PRV00850
C                                                                         PRV00860
      R0H2=R2DSC(1)                                                       PRV00870
      SIGW=SQRT(SIGW2+R0H2/2.)                                            PRV00880
      SIGP=SQRT(SIGP2+R0H2/2.)                                            PRV00890
      FR(I)=1.5*SQRT(SIGW*SIGP)                                           PRV00900
      ACL=CWIND(X,Y,ZREF,T(I))*OWF(NWL,NSOIL)                             PRV00910
C                                                                         PRV00920
C   COMPUTE Z2 APPROXIMATE HEIGHT OF THE CLOUD                            PRV00930
C                                                                         PRV00940
      FZ2(I)=(2.*QDSC(1,1))/PI/FR(I)/ACL                                  PRV00950
      Z22=FZ2(I)                                                          PRV00960
C                                                                         PRV00970
C   COMPUTE THE X POSITION OF THE CLOUD AT A HEIGHT OF Z2 AND A HEIGHT    PRV00980
C   OF 1 METER.                                                           PRV00990
C                                                                         PRV01000
      CALL MOMENT(VGRAV,Z22,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)               PRV01010
      XB(2)=XBAR+XDSC(1)                                                  PRV01020
      Z1=1.0                                                              PRV01030
      CALL MOMENT(VGRAV,Z1,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)                PRV01040
      XB(1)=XBAR+XDSC(1)                                                  PRV01050
      FXC1(I)=(XB(2)-XB(1))/(Z22-Z1)                                      PRV01060
      FXC0(I)=XB(1)-FXC1(I)                                               PRV01070
   20 CONTINUE                                                            PRV01080
C                                                                         PRV01090
C   COMPUTE AND STORE THE QUADRATIC FITS                                  PRV01100
C                                                                         PRV01110
      TIMES(ICOUNT)=T(3)                                                  PRV01120
C                                                                         PRV01130
C   FIT AND STORE THE CLOUD RADIUS                                        PRV01140
C                                                                         PRV01150
      CALL FIT(T,FR,A,B,C)                                                PRV01160
      RT(1,ICOUNT)=A                                                      PRV01170
      RT(2,ICOUNT)=B                                                      PRV01180
      RT(3,ICOUNT)=C                                                      PRV01190
C                                                                         PRV01200
C   FIT AND STORE Z2, APPROXIMATE CLOUD HEIGHT                            PRV01210
C                                                                         PRV01220
      CALL FIT(T,FZ2,A,B,C)                                               PRV01230
      Z2(1,ICOUNT)=A                                                      PRV01240
      Z2(2,ICOUNT)=B                                                      PRV01250
      Z2(3,ICOUNT)=C                                                      PRV01260
C                                                                         PRV01270
C   FIT AND STORE XC1                                                     PRV01280
C                                                                         PRV01290
      CALL FIT(T,FXC1,A,B,C)                                              PRV01300
      XC1(1,ICOUNT)=A                                                     PRV01310
      XC1(2,ICOUNT)=B                                                     PRV01320
      XC1(3,ICOUNT)=C                                                     PRV01330
C                                                                         PRV01340
C   FIT AND STORE XC0                                                     PRV01350
C                                                                         PRV01360
      CALL FIT(T,FXC0,A,B,C)                                              PRV01370
      XC0(1,ICOUNT)=A                                                     PRV01380
      XC0(2,ICOUNT)=B                                                     PRV01390
      XC0(3,ICOUNT)=C                                                     PRV01400
```

```
      TINC=1.2*TINC                                                     PRV01410
      IF(ICOUNT.LT.20.AND.T(3).LT.TMAX) GO TO 10                        PRV01420
      RETURN                                                            PRV01430
      END                                                               PRV01440
```

```
      SUBROUTINE RISE(TPRES,TNEXT,DEL)                                  RIS00010
      REAL M,NDIF,KZ,KX                                                 RIS00020
      LOGICAL SWITCH,CHANGE                                             RIS00030
      DIMENSION WK(12,6)                                                RIS00040
      COMMON/BUOYCL/ Y(8),SPHNS(3),RISTIM                               RIS00050
      COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,M,NDIF,ZINV                      RIS00060
      COMMON /CLOCK/ TIME,TWIND                                         RIS00070
      COMMON/STARS/USTAR,TSTAR,ZSTAR                                    RIS00080
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                                RIS00090
      COMMON/TRAN/YTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE          RIS00100
      COMMON/SIG/SIG02,SIGC                                             RIS00110
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTURIS00120
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),          RIS00130
     1QDSC(20,3)                                                        RIS00140
      COMMON/BURST/ACCEL,TBURST                                         RIS00150
      DATA HMIN,ACCURC,WK,N,ND/.001,.001,72*0.,8,12/                    RIS00160
C ******************************************************************  RIS00170
C                                                                       RIS00180
C                                                                       RIS00190
C     PURPOSE                                                           RIS00200
C                                                                       RIS00210
C     THIS ROUTINE CALLS A RUNGA KUTTA ROUTINE TO INTEGRATE IN TIME     RIS00220
C     THE EQUATIONS FOR THE RISE OF A BUOYANT CLOUD BEGINNING AT TPRES  RIS00230
C     ENDING AT TNEXT UNLESS THE CONDITION FOR SWITCHING TO THE WIND    RIS00240
C     DISPERSION MODEL IS ENCOUNTERED IN WHICH CONVRT IS CALLED.        RIS00250
C     SEE SUBROUTINE DIFEQ FOR THE DEFINITIONS OF Y(I).                 RIS00260
C                                                                       RIS00270
C                                                                       RIS00280
C      ARGUMENTS                                                        RIS00290
C                                                                       RIS00300
C      TPRES      AS INPUT TPRES IS THE INITIAL TIME OF THIS SEGMENT OF RIS00310
C                 INTEGRATION AND IS RETURNED WITH THE VALUE OF THE LAST RIS00320
C                 SUCCESSFUL INTEGRATION STEP.                          RIS00330
C      TNEXT      THE ENDTPOINT OF THE TIME INTERVAL WHICH IS INPUT.     RIS00340
C                                                                       RIS00350
C     REQUIRED SUBROUTINES                                              RIS00360
C                                                                       RIS00370
C          RKM      A RUNGA-KUTTA-MERSON INTEGRATION ROUTINE           RIS00380
C                                                                       RIS00390
C          CONVRT   A SUBROUTINE WHICH CONVERTS THE CURRENT BUOYANT    RIS00400
C                   DUST CLOUD TO A NUMBER OF DISC SOURCES FOR THE     RIS00410
C                   WIND DISPERSION MODEL.  A GAP TIME DURING WHICH THE RIS00420
C                   BUOYANT MODEL IS CONTINUED IS COMPUTED.            RIS00430
C                                                                       RIS00440
C          WNDCAL   COMPUTES SCALED WIND SPEED AT A SPECIFIED HEIGHT   RIS00450
C                                                                       RIS00460
C          DIFFUS   COMPUTES DIFFUSIVITY AT A SPECIFIED HEIGHT         RIS00470
C                                                                       RIS00480
C     CALLED BY DUSTCL                                                  RIS00490
C                                                                       RIS00500
C ******************************************* ***********************  RIS00510
      IF(TNEXT.GT.TWIND)GO TO 999                                       RIS00520
      SWITCH=.FALSE.                                                    RIS00530
      CHANGE=.FALSE.                                                    RIS00540
      T2=TPRES                                                          RIS00550
C                                                                       RIS00560
C     PERFORM INTEGRATION IN SEGMENTS OF TIME                           RIS00570
C                                                                       RIS00580
   10 DO 20 NT=1,300                                                    RIS00590
      T1=T2                                                             RIS00600
      T2=1.2*T1                                                         RIS00610
      IF(T2.LE.0.)T2=.5                                                 RIS00620
      IF(T2.GT.TNEXT)T2=TNEXT                                           RIS00630
      IF (DEL.LT.HMIN)DEL=HMIN                                          RIS00640
      CALL RKM(N,T1,T2,Y,HMIN,DEL,ACCURC,WK,ND)                         RIS00650
C                                                                       RIS00660
C     CHECK TO SEE IF CLOUD GROWTH IS DOMINATED BY WIND DIFFUSION       RIS00670
C     OVER BUOYANT RISE BY COMPARING WIND DIFFUSIVITY, DIFW, TO         RIS00680
C     THE EFFECTIVE BUOYANT DIFFUSIVITY, DIFB AND IF THE HEIGHT         RIS00690
C     OF THE CENTER OF MASS IS LESS THAN ZSTAR SWITCH TO THE            RIS00700
```

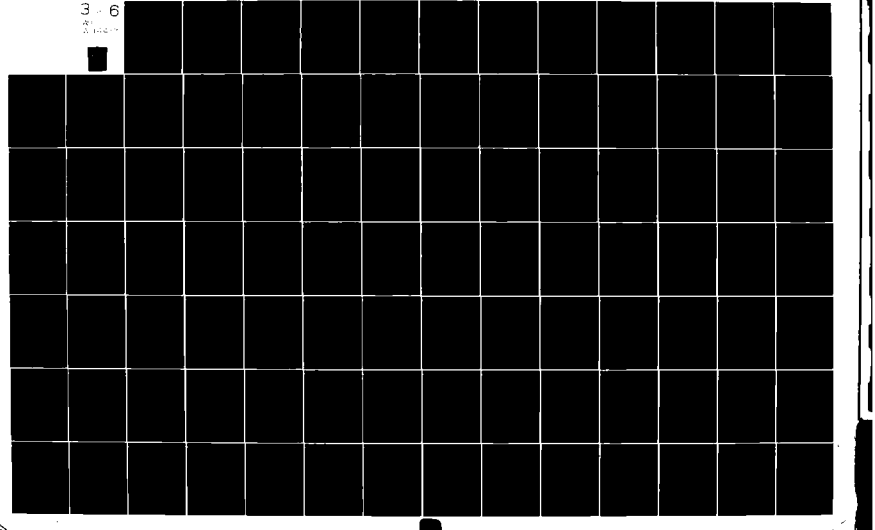
```
C       WIND MODEL.                                                          RIS00710
C                                                                            RIS00720
        IF(T2.LT.TBURST)GO TO 15                                             RIS00730
   5    DIFB=ABS(.1111*Y(1)*Y(3))                                            RIS00740
        TOP=Y(6)+Y(1)                                                        RIS00750
        DIFW=DIFFUS(Z0,ZL,TOP)                                               RIS00760
        IF(DIFB.GT.DIFW)GO TO 15                                             RIS00770
        IF(TOP.GT.ZSTAR)SWITCH=.TRUE.                                        RIS00780
        IF(TOP.GT.ZSTAR)GO TO 15                                             RIS00790
        VTR=USTAR*WNDCAL(Z0,ZL,Y(6))                                         RIS00800
        KZ=DIFFUS(Z0,ZL,Y(6))                                                RIS00810
        KX=DXZ0*KZ                                                           RIS00820
        TTR=T2                                                               RIS00830
        XTR=Y(4)                                                             RIS00840
        ZTR=Y(6)                                                             RIS00850
        CHANGE=.TRUE.                                                        RIS00860
        CALL CONVRT(T2)                                                      RIS00870
        GO TO 200                                                            RIS00880
  15    CONTINUE                                                             RIS00890
        IF(T2.GE.TNEXT)GO TO 200                                             RIS00900
        IF(T2.GT.300.)GO TO 99                                               RIS00910
  20    CONTINUE                                                             RIS00920
  99    WRITE(IOOUT,98)                                                      RIS00930
  98    FORMAT(54H *** DIRTRAN ERROR - 5 MINUTE CUT-OFF ON BUOYANT RISE )    RIS00940
        STOP                                                                 RIS00950
 200    TPRES=T2                                                             RIS00960
        RISTIM=TPRES                                                         RIS00970
 999    RETURN                                                               RIS00980
        END                                                                  RIS00990
```

```
      SUBROUTINE RKM(N,XL,XU,Y,HMIN,DEL,ACCURC,WK,ND)                    RKM00010
C                                                                        RKM00020
C                                                                        RKM00030
C  NUMERICAL INTEGRATION ROUTINE FOR SYSTEMS OF ODE'S                    RKM00040
C        USING THE RUNGE-KUTTA-MERSON TECHNIQUE                          RKM00050
C                                                                        RKM00060
C                                                                        RKM00070
C  INPUT PARAMETERS                                                      RKM00080
C                                                                        RKM00090
C       N - NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS                 RKM00100
C      XL - INITIAL ABCISSA OF THE INTERVAL                              RKM00110
C      XU - THE FINAL ABCISSA OF THE INTEGRATION INTERVAL                RKM00120
C       Y - A SINGLY DIMENSIONED ARRAY OF LENGTH N.  WHEN                RKM00130
C             RKM IS CALLED IT MUST CONTAIN THE VALUES OF                RKM00140
C             THE DEPENDENT VARIABLES AT XL.  UPON RETURN                RKM00150
C             TO THE CALLING PROGRAM Y CONTAINS THE VALUES               RKM00160
C             OF THE DEPENDENT VARIABLES AT XU.                          RKM00170
C    HMIN - THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE              RKM00180
C             INTEGRATION                                                RKM00190
C     DEL - THE INITIAL ESTIMATE OF THE STEP SIZE AND UPON               RKM00200
C             RETURN TO THE CALLING PROGRAM DEL CONTAINS THE             RKM00210
C             FINAL STEP SIZE USED.  THIS VALUE SHOULD BE USED           RKM00220
C             IN THE NEXT CALL TO PRODUCE AN EFFICIENT INTEGRATION.      RKM00230
C             DEL IS RETURNED WITH THE VALUE ZERO IF IT HAS              RKM00240
C             BEEN HALVED BELOW HMIN.                                    RKM00250
C  ACCURC - PREASSIGNED ACCURACY WHICH IS ALSO USED IN ADJUSTING         RKM00260
C             THE STEP SIZE.                                             RKM00270
C      WK - AT LEAST A BLOCK OF N BY 6 FLOATING POINT LOCATIONS          RKM00280
C             USED FOR A WORK ARRAY.                                     RKM00290
C      ND - THE DIMENSION OF ARRAYS Y AND WK.                            RKM00300
C                                                                        RKM00310
C                                                                        RKM00320
C     IT IS REQUIRED THAT THE USER OF RKM WRITE A SUBROUTINE             RKM00330
C  DEFINING THE DIFFERENTIAL EQUATIONS.  THE SUBROUTINE                  RKM00340
C  STATEMENT SHOULD LOOK LIKE - SUBROUTINE DIFEQ(N,X,Y,YP) .             RKM00350
C                                                                        RKM00360
C  WHERE                                                                 RKM00370
C        N - THE NUMBER OF EQUATIONS                                     RKM00380
C        X - THE INDEPENDENT VARIABLE                                    RKM00390
C        Y - SINGLY DIMENSIONED ARRAY OF DEPENDENT VARIABLES             RKM00400
C       YP - SINGLY DIMENSIONED ARRAY OF THE RATES OF Y AT X             RKM00410
C                 YP(I) = D Y(I)/DX                                      RKM00420
C                                                                        RKM00430
C                                                                        RKM00440
      DIMENSION Y(ND),WK(ND,6)                                           RKM00450
      LOGICAL FIRST,QUIT                                                 RKM00460
C                                                                        RKM00470
C  SET UP NEEDED VARIABLES UPON ENTRY                                    RKM00480
C                                                                        RKM00490
      XN=XL                                                              RKM00500
      H=DEL                                                              RKM00510
      FIRST=.TRUE.                                                       RKM00520
      QUIT=.FALSE.                                                       RKM00530
C                                                                        RKM00540
C  CHECK IF XN IS CLOSE TO XU                                            RKM00550
C                                                                        RKM00560
   20 IF(XN+H .LT. XU) GO TO 30                                          RKM00570
      DEL=H                                                              RKM00580
      H=XU-XN                                                            RKM00590
      QUIT=.TRUE.                                                        RKM00600
      IF(FIRST) DEL=H                                                    RKM00610
C                                                                        RKM00620
C  MAKE FIRST CALL TO DIFEQ AT THE BEGINNING OF INTERVAL                 RKM00630
C                                                                        RKM00640
   30 CALL DIFEQ(N,XN,Y,WK(1,1))                                         RKM00650
C                                                                        RKM00660
C  PERFORM THE RUNGE-KUTTA-MERSON ALGORITHM                              RKM00670
C                                                                        RKM00680
   40 H3=H/3.                                                            RKM00690
      DO 50 I=1,N                                                        RKM00700
```

```
      WK(I,3)=H3*WK(I,1)                                                RKM00710
   50 WK(I,6)=Y(I)+WK(I,3)                                              RKM00720
      CALL DIFEQ(N,XN+H3,WK(1,6),WK(1,2))                               RKM00730
      DO 60 I=1,N                                                       RKM00740
   60 WK(I,6)=Y(I)+(WK(I,3)+H3*WK(I,2))/2.                              RKM00750
      CALL DIFEQ(N,XN+H3,WK(1,6),WK(1,2))                               RKM00760
      DO 70 I=1,N                                                       RKM00770
      WK(I,4)=H3*WK(I,2)                                                RKM00780
   70 WK(I,6)=Y(I)+(3.*WK(I,3)+9.*WK(I,4))/8.                           RKM00790
      CALL DIFEQ(N,XN+H/2.,WK(1,6),WK(1,2))                             RKM00800
      DO 80 I=1,N                                                       RKM00810
      WK(I,5)=H3*WK(I,2)                                                RKM00820
   80 WK(I,6)=Y(I)+(3.*WK(I,3)-9.*WK(I,4)+12.*WK(I,5))/2.               RKM00830
      CALL DIFEQ(N,XN+H,WK(1,6),WK(1,2))                                RKM00840
C                                                                       RKM00850
CC  FIND THE LARGEST RELATIVE ERROR                                     RKM00860
C                                                                       RKM00870
      TEST=0.                                                           RKM00880
      DO 90 I=1,N                                                       RKM00890
      YX=Y(I)                                                           RKM00900
      IF(YX .EQ. 0.) YX=ACCURC                                          RKM00910
      E=((WK(I,3)-9.*WK(I,4)/2.+4.*WK(I,5)-H3*WK(I,2)/2.)/5.)/YX        RKM00920
   90 TEST=AMAX1(TEST,ABS(E))                                           RKM00930
      FIRST=.FALSE.                                                     RKM00940
      IF(TEST .LT. ACCURC) GO TO 100                                    RKM00950
C                                                                       RKM00960
CC  IF THE LARGEST ERROR IS GREATER THAN ACCURC HALF THE STEP           RKM00970
CC    SIZE AND TRY AGAIN.                                               RKM00980
C                                                                       RKM00990
      H=H/2.                                                            RKM01000
      IF(H .LT. HMIN) GO TO 10                                          RKM01010
      QUIT=.FALSE.                                                      RKM01020
      GO TO 40                                                          RKM01030
C                                                                       RKM01040
CC  TRUNCATION ERROR LESS THAN ACCURC, RESET THE Y ARRAY TO             RKM01050
CC    SET UP FOR THE NEXT INTERVAL                                      RKM01060
C                                                                       RKM01070
  100 XN=XN+H                                                           RKM01080
      DO 110 I=1,N                                                      RKM01090
  110 Y(I)=Y(I)+(WK(I,3)+4.*WK(I,5)+H3*WK(I,2))/2.                      RKM01100
C                                                                       RKM01110
CC  CHECK FOR STEP SIZE DOUBLING.  DOUBLE IF LARGEST RELATIVE           RKM01120
CC    ERROR IS 32 TIMES LESS THAN ACCURC.                               RKM01130
C                                                                       RKM01140
      IF(.NOT.(TEST .GE. ACCURC/32. .OR. QUIT)) H=H+H                   RKM01150
      IF(.NOT. QUIT) GO TO 20                                           RKM01160
      RETURN                                                            RKM01170
C                                                                       RKM01180
CC  THE VALUE OF H (DEL) IS LESS THAN THE SPECIFIED MINIMUM.            RKM01190
CC    REPORT THIS AND ERROR OUT.                                        RKM01200
C                                                                       RKM01210
   10 CONTINUE                                                          RKM01220
 1000 FORMAT('1  H BELOW HMIN'/'0     INTEGRATION ABORTED')             RKM01230
      DEL=0.                                                            RKM01240
      RETURN                                                            RKM01250
      END                                                               RKM01260
```

```
      SUBROUTINE SETUP(NCHS,SRCBAS,SIDE1,SIDE2,TRNFRM)                  SET00010
      DIMENSION NCHS(2),SRCBAS(2),SIDE1(2),SIDE2(2),TRNFRM(2,2),REF(2)  SET00020
      COMMON /ARRAY/OVRLAP,AREA,PERIM,PRJARY,CENDIF                     SET00030
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,          SET00040
     +           COOR(2,200),TSTAG(200),DMMY(401)                       SET00050
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSET00060
C*****************************************************************    SET00070
C                                                                       SET00080
C     PURPOSE                                                           SET00090
C                                                                       SET00100
C       TO CONVERT THE USER DEFINED COORDINATES OF THE CHARGES TO THE   SET00110
C       LOCAL COORDINATE SYSTEM.                                        SET00120
C                                                                       SET00130
C     INPUTS                                                            SET00140
C                                                                       SET00150
C        NCHS - SINGLY DIMENSIONED ARRAY CONTAINING THE NUMBER OF       SET00160
C               CHARGES.                                                SET00170
C                                                                       SET00180
C        SRCBAS - A REFERENCE CHARGE IN THE USER DEFINED COORDINATES.   SET00190
C                                                                       SET00200
C        SIDE1,SIDE2 - VECTORS DESCRIBING THE BOUNDING PARRALLELOGRAM.  SET00210
C                                                                       SET00220
C        TRNFRM - COORDINATE SYSTEM TRANSFORMATION MATRIX.              SET00230
C                                                                       SET00240
C                                                                       SET00250
C    OUTPUTS  RETURNED IN COMMON /ARRAY/ AND /SEPRTN/                   SET00260
C                                                                       SET00270
C       DIFF- DOUBLY DIMENSIONED ARRAY CONTAINING THE CHARGE COORDINATESSET00280
C             IN THE LOCAL COORDINATE SYSTEM.                           SET00290
C                                                                       SET00300
C       ITOT- TOTAL NUMBER OF CHARGES.                                  SET00310
C                                                                       SET00320
C       NCHTOT- WHEN NARY=1 OR 2 THE TOTAL NUMBER OF CHARGES. WHEN      SET00330
C               WHEN NARY=3 IS SET =1.                                  SET00340
C                                                                       SET00350
C       NTOT- WHEN NARY=1 OR 2 IS SET =1 AND WHEN NARY=3 IS THE TOTAL   SET00360
C             NUMBER OF CHARGES.                                        SET00370
C                                                                       SET00380
C                                                                       SET00390
C    SUBROUTINES AND FUNCTIONS                                          SET00400
C                                                                       SET00410
C       UNIT   COMPUTES THE UNIT VECTOR OF A GIVEN VECTOR               SET00420
C*****************************************************************    SET00430
      IF(NARY.NE.1)GO TO 4                                              SET00440
      NCHTOT=NCHS(1)*NCHS(2)                                            SET00450
      NTOT=1                                                            SET00460
      ITOT=NCHTOT                                                       SET00470
      GO TO 6                                                           SET00480
    4 IF(NARY.NE.2)GO TO 5                                              SET00490
      NCHTOT=NCHS(1)                                                    SET00500
      NTOT=1                                                            SET00510
      ITOT=NCHTOT                                                       SET00520
      GO TO 6                                                           SET00530
    5 IF(NARY.NE.3)GO TO 998                                            SET00540
      NCHTOT=1                                                          SET00550
      NTOT=NCHS(1)                                                      SET00560
      ITOT=NTOT                                                         SET00570
    6 CONTINUE                                                          SET00580
C                                                                       SET00590
C     DETERMINE THE COORDINATE OF THE REFERENCE CHARGE IN THE INTERNAL   SET00600
C     COORDINATE SYSTEM                                                 SET00610
C                                                                       SET00620
      DO 20 I=1,2                                                       SET00630
      REF(I)=0.0                                                        SET00640
      DO 10 J=1,2                                                       SET00650
      REF(I)=REF(I)+TRNFRM(I,J)*SRCBAS(J)                               SET00660
   10 CONTINUE                                                          SET00670
   20 CONTINUE                                                          SET00680
      DO 40 I=1,2                                                       SET00690
      DO 30 J=1,ITOT                                                    SET00700
```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 9/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL



```
      DIFF(I,J)=0.0                                                     SET00710
   30 CONTINUE                                                          SET00720
   40 CONTINUE                                                          SET00730
      IF(NARY.GT.1)GO TO 90                                             SET00740
      NM=0                                                              SET00750
      NC1=NCHS(1)                                                       SET00760
      NC2=NCHS(2)                                                       SET00770
C                                                                       SET00780
CC    COMPUTE LOCATIONS OF CHARGES FOR INTERNAL COORDINATE SYSTEM FOR    SET00790
CC    UNIFORMLY DISTRIBUTED CHARGES.                                    SET00800
C                                                                       SET00810
      DO 80 M=1,NC2                                                     SET00820
      DO 70 N=1,NC1                                                     SET00830
      NM=NM+1                                                           SET00840
      DO 60 I=1,2                                                       SET00850
      DO 50 J=1,2                                                       SET00860
      DIFF(I,NM)=DIFF(I,NM)+FLOAT(N-1)*TRNFRM(I,J)*SIDE1(J)             SET00870
     1 +FLOAT(M-1)*TRNFRM(I,J)*SIDE2(J)                                 SET00880
   50 CONTINUE                                                          SET00890
   60 CONTINUE                                                          SET00900
   70 CONTINUE                                                          SET00910
   80 CONTINUE                                                          SET00920
      GO TO 999                                                         SET00930
   90 CONTINUE                                                          SET00940
C                                                                       SET00950
CC    TRANSFORM CHARGE LOCATIONS TO LOCAL COORDINATE SYSTEM FOR RANDOM   SET00960
CC    CHARGES.                                                          SET00970
C                                                                       SET00980
      NC1=NCHS(1)                                                       SET00990
      DO 120 M=1,NC1                                                    SET01000
      DO 110 I=1,2                                                      SET01010
      DO 100 J=1,2                                                      SET01020
      DIFF(I,M)=DIFF(I,M)+TRNFRM(I,J)*COOR(J,M)                         SET01030
  100 CONTINUE                                                          SET01040
      DIFF(I,M)=DIFF(I,M)-REF(I)                                        SET01050
  110 CONTINUE                                                          SET01060
  120 CONTINUE                                                          SET01070
      GO TO 999                                                         SET01080
  998 WRITE(IOOUT,778)                                                  SET01090
  778 FORMAT(5X,23H *** NARY OUT OF RANGE  )                            SET01100
  999 RETURN                                                            SET01110
      END                                                               SET01120
```

```
      SUBROUTINE SOURCE(W,NCHRG,DD,NSOIL,DSOD)                          SORC0010
C ********************************************************            SORC0020
C                                                                       SORC0030
C                                                                       SORC0040
C     PURPOSE                                                           SORC0050
C                                                                       SORC0060
C     TO CALCULATE EXPLOSIVE DUST SOURCE TERM FOR THE                   SORC0070
C             DIRTRAN CODE                                              SORC0080
C                                                                       SORC0090
C     INPUT                                                             SORC0100
C                                                                       SORC0110
C       W        THE WEIGHT OF THE CHARGE IN KG-TNT                     SORC0120
C       DD       DETONATION DEPTH IN METERS                             SORC0130
C       NSOIL    INTEGER SOIL INDEX                                     SORC0140
C       DSOD     DEPTH OF SOD IN METERS                                 SORC0150
C                                                                       SORC0160
C     OUTPUT  (RETURNED IN COMMON /PRTINF/ , /BUOYCL/ AND /CARB/        SORC0170
C                                                                       SORC0180
C       R0       INITIAL CLOUD RADIUS IN METERS                         SORC0190
C       VGRAV    SINGLY DIMENSIONED ARRAY CONTAINING OPTICALLY WEIGHTED  SORC0200
C                AVERAGE SETTLING VELOCITIES FOR EACH SIZE RANGE IN     SORC0210
C                THE PARTICLE DISTRIBUTION  (METERS/SEC)                SORC0220
C       NPRTS    THE NUMBER OF SIZE RANGES IN THE PARTITIONING OF THE   SORC0230
C                PARTICLE SIZE SPECTRUM                                 SORC0240
C       RSPH     THE INITIAL RADIUS OF THE CLOUD IN METERS              SORC0250
C       DELT     THE INITIAL DIFFERENCE IN TEMPERATURE BETWEEN THE CLOUD SORC0260
C                AND SURROUNDINGS  (DEGREES KELVIN)                     SORC0270
C       VZSPH    THE INITIAL VERTICAL VELOCITY OF THE CLOUD (M/S)       SORC0280
C       XCMSPH   INITIAL HORIZONTAL POSITION OF THE CLOUD (METERS)      SORC0290
C       YCMSPH   INITIAL Y POSITION OF THE CLOUD (METERS)               SORC0300
C       ZCMSPH   INITIAL HEIGHT OF THE CLOUD (METERS)                   SORC0310
C       XTOP     INITIAL X POSITION OF THE TOP OF THE CLOUD (METERS)    SORC0320
C       YTOP     INITIAL Y POSITION OF THE TOP OF THE CLOUD (METERS)    SORC0330
C       RISTIM   TIME LAPSED SINCE DETONATION IN SECONDS                SORC0340
C                                                                       SORC0350
C       RCARB1   PORTION OF BUOYANT CLOUD WHICH IS DIRT PARTICLES       SORC0360
C                                                                       SORC0370
C       RCARB2   PORTION OF BUOYANT CLOUD WHICH IS CARBON PARTICLES     SORC0380
C                                                                       SORC0390
C                                                                       SORC0400
C     CALLED BY DUSTCL                                                  SORC0410
C                                                                       SORC0420
C     SUBROUTINES AND FUNCTIONS                                         SORC0430
C                                                                       SORC0440
C     NONE                                                              SORC0450
C                                                                       SORC0460
C ********************************************************************* SORC0470
      LOGICAL HORIZ,ONCE                                                SORC0480
      DIMENSION CR(5,7),CD(5,7),OWML(3,4),OWSV(3,4),PRTTN(4)            SORC0490
      DIMENSION S(3),BURHTR(5),WTRAT(5)                                 SORC0500
      COMMON/PRTINF/ R0,VGRAV(3),NPRTS                                  SORC0510
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSORC0520
      COMMON/BUOYCL/ RSPH,DELT,VZSPH,XCMSPH,YCMSPH,ZCMSPH,XTOP,YTOP,    SORC0530
     * SPHNS(3),RISTIM                                                  SORC0540
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                                SORC0550
      COMMON/STARS/USTAR,TSTAR,ZSTAR                                    SORC0560
      COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,UM,DN,ZINV                       SORC0570
      COMMON /BURST/ ACCEL,TBURST                                       SORC0580
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)              SORC0590
      COMMON /MODE/ HORIZ                                               SORC0600
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),          SORC0610
     1 QDSC(20,3)                                                       SORC0620
      COMMON/CARB/RCARB1,RCARB2                                         SORC0630
      COMMON/NTAL/TNOT,VOLSPH,TNO,CBLEED                                SORC0640
C                                                                       SORC0650
CC   CR IS THE CRATER RADIUS INDEXED BY COEFFICIENT AND SOIL TYPE       SORC0660
C                                                                       SORC0670
      DATA CR/.271,-.684,.39,.886,0.,.271,-.684,.39,.886,0.,            SORC0680
     1 .386,-.849,.367,.993,0.,.503,-.954,.45,1.19,0.,                  SORC0690
     2 .629,-1.08,.264,1.12,0.,.629,-1.08,.264,1.12,0.,                 SORC0700
```

```
     3 .806,-1.28,-.178,.852,0./                                                SORC0710
C                                                                               SORC0720
CC   CD IS THE CRATER DEPTH INDEXED BY COEFFICIENT AND SOIL TYPE                SORC0730
C                                                                               SORC0740
      DATA CD/.113,-.477,.27,1.84,1.05,.134,-.571,.343,2.24,1.31,               SORC0750
     1 .189,-.84,.447,3.3,2.1,.251,-1.17,.494,4.72,3.34,                        SORC0760
     2 .189,-.84,.447,3.3,2.1,.331,-1.49,.579,4.92,3.13,                        SORC0770
     3 .449,-1.82,.322,4.11,2.02/                                               SORC0780
C                                                                               SORC0790
CC   OWML IS THE OPTICALLY WEIGHTED MASS LOADING COEFFICIENT INDEXED BY         SORC0800
CC        BIN SIZE AND SOIL TYPE                                                SORC0810
C                                                                               SORC0820
      DATA OWML/2.88E3,2*0.,3.08E3,8*0./                                        SORC0830
C                                                                               SORC0840
CC   OWSV IS THE OPTICALLY WEIGHTED PARTICLE SETTLING VELOCITY (CM/SEC)        SORC0850
CC        INDEXED BY BIN SIZE AND SOIL TYPE                                     SORC0860
C                                                                               SORC0870
      DATA OWSV/12*0./                                                          SORC0880
C                                                                               SORC0890
CC   PRTTN IS THE PARTITIONING RATIO INDEXED ON SOIL TYPE                       SORC0900
C                                                                               SORC0910
      DATA PRTTN/4*.8/                                                          SORC0920
C                                                                               SORC0930
CC    BURHTR IS THE RATIO OF BURST HEIGHT TO INITIAL RADIUS AND WTRAT           SORC0940
CC    IS THE FRACTION OF THE TOTAL WEIGHT WHICH IS EFFECTIVE IN THE CLOUD       SORC0950
C                                                                               SORC0960
      DATA BURHTR/0.,4.,2.,4.,3./,WTRAT/.6,1.,.8,1.,.7/                         SORC0970
C                                                                               SORC0980
C                                                                               SORC0990
      RISTM=0.                                                                  SORC1000
      XCMSPH=0.                                                                 SORC1010
      YCMSPH=0.                                                                 SORC1020
      XTOP=0.                                                                   SORC1030
      YTOP=0.                                                                   SORC1040
      TNO=T0                                                                    SORC1050
      NPRTS=1                                                                   SORC1060
C     SCARB IS THE OPTICALLY WEIGHTED CARBON PARTICLE LOADING COEFFICIENT       SORC1070
      SCARB=270.*W                                                              SORC1080
      W3=(W*WTRAT(NCHRG))**.3333333                                             SORC1090
      R0=2.0*W3                                                                 SORC1100
      TAMB=T0+TMPCAL(Z0,ZL,R0)*TSTAR                                            SORC1110
      DELT=.57*TAMB                                                             SORC1120
      RSPH=R0                                                                   SORC1130
      ZCMSPH=R0                                                                 SORC1140
      BURHT=BURHTR(NCHRG)*R0                                                    SORC1150
      BURVZ=1.3*SQRT(R0)                                                        SORC1160
      TBURST=.15*R0                                                             SORC1170
      VZSPH= 2.*BURHT/TBURST-BURVZ                                              SORC1180
      ACCEL=(BURVZ-VZSPH)/TBURST                                                SORC1190
      VOLSPH=(4./3.)*3.141593*R0**3                                             SORC1200
      TNOT=T0+DELT                                                              SORC1210
      CLAM=DD/W3                                                                SORC1220
C                                                                               SORC1230
CC   CALCULATE CRATER RADIUS AND DEPTH                                          SORC1240
C                                                                               SORC1250
      ONCE=.FALSE.                                                              SORC1260
      IF(NSOIL.EQ.1)IDX=4                                                       SORC1270
      IF(NSOIL.EQ.2.)IDX=3                                                      SORC1280
      GO TO 70                                                                  SORC1290
   60 IF(NSOIL.EQ.1)IDX=6                                                       SORC1300
      IF(NSOIL.EQ.2)IDX=4                                                       SORC1310
   70 CONTINUE                                                                  SORC1320
      RC=CR(1,IDX)                                                              SORC1330
      DC=CD(1,IDX)                                                              SORC1340
      IF (CLAM.LT.1.E-30) GO TO 98                                              SORC1350
      TERM=1.                                                                   SORC1360
      DO 100 I=2,5                                                              SORC1370
      TERM=TERM*CLAM                                                            SORC1380
      RC=RC + CR(I,IDX)*TERM                                                    SORC1390
      DC=DC + CD(I,IDX)*TERM                                                    SORC1400
```

```
100   CONTINUE                                                                  SORC1410
 98   CONTINUE                                                                  SORC1420
      RC=RC*W3                                                                  SORC1430
      DC=DC*(W*WTRAT(NCHRG))**.3                                                SORC1440
C                                                                               SORC1450
C  GET CRATER VOLUME                                                            SORC1460
C                                                                               SORC1470
      DSDC=DSOD/DC                                                              SORC1480
      VC=(2.*3.141592/3.)*RC*RC*DC*(1.-1.5*DSDC*(1.-DSDC*DSDC/3.))             SORC1490
      IF(DSOD.GE.DC)VC=0.0                                                      SORC1500
      IF(ONCE)GO TO 110                                                         SORC1510
      ONCE=.TRUE.                                                               SORC1520
      IF(NSOIL.EQ.1)VC1=.5*VC                                                   SORC1530
      IF(NSOIL.EQ.2)VC1=.25*VC                                                  SORC1540
      GO TO 60                                                                  SORC1550
110   IF(NSOIL.EQ.1)VC=VC1+.5*VC                                                SORC1560
      IF(NSOIL.EQ.2)VC=VC1+.75*VC                                               SORC1570
C                                                                               SORC1580
C  CALCULATE OPTICALLY WEIGHTED PARAMETERS                                      SORC1590
C                                                                               SORC1600
      NDSCS=MIN0(10,IFIX(5.*W3/1.8))                                            SORC1610
      CBLEED=0.                                                                 SORC1620
      DO 101 L=1,NPRTS                                                          SORC1630
      S(L)=QWML(L,NSOIL) * VC                                                   SORC1640
      VGRAV(L)=QWSV(L,NSOIL)                                                    SORC1650
      SPHNS(L)=PRTTN(NSOIL) * S(L)                                              SORC1660
      QDSC(1,L) =(1.-PRTTN(NSOIL)) * S(L)/FLOAT(NDSCS)                          SORC1670
      CBLEED=CBLEED+S(L)                                                        SORC1680
101   CONTINUE                                                                  SORC1690
      CBLEED=CBLEED*.03/W3**3                                                   SORC1700
      RCARB=SCARB/SPHNS(1)                                                      SORC1710
      RCARB1=1./(1.+RCARB)                                                      SORC1720
      RCARB2=RCARB/(1.+RCARB)                                                   SORC1730
      SPHNS(1)=SPHNS(1)+SCARB                                                   SORC1740
      DELH=2.*R0/FLOAT(NDSCS)                                                   SORC1750
      Z=-DELH/2.                                                                SORC1760
      DO 200 I=1,NDSCS                                                          SORC1770
      Z=Z+DELH                                                                  SORC1780
      ZDSC(I)=Z                                                                 SORC1790
      DO 201 J=1,NPRTS                                                          SORC1800
      QDSC(I,J)=QDSC(1,J)                                                       SORC1810
201   CONTINUE                                                                  SORC1820
      CON=ALOG(QDSC(I,1)/VISEXT/DELH/(2.*R0)/3.14159)                           SORC1830
      IF(CON.GT.1.)GO TO 210                                                    SORC1840
      D=1.                                                                      SORC1850
      GO TO 230                                                                 SORC1860
210   D=CON                                                                     SORC1870
      DO 220 IT=1,5                                                             SORC1880
      D=(CON-1.+ALOG(D))*D/(D-1)                                                SORC1890
220   CONTINUE                                                                  SORC1900
230   R2DSC(I)=4.*R0*R0/D                                                       SORC1910
      TDSC(I)=-DELH*DELH/D/(DZ0*Z**DN)/4.                                       SORC1920
      SIGZ=DELH*DELH/D                                                          SORC1930
      XDSC(I)=U0*Z**UM * TDSC(I)                                                SORC1940
200   CONTINUE                                                                  SORC1950
999   RETURN                                                                    SORC1960
      END                                                                       SORC1970
```

```
      SUBROUTINE TEMP(Z,TA,DTADZ)                                          TEMP0240
C                                                                          TEMP0010
C   PURPOSE                                                                TEMP0020
C      TO COMPUTE THE AMBIENT ATMOSPHERIC POTENTIAL TEMPERATURE AND        TEMP0030
C      GRADIENT AT A GIVEN HEIGHT.                                         TEMP0040
C                                                                          TEMP0050
C   INPUTS                                                                 TEMP0060
C                                                                          TEMP0070
C      Z    HEIGHT AT WHICH AMBIENT TEMPERATURE AND TEMPERATURE           TEMP0080
C           GRADIENT ARE DESIRED.                                          TEMP0090
C                                                                          TEMP0100
C  OUTPUTS                                                                 TEMP0110
C                                                                          TEMP0120
C    TA     AMBIENT POTENTIAL TEMPERATURE                                  TEMP0130
C                                                                          TEMP0140
C    DTADZ  TEMPERATURE GRADIENT                                           TEMP0150
C                                                                          TEMP0160
C   SUBROUTINES AND FUNCTIONS NEEDED                                       TEMP0170
C                                                                          TEMP0180
C      TMPCAL   COMPUTES SCALED TEMPERATURE AT A GIVEN HEIGHT              TEMP0190
C                                                                          TEMP0200
C   CALLED BY DIFEQ, ATMCAL                                                TEMP0210
C                                                                          TEMP0220
C*************************************************************************TEMP0230
      COMMON/STARS/USTAR,TSTAR,ZSTAR                                       TEMP0250
      COMMON/EKWIND/ALP,C,PYF,PXF,UMAT,VMAT                                TEMP0260
      COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT                                  TEMP0270
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                                   TEMP0280
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU  TEMP0290
      S=Z/ZL                                                               TEMP0300
      TA=TSTAR*TMPCAL(Z0,ZL,Z)+T0                                          TEMP0310
      IF(ABS(ZL).LE.1.E03)GO TO 10                                         TEMP0320
C                                                                          TEMP0330
C NEUTRAL CASE                                                             TEMP0340
C                                                                          TEMP0350
      DTADZ=TSTAR/(Z0+Z)                                                   TEMP0360
      GO TO 999                                                            TEMP0370
   10 IF(ZL.GT.0.0)GO TO 15                                                TEMP0380
C                                                                          TEMP0390
C UNSTABLE CASE                                                            TEMP0400
C                                                                          TEMP0410
      DTADZ=(TSTAR/Z)*(1.-16.*(S))**(-1./2.)                               TEMP0420
      IF(S.LT.-2.0)DTADZ=(TSTAR/ZL)*(AT/3.*(-ZL/Z)**(4./3.))               TEMP0430
      GO TO 999                                                            TEMP0440
C                                                                          TEMP0450
C STABLE CASE                                                              TEMP0460
C                                                                          TEMP0470
   15 DTADZ=(TSTAR/ZL)*(ZL/(Z0+Z)+11.)                                     TEMP0480
      IF(S.GT.1.5)DTADZ=BT*TSTAR/ZL                                        TEMP0490
  999 RETURN                                                               TEMP0500
      END                                                                  TEMP0510
```

```
      FUNCTION TMPCAL(Z0,ZL,Z)                                          TMP00010
C ***************************************************************     TMP00020
CC                                                                      TMP00030
CC                                                                      TMP00040
CC                                                                      TMP00050
CC    PURPOSE                                                           TMP00060
CC                                                                      TMP00070
CC    TO CALCULATE THE POTENTIAL TEMPERATURE SCALED BY THE SCALE        TMP00080
CC    TEMPERATURE, T*, FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV     TMP00090
CC    LENGTH AT A SPECIFIED HEIGHT.                                     TMP00100
CC                                                                      TMP00110
CC                                                                      TMP00120
CC    INPUTS                                                            TMP00130
CC                                                                      TMP00140
CC    Z0      THE FRICTION HEIGHT IN METERS.                            TMP00150
CC    ZL      THE MONIN-OBUKHOV LENGTH IN METERS.                       TMP00160
CC    Z       THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED        TMP00170
CC            IN METERS                                                 TMP00180
CC                                                                      TMP00190
CC                                                                      TMP00200
CC    RETURNS SCALED TEMPERATURE                                        TMP00210
CC                                                                      TMP00220
CC                                                                      TMP00230
CC    CALLED BY ATMCAL, SOURCE AND TEMP                                 TMP00240
CC                                                                      TMP00250
C ***************************************************************     TMP00260
      LOGICAL LOW                                                       TMP00270
      COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT                               TMP00280
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUTMP00290
      PHIM(Z)=(1.-16.*Z)**(-.25)                                        TMP00300
      PSIH(S,S0)=ALOG((S**2.-1.)/(S**2.+1.))*((S0**2.+1.)/(S0**2.-1.)))TMP00310
      PSIHS(Z)=-11.*Z                                                   TMP00320
C     TRACE 999                                                         TMP00330
CC                                                                      TMP00340
CC                                                                      TMP00350
CC    PHIM      THE SHEAR OF MOMENTUM                                   TMP00360
CC    PSIH      THE UNIVERSAL FUNCTION FOR DEVIATION FROM LOGARITHMIC   TMP00370
CC              POTENTIAL TEMPERATURE PROFILE IN THE BOUNDARY LAYER     TMP00380
CC              OF AN UNSTABLE ATMOSPHERE                               TMP00390
CC    PHIHS     THE SAME AS PHIH EXCEPT FOR STABLE ATMOSPHERE           TMP00400
C                                                                       TMP00410
      IF(ABS(ZL).LE.1.E3)GO TO 100                                      TMP00420
      TMPCAL=ALOG(1.+Z/Z0)                                              TMP00430
      GO TO 999                                                         TMP00440
  100 CONTINUE                                                          TMP00450
      P=SIGN(1.,ZL)                                                     TMP00460
      LOW=.TRUE.                                                        TMP00470
      S=Z/ZL                                                            TMP00480
      IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10                                 TMP00490
      S=AMIN1(S,1.5)                                                    TMP00500
      S=AMAX1(S,-2.)                                                    TMP00510
      LOW=.FALSE.                                                       TMP00520
   10 CONTINUE                                                          TMP00530
      IF(P)120,130,130                                                  TMP00540
  120 S=1./PHIM(S)                                                      TMP00550
      S1=Z0/ZL                                                          TMP00560
      S0=1./PHIM(S1)                                                    TMP00570
      TMPCAL=PSIH(S,S0)                                                 TMP00580
C                                                                       TMP00590
CC  FIND CONSTANTS FOR MATCHING IN UNSTABLE CASE AT Z/ZL=-2.            TMP00600
C                                                                       TMP00610
      S2=-2.                                                            TMP00620
      AT=-3.*((1.-16.*S2)**(-1./2.))*(-S2)**(1./3.)                     TMP00630
      CT=-1.*AT*(-S2)**(-1./3.)                                         TMP00640
      GO TO 52                                                          TMP00650
  130 CONTINUE                                                          TMP00660
      PSI=PSIHS(S)                                                      TMP00670
      TMPCAL=ALOG(1.+S*ZL/Z0)-PSI                                       TMP00680
C                                                                       TMP00690
C  FIND CONSTANTS FOR MATCHING IN STABLE CASE AT Z/ZL=1.5               TMP00700
```

```
C                                                                       TMP00710
      S2=1.5                                                            TMP00720
      BT=1./(Z0/ZL+S2)+11.                                              TMP00730
      DT=-1.*BT*S2                                                      TMP00740
   52 CONTINUE                                                          TMP00750
      IF(LOW)GO TO 999                                                  TMP00760
      IF(P)53,53,54                                                     TMP00770
   53 TMPCAL=TMPCAL+CT+AT*(-ZL/Z)**(1./3.)                              TMP00780
      GO TO 999                                                         TMP00790
   54 TMPCAL=TMPCAL+DT+BT*Z/ZL                                          TMP00800
  999 RETURN                                                            TMP00810
      END                                                               TMP00820
```

```
      SUBROUTINE TRAP(TRSK,TRWK,TRSP,H,SIGW,SIG0,TIME,SKT,WAK,SPH)      TRP00510
C                                                                       TRP00010
C  THIS SUBROUTINE PERFORMS A TRAPEZOID INTEGRATION                     TRP00020
C                                                                       TRP00030
C  INPUTS                                                               TRP00040
C                                                                       TRP00050
C     TRSK  - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT     TRP00060
C             TO THE CENTER OF THE SKIRT                                TRP00070
C                                                                       TRP00080
C     TRWK  - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT     TRP00090
C             TO THE CENTER OF THE WAKE                                 TRP00100
C                                                                       TRP00110
C     TRSP  - ESTIMATE TO THE CLOSEST POINT ALONG THE LINE OF SIGHT     TRP00120
C             TO THE CENTER OF THE SPHERE                               TRP00130
C                                                                       TRP00140
C     H     - INTEGRATION STEP SIZE THROUGH THE SKIRT                   TRP00150
C                                                                       TRP00160
C     SIGW  - INTEGRATION STEP SIZE THROUGH THE WAKE                    TRP00170
C                                                                       TRP00180
C     SIG0  - INTEGRATION STEP SIZE THROUGH THE SPHERE                  TRP00190
C                                                                       TRP00200
C     TIME  - TIME TRANSMITTANCE IS DESIRED                             TRP00210
C                                                                       TRP00220
C  ALL OTHER NEEDE INFORMATION IS PASSED VIA COMMON BLOCKS              TRP00230
C                                                                       TRP00240
C                                                                       TRP00250
C  OUTPUTS                                                              TRP00260
C                                                                       TRP00270
C     SKT   - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF       TRP00280
C             SIGHT FROM THE SKIRT                                      TRP00290
C                                                                       TRP00300
C     WAK   - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF       TRP00310
C             SIGHT FROM THE WAKE ONCE THE BUOYANT SPHERE HAS           TRP00320
C             CONVERTED TO THE WIND MODEL                               TRP00330
C                                                                       TRP00340
C     SPH   - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF       TRP00350
C             SIGHT FROM THE SPHERE ONCE IT HAS CONVERTED TO THE        TRP00360
C             WIND MODEL                                                TRP00370
C                                                                       TRP00380
C  FUNCTIONS NEEDED                                                     TRP00390
C                                                                       TRP00400
C     CWIND  - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT      TRP00410
C              (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SKIRT         TRP00420
C                                                                       TRP00430
C     CWAKE  - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT      TRP00440
C              (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE WAKE          TRP00450
C                                                                       TRP00460
C     CSPHER - USED TO FIND THE CONCENTRATION AT A SPECIFIED POINT      TRP00470
C              (X,Y,Z) ALONG THE LINE OF SIGHT DUE TO THE SPHERE        TRP00480
C                                                                       TRP00490
C***********************************************************************TRP00500
      REAL KX,KZ                                                        TRP00520
      LOGICAL SWITCH,CHANGE,ONCE                                        TRP00530
      DIMENSION OWF(5,2),OWFC(5),TRSK(3),TRWK(3),TRSP(3)                TRP00540
      COMMON/LOS/TR(3),RE(3),U(3)                                       TRP00550
      COMMON/ACL/CWINDS,CWINDC,CWINDW                                   TRP00560
      COMMON/CARB/RCARB1,RCARB2                                         TRP00570
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE          TRP00580
      COMMON/SIG/SIG02,SIGC                                             TRP00590
      COMMON/POINTS/XNORM,DOT1,DOT2,DOT3                                TRP00600
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/                TRP00610
      DATA OWFC/1.,.95,.5,.2,1.E-03/                                    TRP00620
      TI=TIME                                                           TRP00630
      SKT=0.0                                                           TRP00640
      WAK=0.0                                                           TRP00650
      SPH=0.0                                                           TRP00660
      SUM=0.0                                                           TRP00670
      SUM1=0.0                                                          TRP00680
      SUM2=0.0                                                          TRP00690
      IND=20                                                            TRP00700
```

```
      ONCE=.FALSE.                                                          TRP00710
      FP=0.0                                                                TRP00720
      F=0.0                                                                 TRP00730
      J=0                                                                   TRP00740
      DO 10 I=1,IND                                                         TRP00750
      XI=H*FLOAT(I-1)                                                       TRP00760
      DIST=DOT1+XI                                                          TRP00770
      IF(DIST.GT.XNORM)GO TO 10                                             TRP00780
      IF(DIST.LT.0.0)GO TO 10                                               TRP00790
      X=TRSK(1)+XI*U(1)                                                     TRP00800
      Y=TRSK(2)+XI*U(2)                                                     TRP00810
      Z=TRSK(3)+XI*U(3)                                                     TRP00820
      FP=F                                                                  TRP00830
      ACL=CWIND(X,Y,Z,TI)                                                   TRP00840
      F=ACL                                                                 TRP00850
C                                                                           TRP00860
C  CHECK TO SEE IF THE CONTRIBUTION IS NEGLIGABLE                           TRP00870
C                                                                           TRP00880
      IF(.NOT.ONCE)ACL1=ACL                                                 TRP00890
      IF(ACL1.LT.1.E-10)GO TO 11                                            TRP00900
      PER=.01*ACL1                                                          TRP00910
      ONCE=.TRUE.                                                           TRP00920
      J=J+1                                                                 TRP00930
      IF(ACL.LT.PER)GO TO 11                                                TRP00940
      IF(J.LE.1)GO TO 10                                                    TRP00950
      SUM=SUM+FP+F                                                          TRP00960
   10 CONTINUE                                                              TRP00970
   11 CONTINUE                                                              TRP00980
      IF((TIME-TTR).LT.1.E-20)GO TO 31                                      TRP00990
C                                                                           TRP01000
C  COMPUTE THE CONTRIBUTION TO CL FROM THE SPHERE USING A TRAPEZOID         TRP01010
C  INTEGRATION ONLY AFTER THE BUOYANT FIREBALL HAS CONVERTED TO THE WINDTRP01020
C  MODEL                                                                    TRP01030
C                                                                           TRP01040
      ONCE=.FALSE.                                                          TRP01050
      FP=0.0                                                                TRP01060
      F=0.0                                                                 TRP01070
      J=0                                                                   TRP01080
      DO 20 I=1,IND                                                         TRP01090
      XI=SIG0*FLOAT(I-1)                                                    TRP01100
      DIST=DOT3+XI                                                          TRP01110
      IF(DIST.GT.XNORM)GO TO 20                                             TRP01120
      IF(DIST.LT.0.0)GO TO 20                                               TRP01130
      X=TRSP(1)+XI*U(1)                                                     TRP01140
      Y=TRSP(2)+XI*U(2)                                                     TRP01150
      Z=TRSP(3)+XI*U(3)                                                     TRP01160
      FP=F                                                                  TRP01170
      ACL=CSPHER(X,Y,Z,TI)                                                  TRP01180
      F=ACL                                                                 TRP01190
      IF(.NOT.ONCE)ACL1=ACL                                                 TRP01200
      IF(ACL1.LT.1.E-05)GO TO 21                                            TRP01210
      PER=.01*ACL1                                                          TRP01220
      ONCE=.TRUE.                                                           TRP01230
      J=J+1                                                                 TRP01240
      IF(ACL.LT.PER)GO TO 21                                                TRP01250
      IF(J.LE.1)GO TO 20                                                    TRP01260
      SUM1=SUM1+FP+F                                                        TRP01270
   20 CONTINUE                                                              TRP01280
   21 CONTINUE                                                              TRP01290
C                                                                           TRP01300
C  COMPUTE CONTRIBUTION TO CL FROM THE WAKE AFTER THE BUOYANT FIREBALL      TRP01310
C  HAS CONVERTED TO THE WIND MODEL USING TRAPEZOID INTEGRATION WITH STEPTRP01320
C  SIZE SIGW.                                                               TRP01330
C                                                                           TRP01340
      ONCE=.FALSE.                                                          TRP01350
      FP=0.0                                                                TRP01360
      F=0.0                                                                 TRP01370
      J=0                                                                   TRP01380
      DO 30 I=1,IND                                                         TRP01390
      XI=SIGW*FLOAT(I-1)                                                    TRP01400
```

```
      DIST=DOT2+XI                                                      TRP01410
      IF(DIST.GT.XNORM)GO TO 30                                         TRP01420
      IF(DIST.LT.0.0)GO TO 30                                           TRP01430
      X=TRWK(1)+XI*U(1)                                                 TRP01440
      Y=TRWK(2)+XI*U(2)                                                 TRP01450
      Z=TRWK(3)+XI*U(3)                                                 TRP01460
      FP=F                                                              TRP01470
      ACL=CWAKE(X,Y,Z,TI)                                               TRP01480
      F=ACL                                                             TRP01490
      IF(.NOT.ONCE)ACL1=ACL                                             TRP01500
      IF(ACL1.LT.1.E-05)GO TO 31                                        TRP01510
      PER=.01*ACL1                                                      TRP01520
      ONCE=.TRUE.                                                       TRP01530
      J=J+1                                                             TRP01540
      IF(ACL.LT.PER)GO TO 31                                            TRP01550
      IF(J.LE.1)GO TO 30                                                TRP01560
      SUM2=SUM2+FP+F                                                    TRP01570
   30 CONTINUE                                                          TRP01580
   31 CONTINUE                                                          TRP01590
      SKT=(ABS(H)/2.)*SUM                                               TRP01600
      WAK=(ABS(SIGW)/2.)*SUM2                                           TRP01610
      SPH=(ABS(SIG0)/2.)*SUM1                                           TRP01620
  999 RETURN                                                            TRP01630
      END                                                               TRP01640
```

```
      SUBROUTINE TRNCAL(TRN,REC,TIME,TRNLOS)                            TRL00520
C                                                                       TRL00010
C  CONTROLING ROUTINE FOR CALCULATING TRANSMITTANCES FOR CHARGE         TRL00020
C  DISTRIBUTION TYPES 1 AND 2                                           TRL00030
C                                                                       TRL00040
C  INPUTS                                                               TRL00050
C                                                                       TRL00060
C  TRN    - TRANSMITTER COORDINATES IN THE LOCAL COORDINATE SYSTEM       TRL00070
C                                                                       TRL00080
C  REC    - RECEIVER COORDINATES IN THE LOCAL COORDINATE SYSTEM         TRL00090
C                                                                       TRL00100
C  TIME   - TIME AFTER THE DETOATION AT WHICH A TRANSMITTANCE IS DESIRED TRL00110
C                                                                       TRL00120
C   ALL OTHER NECESSARY INPUTS ARE PASSED IN COMMON BLOCKS              TRL00130
C                                                                       TRL00140
C                                                                       TRL00150
C  OUTPUT                                                               TRL00160
C                                                                       TRL00170
C    TRNLOS  - TRANSMITTANCE ALONG THE SPECIFIED LINE OF SIGHT          TRL00180
C                                                                       TRL00190
C  SUBROUTINES NEEDED                                                   TRL00200
C                                                                       TRL00210
C     AVRG   - FINDS THE AVERAGE OF THE MOMENTS FOR THE DISCS           TRL00220
C                                                                       TRL00230
C     VSUM   - ADDS TWO VECTORS                                         TRL00240
C                                                                       TRL00250
C     UNIT   - DETERMINE A UNIT VECTOR                                  TRL00260
C                                                                       TRL00270
C     TRNCLD- DETERMINE THE LENGTH OF THE INTERSECTION OF THE LINE OF   TRL00280
C             SIGHT WITH THE WAKE AND SPHERE                            TRL00290
C                                                                       TRL00300
C     TRAP   - DOES A TRAPAZOIDAL INTEGRATION THROUGH SKIRT WAKE AND S  TRL00310
C              SPHERE FOR NON-HORIZONTAL LINES OF SIGHT                 TRL00320
C                                                                       TRL00330
C     TRNCHK- CHECKS TO SEE IF THE OBSCURATION IS SUCH THAT THE         TRL00340
C             TRANSMITTANCE IS LESS THAN A SPECIFIED VALUE              TRL00350
C                                                                       TRL00360
C                                                                       TRL00370
C  FUNCTIONS NEEDED                                                     TRL00380
C                                                                       TRL00390
C     DOTPRD - FINDS THE DOTPRODUCT OF TWO VECTORS                      TRL00400
C                                                                       TRL00410
C     CWIND  - FINDS THE CONCENTRATION ALONG A SPECIFIED HORIZONTAL     TRL00420
C              LINE OF SIGHT OR DETERMINES THE CONCENTRATION AT SOME    TRL00430
C              POINT ALONG THE LINE OF SIGHT FROM THE SKIRT              TRL00440
C                                                                       TRL00450
C     CWAKE  - SAME AS CWIND EXCEPT FOR WAKE                            TRL00460
C                                                                       TRL00470
C     CSPHER - SAME AS CWIND EXCEPT FOR BUOYANT SPHERE                  TRL00480
C                                                                       TRL00490
C***********************************************************************TRL00500
C                                                                       TRL00510
      REAL KZ,KX                                                        TRL00530
      DIMENSION TRN(3),REC(3),OWF(5,2),OWFC(5),TEMP(2)                  TRL00540
      DIMENSION DIR(2),                          XW(3),XS(3)            TRL00550
      DIMENSION TRSK(3),TRWK(3),TRSP(3)                                 TRL00560
      LOGICAL HORIZ,SWITCH,CHANGE,TEST,SKIP                             TRL00570
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUTRL00580
      COMMON/CARB/RCARB1,RCARB2                                         TRL00590
      COMMON/BUOYCL/RSPH,DELT,YZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM  TRL00600
      COMMON/MO5/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,          TRL00610
     +           DMM(600),DMMY(401)                                     TRL00620
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)               TRL00630
      COMMON/MODE/HORIZ                                                 TRL00640
      COMMON/TRAN/VTR,KZ,KX,TTR,XTR,ZTR,QPUFF(3),SWITCH,CHANGE          TRL00650
      COMMON/ACL/CWINDS,CWINDC,CWINDW                                   TRL00660
      COMMON/LOS/T(3),R(3),U(3)                                         TRL00670
      COMMON/SIG/SIG02,SIGC                                             TRL00680
      COMMON/CHARGE/NCHG                                                TRL00690
      COMMON/POINTS/XNORM,DOT1,DOT2,DOT3                                TRL00700
```

```
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL                                TRL00710
      COMMON/SKIPIT/SKIP                                                 TRL00720
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                                 TRL00730
      DATA ONEM/-1.0/                                                    TRL00740
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/                 TRL00750
      DATA OWFC/1.,.95,.5,.2,1.E-03/                                     TRL00760
C                                                                        TRL00770
C  PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER         TRL00780
C                                                                        TRL00790
      NCHG=1                                                             TRL00800
      SKIP=.FALSE.                                                       TRL00810
      TEST=.FALSE.                                                       TRL00820
      HSPH=0.0                                                           TRL00830
      HWAK=0.0                                                           TRL00840
      ACLSKT=0.0                                                         TRL00850
      ACLWAK=0.0                                                         TRL00860
      ACLSPH=0.0                                                         TRL00870
      XNORM=0.0                                                          TRL00880
      DO 10 I=1,3                                                        TRL00890
      R(I)=REC(I)                                                        TRL00900
      T(I)=TRN(I)                                                        TRL00910
      U(I)=R(I)-T(I)                                                     TRL00920
      XNORM=XNORM+U(I)**2                                                TRL00930
   10 CONTINUE                                                           TRL00940
      XNORM=SQRT(XNORM)                                                  TRL00950
      U(1)=U(1)/XNORM                                                    TRL00960
      U(2)=U(2)/XNORM                                                    TRL00970
      U(3)=U(3)/XNORM                                                    TRL00980
      IF((TIME-TTR).LT.1.E-20)GO TO 14                                   TRL00990
C                                                                        TRL01000
C  IF THE BUOYANT SPHERE HAS BECOME WIND BLOWN DETERMINE THE CENTER OF   TRL01010
C  MASS OF THE REFERENCE CHARGE.                                         TRL01020
C                                                                        TRL01030
      XCM=XTR+VTR*(TIME-TTR)                                             TRL01040
      ZCM=ZTR                                                            TRL01050
C                                                                        TRL01060
C  DETERMINE CENTER OF WAKE FOR REFERENCE CHARGE                         TRL01070
C                                                                        TRL01080
   14 ZX=5.0                                                             TRL01090
      CALL AVRG(ZX,TIME,QTOT,XBAVRG,SIG2X,SIG2Y)                         TRL01100
      XW(1)=(XBAVRG+XCM)/2.                                              TRL01110
      XW(2)=YCM/2.                                                       TRL01120
      XW(3)=(5.0+ZCM)/2.                                                 TRL01130
C                                                                        TRL01140
C  IF THE DIFFERENCE BETWEEN THE TRANSMITTER AND RECEIVER IS GREATER     TRL01150
C  THAN 1 PERCENT OF THE DISTANCE BETWEEN THEM THEN THE LOS IS           TRL01160
C  CONSIDERED A SLANT PATH.                                              TRL01170
C                                                                        TRL01180
   13 IF(ABS(U(3)).GT..01)GO TO 20                                       TRL01190
      IF(ABS(TRN(3)-REC(3)).LT.1.E-06)GO TO 9                            TRL01200
C                                                                        TRL01210
C  COMPUTE CLOSEST POINT ALONG THE LOS TO OUR ESTIMATE OF THE CENTER OF  TRL01220
C  THE WAKE                                                              TRL01230
C                                                                        TRL01240
      DOT=-(U(1)*(T(1)-XW(1))+U(2)*(T(2)-XW(2))+U(3)*(T(3)-XW(3)))       TRL01250
      T(3)=T(3)+U(3)*DOT                                                 TRL01260
      R(3)=T(3)                                                          TRL01270
C                                                                        TRL01280
C  COMPUTE CONTRIBUTIONS FOR A HORIZONTAL PATH                           TRL01290
C                                                                        TRL01300
    9 HORIZ=.TRUE.                                                       TRL01310
      CALL VSUM(REC,TRN,ONEM,DIR)                                        TRL01320
      CALL UNIT(DIR,DIR,RANGE)                                           TRL01330
      COSTH=DIR(1)                                                       TRL01340
      SINTH=DIR(2)                                                       TRL01350
      SINTH2=SINTH*SINTH                                                 TRL01360
      COSTH2=COSTH**2                                                    TRL01370
      SCRN(1)=SINTH                                                      TRL01380
      SCRN(2)=-COSTH                                                     TRL01390
      DO 12 J=1,ITOT                                                     TRL01400
```

```
      DO 11 I=1,2                                                       TRL01410
      TEMP(I)=DIFF(I,J)                                                 TRL01420
   11 CONTINUE                                                          TRL01430
      PRSEP(J)=DOTPRD(TEMP,SCRN)                                        TRL01440
   12 CONTINUE                                                          TRL01450
      X=DOTPRD(SCRN,TRN)                                                TRL01460
C                                                                       TRL01470
C  COMPUTE THE CONTRIBUTION FROM THE SKIRT AT A HEIGHT OF T(3) WHERE T(3TRL01480
C  IS THE HEIGHT OF THE TRANSMITTER IF THE DIFFERENCE BETWEEN THE       TRL01490
C  TRANSMITTER AND RECEIVER HEIGHTS IS SMALL AND IS THE Z COMPONENT OF  TRL01500
C  THE POINT ON THE LINE CONNECTING THE TRNSMITTER AND RECEIVER ,WHICH  TRL01510
C  IS CLOSEST TO OUR ESTIMATE OF THE CENTER OF THE WAKE OTHERWISE.      TRL01520
C                                                                       TRL01530
      ACLSKT=CWIND(X,Y,T(3),TIME)                                       TRL01540
      IF(TEST)GO TO 998                                                 TRL01550
C                                                                       TRL01560
C  TEST IS A LOGICAL VARIABLE RETURNED IN COMMON/TRANNY/ FROM SUBROUTINETRL01570
C  TRNCHK WHICH IS CALLED BY CWIND, CWAKE, CSPHER, TRNCLD, AND TRAP     TRL01580
C  EACH TIME A CONTRIBUTION IS MADE TO THE OPTICALL WEIGHTED            TRL01590
C  CONCENTRATION ALONG THE OPTICAL PATH                                 TRL01600
C                                                                       TRL01610
C        TEST=.FALSE.   TRANSMITTANCE IS GREATER THAN TRNMIN            TRL01620
C                       (A TRANSMITTANCE THRESHOLD).                    TRL01630
C            =.TRUE.    TRANSMITTANCE IS LESS THAN TRNMIN              TRL01640
C                                                                       TRL01650
      IF(TIME.GT.TTR)ACLWAK=CWAKE(X,Y,T(3),TIME)                        TRL01660
      IF(TEST)GO TO 998                                                 TRL01670
      IF(TIME.GT.TTR)ACLSPH=CSPHER(X,Y,T(3),TIME)                       TRL01680
      IF(TEST)GO TO 998                                                 TRL01690
      IF(TIME.GT.TTR)GO TO 50                                           TRL01700
      CWINDS=ACLSKT                                                     TRL01710
      CWINDW=ACLWAK                                                     TRL01720
      CWINDC=ACLSPH                                                     TRL01730
      CALL TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)                             TRL01740
      IF(TEST)GO TO 998                                                 TRL01750
      GO TO 50                                                          TRL01760
C                                                                       TRL01770
C  DO TRAPEZOIDAL INTEGRATION FOR SLANT PATH IN BOTH DIRECTIONS FROM    TRL01780
C  AN ESTIMATE OF THE LOCATION OF CENTER OF THE SKIRT USING A STEP SIZE TRL01790
C  OF SIG,THE GEOMETRIC MEAN OF THE AVERAGE OF THE SPREADS OF THE DISCS TRL01800
C  IN BOTH THE X AND Y DIRECTION.THEN IF THE  BUOYANT SPHERE HAS        TRL01810
C  CONVERTED TO THE WIND MODEL DO THE SAME FOR THE WAKE AND SPHERE      TRL01820
C  WITH THE APPROPRIATE STEP SIZE. (CHECK TO SEE IF SPHERE HAS          TRL01830
C  CONVERTED TO THE WIND MODEL IS DONE IN TRAP.)                        TRL01840
C                                                                       TRL01850
   20 HOPIC=.FALSE.                                                     TRL01860
      SIGX=SQRT(SIG2X)                                                  TRL01870
      SIGY=SQRT(SIG2Y)                                                  TRL01880
      SIG=SQRT(SIGX*SIGY)                                               TRL01890
      CALL WIN(2.0,UX,UY)                                               TRL01900
      XS(1)=TIME*UX                                                     TRL01910
      XS(2)=TIME*UY                                                     TRL01920
      XS(3)=2.0                                                         TRL01930
C                                                                       TRL01940
C  FIND THE POINTS ON THE LINE CONNECTING THE TRANSMITTER AND           TRL01950
C  RECEIVER THAT ARE CLOSEST TO OUR ESTIMATE OF THE CENTER OF           TRL01960
C  THE SKIRT,WAKE, AND SPHERE.                                          TRL01970
C                                                                       TRL01980
      DO 48 J=1,ITOT                                                    TRL01990
      XSK=XS(1)+DIFF(1,J)                                               TRL02000
      YSK=XS(2)+DIFF(2,J)                                               TRL02010
      ZSK=XS(3)                                                         TRL02020
      DOT1=-(U(1)*(T(1)-XSK)+U(2)*(T(2)-YSK)+U(3)*(T(3)-ZSK))           TRL02030
      IF(DOT1.LT.0.0)DOT1=0.0                                           TRL02040
      IF(DOT1.GT.XNORM)DOT1=XNORM                                       TRL02050
      XWK=XW(1)+DIFF(1,J)                                               TRL02060
      YWK=XW(2)+DIFF(2,J)                                               TRL02070
      ZWK=XW(3)                                                         TRL02080
      DOT2=-(U(1)*(T(1)-XWK)+U(2)*(T(2)-YWK)+U(3)*(T(3)-ZWK))           TRL02090
      IF(DOT2.LT.0.0)DOT2=0.0                                           TRL02100
```

```
      IF(DOT2.GT.XNORM)DOT2=XNORM                                          TRL02110
      XSP=XCM+DIFF(1,J)                                                    TRL02120
      YSP=YCM+DIFF(2,J)                                                    TRL02130
      ZSP=ZCM                                                              TRL02140
      DOT3=-(U(1)*(T(1)-XSP)+U(2)*(T(2)-YSP)+U(3)*(T(3)-ZSP))              TRL02150
      IF(DOT3.LT.0.0)DOT3=0.0                                              TRL02160
      IF(DOT3.GT.XNORM)DOT3=XNORM                                          TRL02170
      NCHG=J                                                               TRL02180
      DO 45 II=1,3                                                         TRL02190
      TRSK(II)=T(II)+U(II)*DOT1                                            TRL02200
      TRWK(II)=T(II)+U(II)*DOT2                                            TRL02210
      TRSP(II)=T(II)+U(II)*DOT3                                            TRL02220
   45 CONTINUE                                                             TRL02230
      DIFS=DIFFUS(Z0,ZL,2.0)                                               TRL02240
      SIGZ=SQRT(2.*DIFS*TIME)                                              TRL02250
      IF(SIGZ.LT.1.0)SIGZ=1.0                                              TRL02260
      H=U(1)*SIGX+U(2)*SIGY+U(3)*SIGZ                                      TRL02270
      IF(TIME.LT.TTR)GO TO 46                                              TRL02280
      HSPX=SQRT(SIG02+2.*KX*(TIME-TTR))                                    TRL02290
      HSPY=HSPX                                                            TRL02300
      HSPZ=SQRT(SIG02+2.*KZ*(TIME-TTR))                                    TRL02310
      HSPH=(U(1)*HSPX+U(2)*HSPY+U(3)*HSPZ)/2.                              TRL02320
      HWAK=(HSPH+H)/2.                                                     TRL02330
   46 CONTINUE                                                             TRL02340
      CALL TRAP(TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH)               TRL02350
      ACLSKT=ACLSKT+SKT                                                    TRL02360
      ACLWAK=ACLWAK+WAK                                                    TRL02370
      ACLSPH=ACLSPH+SPH                                                    TRL02380
      CALL TRNCHK(ACLSKT,ACLWAK,ACLSPH)                                    TRL02390
      IF(TEST)GO TO 998                                                    TRL02400
      H=-H                                                                 TRL02410
      IF(TIME.LT.TTR)GO TO 47                                              TRL02420
      HSPH=-HSPH                                                           TRL02430
      HWAK=-HWAK                                                           TRL02440
   47 CONTINUE                                                             TRL02450
      CALL TRAP(TRSK,TRWK,TRSP,H,HWAK,HSPH,TIME,SKT,WAK,SPH)               TRL02460
      ACLSKT=ACLSKT+SKT                                                    TRL02470
      ACLWAK=ACLWAK+WAK                                                    TRL02480
      ACLSPH=ACLSPH+SPH                                                    TRL02490
      CWINDS=ACLSKT                                                        TRL02500
      CWINDW=ACLWAK                                                        TRL02510
      CWINDS=ACLSPH                                                        TRL02520
      CALL TRNCHK(ACLSKT,ACLWAK,ACLSPH)                                    TRL02530
      IF(TEST)GO TO 998                                                    TRL02540
   48 CONTINUE                                                             TRL02550
      IF(TIME.GT.TTR)GO TO 50                                              TRL02560
      CALL TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)                                TRL02570
      IF(TEST)GO TO 998                                                    TRL02580
   50 CONTINUE                                                             TRL02590
      ACLC=(ACLWAK+ACLSPH)*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL))        TRL02600
      ACLS=ACLSKT*OWF(NWL,NSOIL)                                           TRL02610
C                                                                          TRL02620
C  COMPUTE THE TRANSMITTANCE ALONG THE LINE OF SIGHT                       TRL02630
C                                                                          TRL02640
      TRNLOS=EXP(-ACLS-ACLC)                                               TRL02650
      GO TO 999                                                            TRL02660
  998 TRNLOS=0.0                                                           TRL02670
  999 RETURN                                                               TRL02680
      END                                                                  TRL02690
```

```
      SUBROUTINE TRNCHK(ACLS,ACLW,ACLC)                                 TRK00310
C                                                                       TRK00010
CCC   THIS IS A SUBROUTINE TO CHECK IF THE CONCENTRATIONS ARE HIGH ENOUGH  TRK00020
CCC   SO THAT THE TRANSMITTANCE WILL BE BELOW A GIVEN LEVEL.            TRK00030
CCC                                                                     TRK00040
CCC   INPUTS                                                            TRK00050
CCC                                                                     TRK00060
CCC      ACLS  - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT TRK00070
CCC              DUE TO THE SKIRT                                       TRK00080
CCC                                                                     TRK00090
CCC      ACLW  - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT TRK00100
CCC              DUE TO THE WAKE                                        TRK00110
CCC                                                                     TRK00120
CCC      ACLC  - CONTRIBUTION TO THE CONCENTRATION ALONG THE LINE OF SIGHT TRK00130
CCC              DUE TO THE BUOYANT SPHERE                              TRK00140
CCC                                                                     TRK00150
CCC      ALL OTHER NECESSARY INFORMATION IS PASSED VIA COMMON BLOCKS    TRK00160
CCC                                                                     TRK00170
CCC   OUTPUT                                                            TRK00180
CCC                                                                     TRK00190
CCC      TEST - LOGIICAL VARIABLE PASSED IN COMMON/TRANNY/ THAT IS .TRUE. TRK00200
CCC             IF THE CONCENTRATION IS SUCH THAT THE TRANSMITTANCE     TRK00210
CCC             ALONG THE LINE OF SIGHT WILL BE LESS THAN THE THRESHOLD TRK00220
CCC             VALUE AND IS .FALSE. OTHERWISE                          TRK00230
CCC                                                                     TRK00240
CCC                                                                     TRK00250
CCC   FUNCTIONS ANT SUBROUTINE NEEDED                                   TRK00260
CCC                                                                     TRK00270
CCC      NONE                                                           TRK00280
C*********************************************************************  TRK00290
C                                                                       TRK00300
      DIMENSION OWF(5,2),OWFC(5)                                        TRK00320
      LOGICAL TEST                                                      TRK00330
      COMMON/CARB/RCARB1,RCARB2                                         TRK00340
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL                               TRK00350
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/                TRK00360
      DATA OWFC/1.,.95,.5,.2,1.E-03/                                    TRK00370
      TEST=.FALSE.                                                      TRK00380
      ACL=(ACLW+ACLC)*(RCARB1*OWF(NWL,NSOIL)+RCARB2*OWFC(NWL))          TRK00390
      ACL=ACL+ACLS*OWF(NWL,NSOIL)                                       TRK00400
      IF(ACL.GT.THRESH)TEST=.TRUE.                                      TRK00410
  999 RETURN                                                            TRK00420
      END                                                               TRK00430
```

```
      SUBROUTINE TRNCLD(XNORM,TIME,ACLWAK,ACLSPH)                       TRD00350
C                                                                       TRD00010
C     ROUTINE FOR DETERMINING CONTRIBUTION FROM SPHERE AND WAKE         TRD00020
C     BEFORE THE BUOYANT SPHERE HAS CONVERTED TO THE WIND MODEL.        TRD00030
C                                                                       TRD00040
C     INPUTS                                                            TRD00050
C                                                                       TRD00060
C        XNORM  -DISTANCE BETWEEN THE TRANSMITTER AND RECEIVER          TRD00070
C                                                                       TRD00080
C        TIME   -TIME AT WHICH TRANSMITTANCE IS DESIRED                 TRD00090
C                                                                       TRD00100
C     OUTPUTS                                                           TRD00110
C                                                                       TRD00120
C        ACLWAK -CONTRIBUTION FROM WAKE TO OPTICALLY WEIGHTED CONCENTRATIONTRD00130
C                ALONG GIVEN LINE OF SIGHT                              TRD00140
C                                                                       TRD00150
C        ACLSPH -CONTRIBUTION FROM BUOYANT SPHERE TO OPTICALLY WEIGHTED TRD00160
C                CONCENTRATION ALONG GIVEN LINE OF SIGHT                TRD00170
C                                                                       TRD00180
C     SUBROUTINES NEEDED                                                TRD00190
C                                                                       TRD00200
C        AVRG   -COMPUTE THE AVERAGE OF THE MOMENTS FOR THE DISCS       TRD00210
C                                                                       TRD00220
C        WIN    -COMPUTE THE WIND SPEED AT A GIVEN HEIGHT               TRD00230
C                                                                       TRD00240
C        AMOUNT-COMPUTE THE DISTRIBUTION OF THE LOADING BETWEEN THE BUOYANTTRD00250
C               SPHERE AND WAKE.                                        TRD00260
C                                                                       TRD00270
C        CONLEN-COMPUTE THE LENGTH OF INTERSECTION OF NON-HORIZONTAL LINE TRD00280
C               OF SIGHT WITH A CONICAL SHAPED WAKE                     TRD00290
C                                                                       TRD00300
C        TRNCHK-ROUTINE TO CHECK IF THE OBSCURATION IS SUCH THAT THE    TRD00310
C               TRANSMITTANCE IS LESS THAN A USET SPECIFIED AMOUNT      TRD00320
C                                                                       TRD00330
C***********************************************************************TRD00340
      LOGICAL HORIZ,TEST                                                TRD00360
      DIMENSION CENTER(3)                                               TRD00370
      COMMON/MODE/HORIZ                                                 TRD00380
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)               TRD00390
      COMMON/BUOYCL/RSPH,DELT,VZ,XCM,YCM,ZCM,XTOP,YTOP,SPHNS(3),RISTIM  TRD00400
      COMMON/PRTINF/R0,VGRAV(3),NPRTS                                   TRD00410
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)TRD00420
      COMMON/MOS/DIFF(2,200),NCHTOT,PRSEP(200),NTOT,NARY,ITOT,          TRD00430
     +           DMM(600),DMMY(401)                                     TRD00440
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUTRD00450
      COMMON/LOS/TR(3),RE(3),U(3)                                       TRD00460
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL                               TRD00470
      COMMON/ACL/CWINDS,CWINDC,CWINDW                                   TRD00480
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     TRD00490
      ACLWAK=0.0                                                        TRD00500
      ACLSPH=0.0                                                        TRD00510
C                                                                       TRD00520
C     DETERMINE THE RADIUS OF THE BASE OF THE CONE                      TRD00530
C                                                                       TRD00540
      ZX=5.0                                                            TRD00550
      CALL AVRG(ZX,TIME,QTOT,XBAVG,SIG2X,SIG2Y)                         TRD00560
      IF(QTOT.LT.1.E-10)GO TO 33                                        TRD00570
      SIGX=SQRT(SIG2X)                                                  TRD00580
      SIGY=SQRT(SIG2Y)                                                  TRD00590
      RB=SQRT(SIGX*SIGY)                                                TRD00600
      GO TO 35                                                          TRD00610
   33 CALL WIN(5.0,UW,V)                                                TRD00620
      XBAVG=UW*TIME                                                     TRD00630
      RB=0.0                                                            TRD00640
   35 VOLWAK=(PI/3.)*(ZCM-5.0)*(RSPH**2+RSPH*RB+RB**2)                  TRD00650
      VOLSPH=(4./3.)*PI*RSPH**3                                         TRD00660
      CALL AMOUNT(VOLSPH,WAKAL,SPHAL)                                   TRD00670
      DO 80 J=1,ITOT                                                    TRD00710
      IF (VOLWAK.LE.0.0) GO TO 68                                       TRD00680
      XB=XBAVG+DIFF(1,J)                                                TRD00720
```

```
      YB=DIFF(2,J)                                                          TRD00730
      XC=XCM+DIFF(1,J)                                                      TRD00740
      YC=YCM+DIFF(2,J)                                                      TRD00750
      IF(ABS(U(3)).LE..01)GO TO 110                                         TRD00760
      ZBOT=5.0                                                              TRD00770
      CALL CONLEN(U,TR,ZCM,ZBOT,XC,YC,RSPH,RB,XB,YB,XNORM,PLENWK)           TRD00780
      ACLW=PLENWK*WAKAL/VOLWAK                                              TRD00790
      GO TO 69                                                              TRD00800
   68 ACLW=0.0                                                              TRD00810
   69 CONTINUE                                                              TRD00820
      ACLWAK=ACLWAK+ACLW                                                    TRD00830
      CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)                                     TRD00840
      IF(TEST)GO TO 999                                                     TRD00850
C                                                                           TRD00860
C  DETERMINE CONTRIBUTION FROM SPHERE FO A SLANT PATH                       TRD00870
C                                                                           TRD00880
      CENTER(1)=XCM+DIFF(1,J)-TR(1)                                         TRD00890
      CENTER(2)=YCM+DIFF(2,J)-TR(2)                                         TRD00900
      CENTER(3)=ZCM-TR(3)                                                   TRD00910
      CLOSE=U(1)*CENTER(1)+U(2)*CENTER(2)+U(3)*CENTER(3)                    TRD00920
      CON=CENTER(1)**2+CENTER(2)**2+CENTER(3)**2-RSPH**2                    TRD00930
      RADIC=CLOSE**2-CON                                                    TRD00940
      IF(RADIC.LT.0.0)GO TO 75                                              TRD00950
      PNEAR=CLOSE-SQRT(RADIC)                                               TRD00960
      PFAR=CLOSE+SQRT(RADIC)                                                TRD00970
      PLENSP=AMIN1(PFAR,XNORM)-AMAX1(PNEAR,0.0)                             TRD00980
      IF(PLENSP.LT.0.0)PLENSP=0.0                                           TRD00990
      ACLS=PLENSP*SPHAL/VOLSPH                                              TRD01000
      GO TO 76                                                              TRD01010
   75 ACLS=0.0                                                              TRD01020
   76 ACLSPH=ACLSPH+ACLS                                                    TRD01030
      CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)                                     TRD01040
      IF(TEST)GO TO 999                                                     TRD01050
      GO TO 80                                                              TRD01060
C                                                                           TRD01070
C  COMPUTE CONTRIBUTIONS FOR SPHERE AND WAKE FOR A HORIZONTAL PATH          TRD01080
C                                                                           TRD01090
  110 IF(TR(3).GT.ZCM+RSPH)GO TO 999                                        TRD01100
      XCEN=XCM+DIFF(1,J)                                                    TRD01110
      YCEN=YCM+DIFF(2,J)                                                    TRD01120
      IF(TR(3).LT.ZCM-RSPH)GO TO 130                                        TRD01130
      RADIUS=SQRT(RSPH**2-(TR(3)-ZCM)**2)                                   TRD01140
      CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENSP)                               TRD01150
      IF(PLENSP.LT.0.0)PLENSP=0.0                                           TRD01160
      ACLSPH=ACLSPH+PLENSP*SPHAL/VOLSPH                                     TRD01170
      CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)                                     TRD01180
      IF(TEST)GO TO 999                                                     TRD01190
  130 IF(TR(3).GT.ZCM)GO TO 999                                             TRD01200
      IF(TR(3).LE.5.0)GO TO 999                                             TRD01210
      ZETA=(TR(3)-5.0)/(ZCM-5.0)                                            TRD01220
      XCEN=ZETA*XCEN+(1.-ZETA)*XB                                           TRD01230
      YCEN=ZETA*YCEN+(1-ZETA)*YB                                            TRD01240
      RADIUS=ZETA*RSPH+(1-ZETA)*RB                                          TRD01250
      CALL PATH(TR,U,XCEN,YCEN,RADIUS,PLENWK)                               TRD01260
      IF(PLENWK.LT.0.0)PLENWK=0.0                                           TRD01270
      ACLWAK=ACLWAK+PLENWK*WAKAL/VOLWAK                                     TRD01280
      CALL TRNCHK(CWINDS,ACLWAK,ACLSPH)                                     TRD01290
      IF(TEST)GO TO 999                                                     TRD01300
   80 CONTINUE                                                              TRD01310
  999 RETURN                                                                TRD01320
      END                                                                   TRD01330
```

```
      SUBROUTINE UNIT(A,B,XNORM)                                        UNIT0010
      DIMENSION A(2),B(2)                                               UNIT0020
C *** B IS THE NORM OF A, AND XNORM IS THE MAGNITUDE                    UNIT0030
      XNORM=SQRT(A(1)**2+A(2)**2)                                       UNIT0040
      B(1)=A(1)/XNORM                                                   UNIT0050
      B(2)=A(2)/XNORM                                                   UNIT0060
      RETURN                                                            UNIT0070
      END                                                               UNIT0080
```

```
      SUBROUTINE VEHCL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,THWND,PHI,NSOIL,   VCL00160
     1                 SILT,NWL,TRNCOR,RECCOR,TIME,DHDT,V0,VEHDIR,     VCL00170
     2                 VEHSPD,VEHWID,VEHWHT,VEHTYP,NEWATM,NEWVEH,      VCL00180
     3                 TRNLOS,NERR)                                    VCL00190
C                                                                       VCL00010
C     THIS ROUTINE CONTROLS THE FLOW OF THE CALCULATION FOR THE         VCL00020
C  VEHICLE GENERATED DUST CLOUD.                                        VCL00030
C                                                                       VCL00040
C  INPUTS                                                               VCL00050
C    SEE DRTRAN FOR DETAILS                                             VCL00060
C                                                                       VCL00070
C  OUTPUTS                                                              VCL00080
C                                                                       VCL00090
C     TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT                    VCL00100
C                                                                       VCL00110
C     NERR   - ERROR CODE                                               VCL00120
C                                                                       VCL00130
C                                                                       VCL00140
C***********************************************************************VCL00150
      DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2)                     VCL00200
      DIMENSION TRNCOR(3),RECCOR(3),TRN(3),REC(3),TRNFRM(2,2),ORIG(2),  VCL00210
     1          V1(2),V0(2),VDIR(2)                                     VCL00220
      LOGICAL DHDT,ERR,NEWATM,NEWVEH                                    VCL00230
      INTEGER VEHTYP                                                    VCL00240
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUVCL00250
      COMMON/GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI,SCRN(2)               VCL00260
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV                         VCL00270
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     VCL00280
      DATA RTPI /1.772453/                                              VCL00290
      THETAX=THWND*PIRAD                                                VCL00300
      IF(.NOT.NEWATM)GO TO 100                                          VCL00310
      CALL ATMCAL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,PHI,THETAX,DHDT,ERR)   VCL00320
      IF(.NOT.ERR)GO TO 100                                             VCL00330
      NERR=7                                                            VCL00340
      GO TO 999                                                         VCL00350
  100 CONTINUE                                                          VCL00360
      IF(.NOT.NEWVEH)GO TO 5                                            VCL00370
      CALL VSRC(VEHSPD,VEHWID,VEHWHT,VEHTYP,NSOIL,SILT)                 VCL00380
      CALL PREVEH(NSOIL,NWL)                                            VCL00390
    5 CONTINUE                                                          VCL00400
C                                                                       VCL00410
C  COMPUTE DIRECTION VECTOR FOR THE VEHICLE MOTION FROM USERS INPUT     VCL00420
C                                                                       VCL00430
      ANGL=VEHDIR*PIRAD                                                 VCL00440
      V1(1)=VEHSPD*COS(ANGL)                                            VCL00450
      V1(2)=VEHSPD*SIN(ANGL)                                            VCL00460
C                                                                       VCL00470
C  COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT THE USER       VCL00480
C  DEFINED COORDINATES INTO LOCAL COORDINATES WITH THE X-AXIS IN         VCL00490
C  THE WIND DIRECTION.                                                  VCL00500
C                                                                       VCL00510
      TRNFRM(1,1)=COS(THETAX)                                           VCL00520
      TRNFRM(2,2)=TRNFRM(1,1)                                           VCL00530
      TRNFRM(1,2)=SIN(THETAX)                                           VCL00540
      TRNFRM(2,1)=-TRNFRM(1,2)                                          VCL00550
      ORIG(1)=V0(1)                                                     VCL00560
      ORIG(2)=V0(2)                                                     VCL00570
C                                                                       VCL00580
C  COMPUTE NEW COORDINATES BY MULTIPLING BY THE TRANSFORMATION MATRIX   VCL00590
C                                                                       VCL00600
      TRN(3)=TRNCOR(3)                                                  VCL00610
      REC(3)=RECCOR(3)                                                  VCL00620
      DO 20 I=1,2                                                       VCL00630
      TRN(I)=0.0                                                        VCL00640
      REC(I)=0.0                                                        VCL00650
      VDIR(I)=0.0                                                       VCL00660
      DO 10 J=1,2                                                       VCL00670
      TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))                     VCL00680
      REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))                     VCL00690
      VDIR(I)=VDIR(I)+TRNFRM(I,J)*V1(J)                                 VCL00700
```

```
   10 CONTINUE                                                          VCL00710
   20 CONTINUE                                                          VCL00720
C                                                                       VCL00730
C  CALL VEHTRN ROUTINE TO USE COMPUTED QUADRATIC FITS TO CALCULATE      VCL00740
C  A TRANSMITTANCE                                                      VCL00750
C                                                                       VCL00760
      CALL VEHTRN(TRN,REC,TIME,VDIR,TRNLOS)                             VCL00770
  999 RETURN                                                            VCL00780
      END                                                               VCL00790
```

```
      SUBROUTINE VEHTRN(TRN,REC,TIME,VDIR,TRNLOS)                       VTN00340
C                                                                       VTN00010
C     THIS ROUTINE PARAMETERIZES THE LINE CONNECTING THE TRANSMITTER    VTN00020
C   AND RECEIVER IN THE LOCAL COORDINATE SYSTEM AND DOES A TRAPEZOIDAL  VTN00030
C   INTEGRATION FROM VEHICLE TIME=0.0 TO VEHICLE TIME=TIME THE          VTN00040
C   TRANSMITTANCE IS DESIRED.                                           VTN00050
C                                                                       VTN00060
C   INPUTS                                                              VTN00070
C                                                                       VTN00080
C     TRN    - THE COORDINATES OF THE TRANSMITTER IN THE LOCAL COORDINATE VTN00090
C              SYSTEM                                                   VTN00100
C                                                                       VTN00110
C     REC    - THE COORDINATES OF THE RECEIVER IN THE LOCAL COORDINATE  VTN00120
C              SYSTEM                                                   VTN00130
C                                                                       VTN00140
C     TIME   - THE PRESENT TIME AT WHICH A TRANSMITTANCE IS DESIRED     VTN00150
C                                                                       VTN00160
C     NWL    - INTEGER INDEX FOR WAVELENGTH BEING USED                  VTN00170
C                                                                       VTN00180
C     NSOIL  - INTEGER INDEX FOR SOIL TYPE                              VTN00190
C                                                                       VTN00200
C     VDIR   - VECTOR INDICATING THE DIRECTION AND SPEED OF THE VEHICLE VTN00210
C              IN THE LOCAL COODINATE SYSTEM                            VTN00220
C                                                                       VTN00230
C   OUTPUT                                                              VTN00240
C                                                                       VTN00250
C     TRNLOS - TRANSMITTANCE ALONG THE LINE OF SIGHT AT THE INDICATED   VTN00260
C              TIME                                                     VTN00270
C                                                                       VTN00280
C   FUNCTIONS AND SUBROUTINES NEEDED                                    VTN00290
C                                                                       VTN00300
C     GRAND EVALUATES THE INTEGRAND                                     VTN00310
C                                                                       VTN00320
C***********************************************************************VTN00330
      LOGICAL TEST                                                      VTN00350
      DIMENSION TRN(3),REC(3),TR(3),RE(3),VDIR(2),OWF(5,2),U(3)         VTN00360
      COMMON/MOS/DMMMY(604),DMM(600),                                   VTN00370
     +           ICOUNT,TIMES(25),XC0(3,25),XC1(3,25),RT(3,25),         VTN00380
     +           RB(3,25),Z2(3,25)                                      VTN00390
      COMMON/TRANNY/THRESH,TEST,NWL,NSOIL                               VTN00400
      DATA OWF/1.,.93,.52,.44,2.E-03,1.,1.,1.,1.,4.E-03/                VTN00410
      TEST=.FALSE.                                                      VTN00420
C                                                                       VTN00430
C   PARAMETERIZE THE LINE CONNECTING THE TRANSMITTER AND RECEIVER       VTN00440
C                                                                       VTN00450
      XNORM=0.0                                                         VTN00460
      DO 10 I=1,3                                                       VTN00470
      RE(I)=REC(I)                                                      VTN00480
      TR(I)=TRN(I)                                                      VTN00490
      U(I)=RE(I)-TR(I)                                                  VTN00500
      XNORM=XNORM+U(I)**2                                               VTN00510
   10 CONTINUE                                                          VTN00520
      XNORM=SQRT(XNORM)                                                 VTN00530
      U(1)=U(1)/XNORM                                                   VTN00540
      U(2)=U(2)/XNORM                                                   VTN00550
      U(3)=U(3)/XNORM                                                   VTN00560
C                                                                       VTN00570
C     INCREMENT VEHICLE TRAVEL TIME AND CALL GRAND TO COMPUTE THE VALUE VTN00580
C   OF THE INTEGRAND.  IF THE VALUE OF THE VARIABLE TIME IS GREATER     VTN00590
C   THAN THE MAXIMUM TIME THAT HAS BEEN STORED (APPROX. 373 SEC) THEN   VTN00600
C   ANY DUST PRODUCED MORE THAN TIMES(20) SECONDS IS ASSUMED TO HAVE    VTN00610
C   NO EFFECT ON THE TRANSMITTANCE.                                     VTN00620
C                                                                       VTN00630
C                                                                       VTN00640
      IF(TIME.LE.TIMES(ICOUNT))GO TO 11                                 VTN00650
      TINC=TIMES(ICOUNT)/400.                                           VTN00660
      TSTART=TIME-TIMES(ICOUNT)                                         VTN00670
      GO TO 15                                                          VTN00680
   11 TINC=TIME/400.                                                    VTN00690
      TSTART=0.0                                                        VTN00700
```

```
 15 SUM=0.0                                                              VTN00710
    DO 50 I=1,401                                                        VTN00720
    TIVEH=TSTART+FLOAT(I-1)*TINC                                         VTN00730
    CALL GRAND(U,TR,XNORM,TIME,TIVEH,VDIR,VALUE)                         VTN00740
    IF(I.EQ.1.OR.I.EQ.401)GO TO 20                                       VTN00750
    SUM=SUM+VALUE                                                        VTN00760
    GO TO 40                                                             VTN00770
 20 SUM=SUM+VALUE/2.                                                     VTN00780
 40 CONTINUE                                                             VTN00790
    SUM1=SUM*TINC                                                        VTN00800
    ACLWAK=0.0                                                           VTN00810
    ACLSPH=0.0                                                           VTN00820
    CALL TRNCHK(SUM1,ACLWAK,ACLSPH)                                      VTN00830
    IF(TEST)GO TO 998                                                    VTN00840
 50 CONTINUE                                                             VTN00850
    SUM=SUM*TINC                                                         VTN00860
    ACL=OWF(NWL,NSOIL)*SUM                                               VTN00870
    TRNLOS=EXP(-ACL)                                                     VTN00880
    GO TO 999                                                            VTN00890
998 TRNLOS=0.0                                                           VTN00900
999 RETURN                                                               VTN00910
    END                                                                  VTN00920
```

```
      SUBROUTINE VSRC(VSPD,VWID,VWHT,VEHTYP,NSOIL,SILT)                 VRS00320
C                                                                       VRS00010
CCC   THIS SUBROUTINE INITIALIZES THE DUST CLOUD PRODUCED BY A VEHICLE  VRS00020
CCC                                                                     VRS00030
CCC   INPUTS                                                            VRS00040
CCC                                                                     VRS00050
CCC      VSPD  - THE VELOCITY OF THE VEHICLE                            VRS00060
CCC                                                                     VRS00070
CCC      VWID  - WIDTH OF THE VEHICLE                                   VRS00080
CCC                                                                     VRS00090
CCC      VWHT  - WEIGHT OF THE VEHICLE IN KGS.                          VRS00100
CCC                                                                     VRS00110
CCC      NSOIL - SOIL TYPE                                              VRS00120
CCC                                                                     VRS00130
CCC      SILT  - SILT CONTENT OF THE SOIL                               VRS00140
CCC                                                                     VRS00150
CCC                                                                     VRS00160
CCC   OUTPUTS                                                           VRS00170
CCC                                                                     VRS00180
CCC    STOPED IN COMMON/DISCS/ AND COMMON/PRE/                          VRS00190
CCC                                                                     VRS00200
CCC      NDSCS - NUMBER OF DISCS (FOR A VEHICLE ONLY ONE)               VRS00210
CCC                                                                     VRS00220
CCC      TDSC -  TIME OF RELEASE OF THE DISCS                           VRS00230
CCC                                                                     VRS00240
CCC      XDSC -  X POSITION OF DISC AT TIME OF RELEASE                  VRS00250
CCC                                                                     VRS00260
CCC      ZDSC -  HEIGHT OF RELEASE OF THE DISC                          VRS00270
CCC                                                                     VRS00280
CCC      R2DSC - SQUARE OF THE RADIUS OF THE DISC                       VRS00290
CCC                                                                     VRS00300
CCC   ***************************************************************** VRS00310
      INTEGER VEHTYP                                                    VRS00330
      COMMON/PRTINF/R0,VGRAV(3),NPRTS                                   VRS00340
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUVRS00350
      COMMON/DISCS/NDSCS,TDSC(20),XDSC(20),ZDSC(20),R2DSC(20),QDSC(20,3)VRS00360
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV                         VRS00370
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                                VRS00380
      COMMON/PRE/Z,RT2DZ                                                VRS00390
      COMMON/VL/VLOAD                                                   VRS00400
      NDSCS=1                                                           VRS00410
      NPRTS=1                                                           VRS00420
      VGRAV(1)=0.0                                                      VRS00430
      QDSC(1,1)=1.0                                                     VRS00440
C                                                                       VRS00450
CC    INITIALIZE THE VEHICLE SOURCE FOR WHEELED VEHICLES                VRS00460
C                                                                       VRS00470
      ZDSC(1)=VWID/8.                                                   VRS00480
      Z=VWID/4.                                                         VRS00490
      ZZ=VWID/4.                                                        VRS00500
      DZ=DIFFUS(Z0,ZL,ZZ)                                               VRS00510
      DX=DXZ0*DZ                                                        VRS00520
      TDSC(1)=-9.*(VWID**2)/512./DZ                                     VRS00530
      TOF=-TDSC(1)                                                      VRS00540
      CALL MOMENT(VGRAV,Z,ZDSC(1),TOF,Q,XBAR,SIGW2,SIGP2)               VRS00550
      XDSC(1)=-XBAR                                                     VRS00560
      A=1.0                                                             VRS00570
      B=SIGW2+SIGP2                                                     VRS00580
      C=4.*((SIGW2*SIGP2)-((VWID/3.)**4))                               VRS00590
      RAD=B**2-A*C                                                      VRS00600
      R2DSC(1)=-B+SQRT(RAD)                                             VRS00610
      RT2DZ=SQRT(2.*DZ)                                                 VRS00620
C                                                                       VRS00630
CC    INTIALIZE LOADING FOR VEHICLE (VLOAD IN KG/SEC)                   VRS00640
C                                                                       VRS00650
   20 SILTPC=100.*SILT                                                  VRS00660
      A=3.8E-9                                                          VRS00670
      IF(VEHTYP.GT.0)A=1.52E-08                                         VRS00680
      Q=A*VSPD*VWHT*SILTPC                                              VRS00690
      ALPHA=240.                                                        VRS00700
```

```
      VLOAD=ALPHA*VSPD*Q                                                VRS00710
  999 RETURN                                                            VRS00720
      END                                                               VRS00730
```

```
      SUBROUTINE VSUM(A,B,S,C)                                          VSU00010
      DIMENSION A(2),B(2),C(2)                                          VSU00020
C *** C=A+S*B WHERE A,B,C ARE VECTORS AND S IS SCALAR                   VSU00030
      DO 14 J=1,2                                                       VSU00040
   14 C(J)=A(J)+S*B(J)                                                  VSU00050
      RETURN                                                            VSU00060
      END                                                               VSU00070
```

```
      SUBROUTINE WIN(Z,U,V)                                              WIN00100
      COMMON/STARS/USTAR,TSTAR,ZSTAR                                     WIN00200
      COMMON/EKWIND/ALP,C,PYF,PXF,UHAT,VHAT                              WIN00300
      COMMON/EKTEMP/Z0,ZL,T0,TC1,TC2,TC3                                 WIN00400
      COMMON/BUOYCL/Y(8),SPHNS(3),RISTIM                                 WIN00500
C*********************************************************************  WIN00600
C                                                                        WIN00700
C  PURPOSE                                                               WIN00800
C                                                                        WIN00900
C     TO COMPUTE THE WIND SPEED AT A SPECIFIED HEIGHT                    WIN01000
C                                                                        WIN01100
C     INPUTS                                                             WIN01200
C       Z    HEIGHT AT WHICH WIND SPEEDS ARE DESIRED                     WIN01400
C                                                                        WIN01500
C     OUTPUTS                                                            WIN01600
C                                                                        WIN01700
C       U    WIND SPEED IN THE DIRECTION OF THE GROUND WIND              WIN01800
C                                                                        WIN01900
C       V    WIND SPEED PERPENDICULAR TO THE GROUND WIND                 WIN02000
C                                                                        WIN02100
C     CALLED BY DIFEQ                                                    WIN02200
C                                                                        WIN02300
C     SUBROUTINES AND FUNCTIONS NEEDED                                   WIN02400
C                                                                        WIN02500
C        WNDCAL    CALCULATES SCALED WIND SPEED                          WIN02600
C                                                                        WIN02700
C*********************************************************************  WIN02800
      IF(Z.GT.ZSTAR)GO TO 100                                            WIN02900
      U=USTAR*WNDCAL(Z0,ZL,Z)                                            WIN03000
      V=0.0                                                              WIN03100
      GO TO 999                                                          WIN03200
  100 UE=C*EXP(-ALP*Z)*COS(ALP*Z)-PYF                                    WIN03300
      VE=-C*EXP(-ALP*Z)*SIN(ALP*Z)+PXF                                   WIN03400
      U=UHAT*UE+VHAT*VE                                                  WIN03500
      V=-VHAT*UE+UHAT*VE                                                 WIN03600
  999 RETURN                                                             WIN03700
      END                                                                WIN03800
```

```
      FUNCTION WNDCAL(Z0,ZL,Z)                                          WND00010
C ***************************************************************     WND00020
C                                                                       WND00030
C                                                                       WND00040
C     PURPOSE                                                           WND00050
C                                                                       WND00060
C     TO CALCULATE THE WIND SPEED, U/U*, SCALED BY THE FRICTION         WND00070
C     VELOCITY FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV LENGTH AT A  WND00080
C     SPECIFIED HEIGHT.                                                 WND00090
C                                                                       WND00100
C                                                                       WND00110
C     INPUTS                                                            WND00120
C                                                                       WND00130
C     Z0       THE FRICTION HEIGHT IN METERS.                           WND00140
C     ZL       THE MONIN-OBUKHOV LENGTH IN METERS.                      WND00150
C     Z        THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED       WND00160
C              IN METERS                                                WND00170
C                                                                       WND00180
C                                                                       WND00190
C     RETURNS VELOCITY SCALED BY FRICTION VELOCITY                      WND00200
C                                                                       WND00210
C                                                                       WND00220
C     CALLED BY ATMCAL,WIN AND RISE                                     WND00230
C                                                                       WND00240
C ***************************************************************     WND00250
      LOGICAL LOW                                                       WND00260
      COMMON/COEF/AW,CW,BW,DW,AT,CT,BT,DT                               WND00270
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUWND00280
      PSIM(Z,Z1)=ALOG((1.-Z)/(1.-Z1))-ALOG((1.+Z)/(1.+Z1))+             WND00290
     $2.*(ATAN(Z)-ATAN(Z1))                                             WND00300
      PSIMS(Z)=-7.*Z                                                    WND00310
      PHIM(Z)=(1.-16.*Z)**(-.25)                                        WND00320
C                                                                       WND00330
C     PHIM       THE SHEAR OF MOMENTUM                                  WND00340
C     PSIM       THE UNIVERSAL FUNCTION FOR THE DEVIATION FROM          WND00350
C                LOGARITHMIC WIND VELOCITY BOUNDARY LAYER PROFILE IN AN WND00360
C                UNSTABLE ATMOSPHERE                                    WND00370
C     PSIMS      THE SAME AS PSIM FOR A STABLE ATMOSPHERE               WND00380
C                                                                       WND00390
      IF(ABS(ZL).LE.1.E3)GO TO 100                                      WND00400
      WNDCAL=ALOG(1.+Z/Z0)                                              WND00410
      GO TO 999                                                         WND00420
  100 CONTINUE                                                          WND00430
      P=SIGN(1.,ZL)                                                     WND00440
      LOW=.TRUE.                                                        WND00450
      S=Z/ZL                                                            WND00460
      IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10                                 WND00470
      S=AMIN1(S,1.5)                                                    WND00480
      S=AMAX1(S,-2.)                                                    WND00490
      LOW=.FALSE.                                                       WND00500
   10 CONTINUE                                                          WND00510
      IF(P)120,130,130                                                  WND00520
  120 S=1./PHIM(S)                                                      WND00530
      S1=Z0/ZL                                                          WND00540
      S0=1./PHIM(S1)                                                    WND00550
      WNDCAL=PSIM(S,S0)                                                 WND00560
C                                                                       WND00570
C DETERMINE THE CONSTANTS FOR MATCHING AT Z/ZL=-2.                      WND00580
C                                                                       WND00590
      S2=-2.                                                            WND00600
      AW=-3.*((1-16.*(S2))**(-.25))*((-S2)**(1./3.))                    WND00610
      CW=-1.*AW*(-S2)**(-1./3.)                                         WND00620
      GO TO 52                                                          WND00630
  130 CONTINUE                                                          WND00640
      PSI=PSIMS(S)                                                      WND00650
      WNDCAL=ALOG(1.+S*ZL/Z0)-PSI                                       WND00660
C                                                                       WND00670
C  FIND THE CONSTANTS FOR MATCHING OF STABLE PROFILE AT Z/ZL=1.5        WND00680
C                                                                       WND00690
                                                                        WND00700
```

```
      S2=1.5                                                            WND00710
      BW=1./(Z0/ZL+S2)+7.                                               WND00720
      DW=-1.*S2*BW                                                      WND00730
   52 CONTINUE                                                          WND00740
      IF(LOW)GO TO 999                                                  WND00750
      IF(P)53,53,54                                                     WND00760
   53 WNDCAL=WNDCAL+CW+AW*(-ZL/Z)**(1./3.)                              WND00770
      GO TO 999                                                         WND00780
   54 WNDCAL=WNDCAL+DW+BW*Z/ZL                                          WND00790
  999 WNDCAL=WNDCAL/.4                                                  WND00800
  998 CONTINUE                                                          WND00810
      RETURN                                                            WND00820
      END                                                               WND00830
```

```
      SUBROUTINE NMMW(FREQGH,ICLMAT,MMTRAN,IERR)                        NMM00010
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     NMM00020
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,             NMM00030
     +        FOGPRB,WNDVEL,WNDDIR,IPASCT                               NMM00040
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUNMM00050
      COMMON /GEOMET/PTS(15),IGEOSW                                     NMM00060
C-----------------------------------------------------------------    NMM00070
C   PURPOSE: TO CALCULATE THE EXTINCTION AND BACKSCATTER AT MILLIMETER  NMM00080
C            FREQUENCIES 10 TO 1000 GHZ, DUE TO                         NMM00090
C            GASEOUS ABSORPTION, FOG(ICE AND WATER) AND                 NMM00100
C            CLOUD BULK ATTENUATION, RAIN, SNOW EXTINCTION.             NMM00110
C-----------------------------------------------------------------    NMM00120
C                                                                       NMM00130
C** INPUT TO THE NMMW MODULE IS PERFORMED THROUGH A CARD ORDER-         NMM00140
C** INDEPENDENT INPUT TECHNIQUE. A FOUR-LETTER IDENTIFIER IN COLS.      NMM00150
C** 1-4 OF EACH RECORD SPECIFIES THE TYPE OF DATA BEING READ BY THE     NMM00160
C** MODULE. THE INPUT CARDS MAY APPEAR IN ANY ORDER WITH THE EXCEPTION  NMM00170
C** OF OF THE (GO) END OF READ SENTINEL, WHICH MUST BE THE LAST CARD    NMM00180
C** READ. ALL OF THE FOLLOWING CARDS ARE READ IN UNDER THE FORMAT       NMM00190
C** (A4,1X,3(E10.4,1X)) :                                              NMM00200
C                                                                       NMM00210
C-----------------------------------------------------------------    NMM00220
C       CARD IDENTIFIER : PATH                                          NMM00230
C       VARIABLES READ : MMWPTH       (PATH LENGTH (KM))                NMM00240
C-----------------------------------------------------------------    NMM00250
C       CARD IDENTIFIER : ATMO                                          NMM00260
C       VARIABLES READ : TEMP1        (TEMPERATURE (DEG C))             NMM00270
C                        PRESS1       (PRESSURE (MB))                   NMM00280
C                        ABSHUM       (IF .GT. 0. ABSOLUTE              NMM00290
C                                      HUMIDITY (GM/M**3))              NMM00300
C                                     (IF .LT. 0. RELATIVE              NMM00310
C                                      HUMIDITY (%))                    NMM00320
C    ***NOTE: TEMP1,PRESS1,ABSHUM WILL BE PASSED FROM CLIMAT IF         NMM00330
C             ICLMAT=1. IN THAT EVENT, THE (ATMO) CARD IS NOT NEEDED.   NMM00340
C-----------------------------------------------------------------    NMM00350
C       CARD IDENTIFIER : FOGD                                          NMM00360
C       VARIABLES READ : FOGDEN       (FOG DENSITY (LIQUID WATER,       NMM00370
C                                     GM/M**3))                         NMM00380
C-----------------------------------------------------------------    NMM00390
C       CARD IDENTIFIER : RAIN                                          NMM00400
C       VARIABLES READ : RAINRT       (RAIN RATE (MM/HR))               NMM00410
C-----------------------------------------------------------------    NMM00420
C       CARD IDENTIFIER : SNOW                                          NMM00430
C       VARIABLES READ : SNOWRT       (SNOW RATE (MM/HR) WATER EQUIV.)  NMM00440
C-----------------------------------------------------------------    NMM00450
C       CARD IDENTIFIER : GO                                            NMM00460
C       VARIABLES READ : NONE         (END OF READ SENTINEL)            NMM00470
C-----------------------------------------------------------------    NMM00480
C                                                                       NMM00490
C  MAIN ROUTINE CALLS NMMW. NMMW THEN CALLS                             NMM00500
C    (1)MMWGS(GAS ABSORPTION), (2)MMWFG(FOG ABSORPTION,BACKSCATTER),    NMM00510
C    (3)MMRAN(RAIN EXTINCTION,BACKSCATTER),                             NMM00520
C    (4)MMSNO(SNOW EXTINCTION,BACKSCATTER), AND RETURNS TO MAIN NMMW    NMM00530
C                                                                       NMM00540
C  LOCAL VARIABLES                                                      NMM00550
C-----------------------------------------------------------------    NMM00560
      REAL MMWPTH,MMTRAN,MMBSXS                                         NMM00570
      DIMENSION DAT(3),IAL(6)                                           NMM00580
      DATA IAL/2HPA  ,2HAT  ,2HFO  ,2HRA  ,2HSN  ,2HGO  /               NMM00590
      IERR=0                                                            NMM00600
      GASABS=0.                                                         NMM00610
      FOGEXT=0.                                                         NMM00620
      RAINEX=0.                                                         NMM00630
      FOGBS=0.                                                          NMM00640
      RAINBS=0.                                                         NMM00650
      SNOWBS=0.                                                         NMM00660
    5 READ(IOIN,400)IALFA,IALFA2,(DAT(L),L=1,3)                         NMM00670
  400 FORMAT(2A2,1X,3(E10.4,1X))                                        NMM00680
      IF(IALFA.EQ.IAL(1)) GO TO 10                                      NMM00690
      IF(IALFA.EQ.IAL(2)) GO TO 20                                      NMM00700
```

```
      IF(IALFA.EQ.IAL(3)) GO TO 30                                      NMM00710
      IF(IALFA.EQ.IAL(4)) GO TO 40                                      NMM00720
      IF(IALFA.EQ.IAL(5)) GO TO 50                                      NMM00730
      IF(IALFA.EQ.IAL(6)) GO TO 60                                      NMM00740
      WRITE(IOOUT,450)IALFA,IALFA2                                      NMM00750
  450 FORMAT(1H0,20X,2A2,22H IS AN INCORRECT LABEL //)                  NMM00760
      GO TO 300                                                         NMM00770
   10 MMWPTH=DAT(1)                                                     NMM00780
      GO TO 5                                                           NMM00790
   20 TEMP1=DAT(1)                                                      NMM00800
      PRESS1=DAT(2)                                                     NMM00810
      ABSHUM=DAT(3)                                                     NMM00820
      GO TO 5                                                           NMM00830
   30 FOGDEN=DAT(1)                                                     NMM00840
      GO TO 5                                                           NMM00850
   40 RAINRT=DAT(1)                                                     NMM00860
      GO TO 5                                                           NMM00870
   50 SNOWRT=DAT(1)                                                     NMM00880
      GO TO 5                                                           NMM00890
   60 CONTINUE                                                          NMM00900
      IF(IGEOSW.NE.1)GO TO 99                                           NMM00910
      MMWPTH=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+                NMM00920
     +(PTS(6)-PTS(3))**2)                                               NMM00930
   99 CONTINUE                                                          NMM00940
      IF(ICLMAT.EQ.1) TEMP1=TEMP                                        NMM00950
      IF(ICLMAT.EQ.1) PRESS1=PRESS                                      NMM00960
      IF(ICLMAT.EQ.1) ABSHUM=AH                                         NMM00970
      WRITE(IOOUT,500) TEMP1,PRESS1,ABSHUM,FOGDEN                       NMM00980
      WRITE(IOOUT,600) RAINRT,SNOWRT,FREQGH,MMWPTH                      NMM00990
C CHECK MODEL INPUTS FOR RANGE OF VALIDITY.                             NMM01000
      IF(PRESS1.GE.500.) GO TO 100                                      NMM01010
      IERR=1                                                            NMM01020
      WRITE(IOOUT,800)                                                  NMM01030
  100 IF(FREQGH.LE.1000..AND.FREQGH.GE.10.) GO TO 150                   NMM01040
      IERR=1                                                            NMM01050
      WRITE(IOOUT,900)                                                  NMM01060
  150 IF(IERR.EQ.1) GO TO 300                                           NMM01070
C CHANGE UNITS                                                          NMM01080
      PRSTOR=PRESS1/TORRMB                                              NMM01090
      TEMPDK=TEMP1+CDEGK                                                NMM01100
C CALL INDIVIDUAL MODULES FOR GAS, FOG/CLOUD, SNOW, RAIN EXTINCTION     NMM01110
      CALL MMWGS(TEMPDK,PRSTOR,ABSHUM,FREQGH,GASABS)                    NMM01120
      IF(FOGDEN.GT.1.E-10)                                              NMM01130
     +    CALL MMWFG(FOGDEN,TEMPDK,FREQGH,FOGEXT,FOGBS)                NMM01140
      IF(RAINRT.GT.1.E-10)                                              NMM01150
     +    CALL MMRAN(RAINRT,TEMPDK,FREQGH,2.,RAINEX,RAINBS)            NMM01160
      IF(SNOWRT.GT.1.E-10)                                              NMM01170
     +    CALL MMSNO(SNOWRT,TEMPDK,FREQGH,SNOWEX,SNOWBS)               NMM01180
C COMPUTE TRANSMISSION                                                  NMM01190
      TOTEXT=GASABS+FOGEXT+RAINEX+SNOWEX                                NMM01200
      MMBSXS=FOGBS+RAINBS+SNOWBS                                        NMM01210
      MMTRAN=EXP(-MMWPTH*TOTEXT)                                        NMM01220
C CHANGE UNITS FOR ABSORPTION/EXTINCTION FROM 1/KM TO DB/KM.            NMM01230
      DBKM=4.343                                                        NMM01240
      GASABS=GASABS*DBKM                                                NMM01250
      FOGEXT=FOGEXT*DBKM                                                NMM01260
      RAINEX=RAINEX*DBKM                                                NMM01270
      SNOWEX=SNOWEX*DBKM                                                NMM01280
      WRITE(IOOUT,700) GASABS,FOGEXT,RAINEX,SNOWEX,MMTRAN               NMM01290
      WRITE(IOOUT,750) FOGBS,RAINBS,SNOWBS,MMBSXS                       NMM01300
C  COMPUTATION COMPLETED                                                NMM01310
  300 RETURN                                                            NMM01320
C                                                                       NMM01330
500   FORMAT(1H0,///,47X,12HTEMPERATURE ,14X,F8.3,                      NMM01340
     +       10H DEGREES C,/,47X,9HPRESSURE ,17X,F8.3,                  NMM01350
     +       3H MB,/,47X,17HABSOLUTE HUMIDITY,9X,F8.3,                  NMM01360
     +       7H G/M**3/,47X,11HFOG DENSITY,15X,F8.3,                    NMM01370
     +       7H G/M**3)                                                 NMM01380
600   FORMAT(1H ,46X,9HRAIN RATE,17X,F8.3,6H MM/HR,/                    NMM01390
     +       ,47X,9HSNOW RATE,17X,F8.3,6H MM/HR,/                       NMM01400
```

```
     +        ,47X,9HFREQUENCY,17X,F8.3,4H GHZ/                              NMM01410
     +        ,47X,11HPATH LENGTH,15X,F8.3,3H KM)                            NMM01420
700   FORMAT(1H0,46X,14HGAS ABSORPTION,10X,E10.4,6H DB/KM,/                  NMM01430
     +        ,47X,15HFOG EXTINCTION ,9X,E10.4,6H DB/KM,/                    NMM01440
     +        ,47X,16HRAIN EXTINCTION ,8X,E10.4,6H DB/KM,/                   NMM01450
     +        ,47X,16HSNOW EXTINCTION ,8X,E10.4,6H DB/KM,/                   NMM01460
     +        ,47X,13HTRANSMISSION ,11X,E10.4/)                              NMM01470
 750   FORMAT(1H ,46X,15HFOG BACKSCATTER,9X,E10.4,10H M**2/M**3,/             NMM01480
     +        ,47X,16HRAIN BACKSCATTER,8X,E10.4,10H M**2/M**3,/              NMM01490
     +        ,47X,16HSNOW BACKSCATTER,8X,E10.4,10H M**2/M**3,/              NMM01500
     +        ,47X,17HTOTAL BACKSCATTER,7X,E10.4,10H M**2/M**3)              NMM01510
800   FORMAT(1H0,47X,41HPRESSURE LESS THAN 500 MB, GAS ABSORPTION,           NMM01520
     +       19H WILL BE INACCURATE)                                         NMM01530
900   FORMAT(1H0,47X,31HFREQUENCY<10. GHZ, OR >1000 GHZ,                     NMM01540
     +       42H CALULATION WILL FAIL, USE OTHER FREQUENCY)                  NMM01550
      END                                                                    NMM01560
```

```
      FUNCTION AB(WA,A,CE,B,C)                                          AB 00010
      AB=A*EXP(-ABS((ALOG10(1.E4*WA/CE)/B))**C)                         AB 00020
      RETURN                                                            AB 00030
      END                                                               AB 00040
```

```
FUNCTION DOP(WA,A,CE1,B,C,CE2,D,E,CE3,F,G)                                    DOP00010
V=1./WA                                                                       DOP00020
V2=V*V                                                                        DOP00030
H1=CE1*CE1-V2                                                                 DOP00040
H2=CE2*CE2-V2                                                                 DOP00050
H3=CE3*CE3-V2                                                                 DOP00060
DOP=SQRT(A+B*H1/(H1*H1+C*V2)+D*H2/(H2*H2+E*V2)+F*H3/(H3*H3+G*V2))             DOP00070
RETURN                                                                        DOP00080
END                                                                           DOP00090
```

```
      SUBROUTINE INTRP(A,B,T,F,TT,FF,AA,BB,J)                          INT00010
C                                                                      INT00020
C-----------------------------------------------------------------     INT00030
C     PURPOSE: TO DO FREQUENCY AND TEMPERATURE INTERPOLATION           INT00040
C-----------------------------------------------------------------     INT00050
      DIMENSION A(9,3), B(9,3), F(9), T(3)                             INT00060
C                                                                      INT00070
      IF(TT.LT.T(1)) TT=T(1)                                           INT00080
      DO 11 J=2,3                                                      INT00090
         IF(TT.LT.T(J)) GO TO 14                                       INT00100
   11 CONTINUE                                                         INT00110
      TT=T(3)                                                          INT00120
      J=3                                                              INT00130
   14 CONTINUE                                                         INT00140
      DO 15 I=2,9                                                      INT00150
      IF(FF.LT.F(I)) GO TO 16                                          INT00160
   15 CONTINUE                                                         INT00170
      FF=F(9)                                                          INT00180
      I=9                                                              INT00190
   16 FF=ALOG10(FF)                                                    INT00200
      F0=ALOG10(F(I))                                                  INT00210
      F1=ALOG10(F(I-1))                                                INT00220
      FF0=(F0-FF)/(F0-F1)                                              INT00230
      FF1=(FF-F1)/(F0-F1)                                              INT00240
      TF0=(T(J)-TT)/(T(J)-T(J-1))                                      INT00250
      TF1=(TT-T(J-1))/(T(J)-T(J-1))                                    INT00260
      A11=ALOG10(A(I-1,J-1))                                           INT00270
      A01=ALOG10(A(I,J-1))                                             INT00280
      A10=ALOG10(A(I-1,J))                                             INT00290
      A00=ALOG10(A(I,J))                                               INT00300
      APJ1=A11*FF0+A01*FF1                                             INT00310
      APJ0=A10*FF0+A00*FF1                                             INT00320
      AA=APJ1*TF0+APJ0*TF1                                             INT00330
      B11=ALOG10(B(I-1,J-1))                                           INT00340
      B10=ALOG10(B(I-1,J))                                             INT00350
      B01=ALOG10(B(I,J-1))                                             INT00360
      B00=ALOG10(B(I,J))                                               INT00370
      BPJ1=B11*FF0+B01*FF1                                             INT00380
      BPJ0=B10*FF0+B00*FF1                                             INT00390
      BB=BPJ1*TF0+BPJ0*TF1                                             INT00400
      AA=10.**AA                                                       INT00410
      BB=10.**BB                                                       INT00420
      RETURN                                                           INT00430
      END                                                              INT00440
```

```
      SUBROUTINE MMH2O(V,T,PTOT,PH2O,DATH2O,ABH2O)                      MMH00010
C                                                                       MMH00020
C*************************************************************        MMH00030
C                                                                       MMH00040
C  ROUTINE TO CALCULATE H2O VAPOR ABSORPTION FOR 0 TO 350 GHZ.          MMH00050
C                                                                       MMH00060
C  INPUTS ARE: WAVENUMBER(/CM), TEMPERATURE(KELVIN), TOTAL              MMH00070
C    PRESSURE(TORR), H2O VAPOR PRESSURE(TORR), LINE DATA ARRAY.         MMH00080
C  OUTPUT IS: H2O VAPOR ABSORPTION                                      MMH00090
C                                                                       MMH00100
C  CALLED BY MMWGS   MAKES NO CALLS                                     MMH00110
C                                                                       MMH00120
C  LOCAL VARIABLES:                                                     MMH00130
C    WCD       VAPOR COLUMN DENSITY(/CM/CM/KM)                         MMH00140
C    CT        LINE STRENGTH TEMPERATURE CORRECTION                    MMH00150
C    CA        LINE WIDTH SELF BROADENING AND TEMP. CORRECTION         MMH00160
C    SA        CORRECTED LINE STRENGTH                                 MMH00170
C    GA        CORRECTED LINE WIDTH                                    MMH00180
C    ABS       SINGLE LINE ABSORPTION(/KM)                             MMH00190
C*************************************************************        MMH00200
      DIMENSION DATH2O(37,4)                                            MMH00210
C                                                                       MMH00220
      ABH2O=0.                                                          MMH00230
      WCD=7.33994E26*PH2O/760./T*(PFR(T))                               MMH00240
      CT=4.860773E-3*(T-296.)/T                                         MMH00250
      CA=((296./T)**.62)*(PTOT+(4.*PH2O))/760.                          MMH00260
C                                                                       MMH00270
      DO 500 L=1,37                                                     MMH00280
         SA=DATH2O(L,2)*WCD*EXP(DATH2O(L,4)*CT)                         MMH00290
         GA=DATH2O(L,3)*CA                                              MMH00300
         ABS=SA*SUPK(V,DATH2O(L,1),GA)                                  MMH00310
         ABH2O=ABH2O+ABS                                                MMH00320
500   CONTINUE                                                          MMH00330
      RETURN                                                            MMH00340
      END                                                               MMH00350
```

```
      SUBROUTINE MMIDX(XL,T,ICE,H2OAB,H2OK2)                            MMI00010
C   REF: RAY, APPLIED OPTICS, VOL. 11, P. 1836,(1972)                   MMI00020
      COMPLEX CINDX,XK                                                  MMI00030
      XXL=XL/10.                                                        MMI00040
      TT=T-273.16                                                       MMI00050
      T3=TT-25.                                                         MMI00060
      IF(ICE.NE.0) GO TO 150                                            MMI00070
C  PARAMETERS FOR WATER                                                 MMI00080
100   EFIN=5.27137+.0216474*TT-1.31198E-3*TT*TT                         MMI00090
      ALFA=-16.8129/T+.0609265                                          MMI00100
      XLS=3.3836E-4*EXP(2513.98/T)                                      MMI00110
      SIGMA=1.25664E9                                                   MMI00120
      ES=78.54*(1.-4.579E-3*T3+1.19E-5*T3*T3-2.8E-8*T3*T3*T3)           MMI00130
      GO TO 200                                                         MMI00140
C                                                                       MMI00150
C     PARAMETERS FOR ICE                                                MMI00160
  150 EFIN=3.168                                                        MMI00170
      ALFA=0.288+0.0052*TT+2.3E-4*TT*TT                                 MMI00180
      XLS=9.990288E-5*EXP(1.32E4/(1.9869*T))                            MMI00190
      SIGMA=1.26*EXP(-1.25E4/(1.9869*T))                                MMI00200
      ES=203.168+2.5*TT+0.15*TT*TT                                      MMI00210
200   U=(ES-EFIN)*(XLS/XXL)**(1.-ALFA)                                  MMI00220
      Y=1.+2.*(XLS/XXL)**(1.-ALFA)*SIN(ALFA*1.57079633)+                MMI00230
     1   (XLS/XXL)**(2-2*ALFA)                                          MMI00240
      EP=EFIN+((ES-EFIN)+U*SIN(ALFA*1.57079633))/Y                      MMI00250
      EPP=(U*COS(ALFA*1.57079633))/Y+SIGMA*XXL/1.88496E11               MMI00260
      RE=SQRT((EP+SQRT(EP*EP+EPP*EPP))/2.)                              MMI00270
      AI=-EPP/2./RE                                                     MMI00280
      IF(ICE.NE.0) GO TO 400                                            MMI00290
C                                                                       MMI00300
      IF(XXL.LE..034) GO TO 307                                         MMI00310
      IF(XXL.GT..1)  GO TO 311                                          MMI00320
306   R2=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,          MMI00330
     +       259913.,161.29,43319.7,27661.2)                            MMI00340
      R2=R2+R2*T3*1.E-3*EXP((2.5E-5*XXL)**.25)                          MMI00350
      RE=RE*(XXL-.034)/.066+R2*(.1-XXL)/.066                            MMI00360
      GO TO 311                                                         MMI00370
307   RE=DOP(XXL,1.83899,1639.,52340.4,10399.2,588.24,345005.,          MMI00380
     +       259913.,161.29,43319.7,27661.2)                            MMI00390
      RE=RE+RE*T3*1.E-3*EXP((2.5E-5*XXL)**.25)                          MMI00400
311   CONTINUE                                                          MMI00410
      IF(XXL.GT. .3) GO TO 500                                          MMI00420
      AI=AI+AB(XXL,.25,300.,.47,3.)+AB(XXL,.39,17.,.45,1.3)             MMI00430
     +     +AB(XXL,.41,62.,.35,1.7)                                     MMI00440
      GO TO 500                                                         MMI00450
C                                                                       MMI00460
400   CONTINUE                                                          MMI00470
      IF(XXL.GT.0.08) GO TO 500                                         MMI00480
  405 RC=DOP(XXL,1.225,1652.9,1.12082E6,46E-11,909.09,416441.,118852.,  MMI00490
     +       223.2,47031.8,126834.)                                     MMI00500
      RE=RE*(XXL-0.02)/0.06+R2*(0.08-XXL)/0.06                          MMI00510
      AI=AI+AB(XXL,.242,62.,.23,1.6)+AB(XXL,.581,44.8,0.055,1.)         MMI00520
C                                                                       MMI00530
500   CINDX=CMPLX(RE,AI)                                                MMI00540
      XK=(CINDX*CINDX-1)/(CINDX*CINDX+2)                                MMI00550
      H2OAB=AIMAG(-XK)                                                  MMI00560
      H2OK2=XK*CONJG(XK)                                                MMI00570
      RETURN                                                            MMI00580
      END                                                               MMI00590
```

```
      SUBROUTINE MMOXY(V,T,PTOT,PH2O,DATA02,ABS02)                      MM000010
C                                                                       MM000020
C*****************************************************                  MM000030
C  ROUTINE TO CALCULATE ABSORPTION DUE TO OXYGEN, METHOD IS THAT OF     MM000040
C      LIEBE, GIMMESTAD,& HOPPONEN. IEEE TRANS. ANT. PROP. V.25, P327.  MM000050
C                                                                       MM000060
C  INPUTS ARE:  FREQUENCY(GHZ), TEMPERATURE(KELVIN), TOTAL              MM000070
C      PRESSURE(TORR), H2O VAPOR PRESSURE(TORR), O2 LINE DATA           MM000080
C      ARRAY.                                                           MM000090
C  OUTPUTS ARE: O2 ABSORPTION (1/KM)                                    MM000100
C                                                                       MM000110
C  CALLED FROM MMWGS  CALLS NO OTHER ROUTINES.                          MM000120
C                                                                       MM000130
C  LOCAL VARIABLES:                                                     MM000140
C      T2          300./T                                            MM000150
C      PHI         TEMPERATURE CORRECTION FOR LINE STRENGTHS          MM000160
C      S           CORRECTED LINE STRENGTH(HZ TORR)                   MM000170
C      GAMMA       CORRECTED LINE WIDTH (1/GHZ.)                      MM000180
C      XIF         LINE INTERFERENCE FACTOR                           MM000190
C      VMI         DATA02(L,1)-V                                      MM000200
C      VPL         DATA02(L,1)+V                                      MM000210
C      PROFIL      MODIFIED VANVLECK-WEISSKOPF LINE SHAPE             MM000220
C                                                                       MM000230
      DIMENSION DATA02(42,6)                                            MM000240
C                                                                       MM000250
      ABS02=0.                                                          MM000260
      T2=300./T                                                         MM000270
C                                                                       MM000280
      DO 500 L=1,42                                                     MM000290
         K=IFIX(DATA02(L,6))                                            MM000300
         PHI=T2*T2*T2*EXP(-6.895E-3*K*(K+1)*(T2-1.))                    MM000310
         S=0.2095*PTOT*DATA02(L,2)*PHI                                  MM000320
         GAMMA=DATA02(L,3)*(.929*PTOT*T2**.9+1.3*T2*PH2O)*1.E-3         MM000330
         XIF=DATA02(L,4)*T2**DATA02(L,5)*PTOT*1.E-3                     MM000340
         VMI=DATA02(L,1)-V                                              MM000350
         VPL=VMI+2*V                                                    MM000360
         PROFIL=(V/DATA02(L,1))*((GAMMA-VMI*XIF)/(VMI*VMI+GAMMA*GAMMA)+ MM000370
     *          (GAMMA-VPL*XIF)/(VPL*VPL+GAMMA*GAMMA))                  MM000380
C                                                                       MM000390
         ABS02=ABS02+S*PROFIL*V*4.192E-5                                MM000400
C                                                                       MM000410
500   CONTINUE                                                          MM000420
      RETURN                                                            MM000430
      END                                                               MM000440
```

```
      SUBROUTINE MMRAN(RAINRT,T,FREQ,RTYPE,GRAIN,BSRAIN)                MMR00010
C  REF: OLSEN, ET.AL., IEEE ANT. PROP., VOL.26, P.318(1978)             MMR00020
C-----------------------------------------------------------------      MMR00030
C                                                                       MMR00040
C  ROUTINE TO COMPUTE THE ATTENUATION DUE TO RAIN, FOR                  MMR00050
C    FREQUENCIES BETWEEN 10 & 1000 GHZ; BACKSCATTER ALSO.               MMR00060
C                                                                       MMR00070
C  INPUTS ARE: FREQUENCY(GHZ), TEMPERATURE(KELVIN), RAIN RATE(MM/HR)    MMR00080
C  OUTPUT IS ATTENUATION (1/KM), BACKSCATTER (MM**2/MM**3)              MMR00090
C                                                                       MMR00100
C  CALLED FROM MMWMOD  CALLS NO SUBROUTINES.                            MMR00110
C                                                                       MMR00120
C  LOCAL VARIABLES:                                                     MMR00130
C      ALFA,BETA                  A AND B PARAMETERS FUNCTION OF FREQ,  MMR00140
C                                  TEMPERATURE, AND RAIN TYPE           MMR00150
C      BSAT,BSBT                  POLYNOMIAL COEF.S FOR LINEAR FIT     MMR00160
C                                  TO BACKSCATTER= ALF*RAINRT**BET      MMR00170
C-----------------------------------------------------------------      MMR00180
      DIMENSION ALFA(9,3,3), BETA(9,3,3), F(9), TK(3),A(9,3),B(9,3),    MMR00190
     +          BSAT(6,2), BSBT(6,2)                                    MMR00200
      DATA ALFA/1.42E-2,3.34E-2,.197,.404,1.11,1.76,2.36,2.72,2.89,     MMR00210
     D         1.14E-2,2.82E-2,.180,.387,1.18,1.89,2.46,2.78,2.98,     MMR00220
     D         7.52E-3,2.20E-2,.167,.368,1.24,2.05,2.58,2.84,2.89,     MMR00230
     M         1.59E-2,3.94E-2,.275,.579,1.35,1.80,2.07,2.22,2.00,     MMR00240
     M         1.36E-2,3.68E-2,.268,.579,1.42,1.88,2.10,2.24,2.00,     MMR00250
     M         1.01E-2,3.57E-2,.269,.572,1.46,1.96,2.13,2.24,2.01,     MMR00260
     T         1.51E-2,4.20E-2,.376,.619,.871,1.05,1.02,.972,.857,     MMR00270
     T         1.69E-2,4.66E-2,.372,.629,.909,1.04,.997,.976,.856,     MMR00280
     T         1.90E-2,5.89E-2,.360,.610,.936,1.02,.992,.997,.856/,    MMR00290
     +    BETA/.932,.954,1.016,1.027,.943,.846,.766,.704,.613,         MMR00300
     D         .968,1.003,1.053,1.053,.941,.829,.749,.692,.612,        MMR00310
     D         1.024,1.106,1.107,1.082,.936,.807,.725,.679,.613,       MMR00320
     M         1.094,1.088,1.002,.904,.753,.686,.645,.618,.611,        MMR00330
     M         1.150,1.118,1.007,.905,.742,.677,.641,.614,.610,        MMR00340
     M         1.260,1.160,.999,.900,.732,.667,.635,.612,.610,         MMR00350
     T         1.087,1.027,.784,.714,.657,.610,.608,.612,.614,         MMR00360
     T         1.076,1.010,.783,.709,.650,.610,.613,.611,.615,         MMR00370
     T         1.079,.966,.782,.708,.642,.615,.615,.607,.615/          MMR00380
      DATA TK/263.16,273.16,293.16/,                                    MMR00390
     +     F/10.,15.,35.,50.,95.,140.,225.,310.,1000./                 MMR00400
      DATA BSAT/-.8824881E+01,-.1029998E-01,+.2451205E-04,              MMR00410
     +       -.2462900E-07,+.6507628E-10,-.1856080E-12,                 MMR00420
     +       -.2127020E+02,+.6906017E+00,-.1924260E-01,                 MMR00430
     +       +.3035233E-03,-.2545323E-05,+.8673581E-08/,                MMR00440
     + BSBT/+.7901887E+00,-.1900189E-02,+.6341350E-05,                  MMR00450
     +       +.3186429E-08,-.5933950E-10,+.9056715E-13,                 MMR00460
     +       +.1361993E+01,+.3628100E-01,-.2461284E-02,                 MMR00470
     +       +.4805257E-04,-.3988064E-06,+.1193458E-08/                 MMR00480
C                                                                       MMR00490
      TR=T                                                              MMR00500
      GRAIN=0.                                                          MMR00510
      BSRAIN=0.                                                         MMR00520
      IF(FREQ.LT.10.) GO TO 200                                         MMR00530
      FR=FREQ                                                           MMR00540
C                                                                       MMR00550
      ITYPE=IFIX(RTYPE+0.1)                                             MMR00560
      DO 10 I=1,3                                                       MMR00570
        DO 10 J=1,9                                                     MMR00580
          A(J,I)=ALFA(J,I,ITYPE)                                        MMR00590
          B(J,I)=BETA(J,I,ITYPE)                                        MMR00600
   10 CONTINUE                                                          MMR00610
C                                                                       MMR00620
      CALL INTRP(A,B,TK,F,TR,FR,AA,BB,J)                                MMR00630
      GRAIN=(AA*RAINRT**BB)/4.343                                       MMR00640
C                                                                       MMR00650
CC    CALCULATIONS FOR RAIN BACKSCATTER.                                MMR00660
C                                                                       MMR00670
      AA=0.                                                             MMR00680
      BB=0.                                                             MMR00690
      IA=2                                                              MMR00700
      IB=2
```

```
      IF(FREQ.GT.87.) IA=1                                              MMR00710
      IF(FREQ.GT.82.) IB=1                                              MMR00720
      DO 100 K=1,6                                                      MMR00730
      AA=AA+BSAT(K,IA)*(FREQ**(K-1))                                    MMR00740
      BB=BB+BSBT(K,IB)*(FREQ**(K-1))                                    MMR00750
100   CONTINUE                                                          MMR00760
      AA=EXP(AA)                                                        MMR00770
      BSRAIN=AA*RAINRT**BB                                              MMR00780
200   RETURN                                                            MMR00790
      END                                                               MMR00800
```

```
      SUBROUTINE MMSNO(SNRT,TK,FQ,SNEX,SNBS)                                MMS00010
C                                                                           MMS00020
C-----------------------------------------------------------------------    MMS00030
C     PURPOSE: TO COMPUTE SNOW EXTINCTION AND BACKSCATTER X-SECTION.        MMS00040
C-----------------------------------------------------------------------    MMS00050
C     INPUTS: SNOWFALL RATE(MM/HR), TEMPERATURE(KELVIN), FREQUENCY(GHZ)     MMS00060
C                                                                           MMS00070
C     OUTPUT: SNOW EXTINCTION (1/KM), BACKSCATTER(M**2/M**3).               MMS00080
C                                                                           MMS00090
C     CALLED FROM MMMW.                                                     MMS00100
C                                                                           MMS00110
C     LOCAL VARIABLES:                                                      MMS00120
C        FR,TS          LOCAL FREQUENCY,TEMPERATURE                       MMS00130
C        ITYPE          INTEGER SNOW TYPE                                 MMS00140
C        XLMDA          WAVELENGTH(MM)                                    MMS00150
C        FF0,1          FREQUENCY FITTING FACTORS                         MMS00160
C        TF0,1          TEMPERATURE FITTING FACTORS                       MMS00170
C        AA0,1          INTERMEDIATE A VALUES                             MMS00180
C        BB0,1          INTERMEDIATE B VALUES                             MMS00190
C        AA,BB          TERMS IN EXT(SNOW)=AA*SNOWRATE**BB                MMS00200
C        AABSLT           TERM IN BSCAT(SNOW)=AA*SNOWRATE**1.8            MMS00210
C                                                                           MMS00220
C-----------------------------------------------------------------------    MMS00230
      DIMENSION A(9,3),B(9,3),F(9),T(3),BSAT(6,2),BSBT(6,2),SFCT(3)         MMS00240
      DATA A/1.30E-3,2.75E-3,1.25E-2,2.50E-2,8.00E-2,1.65E-1,               MMS00250
     1       3.60E-1,6.91E-1,1.68E+0,                                       MMS00260
     2       2.07E-2,4.34E-2,1.60E-1,2.00E-1,3.10E-1,4.00E-1,               MMS00270
     3       5.81E-1,6.50E-1,1.11E+0,                                       MMS00280
     4       6.22E-2,9.67E-2,2.35E-1,3.41E-1,6.10E-1,8.52E-1,               MMS00290
     5       7.83E-1,7.37E-1,5.76E-1/                                       MMS00300
      DATA B/1.3,1.46,1.6,1.54,1.26,1.1,.89,.79,.6,                         MMS00310
     1       1.3,1.2,.95,.80,.75,.67,.65,.64,.60,                          MMS00320
     2       1.3,1.2,.95,.80,.75,.67,.65,.64,.60/                          MMS00330
      DATA F/10.,15.,35.,50.,95.,140.,225.,312.,1000./                      MMS00340
      DATA T/271.,273.,275./,SFCT/1.,3.,4./                                 MMS00350
      DATA BSAT/-.8824881E+01,-.1029998E-01,+.2451205E-04,                  MMS00360
     +       -.2462900E-07,+.6507628E-10,-.1856080E-12,                     MMS00370
     +       -.2127020E+02,+.6906017E+00,-.1924260E-01,                     MMS00380
     +       +.3035233E-03,-.2545323E-05,+.8673581E-08/,                    MMS00390
     + BSBT/+.7901887E+00,-.1900189E-02,+.6341350E-05,                      MMS00400
     +       +.3186429E-08,-.5933950E-10,+.9056715E-13,                     MMS00410
     +       +.1361993E+01,+.3628100E-01,-.2461284E-02,                     MMS00420
     +       +.4805257E-04,-.3988064E-06,+.1193458E-08/                     MMS00430
C                                                                           MMS00440
      TS=TK                                                                 MMS00450
      FS=FQ                                                                 MMS00460
      ICE=1                                                                 MMS00470
      SNEX=0.                                                               MMS00480
      SNBS=0.                                                               MMS00490
      IF(FQ.LT.10.) GO TO 200                                               MMS00500
      XLMDA=299.79/FQ                                                       MMS00510
      CALL INTRP(A,B,T,F,TS,FS,AA,BB,J)                                     MMS00520
      SNEX=AA*SNRT**BB                                                      MMS00530
C                                                                           MMS00540
C     CALCULATIONS FOR SNOW BACKSCATTER.                                    MMS00550
C                                                                           MMS00560
      AA=0.                                                                 MMS00570
      BB=0.                                                                 MMS00580
      IA=2                                                                  MMS00590
      IB=2                                                                  MMS00600
      IF(FQ.GT.87.) IA=1                                                    MMS00610
      IF(FQ.GT.82.) IB=1                                                    MMS00620
      DO 100 K=1,6                                                          MMS00630
      AA=AA+BSAT(K,IA)*(FQ**(K-1))                                          MMS00640
      BB=BB+BSBT(K,IB)*(FQ**(K-1))                                          MMS00650
100   CONTINUE                                                              MMS00660
      AA=EXP(AA)                                                            MMS00670
      BB=1.2*BB                                                             MMS00680
      F1=(FQ-10.)/85.                                                       MMS00690
      FCT=.367+F1*.633                                                      MMS00700
```

```
      IF(FQ.GT.95.) FCT=1.                                              MMS00710
      AA=FCT*AA                                                         MMS00720
  110 IF(TK.GE.275.) J=4                                                MMS00730
      SNBS=SFCT(J-1)*AA*SNRT**BB                                        MMS00740
C                                                                       MMS00750
  200 RETURN                                                            MMS00760
C                                                                       MMS00770
      END                                                               MMS00780
```

```
      SUBROUTINE MMWFG(FD,T,FREQ,GFOG,BSFOG)                            MMF00010
C                                                                       MMF00020
C*****************************************************                  MMF00030
C                                                                       MMF00040
C   CALCULATES ABSORPTION DUE TO WATER  FOGS/CLOUDS,                    MMF00050
C    AND BACKSCATTER CROSS SECTION IN M**2/M**3.                        MMF00060
C                                                                       MMF00070
C   INPUTS ARE: FOG DENSITY(GM/M**3), TEMPERTURE(KELVIN),               MMF00080
C     FREQUENCY(GHZ).                                                   MMF00090
C   OUTPUTS ARE: FOG ABSORPTION(/KM), BACKSCATTER X-SECTION(M**2 M**3).  MMF00100
C                                                                       MMF00110
C   MMFOG IS CALLED FROM MMWMOD  CALLS MMIDX SUBROUTINE.                MMF00120
C                                                                       MMF00130
C   LOCAL VARIABLES:                                                    MMF00140
C      XLMDA          WAVELENGTH(MM)                                    MMF00150
C*****************************************************                  MMF00160
C                                                                       MMF00170
      ICE=0                                                             MMF00180
      IF(T.LT.243.) ICE=1                                               MMF00190
C COMPUTE FOG EXTINCTION                                                MMF00200
      XLMDA=10./(FREQ/29.98)                                            MMF00210
      CALL MMIDX(XLMDA,T,ICE,H2OAB,H2OK2)                               MMF00220
      GFOG=18.8498*H2OAB*FD/XLMDA                                       MMF00230
      BSFOG=1.162E-06*H2OK2*FD**1.75/(XLMDA**4)                         MMF00240
      RETURN                                                            MMF00250
C                                                                       MMF00260
      END                                                               MMF00270
```

```
      SUBROUTINE MMWGS(T,P,AH,FRQ,GAS)                                          MMG00010
C                                                                               MMG00020
C*************************************************************                  MMG00030
C                                                                               MMG00040
C   SUBROUTINE COMPUTES GASEOUS ABSORPTION FROM 0 TO 1000 GHZ FOR               MMG00050
C     H2O VAPOR AND O2.                                                         MMG00060
C                                                                               MMG00070
C   INPUTS INCLUDE: TEMPEATURE(KELVIN), PRESSURE(TORR), ABSOLUTE HUMIDITYMMG00080
C     IN GM/M**3, FREQUENCY(GIGAHERTZ).                                         MMG00090
C   OUTPUTS ARE:  GAS ABSORPTION (1/KM)                                         MMG00100
C                                                                               MMG00110
C   MMWGS IS CALLED FROM NMMW, CALLS (1)MMOXY(OXYGEN ABSORPTION),               MMG00120
C     (2) MMH2O(H2O VAPOR ABSORPTION).                                          MMG00130
C                                                                               MMG00140
C   LOCAL VARIABLES:                                                            MMG00150
C     DATA02(L,J): OXYGEN LINE DATA, L=LINE NUMBER, J=TYPE:                     MMG00160
C                  J=1  : LINE FREQUENCY(GHZ)                                   MMG00170
C                    2  : LINE STRENGTH AT 300K                                 MMG00180
C                    3  : LINE WIDTH AT 300K (GHZ/TORR)                         MMG00190
C                    4  : INTERFERENCE PARAMETER AT 300K                        MMG00200
C                    5  : INTERFERENCE TEMPERATURE CORRECTION                   MMG00210
C                    6  : LINE QUANTUM PARAMETER                                MMG00220
C     DATH2O(L,J) : H2O LINE DATA, L=LINE NUMBER, J=TYPE:                       MMG00230
C                    1  : WAVENUMBER(1/CM)                                      MMG00240
C                    2  : STRENGTH                                              MMG00250
C                    3  : WIDTH(1/CM/TORR)                                      MMG00260
C                    4  : GROUND STATE ENERGY                                   MMG00270
C     GH2O        : H2O VAPOR ABSORPTION (1/KM)                                 MMG00280
C     GO2         : O2  ABSORPTION (1/KM)                                       MMG00290
C     PH2O        : H2O VAPOR PRESSURE(TORR)                                    MMG00300
C     WVNMB       : WAVENUMBER(1/CM)                                            MMG00310
C*************************************************************                  MMG00320
C                                                                               MMG00330
      DIMENSION DATA02(42,6), DATH2O(37,4)                                      MMG00340
      DIMENSION D1O2(42,3),D2O2(42,3),D1H2O(37,2),D2H2O(37,2)                   MMG00350
      EQUIVALENCE (DATA02(1,1),D1O2(1,1)),(DATA02(1,4),D2O2(1,1))               MMG00360
      EQUIVALENCE (DATH2O(1,1),D1H2O(1,1)),(DATH2O(1,3),D2H2O(1,1))             MMG00370
      COMMON /IOUN1T/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUMMG00380
C*****************************************************************           MMG00390
      DATA D1O2/                                                                MMG00400
     1 49.451,49.961,50.473,50.987,51.503,52.021,52.542,53.066,                 MMG00410
     2 53.595,54.129,54.671,55.221,55.783,56.264,56.363,56.968,                 MMG00420
     3 57.612,58.323,58.446,59.164,59.590,60.306,60.434,61.150,                 MMG00430
     4 61.800,62.411,62.486,62.998,63.568,64.127,64.678,65.224,                 MMG00440
     5 65.764,66.302,66.836,67.369,67.900,68.430,68.960,69.488,                 MMG00450
     6 70.016,118.75,                                                           MMG00460
     1 7.E-5,2.2E-4,6.E-4,1.56E-3,3.86E-3,8.99E-3,1.971E-2,.04072,              MMG00470
     2 .07919,.1448,.2489,.4012,.6056,.3487,.8539,1.1204,1.3595,                MMG00480
     3 1.515,.9251,1.5263,1.341,1.3487,1.5626,1.5899,1.4588,1.2272,             MMG00490
     4 .9634,.954,.6898,.4656,.2942,.1744,.0971,.0508,.025,.0116,               MMG00500
     5 5.08E-3,2.1E-3,8.2E-4,3.E-4,1.E-4,.5973,                                 MMG00510
     1 1.260,1.310,1.330,1.360,1.380,1.410,1.440,1.460,1.490,                   MMG00520
     2 1.510,1.540,1.570,1.601,2.212,1.635,1.672,1.714,1.762,                   MMG00530
     3 1.964,1.819,1.859,1.890,1.789,1.736,1.694,1.658,1.990,                   MMG00540
     4 1.627,1.598,1.568,1.540,1.510,1.490,1.460,1.440,1.410,                   MMG00550
     5 1.380,1.360,1.330,1.310,1.280,2.140/                                     MMG00560
      DATA D2O2/                                                                MMG00570
     1 0.000,0.000,0.000,1.040,0.802,0.897,0.825,0.780,0.764,                   MMG00580
     2 0.666,0.651,0.550,0.481,0.931,0.371,0.254,0.100,-.087,                   MMG00590
     3 0.729,-.318,0.433,-.543,0.179,-.028,-.183,-.324,-.615,                   MMG00600
     4 -.419,-.537,-.591,-.693,-.703,-.796,-.808,-.849,-.916,                   MMG00610
     5 -.822,-1.05,0.000,0.000,0.000,-.054,                                     MMG00620
     1 1.00,1.00,1.00,1.38,2.04,1.69,1.91,1.88,1.90,2.01,1.95,                  MMG00630
     2 2.11,2.13,0.89,2.36,2.66,4.20,-5.8,0.79,0.11,0.50,0.69,                  MMG00640
     3 -.99,7.60,3.04,2.34,0.85,2.24,2.02,2.04,1.89,1.95,1.85,                  MMG00650
     4 1.83,1.86,1.66,1.99,1.36,1.00,1.00,1.00,0.89,                            MMG00660
     1 41.,39.,37.,35.,33.,31.,29.,27.,25.,23.,21.,19.,17.,1.,15.,13.,          MMG00670
     2 11.,9.,3.,7.,5.,5.,7.,9.,11.,13.,3.,15.,17.,19.,21.,23.,25.,             MMG00680
     3 27.,29.,31.,33.,35.,37.,39.,41.,1./                                      MMG00690
C*****************************************************************           MMG00700
```

```
      DATA D1H2O/                                                       MMG00710
     1  00.742,06.115,06.790,10.715,10.846,12.682,14.778,14.944,        MMG00720
     2  15.707,15.834,16.294,18.270,18.577,20.704,21.960,24.860,        MMG00730
     3  25.085,30.560,32.366,32.954,36.604,37.137,37.910,38.638,        MMG00740
     4  38.791,39.112,40.282,40.520,42.639,43.243,43.631,44.100,        MMG00750
     5  46.750,47.053,48.059,49.765,49.820,                             MMG00760
     1  .436E-24,.775E-22,.186E-24,.250E-23,.906E-22,.827E-21,          MMG00770
     2  .145E-22,.863E-21,.270E-22,.108E-21,.219E-22,.985E-22,          MMG00780
     3  .526E-19,.565E-21,.531E-22,.664E-22,.347E-19,.143E-20,          MMG00790
     4  .160E-20,.252E-19,.164E-18,.502E-19,.333E-21,.242E-20,          MMG00800
     5  .179E-18,.197E-21,.558E-19,.155E-21,.707E-21,.687E-21,          MMG00810
     6  .511E-22,.568E-20,.302E-21,.142E-18,.930E-21,.399E-22,          MMG00820
     7  .478E-22/                                                       MMG00830
      DATA D2H2O/                                                       MMG00840
     1  .081,.094,.095,.063,.087,.091,.050,.083,.061,.071,.075,         MMG00850
     2  .111,.107,.072,.111,.103,.102,.084,.083,.101,.097,.099,         MMG00860
     3  .094,.073,.094,.063,.093,.098,.066,.074,.060,.081,.084,         MMG00870
     4  .091,.078,.096,.097,                                            MMG00880
     1   446.512, 136.164, 134.800,1284.921, 315.780, 212.156,          MMG00890
     2 1045.069, 285.419, 742.079, 488.136, 586.482,  23.750,          MMG00900
     3   23.794, 488.110,1618.550,  69.920,  70.091, 285.219,          MMG00910
     4  383.843,  37.137, 136.761,   0.000, 172.880, 610.345,          MMG00920
     5  173.366, 888.641, 275.498,1731.890, 888.607, 842.361,          MMG00930
     6 1079.088, 508.814, 398.392, 399.459, 601.553, 100.391,          MMG00940
     7 1693.650/                                                       MMG00950
C*********************************************************************MMG00960
C                                                                       MMG00970
C   COMPUTE WATER VAPOR PRESSURE, FREQUENCY IN WAVENUMBERS              MMG00980
      PH2O=AH*T*3.462977E-3                                             MMG00990
      IF(AH.LT.0.) PH2O=-PSAT(T)*AH/100.                                MMG01000
      WVNMB=FRQ/29.98                                                   MMG01010
C                                                                       MMG01020
C   COMPUTE H2O ABSORPTION                                              MMG01030
      CALL MMH2O(WVNMB,T,P,PH2O,DATH2O,GH2O)                            MMG01040
C                                                                       MMG01050
C   COMPUTE O2 ABSORPTION                                               MMG01060
      IF(FRQ.LT.140.) CALL MMOXY(FRQ,T,P,PH2O,DATA02,GO2)               MMG01070
C                                                                       MMG01080
C   SUM ABSORPTION                                                      MMG01090
      GAS=GH2O+GO2                                                      MMG01100
C                                                                       MMG01110
C*********************************************************************MMG01120
      RETURN                                                            MMG01130
      END                                                               MMG01140
```

```
      FUNCTION PFR(T)                                                   PFR00010
C       COMPUTE H2O PARTITIAN FUNCTION CORRECTIONS                      PFR00020
C                                                                       PFR00030
      DIMENSION VIB(3)                                                  PFR00040
      DATA VIB/3693.9,1614.5,3801.8/                                    PFR00050
C                                                                       PFR00060
      QJ=296./T                                                         PFR00070
      QJ=QJ*SQRT(QJ)                                                    PFR00080
      T1=-1.43879/296.                                                  PFR00090
      T2=-1.43879/T                                                     PFR00100
      T1S=1.                                                            PFR00110
      T2S=1.                                                            PFR00120
C                                                                       PFR00130
      DO 10 J=1,3                                                       PFR00140
         V=VIB(J)                                                       PFR00150
         T11=1.-EXP(T1*V)                                               PFR00160
         T22=1.-EXP(T2*V)                                               PFR00170
         T1S=T1S*T11                                                    PFR00180
10       T2S=T2S*T22                                                    PFR00190
      PFR=QJ*T2S/T1S                                                    PFR00200
      RETURN                                                            PFR00210
C                                                                       PFR00220
      END                                                               PFR00230
```

```
      FUNCTION PSAT(T)                                                  PSAT0010
C                                                                       PSAT0020
      DATA C1,C2,C3,C4,C5,C6,C7/-7.90298,5.02808,-1.3816E-7,11.344,     PSAT0030
     +     8.1328E-3,-3.49149,3.005715/                                 PSAT0040
      DATA D1,D2,D3,D4/-9.09718,-3.56654,.876793,.785835/               PSAT0050
      DATA TS,TO/373.16,273.16/,CONV/.7500646/                          PSAT0060
      IF(T.LE.TO) GO TO 100                                             PSAT0070
       TR=TS/T                                                          PSAT0080
      TRI=T/TS                                                          PSAT0090
      EW=C1*(TR-1.)+C2*ALOG10(TR)+C3*(10.**(C4*(1.-TRI))-1.)+           PSAT0100
     +    C5*(10.**(C6*(TR-1.))-1.)+C7                                 PSAT0110
      GO TO 200                                                         PSAT0120
100   TR=TO/T                                                           PSAT0130
      TRI=T/TO                                                          PSAT0140
      EW=D1*(TR-1.)+D2*ALOG10(TR)+D3*(1.-TRI)+D4                        PSAT0150
200   PSAT=(10.**EW)*CONV                                               PSAT0160
      RETURN                                                            PSAT0170
      END                                                               PSAT0180
```

```
      FUNCTION SUPK(A,B,C)                                                  SUPK0010
C        COMPUTES THE SUPER KINETIC LINE PROFILE FACTOR.                    SUPK0020
C                                                                           SUPK0030
      PI=3.14159265                                                         SUPK0040
      XNORM=.998776                                                         SUPK0050
      VM=10.*C                                                              SUPK0060
      X=ABS((B*B-A*A)/(2.*C*A))                                             SUPK0070
      IF(ABS(B-A).GT.VM) GO TO 10                                           SUPK0080
C                                                                           SUPK0090
      SUPK=XNORM/(PI*C)/(X*X+1.)                                            SUPK0100
      RETURN                                                                SUPK0110
C                                                                           SUPK0120
10    CXI=10.*(B+10.*C/2.)/(B+10.*C)                                        SUPK0130
      CXI=((CXI**1.88)+1.)/((CXI*CXI)+1.)                                   SUPK0140
      SUPK=XNORM*CXI/(PI*C)/((X**1.88)+1.)                                  SUPK0150
      RETURN                                                                SUPK0160
      END                                                                   SUPK0170
```

```
      SUBROUTINE CLTRAN(CTRANS,WAVE,IRUN,IERR)                          CLT00010
      DIMENSION TAU(20)                                                 CLT00020
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)               CLT00030
      COMMON /CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,
     +       WNDVEL,WNDDIR,IPASCT
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT      CLT00040
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)        CLT00050
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)            CLT00060
      COMMON /CYL/XC,YC,RADIUS                                          CLT00070
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCLT00080
      COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)         CLT00090
     +,RADICL(10)                                                       CLT00100
      COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)           CLT00110
C-----------------------------------------------------------------------CLT00120
C** DATA IS READ FROM INPUT RECORDS AND THEN TRANSFERRED TO CLTRAN       CLT00130
C** BY SUBROUTINE CLREAD. INPUT TO CLTRAN IS CARD ORDER-INDEPENDENT,     CLT00140
C** WITH A FOUR-LETTER IDENTIFIER IN COLUMNS 1-4 OF EACH INPUT RECORD.   CLT00150
C** THE ONLY EXCEPTION TO THIS ORDER-INDEPENDENCE IS THE GO SENTINEL     CLT00160
C** CARD, WHICH MUST BE THE LAST RECORD READ. ALL CARDS ARE READ IN      CLT00170
C** UNDER THE FORMAT (A4,1X,5(E10.5,1X)). THE IDENTIFICATION AND MEANINGCLT00180
C** OF EACH INPUT RECORD ARE AS FOLLOWS :                               CLT00190
C-----------------------------------------------------------------------CLT00200
C     CARD IDENTIFIER : SEEK                                            CLT00210
C     VARIABLES READ : XS,YS,ZS                                         CLT00220
C       XS,YS,ZS = POSITION COORDINATES OF SEEKER (OR RECEIVER) (KM)    CLT00230
C-----------------------------------------------------------------------CLT00240
C     CARD IDENTIFIER : TARG                                            CLT00250
C     VARIABLES READ : XT,YT,ZT                                         CLT00260
C       XT,YT,ZT = POSITION COORDINATES OF TARGET  (KM)                 CLT00270
C-----------------------------------------------------------------------CLT00280
C                                                                       CLT00290
C** THE FOLLOWING FOUR CARDS REPRESENT THE STRATIFORM CLOUDS TREATED     CLT00300
C** BY CLTRAN. THEY ARE TREATED AS INFINITE LAYERS IN A PLANE-PARALLEL   CLT00310
C** ATMOSPHERE.                                                         CLT00320
C                                                                       CLT00330
C-----------------------------------------------------------------------CLT00340
C     CARD IDENTIFIER : CLST  (STRATUS CLOUD TYPE)                      CLT00350
C     VARIABLES READ : ZLBASE,ZLTHIC                                    CLT00360
C       ZLBASE = HEIGHT OF CLOUD BASE  (KM)                             CLT00370
C       ZLTHIC = VERTICAL THICKNESS OF CLOUD LAYER  (KM)                CLT00380
C-----------------------------------------------------------------------CLT00390
C     CARD IDENTIFIER : CLAS  (ALTOSTRATUS CLOUD TYPE)                  CLT00400
C     VARIABLES READ : ZLBASE,ZLTHIC                                    CLT00410
C-----------------------------------------------------------------------CLT00420
C     CARD IDENTIFIER : CLNS  (NIMBOSTRATUS CLOUD TYPE)                 CLT00430
C     VARIABLES READ : ZLBASE,ZLTHIC                                    CLT00440
C-----------------------------------------------------------------------CLT00450
C     CARD IDENTIFIER : CLSC  (STRATOCUMULUS CLOUD TYPE)                CLT00460
C     VARIABLES READ : ZLBASE,ZLTHIC                                    CLT00470
C-----------------------------------------------------------------------CLT00480
C                                                                       CLT00490
C** THE NEXT TWO CARDS REPRESENT THE CUMULUS CLOUD TYPES ADDRESSED       CLT00500
C** BY CLTRAN. CLOUDS OF THIS KIND ARE MODELLED AS CYLINDERS WHICH       CLT00510
C** HAVE VERTICAL SYMMETRY AXES. THESE TYPES ARE REPRESENTED BY          CLT00520
C** THE FOLLOWING RECORDS :                                             CLT00530
C                                                                       CLT00540
C-----------------------------------------------------------------------CLT00550
C     CARD IDENTIFIER : CLCH  (CUMULUS HUMILIS CLOUD TYPE)              CLT00560
C     VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD               CLT00570
C       ZCBASE = HEIGHT OF CLOUD CYLINDER'S LOWER BASE (KM)             CLT00580
C       ZCTHIC = VERTICAL THICKNESS OF CLOUD CYLINDER  (KM)             CLT00590
C       RADICL = RADIUS OF CLOUD CYLINDER  (KM)                         CLT00600
C       XCLOUD = X-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM)   CLT00610
C       YCLOUD = Y-COORDINATE OF VERTICAL AXIS OF CLOUD CYLINDER (KM)   CLT00620
C-----------------------------------------------------------------------CLT00630
C     CARD IDENTIFIER : CLCC  (CUMULUS CONGESTUS CLOUD TYPE)            CLT00640
C     VARIABLES READ : ZCBASE,ZCTHIC,RADICL,XCLOUD,YCLOUD               CLT00650
C-----------------------------------------------------------------------CLT00660
C                                                                       CLT00670
C** THE FOLLOWING CARD MUST BE THE LAST RECORD READ :                   CLT00680
```

```
C                                                                               CLT00690
C-------------------------------------------------------------------------------CLT00700
C     CARD IDENTIFIER : GO                                                      CLT00710
C     VARIABLES READ : NONE                                                     CLT00720
C-------------------------------------------------------------------------------CLT00730
      IF(IRUN.GT.1) GO TO 5                                                     CLT00740
      NLAYER=0                                                                  CLT00750
      NCCLDS=0                                                                  CLT00760
    5 CONTINUE                                                                  CLT00770
      IWV=0                                                                     CLT00780
      ZBMIN=999.
      FOGPRB=0.
      IF(WAVE.GT.0.20.AND.WAVE.LT.2.00) IWV=1                                   CLT00790
      IF(WAVE.GT.3.00.AND.WAVE.LT.5.00) IWV=2                                   CLT00800
      IF(WAVE.GT. 8.0.AND.WAVE.LT.12.0) IWV=3                                   CLT00810
      IF(IWV.NE.0) GO TO 50                                                     CLT00820
      IERR=1                                                                    CLT00830
      WRITE(IOOUT,22)                                                           CLT00840
   22 FORMAT(1H0,20X,94H***CLTRAN ERROR*** INPUT WAVELENGTH DOES NOT LIECLT00850
     + WITHIN ALLOWABLE LIMITS, EXECUTION TERMINATED /)                         CLT00860
      GO TO 900                                                                 CLT00870
   50 CALL CLREAD(NLAYER,NCCLDS,IERR)                                           CLT00880
      IF(IERR.EQ.1) GO TO 900                                                   CLT00890
      ISLTUP=0                                                                  CLT00900
      ISLTDN=0                                                                  CLT00910
      IHORIZ=0                                                                  CLT00920
      IVERT=0                                                                   CLT00930
C                                                                               CLT00940
C**** DETERMINE SENSE OF L-O-S SLOPE FROM SEEKER'S POINT OF VIEW                CLT00950
C                                                                               CLT00960
      TESVER=(XS-XT)**2+(YS-YT)**2                                              CLT00970
      IF(TESVER.EQ.0.0)IVERT=1                                                  CLT00980
      IF(IVERT.EQ.1)GO TO 300                                                   CLT00990
      IF(ZS-ZT)200,210,220                                                      CLT01000
  200 ISLTUP=1                                                                  CLT01010
      GO TO 300                                                                 CLT01020
  210 IHORIZ=1                                                                  CLT01030
      GO TO 300                                                                 CLT01040
  220 ISLTDN=1                                                                  CLT01050
  300 CONTINUE                                                                  CLT01060
      IF(IHORIZ.EQ.1)GO TO 310                                                  CLT01070
C                                                                               CLT01080
C**** COMPUTE L-O-S SLOPES IN X-Z AND Y-Z VERTICAL PLANES                       CLT01090
C                                                                               CLT01100
      XIX=(XS-XT)/(ZS-ZT)                                                       CLT01110
      XIY=(YS-YT)/(ZS-ZT)                                                       CLT01120
  310 CONTINUE                                                                  CLT01130
C                                                                               CLT01140
C**** STRATIFORM CLOUD BLOCK                                                    CLT01150
C                                                                               CLT01160
      IF(NLAYER.EQ.0)GO TO 500                                                  CLT01170
C                                                                               CLT01180
C**** UTILIZE DEFAULT BASE OR THICKNESS VALUES IF NECESSARY                     CLT01190
C                                                                               CLT01200
      CALL DEFSET(1,NLAYER)                                                     CLT01210
      DO 400 N=1,NLAYER                                                         CLT01220
      NLINT(N)=0                                                                CLT01230
      Z0(N)=ZLBASE(N)                                                           CLT01240
      Z1(N)=Z0(N)+ZLTHIC(N)                                                     CLT01250
      ZBAS=Z0(N)                                                                CLT01260
      ZTOP=Z1(N)                                                                CLT01270
      ICL=ILCTYP(N)                                                             CLT01280
      IF(ZBMIN.LT.ZLBASE(N)) GO TO 320
      ZBMIN=ZLBASE(N)
      FOGPRB=FLOAT(ICL)
  320 CONTINUE
C                                                                               CLT01290
C**** DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CLOUD LAYER (IF                 CLT01300
C**** THERE ARE ANY): (X1(N),Y1(N),Z1(N))= UPPER INTERSECTION POINT,             CLT01310
C**** (X0(N),Y0(N),Z0(N))= LOWER INTERSECTION POINT                             CLT01320
```

```
C                                                                       CLT01330
      CALL LAYRXY(XIX,XIY,N)                                            CLT01340
      IF(TESALL.LE.0.0)GO TO 350                                        CLT01350
      NLINT(N)=N                                                        CLT01360
C                                                                       CLT01370
C**** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH           CLT01380
C                                                                       CLT01390
      CALL CLEXTN(TAUN,N)                                               CLT01400
      TAU(N)=TAUN                                                       CLT01410
      ATTL(N)=SQRT((X1(N)-X0(N))**2+(Y1(N)-Y0(N))**2+(Z1(N)-Z0(N))**2)  CLT01420
      GO TO 400                                                         CLT01430
C                                                                       CLT01440
C**** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH                 CLT01450
C**** IS NOW SET TO ZERO                                                CLT01460
C                                                                       CLT01470
  350 TAU(N)=0.0                                                        CLT01480
      ATTL(N)=0.0                                                       CLT01490
  400 CONTINUE                                                          CLT01500
C                                                                       CLT01510
C**** END STRATIFORM CLOUD BLOCK                                        CLT01520
C                                                                       CLT01530
C**** CUMULIFORM CLOUD BLOCK                                            CLT01540
C                                                                       CLT01550
  500 IF(NCCLDS.EQ.0)GO TO 805                                          CLT01560
C                                                                       CLT01570
C**** UTILIZE DEFAULT BASE, THICKNESS, OR RADIUS IF NECESSARY           CLT01580
C                                                                       CLT01590
      CALL DEFSET(2,NCCLDS)                                             CLT01600
      DO 800 N=1,NCCLDS                                                 CLT01610
      NN=NLAYER+N                                                       CLT01620
      NCINT(N)=0                                                        CLT01630
      Z0(NN)=ZCBASE(N)                                                  CLT01640
      Z1(NN)=Z0(NN)+ZCTHIC(N)                                           CLT01650
      ICL=ICCTYP(N)                                                     CLT01660
      IF(ZBMIN.LT.ZCBASE(N)) GO TO 520
      ZBMIN=ZCBASE(N)
      FOGPRB=FLOAT(ICL)
  520 CONTINUE
      RADIUS=RADICL(N)                                                  CLT01670
      XC=XCLOUD(N)                                                      CLT01680
      YC=YCLOUD(N)                                                      CLT01690
      ZBAS=Z0(NN)                                                       CLT01700
      ZTOP=Z1(NN)                                                       CLT01710
      TESARG=0.0                                                        CLT01720
      TESVT=0.0                                                         CLT01730
C                                                                       CLT01740
C**** DETERMINE X,Y,Z INTERSECTIONS OF L-O-S AND CUMULUS CLOUD CYLINDER CLT01750
C**** (IF THERE ARE ANY): (X1(N),Y1(N),Z1(N))= UPPER INTERSECTION POINT,CLT01760
C**** (X0(N),Y0(N),Z0(N))= LOWER INTERSECTION POINT                     CLT01770
C                                                                       CLT01780
      CALL CYLXY(XIX,XIY,NN)                                            CLT01790
      IF(TESALL.LE.0.0.OR.TESARG.LT.0.0.OR.TESVT.LT.0.0)GO TO 650       CLT01800
      NCINT(N)=N                                                        CLT01810
C                                                                       CLT01820
C**** IF THERE ARE ANY INTERSECTIONS, DETERMINE OPTICAL DEPTH           CLT01830
C                                                                       CLT01840
      CALL CLEXTN(TAUN,NN)                                              CLT01850
      TAU(NN)=TAUN                                                      CLT01860
      ATTL(NN)=SQRT((X1(NN)-X0(NN))**2+(Y1(NN)-Y0(NN))**2+              CLT01870
     +(Z1(NN)-Z0(NN))**2)                                               CLT01880
      GO TO 800                                                         CLT01890
C                                                                       CLT01900
C**** IF NO INTERSECTIONS WERE FOUND, THE OPTICAL DEPTH                 CLT01910
C**** IS NOW SET TO ZERO                                                CLT01920
C                                                                       CLT01930
  650 TAU(NN)=0.0                                                       CLT01940
      ATTL(NN)=0.0                                                      CLT01950
  800 CONTINUE                                                          CLT01960
C                                                                       CLT01970
C**** END CUMULIFORM CLOUD BLOCK                                        CLT01980
```

```
C                                                                       CLT01990
  805 CONTINUE                                                          CLT02000
C**** DETERMINE CUMULATIVE CLOUD OPTICAL DEPTH AND TRANSMITTANCE        CLT02010
      TAUTOT=0.0                                                        CLT02020
      NN=NLAYER+NCCLDS                                                  CLT02030
      IF(NN.EQ.0)GO TO 820                                              CLT02040
      DO 810 N=1,NN                                                     CLT02050
  810 TAUTOT=TAUTOT+TAU(N)                                              CLT02060
  820 CTRANS=EXP(-TAUTOT)                                               CLT02070
  850 CALL LISOUT(NN,CTRANS,TAUTOT,TAU,NLAYER,IRUN)                     CLT02080
  900 RETURN                                                            CLT02090
      END                                                               CLT02100
```

```
      SUBROUTINE LAYRXY(XIX,XIY,IN)                                          LAY00010
      LOGICAL INSEEK,INTARG                                                  LAY00020
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)                    LAY00030
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT           LAY00040
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)             LAY00050
C**** CHECK LOS TO SEE IF IT INTERSECTS THE CLOUD LAYER                       LAY00060
      UP1=ISLTUP*(ZS-Z1(IN))                                                 LAY00070
      UP2=ISLTUP*(Z0(IN)-ZT)                                                 LAY00080
      DN1=ISLTDN*(ZT-Z1(IN))                                                 LAY00090
      DN2=ISLTDN*(Z0(IN)-ZS)                                                 LAY00100
      HOR12=IHORIZ*(ZS-Z1(IN))*(ZS-Z0(IN))*(-1.0)                            LAY00110
      UPVER1=IVERT*(ZT-ZS)*(ZS-Z1(IN))                                       LAY00120
      UPVER2=IVERT*(ZT-ZS)*(Z0(IN)-ZT)                                       LAY00130
      TESALL=UP1*UP2+DN1*DN2+HOR12+UPVER1*UPVER2                             LAY00140
      IF(TESALL.LT.0.0)RETURN                                                LAY00150
      IF(IHORIZ.EQ.1)RETURN                                                  LAY00160
C                                                                            LAY00170
C**** COMPUTE X,Y INTERSECTIONS OF CLOUD PLANES AND LOS                      LAY00180
C**** ALSO, CHECK FOR THE CASE WHERE EITHER THE SEEKER                       LAY00190
C**** OR TARGET IS INSIDE OF THE CLOUD                                       LAY00200
C                                                                            LAY00210
      INSEEK=ZS.LE.Z1(IN).AND.ZS.GE.Z0(IN)                                   LAY00220
      INTARG=ZT.LE.Z1(IN).AND.ZT.GE.Z0(IN)                                   LAY00230
      IF(ISLTUP.EQ.1.AND.INTARG)Z1(IN)=ZT                                    LAY00240
      IF(ISLTDN.EQ.1.AND.INTARG)Z0(IN)=ZT                                    LAY00250
      IF(ISLTUP.EQ.1.AND.INSEEK)Z0(IN)=ZS                                    LAY00260
      IF(ISLTDN.EQ.1.AND.INSEEK)Z1(IN)=ZS                                    LAY00270
      X1(IN)=XS+XIX*(Z1(IN)-ZS)                                              LAY00280
      Y1(IN)=YS+XIY*(Z1(IN)-ZS)                                              LAY00290
      X0(IN)=XS+XIX*(Z0(IN)-ZS)                                              LAY00300
      Y0(IN)=YS+XIY*(Z0(IN)-ZS)                                              LAY00310
      RETURN                                                                 LAY00320
      END                                                                    LAY00330
```

```
      SUBROUTINE CYLXY(XIX,XIY,IN)                                      CYL00010
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)               CYL00020
      COMMON /CYL/XC,YC,RADIUS                                          CYL00030
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT      CYL00040
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)        CYL00050
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)            CYL00060
C**** FIRST DETERMINE INTERSECTION PTS OF LOS WITH UPPER                 CYL00070
C**** AND LOWER BASE PLANES OF CLOUD CYLINDER                           CYL00080
      TESALL=0.0                                                        CYL00090
      CALL LAYRXY(XIX,XIY,IN)                                           CYL00100
      IF(TESALL.LT.0.0)RETURN                                           CYL00110
      X0LAYR=X0(IN)                                                     CYL00120
      Y0LAYR=Y0(IN)                                                     CYL00130
      Z0LAYR=Z0(IN)                                                     CYL00140
      X1LAYR=X1(IN)                                                     CYL00150
      Y1LAYR=Y1(IN)                                                     CYL00160
      Z1LAYR=Z1(IN)                                                     CYL00170
C**** CALCULATE NEXT THE INTERSECTION PTS OF THE LOS                     CYL00180
C**** WITH THE SURFACE OF AN INFINITE VERTICAL CYLINDER WITH             CYL00190
C**** THE SAME RADIUS (RADIUS)AND LATERAL POSITION (XC,YC) AS THE CLOUD  CYL00200
      IF(IVERT.NE.1)GO TO 40                                            CYL00210
      VERTES=SQRT((XT-XC)**2+(YT-YC)**2)                                CYL00220
      TESVT=RADIUS-VERTES                                               CYL00230
      IF(VERTES.LT.RADIUS)GO TO 400                                     CYL00240
      RETURN                                                            CYL00250
   40 IF((XT-XS).EQ.0.0)GO TO 60                                        CYL00260
      ALPHA=(YT-YS)/(XT-XS)                                             CYL00270
      A2=ALPHA**2                                                       CYL00280
      C2=1.0+A2                                                         CYL00290
      C1=2.0*(ALPHA*(YT-YC-ALPHA*XT)-XC)                                CYL00300
      C0=XC**2+(YT-YC-ALPHA*XT)**2-RADIUS**2                            CYL00310
      TESARG=C1**2-4.0*C2*C0                                            CYL00320
      IF(TESARG.LT.0.0)RETURN                                           CYL00330
      XP=(-C1+SQRT(TESARG))/(2.0*C2)                                    CYL00340
      XM=(-C1-SQRT(TESARG))/(2.0*C2)                                    CYL00350
      YP=YT+ALPHA*(XP-XT)                                               CYL00360
      YM=YT+ALPHA*(XM-XT)                                               CYL00370
      IF(IHORIZ.EQ.1)GO TO 300                                          CYL00380
      ZM=ZS+(1.0/XIX)*(XM-XS)                                           CYL00390
      ZP=ZS+(1.0/XIX)*(XP-XS)                                           CYL00400
C**** CHECK FOR SKEW MISS OF CLOUD                                      CYL00410
      TOPP=ZP-ZTOP                                                      CYL00420
      TOPM=ZM-ZTOP                                                      CYL00430
      BASP=ZBAS-ZP                                                      CYL00440
      BASM=ZBAS-ZM                                                      CYL00450
      IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50              CYL00460
      TESALL=-1.0                                                       CYL00470
      RETURN                                                            CYL00480
   60 XP=XT                                                             CYL00490
      TESARG=RADIUS**2-(XP-XT)**2                                       CYL00500
      IF(TESARG.LT.0.0)RETURN                                           CYL00510
      YP=YC+SQRT(TESARG)                                                CYL00520
      XM=XT                                                             CYL00530
      YM=YC+SQRT(TESARG)                                                CYL00540
      IF(IHORIZ.EQ.1)GO TO 300                                          CYL00550
      ZM=ZS+(1.0/XIY)*(YM-YS)                                           CYL00560
      ZP=ZS+(1.0/XIY)*(YP-YS)                                           CYL00570
C**** CHECK FOR SKEW MISS OF CLOUD                                      CYL00580
      TOPP=ZP-ZTOP                                                      CYL00590
      TOPM=ZM-ZTOP                                                      CYL00600
      BASP=ZBAS-ZP                                                      CYL00610
      BASM=ZBAS-ZM                                                      CYL00620
      IF((TOPP*TOPM).LT.0.0.OR.(BASP*BASM).LT.0.0)GO TO 50              CYL00630
      TESALL=-1.0                                                       CYL00640
      RETURN                                                            CYL00650
   50 IF(ZP.LT.ZM)GO TO 100                                             CYL00660
      X0CYL=XM                                                          CYL00670
      Y0CYL=YM                                                          CYL00680
      Z0CYL=ZM                                                          CYL00690
      X1CYL=XP                                                          CYL00700
```

```
      Y1CYL=YP                                                          CYL00710
      Z1CYL=ZP                                                          CYL00720
      GO TO 200                                                         CYL00730
  100 X0CYL=XP                                                          CYL00740
      Y0CYL=YP                                                          CYL00750
      Z0CYL=ZP                                                          CYL00760
      X1CYL=XM                                                          CYL00770
      Y1CYL=YM                                                          CYL00780
      Z1CYL=ZM                                                          CYL00790
  200 CONTINUE                                                          CYL00800
      GO TO 450                                                         CYL00810
  300 X0(IN)=XM                                                         CYL00820
      Y0(IN)=YM                                                         CYL00830
      Z0(IN)=ZT                                                         CYL00840
      X1(IN)=XP                                                         CYL00850
      Y1(IN)=YP                                                         CYL00860
      Z1(IN)=ZT                                                         CYL00870
      GO TO 500                                                         CYL00880
  400 X0(IN)=XT                                                         CYL00890
      Y0(IN)=YT                                                         CYL00900
      X1(IN)=XT                                                         CYL00910
      Y1(IN)=YT                                                         CYL00920
      GO TO 500                                                         CYL00930
  450 IF(Z0CYL.LT.Z0LAYR)GO TO 460                                      CYL00940
      X0(IN)=X0CYL                                                      CYL00950
      Y0(IN)=Y0CYL                                                      CYL00960
      Z0(IN)=Z0CYL                                                      CYL00970
      GO TO 470                                                         CYL00980
  460 X0(IN)=X0LAYR                                                     CYL00990
      Y0(IN)=Y0LAYR                                                     CYL01000
      Z0(IN)=Z0LAYR                                                     CYL01010
  470 IF(Z1CYL.GT.Z1LAYR)GO TO 480                                      CYL01020
      X1(IN)=X1CYL                                                      CYL01030
      Y1(IN)=Y1CYL                                                      CYL01040
      Z1(IN)=Z1CYL                                                      CYL01050
      GO TO 500                                                         CYL01060
  480 X1(IN)=X1LAYR                                                     CYL01070
      Y1(IN)=Y1LAYR                                                     CYL01080
      Z1(IN)=Z1LAYR                                                     CYL01090
  500 RETURN                                                            CYL01100
      END                                                               CYL01110
```

```
      SUBROUTINE CLEXTN(TAUN,IN)                                        CLE00010
      DIMENSION AA(42),BB(42),CC(42),A(7,6,3)                           CLE00020
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)               CLE00030
      EQUIVALENCE (AA(1),A(1,1,1)),(BB(1),A(1,1,2)),(CC(1),A(1,1,3))    CLE00040
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)        CLE00050
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT      CLE00060
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV                                  CLE00070
      DATA AA/5.3744E-2,2.9313E-4,-1.1300E-6,1.6228E-9,-1.0421E-12,     CLE00080
     +2.4192E-16,0.0000E-00,8.4451E-2,1.6549E-4,-4.7300E-7,             CLE00090
     +4.4334E-10,-1.3851E-13,-2.0644E-17,1.2569E-20,1.0240E-1,          CLE00100
     +4.8678E-5,-2.2195E-7,1.4506E-10,5.9393E-14,-8.7250E-17,           CLE00110
     +2.1383E-20,3.6775E-2,1.6160E-4,-6.4666E-7,9.5298E-10,             CLE00120
     +-6.4023E-13,1.6916E-16,-7.9924E-21,1.8410E-2,6.6870E-4,           CLE00130
     +-2.8406E-6,5.4478E-9,-5.4854E-12,2.7932E-15,-5.8271E-19,          CLE00140
     +3.3458E-2,1.3098E-4,-4.1528E-7,6.1166E-10,-4.6681E-13,            CLE00150
     +1.7383E-16,-2.5043E-20/                                           CLE00160
      DATA BB/7.5099E-2,3.2061E-4,-1.7060E-6,3.3538E-9,-3.2719E-12,     CLE00170
     +1.5721E-15,-3.0123E-19,1.1808E-1,2.2387E-4,-8.7996E-7,            CLE00180
     +1.1298E-9,-6.6823E-13,1.7514E-16,-1.5665E-20,1.4155E-1,           CLE00190
     +-5.0592E-5,-1.2280E-7,1.2715E-10,2.7470E-14,-6.6940E-17,          CLE00200
     +1.7734E-20,4.9533E-2,2.0904E-4,-1.1626E-6,2.3531E-9,              CLE00210
     +-2.3463E-12,1.1472E-15,-2.2267E-19,3.8315E-3,1.1837E-3,           CLE00220
     +-5.1096E-6,9.4660E-9,-8.6426E-12,3.6778E-15,-5.5775E-19,          CLE00230
     +4.1534E-2,2.0220E-4,-8.1463E-7,1.3590E-9,-1.1142E-12,             CLE00240
     +4.3867E-16,-6.6426E-20/                                           CLE00250
      DATA CC/1.1269E-2,2.8659E-4,-6.0210E-8,-1.5274E-9,2.7747E-12,     CLE00260
     +-1.8946E-15,4.5539E-19,1.9856E-2,1.2292E-4,-8.4479E-8,            CLE00270
     +8.6563E-12,0.0000E-0,0.0000E-0,0.0000E-0,3.1907E-2,               CLE00280
     +1.9632E-4,-1.1114E-7,-3.3987E-10,5.2528E-13,-2.7799E-16,          CLE00290
     +5.0469E-20,6.8522E-3,1.5362E-4,7.5813E-8,-1.1430E-9,              CLE00300
     +1.8885E-12,-1.2477E-15,2.9522E-19,8.5792E-4,6.4122E-5,            CLE00310
     +7.7271E-7,-2.9750E-9,4.4014E-12,-3.0626E-15,8.1541E-19,           CLE00320
     +3.7151E-3,1.4919E-4,-9.3486E-8,-1.2183E-10,1.8733E-13,            CLE00330
     +-9.2167E-17,1.5939E-20/                                           CLE00340
      TAUN=0.0                                                          CLE00350
      ZZA=Z0(IN)-ZBAS                                                   CLE00360
      ZZB=Z1(IN)-ZBAS                                                   CLE00370
      VERDIS=ZZB-ZZA                                                    CLE00380
      HORDIS=SQRT((X1(IN)-X0(IN))**2+(Y1(IN)-Y0(IN))**2)                CLE00390
      EL=SQRT(HORDIS**2+VERDIS**2)                                      CLE00400
      IF(EL.EQ.0.0)RETURN                                               CLE00410
      IF(IHORIZ.EQ.1)GO TO 200                                          CLE00420
      XI=VERDIS/EL                                                      CLE00430
      ELA=ZZA*1000.0                                                    CLE00440
      ELB=(XI*EL+ZZA)*1000.0                                            CLE00450
      POLYA=0.0                                                         CLE00460
      POLYB=0.0                                                         CLE00470
      DO 100 N=1,7                                                      CLE00480
      EN=FLOAT(N)                                                       CLE00490
      AN=A(N,ICL,IWV)                                                   CLE00500
      TERMB=(1.0/(XI*EN))*AN*ELB**N                                     CLE00510
      TERMA=(1.0/(XI*EN))*AN*ELA**N                                     CLE00520
      POLYA=POLYA+TERMA                                                 CLE00530
  100 POLYB=POLYB+TERMB                                                 CLE00540
      TAUN=POLYB-POLYA                                                  CLE00550
      GO TO 300                                                         CLE00560
  200 ZZA=ZZA*1000.0                                                    CLE00570
      DO 250 N=1,7                                                      CLE00580
      TERMK=A(N,ICL,IWV)*HORDIS*1000.0*ZZA**(N-1)                       CLE00590
  250 TAUN=TAUN+TERMK                                                   CLE00600
  300 RETURN                                                            CLE00610
      END                                                               CLE00620
```

```
      SUBROUTINE LISOUT(NN,CTRANS,TAUTOT,TAU,NLAYER,IRUN)                    LIS00010
C                                                                            LIS00020
C**** OUTPUT CONTROL ROUTINE                                                 LIS00030
C                                                                            LIS00040
      DIMENSION ALPH(6),T(20),TAU(20)                                        LIS00050
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)                    LIS00060
      COMMON /PATHL/ X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)            LIS00070
      COMMON /LEVEL/ISLTUP,ISLTDN,IHORIZ,IVERT,TESALL,TESARG,TESVT           LIS00080
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)                 LIS00090
      COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)              LIS00100
     +,RADICL(10)                                                            LIS00110
      COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)                LIS00120
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTULIS00130
      DATA ALPH/2HST,2HAS,2HNS,2HSC,2HCH,2HCC/                               LIS00140
      IF(IWV.EQ.1)WVL=0.55                                                   LIS00150
      IF(IWV.EQ.2)WVL=3.80                                                   LIS00160
      IF(IWV.EQ.3)WVL=10.60                                                  LIS00170
      IF(IRUN.LT.2) WRITE(IOOUT,150)WVL                                      LIS00180
      IF(IRUN.GT.1) WRITE(IOOUT,160)WVL                                      LIS00190
  150 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS /)                     LIS00200
  160 FORMAT(1H0,40X,13HWAVELENGTH = ,F5.2,9H MICRONS /)                     LIS00210
      IF(IHORIZ.EQ.1)WRITE(IOOUT,200)                                        LIS00220
      IF(IVERT.EQ.1)WRITE(IOOUT,210)                                         LIS00230
      IF(ISLTUP.EQ.1)WRITE(IOOUT,220)                                        LIS00240
      IF(ISLTDN.EQ.1)WRITE(IOOUT,230)                                        LIS00250
  200 FORMAT(1H0,40X,28HLINE-OF-SIGHT IS HORIZONTAL //)                      LIS00260
  210 FORMAT(1H0,40X,26HLINE-OF-SIGHT IS VERTICAL //)                        LIS00270
  220 FORMAT(1H0,40X,28HLINE-OF-SIGHT SLANTS UPWARD //)                      LIS00280
  230 FORMAT(1H0,40X,30HLINE-OF-SIGHT SLANTS DOWNWARD //)                    LIS00290
      PTHLEN=SQRT((XS-XT)**2+(YS-YT)**2+(ZS-ZT)**2)                          LIS00300
      WRITE(IOOUT,300)PTHLEN                                                 LIS00310
  300 FORMAT(1H0,40X,29HTOTAL LINE-OF-SIGHT LENGTH = ,F7.3,4H KM )           LIS00320
      ATTLEN=0.0                                                             LIS00330
      IF(TAUTOT.EQ.0.0)GO TO 780                                             LIS00340
      DO 310 N=1,NN                                                          LIS00350
  310 ATTLEN=ATTLEN+ATTL(N)                                                  LIS00360
      WRITE(IOOUT,400)ATTLEN                                                 LIS00370
  400 FORMAT(1H0,40X,50HTOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD      LIS00380
     += ,F7.3,4H KM )                                                        LIS00390
      WRITE(IOOUT,500)TAUTOT                                                 LIS00400
  500 FORMAT(1H0,40X,22HTOTAL OPTICAL DEPTH = ,F7.2/)                        LIS00410
      WRITE(IOOUT,600)CTRANS                                                 LIS00420
  600 FORMAT(1H0,40X,36HTRANSMITTANCE ALONG LINE-OF-SIGHT = ,E11.5,//)       LIS00430
      WRITE(IOOUT,605)                                                       LIS00440
  605 FORMAT(1H0,40X,50HSEEKER COORDINATES (KM)   TARGET COORDINATES (KMLIS00450
     +) )                                                                    LIS00460
      WRITE(IOOUT,606)                                                       LIS00470
  606 FORMAT(1H ,40X,23HXSEEKER YSEEKER ZSEEKER,3X,23HXTARGET YTARGET ZTLIS00480
     +ARGET)                                                                 LIS00490
      WRITE(IOOUT,607)                                                       LIS00500
  607 FORMAT(1H ,40X,3(8H------- ),2X,3(8H------- ))                         LIS00510
      WRITE(IOOUT,609)XS,YS,ZS,XT,YT,ZT                                      LIS00520
  608 FORMAT(1H0,40X,3(F6.3,2X),2X,3(F6.3,2X))                               LIS00530
      WRITE(IOOUT,613)                                                       LIS00540
  613 FORMAT(1H ,//)                                                         LIS00550
      WRITE(IOOUT,615)                                                       LIS00560
  615 FORMAT(1H0,40X,62HCLOUD TYPE    LINE-OF-SIGHT  INTERSECTION  COORDLIS00570
     +INATES  (KM) )                                                         LIS00580
      WRITE(IOOUT,620)                                                       LIS00590
  620 FORMAT(1H ,40X,10H/ID NUMBER,4X,47HXUPPER  YUPPER  ZUPPER   XLOWERLIS00600
     +  YLOWER  ZLOWER)                                                      LIS00610
      WRITE(IOOUT,625)                                                       LIS00620
  625 FORMAT(1H ,40X,10H----------,4X,3(8H------  ),1X,3(8H------  ))       LIS00630
      IF(NLAYER.EQ.0)GO TO 685                                               LIS00640
      DO 670 N=1,NLAYER                                                      LIS00650
      IND=ILCTYP(N)                                                          LIS00660
      IF(NLINT(N).EQ.0)GO TO 670                                             LIS00670
      WRITE(IOOUT,680)ALPH(IND),N,X1(N),Y1(N),Z1(N),X0(N),Y0(N),Z0(N)        LIS00680
  670 CONTINUE                                                               LIS00690
  680 FORMAT(1H0,42X,A2,1H/,I2,6X,3(F7.3,1X),1X,3(F7.3,1X))                  LIS00700
```

```
685 NLO=NLAYER+1                                                        LIS00710
    IF(NLAYER.EQ.NN)GO TO 699                                           LIS00720
    LL=0                                                                LIS00730
    DO 690 N=NLO,NN                                                     LIS00740
    LL=LL+1                                                             LIS00750
    IND=ICCTYP(LL)                                                      LIS00760
    IF(NCINT(LL).EQ.0)GO TO 690                                         LIS00770
    WRITE(IOOUT,680)ALPH(IND),N,X1(N),Y1(N),Z1(N),X0(N),Y0(N),Z0(N)     LIS00780
690 CONTINUE                                                            LIS00790
    WRITE(IOOUT,613)                                                    LIS00800
    WRITE(IOOUT,691)                                                    LIS00810
691 FORMAT(1H0,40X,59HCLOUD TYPE   CUMULUS CENTER POSITIONS (FOR CC ORLIS00820
   + CH TYPES) )                                                        LIS00830
    WRITE(IOOUT,692)                                                    LIS00840
692 FORMAT(1H ,40X,10H/ID NUMBER,6X,26HXCLOUD (KM)    YCLOUD (KM))      LIS00850
    WRITE(IOOUT,693)                                                    LIS00860
693 FORMAT(1H ,40X,10H----------,6X,26H----------    ----------)       LIS00870
    LL=0                                                                LIS00880
    DO 695 N=NLO,NN                                                     LIS00890
    LL=LL+1                                                             LIS00900
    IND=ICCTYP(LL)                                                      LIS00910
695 WRITE(IOOUT,696)ALPH(IND),N,XCLOUD(LL),YCLOUD(LL)                   LIS00920
696 FORMAT(1H0,42X,A2,1H/,I2,10X,F7.3,8X,F7.3)                          LIS00930
699 CONTINUE                                                            LIS00940
    WRITE(IOOUT,613)                                                    LIS00950
    DO 610 N=1,NN                                                       LIS00960
610 T(N)=EXP(-TAU(N))                                                   LIS00970
    IF(NCCLDS.GT.0) WRITE(IOOUT,700)                                    LIS00980
    IF(NCCLDS.LT.1) WRITE(IOOUT,705)                                    LIS00990
700 FORMAT(1H1,20X,11HCLOUD TYPE ,2X,15HHEIGHT OF BASE ,2X,             LIS01000
   +10HTHICKNESS ,2X,16HRADIUS OF CLOUD ,2X,14HOPTICAL DEPTH ,2X,       LIS01010
   +14HTRANSMITTANCE )                                                  LIS01020
705 FORMAT(1H0,20X,11HCLOUD TYPE ,2X,15HHEIGHT OF BASE ,2X,             LIS01030
   +10HTHICKNESS ,2X,16HRADIUS OF CLOUD ,2X,14HOPTICAL DEPTH ,2X,       LIS01040
   +14HTRANSMITTANCE )                                                  LIS01050
    WRITE(IOOUT,710)                                                    LIS01060
710 FORMAT(1H ,20X,10H/ID NUMBER,9X,4H(KM),9X,4H(KM),12X,4H(KM),9X,     LIS01070
   +11HALONG L-O-S,5X,11HALONG L-O-S)                                   LIS01080
    WRITE(IOOUT,720)                                                    LIS01090
720 FORMAT(1H ,20X,10H----------,3X,14H--------------,3X,9H---------,   LIS01100
   +3X,15H---------------,3X,13H-------------,3X,13H-------------)      LIS01110
    IF(NLAYER.EQ.0)GO TO 735                                            LIS01120
    BLANK=0.0                                                           LIS01130
    DO 730 N=1,NLAYER                                                   LIS01140
    IND=ILCTYP(N)                                                       LIS01150
730 WRITE(IOOUT,760)ALPH(IND),N,ZLBASE(N),ZLTHIC(N),BLANK,TAU(N),T(N)   LIS01160
735 NLO=NLAYER+1                                                        LIS01170
740 IF(NN.EQ.NLAYER)GO TO 800                                           LIS01180
    LL=0                                                                LIS01190
    DO 750 N=NLO,NN                                                     LIS01200
    LL=LL+1                                                             LIS01210
    IND=ICCTYP(LL)                                                      LIS01220
750 WRITE(IOOUT,760)ALPH(IND),N,ZCBASE(LL),ZCTHIC(LL),RADICL(LL),       LIS01230
   +TAU(N),T(N)                                                         LIS01240
760 FORMAT(1H0,22X,A2,1H/,I2,10X,F7.3,7X,F7.3,8X,F7.3,10X,F7.2,7X,      LIS01250
   +E11.5)                                                              LIS01260
    GO TO 800                                                           LIS01270
780 WRITE(IOOUT,790)                                                    LIS01280
790 FORMAT(1H0,20X,59HNO CLOUD OBSCURATION : L-O-S DOES NOT INTERSECT   LIS01290
   +ANY CLOUDS //)                                                      LIS01300
800 RETURN                                                              LIS01310
    END                                                                 LIS01320
```

```
      SUBROUTINE DEFSET(ISTEP,NMAX)                                     DEF00010
C                                                                       DEF00020
C**** THIS ROUTINE RESETS THE CLOUD BASE HEIGHT, THICKNESS,             DEF00030
C**** AND RADIUS VALUES TO THE NEAREST REALISTIC BOUNDARIES             DEF00040
C**** IF THEY DO NOT LIE CLOSE TO THE RANGES SPECIFIED IN               DEF00050
C**** R. D. H. LOW'S PAPER.                                             DEF00060
C                                                                       DEF00070
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)            DEF00080
      COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)         DEF00090
     +,RADICL(10)                                                       DEF00100
      IF(ISTEP.NE.1)GO TO 10                                            DEF00110
      DO 5 N=1,NMAX                                                     DEF00120
      ITYPE=ILCTYP(N)                                                   DEF00130
      IF(ITYPE.EQ.2)GO TO 3                                             DEF00140
    1 IF(ZLBASE(N).LT.0.1)ZLBASE(N)=0.1                                 DEF00150
      IF(ZLBASE(N).GT.1.5)ZLBASE(N)=1.5                                 DEF00160
      IF(ITYPE.EQ.3)GO TO 4                                             DEF00170
    2 IF(ZLTHIC(N).LT.0.2)ZLTHIC(N)=0.2                                 DEF00180
      IF(ZLTHIC(N).GT.1.0)ZLTHIC(N)=1.0                                 DEF00190
      GO TO 5                                                           DEF00200
    3 IF(ZLBASE(N).LT.2.0)ZLBASE(N)=2.0                                 DEF00210
      IF(ZLBASE(N).GT.5.0)ZLBASE(N)=5.0                                 DEF00220
    4 IF(ZLTHIC(N).LT.1.0)ZLTHIC(N)=1.0                                 DEF00230
      IF(ZLTHIC(N).GT.4.0)ZLTHIC(N)=4.0                                 DEF00240
    5 CONTINUE                                                          DEF00250
   10 IF(ISTEP.NE.2)GO TO 20                                            DEF00260
      DO 15 N=1,NMAX                                                    DEF00270
      IF(ZCBASE(N).LT.0.8)ZCBASE(N)=0.8                                 DEF00280
      IF(ZCBASE(N).GT.1.5)ZCBASE(N)=1.5                                 DEF00290
      IF(ZCTHIC(N).LT.0.2)ZCTHIC(N)=0.2                                 DEF00300
      IF(ZCTHIC(N).GT.5.0)ZCTHIC(N)=5.0                                 DEF00310
      IF(RADICL(N).LT.0.05)RADICL(N)=0.05                               DEF00320
      IF(RADICL(N).GT.0.6)RADICL(N)=0.6                                 DEF00330
   15 CONTINUE                                                          DEF00340
   20 RETURN                                                            DEF00350
      END                                                               DEF00360
```

```
      SUBROUTINE CLREAD(NLAYER,NCCLDS,IERR)                                 CLR00010
      DIMENSION IALPHA(18),DATELT(5)                                        CLR00020
      DIMENSION X1(20),Y1(20),Z1(20),X0(20),Y0(20),Z0(20)                   CLR00030
      COMMON /PATHL/X0,Y0,Z0,X1,Y1,Z1,XS,YS,ZS,XT,YT,ZT,ATTL(20)            CLR00040
      COMMON /GEOMET/PTS(15),IGEOSW                                         CLR00050
      COMMON /BASTOP/ZBAS,ZTOP,ICL,IWV,ILCTYP(10),ICCTYP(10)                CLR00060
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCLR00070
      COMMON /BASTH/ZLBASE(10),ZLTHIC(10),ZCBASE(10),ZCTHIC(10)             CLR00080
     +,RADICL(10)                                                           CLR00090
      COMMON /INTCL/XCLOUD(10),YCLOUD(10),NLINT(10),NCINT(10)               CLR00100
      DATA IALPHA/2HCL,2HST,2HCL,2HAS,2HCL,2HNS,2HCL,2HSC,2HCL,2HCH,        CLR00110
     +2HCL,2HCC,2HSE,2HEK,2HTA,2HRG,2HGO,2H  /                              CLR00120
      ISFLAG=0                                                              CLR00130
      ITFLAG=0                                                              CLR00140
      IF(IGEOSW.NE.1)GO TO 60                                               CLR00150
      XS=PTS(4)                                                             CLR00160
      YS=PTS(5)                                                             CLR00170
      ZS=PTS(6)                                                             CLR00180
      XT=PTS(1)                                                             CLR00190
      YT=PTS(2)                                                             CLR00200
      ZT=PTS(3)                                                             CLR00210
   60 CONTINUE                                                              CLR00220
      DO 900 N=1,23                                                         CLR00230
      READ(IOIN,100)IALPH,IALP,(DATELT(I),I=1,5)                            CLR00240
  100 FORMAT(2A2,1X,5(E10.5,1X))                                            CLR00250
      INDEX=10                                                              CLR00260
      DO 200 K=1,17,2                                                       CLR00270
      IF((IALPH.EQ.IALPHA(K)).AND.(IALP.EQ.IALPHA(K+1))) GO TO 50           CLR00280
      GO TO 200
   50 REALK=FLOAT(K)
      RQT=REALK/2.0
      KJ=IFIX(RQT)+1
      INDEX=KJ
      GO TO 300                                                             CLR00300
  200 CONTINUE                                                              CLR00310
  300 IF(INDEX.LT.7)GO TO (350,350,350,350,400,400),INDEX                   CLR00320
      INM6=INDEX-6                                                          CLR00330
      GO TO (500,600,999,997),INM6                                          CLR00340
  350 NLAYER=NLAYER+1                                                       CLR00350
      IF(NLAYER.GT.10) GO TO 993                                            CLR00360
      ILCTYP(NLAYER)=INDEX                                                  CLR00370
      ZLBASE(NLAYER)=DATELT(1)                                              CLR00380
      ZLTHIC(NLAYER)=DATELT(2)                                              CLR00390
      GO TO 900                                                             CLR00400
  400 NCCLDS=NCCLDS+1                                                       CLR00410
      IF(NCCLDS.GT.10) GO TO 995                                            CLR00420
      ICCTYP(NCCLDS)=INDEX                                                  CLR00430
      ZCBASE(NCCLDS)=DATELT(1)                                              CLR00440
      ZCTHIC(NCCLDS)=DATELT(2)                                              CLR00450
      RADICL(NCCLDS)=DATELT(3)                                              CLR00460
      XCLOUD(NCCLDS)=DATELT(4)                                              CLR00470
      YCLOUD(NCCLDS)=DATELT(5)                                              CLR00480
      GO TO 900                                                             CLR00490
  500 IF(ISFLAG.EQ.1) GO TO 997                                             CLR00500
      XS=DATELT(1)                                                          CLR00510
      YS=DATELT(2)                                                          CLR00520
      ZS=DATELT(3)                                                          CLR00530
      ISFLAG=1                                                              CLR00540
      GO TO 900                                                             CLR00550
  600 IF(ITFLAG.GT.1) GO TO 997                                             CLR00560
      XT=DATELT(1)                                                          CLR00570
      YT=DATELT(2)                                                          CLR00580
      ZT=DATELT(3)                                                          CLR00590
      ITFLAG=1                                                              CLR00600
  900 CONTINUE                                                              CLR00610
      GO TO 999                                                             CLR00620
  993 IERR=1                                                                CLR00630
      WRITE(IOOUT,994)                                                      CLR00640
  994 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF STRATIFORM CLOUDS ICLR00650
     +NPUT EXCEEDS THE LIMIT OF 10 //)                                      CLR00660
```

```
      GO TO 999                                                                 CLR00670
  995 IERR=1                                                                    CLR00680
      WRITE(IOOUT,996)                                                          CLR00690
  996 FORMAT(1H0,20X,76H***CLREAD ERROR*** NUMBER OF CUMULIFORM CLOUDS ICLR00700
     +NPUT EXCEEDS THE LIMIT OF 10 //)                                          CLR00710
      GO TO 999                                                                 CLR00720
  997 IERR=1                                                                    CLR00730
      WRITE(IOOUT,998)                                                          CLR00740
  998 FORMAT(1H0,20X,64H***CLREAD ERROR*** IMPROPER INPUT FORMAT OR ABSECLR00750
     +NT GO   SENTINEL //)                                                      CLR00760
  999 RETURN                                                                    CLR00770
      END                                                                       CLR00780
```

```
      SUBROUTINE SCREEN(IERR,ICLMAT)                                    SCR00010
C                                                                       SCR00020
C  THIS MODULE  1) COMPUTES TRANSMITTANCE REQUIRED TO REDUCE THE        SCR00030
C                  PROBABILITY OF STATIC TARGET DETECTION BELOW A       SCR00040
C                  GIVEN LEVEL FOR CERTAIN TYPES OF IMAGERS. (ITAM)     SCR00050
C                                                                       SCR00060
C       - AND/OR -                                                      SCR00070
C                                                                       SCR00080
C              2) COMPUTES HC AND WP 105 MM/155 MM SMOKE MUNITIONS      SCR00090
C                 REQUIRED TO PRODUCE A SMOKE SCREEN OF USER-DEFINED    SCR00100
C                 LENGTH AND DURATION FOR VISIBLE, NEAR, MID AND FAR    SCR00110
C                 IR WAVELENGTHS.                                       SCR00120
C                                                                       SCR00130
C   SUBROUTINES CALLED BY SCREEN ARE - CWIC AND ITAM.                   SCR00140
C                                                                       SCR00150
C   ALL OUTPUT FROM SCREEN IS TABULAR.                                  SCR00160
C                                                                       SCR00170
C  THE PRESENT VERSION OF CWIC USES THE XSCALE MODULE TO OPTIONALLY     SCR00180
C     CORRECT FOR EXTINCTION DUE TO FOG, RAIN AND/OR SNOW AT IR         SCR00190
C     WAVELENGTHS BASED ON VISIBILITY IN THE .5 MICROMETER REGION.      SCR00200
C  CWIC CAN ALSO COMPUTE PASQUILL CATEGORY FROM FUNDAMENTAL MET INPUTS  SCR00210
C     IF THE USER CHOOSES NOT TO PROVIDE THE CATEGORY DIRECTLY.         SCR00220
C  CLIMATOLOGICAL VALUES FROM THE CLIMAT MODULE CAN OPTIONALLY BE       SCR00230
C     USED AS "TYPICAL" MET INPUTS.                                     SCR00240
C                                                                       SCR00250
C  THE ITAM MODULE CAN BE USED IN A LOOPING MODE THROUGH MULTIPLE SETS  SCR00260
C     OF INPUT RECORDS TO GENERATE TABLES.  THE LAST VALUE OF TRANS-    SCR00270
C     MITTANCE COMPUTED IS THAT VALUE WHICH CAN (OPTIONALLY) BE PASSED  SCR00280
C     TO CWIC AS THE THRESHOLD LEVEL FOR TOTAL PATH TRANSMITTANCE.      SCR00290
C                                                                       SCR00300
C  TWO RECORDS MUST BE PROVIDED TO SCREEN:                              SCR00310
C  ONE CARD MUST BE INPUT TO SCREEN TO SELECT OPTIONS: (3(1X,I1))       SCR00320
C   COL 2    ICITAM   =1 CALL ITAM, OR 0 (NO CALL).                    SCR00330
C   COL 4    ICCWIC   =1 CALL CWIC, OR 0 (NO CALL).                    SCR00340
C   COL 6    ICCLIM   =1 USE CLIMAT FOR MET INPUTS, OR 0 USE USER VALUES. SCR00350
C                                                                       SCR00360
C  IF CHOSEN, ALL INPUT RECORDS TO ITAM ARE READ FOLLOWING THE ABOVE    SCR00370
C     RECORD. (SEE INPUT DESCRIPTION IN ITAM)                           SCR00380
C                                                                       SCR00390
C     AND, IF CHOSEN, INPUT RECORDS TO CWIC ARE THEN READ. (SEE INPUT   SCR00400
C     DESCRIPTION IN CWIC)                                              SCR00410
C                                                                       SCR00420
C  FINALLY, ONE RECORD IS READ BY SCREEN WITH THE WORD "END" IN COLUMNS  SCR00430
C     1-3.  THIS RETURNS CONTROL TO THE EOSAEL EXECUTIVE MODULE.        SCR00440
C                                                                       SCR00450
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSCR00460
      READ (IOIN,10) ICITAM,ICCWIC,ICCLIM                               SCR00470
10    FORMAT(1X,I1,1X,I1,1X,I1)                                         SCR00480
      IF (ICITAM.NE.0) CALL ITAM(IERR,TFNL)                             SCR00490
      IF (IERR.EQ.1) GOTO 15                                            SCR00500
      IF (ICCLIM.EQ.0) GOTO 20                                          SCR00510
      IF (ICLMAT.EQ.1) GOTO 20                                          SCR00520
      IERR=1                                                            SCR00530
      WRITE (IOOUT,30)                                                  SCR00540
30    FORMAT(1X,62H*** IN SCREEN ROUTINE, MET SOURCE SPECIFIED AS CLIMATSCR00550
     *OLOGICAL,/5X,39HBUT CLIMAT ROUTINE HAD NOT BEEN CALLED.)          SCR00560
      RETURN                                                            SCR00570
15    WRITE (IOOUT,40)                                                  SCR00580
      TFNL=1.                                                           SCR00590
40    FORMAT(1X,21HIERR FLAG SET IN ITAM)                               SCR00600
20    IF  (ICCWIC.NE.0) CALL CWIC (IERR,ICITAM,ICCLIM,TFNL)             SCR00610
      RETURN                                                            SCR00620
      END                                                               SCR00630
```

```
      SUBROUTINE CWIC (IERR,ICITAM,ICCLIM,TFNL)                          CWC00010
C  CWIC COMPUTES THE REQUIRED SMOKE MUNITIONS EXPENDITURE (NUMBER,        CWC00020
C     RATE OF FIRE, PLACEMENT) TO PRODUCE A SCREEN OF DEFINED             CWC00030
C     LENGTH AND DURATION USING HC OR WP 105MM OR 155MM SMOKE             CWC00040
C     MUNITIONS.  WAVELENGTHS ARE FOR RANGES OF VISIBLE, NEAR,            CWC00050
C     MID AND FAR IR.                                                     CWC00060
C                                                                         CWC00070
C  ON INPUT, IF ICITAM IS NON-ZERO, ALL COMPUTATIONS WILL USE             CWC00080
C     THE INPUT TFNL AS THE TOTAL SCREEN TRANSMISSION THRESHOLD.  IF      CWC00090
C     ICITAM IS 0, THEN BUILT-IN TOTAL THRESHOLDS OF .05 ARE USED         CWC00100
C     FOR ALL WAVELENGTH REGIONS.                                         CWC00110
C                                                                         CWC00120
C  INPUTS TO CWIC ARE ON STANDARDIZED RECORDS CONTAINING KEY-WORDS        CWC00130
C     IN COLUMNS 1-4 AND REAL (IE DECIMAL) VALUES FOR ALL INPUTS,         CWC00140
C     PLACED IN FIELDS 11-20, 21-30,..., 71-80.  KEY-WORD TYPES           CWC00150
C     ARE    SCRN   FOR SCREEN, OBSERVER/TARGET LOS AND ADVERSE           CWC00160
C                   WEATHER CORRECTIONS.                                  CWC00170
C            METR   FOR METEOROLOGICAL CONDITIONS (MAY BE OMITTED         CWC00180
C                      IF ICCLIM IS NON-ZERO, IN WHICH CASE THE           CWC00190
C                      CLIMATOLOGICAL VALUES FROM CLIMAT ARE USED.        CWC00200
C            PASQ   FOR (OPTIONAL) MET PARAMETERS REQUIRED TO COMPUTE     CWC00210
C                      THE PASQUILL STABILITY CATEGORY.  NOT REQUIRED     CWC00220
C                      IF PASQUILL CATEGORY ITSELF IS INPUT.              CWC00230
C            DONE   WHICH RETURNS EXECUTIVE CONTROL BACK TO THE SCREEN    CWC00240
C                      MODULE.                                            CWC00250
C                                                                         CWC00260
C  THE ORDER OF THE INPUT RECORDS IS IMMATERIAL, EXCEPT THAT THE (DONE)   CWC00270
C     CARD MUST BE LAST.                                                  CWC00280
C                                                                         CWC00290
C  INPUTS:   (ALL VALUES REAL)   FORMAT (2A2,6X,7F10.3)                   CWC00300
C                                                                         CWC00310
C  KEYWORD    COLS.    VARIABLE              DESCRIPTION                  CWC00320
C  -------    -------  --------  ----------------------------------------CWC00330
C    SCRN       1-4               SCREEN/LOS DEFINITION (REQUIRED)       CWC00340
C              11-20     TIME     -  SCREEN DURATION (MINUTES)           CWC00350
C              21-30     X0       -  SCREEN LENGTH (METERS)              CWC00360
C              31-40     H3       -  SLANT RANGE OBS-TARGET (KM)         CWC00370
C              41-50     AST      -  ELEVATION ANGLE OF TARGET FROM      CWC00380
C                                     OBSERVER WRT HORIZONTAL (DEG.)      CWC00390
C              51-60     DLS      -  COMPASS DIRECTION (CLOCKWISE WRT    CWC00400
C                                     NORTH) OF LINE-OF-SIGHT (DEG.)      CWC00410
C              61-70     ARE      -  TERRAIN ROUGHNESS ELEMENT (CM)      CWC00420
C              71-80     FOG      -  ADVERSE WEATHER/HAZE SELECTION      CWC00430
C                                     0. = NO ADVERSE WEATHER             CWC00440
C                                     1. = ONLY CORRECT VISIBLE WAVE-     CWC00450
C                                          LENGTHS FOR INPUT VISIBILITY.  CWC00460
C                                     2. = CORRECT FOR FOG/HAZE FOR       CWC00470
C                                          MARITIME ARCTIC AIR MASS       CWC00480
C                                     3. = CORRECT FOR FOG/HAZE FOR       CWC00490
C                                          MARITIME POLAR AIR MASS        CWC00500
C                                     4. = CORRECT FOR FOG/HAZE FOR       CWC00510
C                                          CONTINENTAL POLAR AIR MASS     CWC00520
C                                     5. = CORRECT FOR RAIN.              CWC00530
C                                     6. = CORRECT FOR SNOW.              CWC00540
C                                                                         CWC00550
C    METR       1-4               MET INPUTS (NOT REQUIRED IF ICCLIM     CWC00560
C                                  NON-ZERO FROM SCREEN MODULE.)          CWC00570
C              11-20     S3       -  WINDSPEED (METERS/SEC)              CWC00580
C              21-30     D0       -  WIND DIRECTION (DEG) CLOCKWISE      CWC00590
C                                     WRT NORTH, USUAL MET CONVENTION     CWC00600
C              31-40     PCAT     -  PASQUILL CATEGORY. IF INPUT AS 0.   CWC00610
C                                     SEE PASQ RECORD BELOW.  OTHERWISE,  CWC00620
C                                     1.=A, 2.=B, 3.=C, 4.=D, 5.=E, 6.=F  CWC00630
C              41-50     VS       -  VISIBILITY (KM) NOT REQUIRED IF     CWC00640
C                                     FOG = 0. IS SPECIFIED.              CWC00650
C              51-60     R0       -  RELATIVE HUMIDITY (PERCENT). IF 0., CWC00660
C                                     THEN DEW POINT AND TEMPERATURE ARE  CWC00670
C                                     REQUIRED BELOW TO COMPUTE R0.       CWC00680
C              61-70     T0       -  AIR TEMP. (DEG C) REQUIRED IF R0 NOT CWC00690
C                                     GIVEN.                              CWC00700
```

```
C                71-80      T1          -  DEW POINT TEMP. (DEG C) REQUIRED IF      CWC00710
C                                          R0 NOT GIVEN.                           CWC00720
C                                                                                  CWC00730
C      PASQ       1-4                     PASQUILL CATEGORY DETERMINATION (REQUI   CWC00740
C                                          RED ONLY IF PCAT ABOVE IS 0.)           CWC00750
C                11-20      SLAT        -  LATITUDE OF SITE (DEG.). POSITIVE IC    CWC00760
C                                          NORTH LATITUDE.                         CWC00770
C                21-30      SLONG       -  LONGITUDE OF SITE (DEG.). POSITIVE      CWC00780
C                                          EAST LONGITUDE.                         CWC00790
C                31-40      SJDATE      -  JULIAN DATE. (DECIMAL DAYS)             CWC00800
C                41-50      SZHOUR      -  GMT TIME OF DAY (DECIMAL HOURS AND      CWC00810
C                                          FRACTIONAL HOURS)                       CWC00820
C                51-60      C0          -  CEILING CLOUD HEIGHT (METERS)           CWC00830
C                61-70      C1          -  CLOUD COVER (PERCENT)                   CWC00840
C                                                                                  CWC00850
C      DONE       1-4                     END OF CWIC EXECUTION (REQUIRED)        CWC00860
C                                                                                  CWC00870
C     ALL OUTPUT FROM CWIC IS TABULAR.  SUBROUTINES CALLED ARE -                   CWC00880
C          CWIC1, CWIC3, CWIC4, JPASCT, XSCALE.                                    CWC00890
C                                                                                  CWC00900
      DIMENSION CS(4,2),TRSH(4),WAVE(4),ISLANT(4),IR(8),                           CWC00910
     *H(2,2),U(2,2),LNGTH(16),ISMOKE(2),IGUN(4),JANS(4)                           CWC00920
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN, LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCWC0093
      COMMON /GEOMET/PTS(15),IGEOSW                                                CWC00940
      COMMON /CLYMAT/ TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,                CWC00950
     *WNDVEL,WNDDIR,IPASCT                                                        CWC00960
      COMMON /MO6/ P(4,2,2),Q1(4,2,2),Q2(4,2,2),Y(4,2,2),Z(4,2,2),                 CWC00970
     *             C(4,2),R(4,2),V(2),RV(7),T(4),ARE,C0,C1,C2,D0,D2,               CWC00980
     *             DLS,R0,R2,S0,S3,SJDATE,SLAT,SLONG,SZHOUR,TIME,                  CWC00990
     *             TR(4),X0,Y1,IT(2),IP0,IDUMMY                                    CWC01000
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                               CWC01010
      EXTERNAL JPASCT                                                              CWC01020
      DATA IR/2HSC,2HRN,2HME,2HTR,2HPA,2HSW,2HDO,2HNE/                             CWC01030
      DATA H /18.7,77.1,1737.3,7076.2/                                             CWC01040
      DATA U /5.4,7.9,1.8,2.6/                                                     CWC01050
      DATA LNGTH /2HVI,2HSI,2HBL,2HE ,2HNE,2HAR,2H I,2HR ,2HMI,                    CWC01060
     *2HD ,2HIR,2H  ,2HFA,2HR ,2HIR,2H  / CWC01070
      DATA JANS /2HNO,2H  ,2HYE,2HS /                                              CWC01080
      DATA ISMOKE /2HHC,2HWP/                                                      CWC01090
      DATA IGUN /2H10,2H5 ,2H15,2H5 /                                              CWC01100
C *** INITIALIZE VALUES FOR DEFAULTS.                                              CWC01110
C     EXTINCTION COEF HC (VIS.,NEAR,MID,FAR), WP (VIS.,NEAR,MID,FAR IR)           CWC01120
      DATA CS /3.3,1.5,0.53,0.13,2.46,2.0,.25,.32/                                 CWC01130
C     BUILT IN TRANSMISSION THRESHOLDS, WAVELENGTHS, XSCALE SLANTS                 CWC01140
      DATA TRSH /.05, .05, .05, .05/                                               CWC01150
      DATA WAVE /.55,1.06,3.5,10.6/,ISLANT/0,1,1,1/                                CWC01160
      ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)                                         
      TIME=10.                                                                     CWC01170
      X0=1000.                                                                     CWC01180
      H3=3.                                                                        CWC01190
      AST=0.                                                                       CWC01200
      DLS=90.                                                                      CWC01210
      ARE=1.                                                                       CWC01220
      FOG=0.                                                                       CWC01230
      S3=4.                                                                        CWC01240
      D0=0.                                                                        CWC01250
      PCAT=3.                                                                      CWC01260
      R0=50.                                                                       CWC01270
      T0=24.                                                                       CWC01280
      T1=12.94                                                                     CWC01290
      VS=10.                                                                       CWC01300
C READ DATA CARDS                                                                  CWC01310
      ICOU=0                                                                       CWC01320
10    READ (IOIN,900) IT(1),IT(2),(RV(I),I=1,7)                                    CWC01330
      ICOU=ICOU+1                                                                  CWC01340
      IF (IT(1).EQ.IR(1).AND.IT(2).EQ.IR(2)) GOTO 20                               CWC01350
      IF (IT(1).EQ.IR(3).AND.IT(2).EQ.IR(4)) GOTO 40                               CWC01360
      IF (IT(1).EQ.IR(5).AND.IT(2).EQ.IR(6)) GOTO 50                               CWC01370
      IF (IT(1).EQ.IR(7).AND.IT(2).EQ.IR(8)) GOTO 60                               CWC01380
      WRITE (IOOUT,901) IT(1),IT(2),(RV(I),I=1,7)                                  CWC01390
```

```
900    FORMAT(2A2,6X,7F10.3)                                                  CWC01400
901    FORMAT(1X,51HIN CWIC, THE FOLLOWING CARD DOES NOT CONFORM TO PRO,      CWC01410
      *15HPER CONVENTIONS/1X,2A2,6X,7F10.3)                                   CWC01420
       IF (ICOU.LE.4) GOTO 10                                                 CWC01430
       WRITE (IOOUT,902)                                                      CWC01440
902    FORMAT(1X,39HINVALID INPUTS TO CWIC. IERR=1 RETURNED)                  CWC01450
       IERR=1                                                                 CWC01460
       RETURN                                                                 CWC01470
20     TIME=RV(1)                                                             CWC01480
       X0=RV(2)                                                               CWC01490
       H3=RV(3)                                                               CWC01500
       AST=RV(4)                                                              CWC01510
       DLS=RV(5)                                                              CWC01520
       ARE=RV(6)                                                              CWC01530
       FOG=RV(7)                                                              CWC01540
       IF(IGEOSW.NE.1) GO TO 22                                               CWC01550
       DELX=PTS(1)-PTS(4)                                                     CWC01560
       DELY=PTS(2)-PTS(5)                                                     CWC01570
       DELZ=PTS(3)-PTS(6)                                                     CWC01580
       H3=SQRT(DELX**2+DELY**2+DELZ**2)                                       CWC01590
       HDIS=SQRT(DELX**2+DELY**2)                                             CWC01600
       RTDCON=57.29577951                                                     CWC01610
       AST=RTDCON*ACOS(HDIS/H3)                                               CWC01620
       IF(HDIS.GT.1.E-20) DLS=RTDCON*ACOS(DELY/HDIS)                          CWC01630
       IF(DELX.LT.0.) DLS=360.-DLS                                            CWC01640
   22  CONTINUE                                                               CWC01650
       GOTO 10                                                                CWC01660
40     S3=RV(1)                                                               CWC01670
       D0=RV(2)                                                               CWC01680
       PCAT=RV(3)                                                             CWC01690
       VS=RV(4)                                                               CWC01700
       R0=RV(5)                                                               CWC01710
       T0=RV(6)                                                               CWC01720
       T1=RV(7)                                                               CWC01730
       GOTO 10                                                                CWC01740
50     SLAT=RV(1)                                                             CWC01750
       SLONG=RV(2)                                                            CWC01760
       SJDATE=RV(3)                                                           CWC01770
       SZHOUR=RV(4)                                                           CWC01780
       C0=RV(5)                                                               CWC01790
       C1=RV(6)                                                               CWC01800
       GOTO 10                                                                CWC01810
C***   BEGIN COMPUTATIONS.  -- FIRST MET VALUES.                              CWC01820
60     IF (ICCLIM.EQ.0) GOTO 70                                               CWC01830
C*** USE CLIMAT PASSED VALUES:                                                CWC01840
       T0=TEMP                                                                CWC01850
       T1=DP                                                                  CWC01860
       R0=RH                                                                  CWC01870
       PCAT=FLOAT(IPASCT)                                                     CWC01880
       VS=VIS                                                                 CWC01890
       D0=WNDDIR                                                              CWC01900
       S3=WNDVEL                                                              CWC01910
C*** PROVIDE WINDSPEED IN KNOTS AS S0:                                        CWC01920
70     S0=S3/.515                                                             CWC01930
       IF (S0.LE.1.) S3=.515                                                  CWC01940
       IF (S0.LE.1.) S0=1.                                                    CWC01950
C*** NOW CHECK RH AND COMPUTE IF NECESSARY.                                   CWC01960
       IRNOT=0                                                                CWC01970
       IF (R0.GT.0.) GOTO 80                                                  CWC01980
       IF(T0 .GT. 0.) GO TO 76                                                CWC01990
       A0=9.5                                                                 CWC02000
       B0=265.5                                                               CWC02010
       IF(T0 .LE. 0.) GO TO 78                                                CWC02020
76     CONTINUE                                                               CWC02030
       A0=7.5                                                                 CWC02040
       B0=237.3                                                               CWC02050
78     CONTINUE                                                               CWC02060
       IF(T1 .GE. 0.) GO TO 79                                                CWC02070
       A1=9.5                                                                 CWC02080
       B1=265.5                                                               CWC02090
```

```
      IF(T1 .LE. 0.) GO TO 75                                                  CWC02100
79    CONTINUE                                                                 CWC02110
      A1=7.5                                                                   CWC02120
      B1=237.3                                                                 CWC02130
75    CONTINUE                                                                 CWC02140
      E0=6.11*10.**((A0*T0)/(B0+T0))                                           CWC02150
      E1=6.11*10.**((A1*T1)/(B1+T1))                                           CWC02160
      R0=(E1/E0)*100.                                                          CWC02170
      IRNOT=1                                                                  CWC02180
C*** NOW CHECK PASQUILL CATEGORY..                                             CWC02190
80    IP0=IFIX(PCAT+.001)                                                      CWC02200
      IF (IP0.GT.6) IP0=6                                                      CWC02210
      IPNOT=0                                                                  CWC02220
      IF (IP0.GT.0) GOTO 90                                                    CWC02230
      CALL CWIC1 (IP0,C0,C1,SLAT,SLONG,SJDATE,SZHOUR,S0)                       CWC02240
      IPNOT=1                                                                  CWC02250
C*** NEXT COMPUTE CL VALUE FOR THRESHOLD TRANSMITTANCE, CORRECTED FOR          CWC02260
C    FOG,RAIN,...                                                             CWC02270
C*** VISIBILITY EXTINCTION (KM-1) .. NOTE THAT IF VS=0., THEN SET TO           CWC02280
C                                   CLEAR DAY AND COMPUTATIONS CONTINUE.       CWC02290
90    EX55=0.                                                                  CWC02300
      IF (VS.GT.0.) EX55=3.912/VS                                              CWC02310
C*** TRANSMISSION THRESHOLDS..                                                 CWC02320
      DO 92 I=1,4                                                              CWC02330
      T(I)=TRSH(I)                                                             CWC02340
      IF (ICITAM.NE.0) T(I)=TFNL                                               CWC02350
      TR(I)=T(I)                                                               CWC02360
92    CONTINUE                                                                 CWC02370
C***  CORRECTIONS FOR WEATHER                                                  CWC02380
      IFOG=-1+IFIX(FOG+.0001)                                                  CWC02390
      IF (IFOG.LE.-1) GOTO 100                                                 CWC02400
C*** CORRECT VISIBLE FOR VISIBILITY (H3 SLANT RNG, AST ELEV. ANG.)             CWC02410
      XSTRN=0.                                                                 CWC02420
      EPTH=EX55*H3                                                             CWC02430
      IF (EPTH.LT.12.) XSTRN=EXP(-EPTH)                                        CWC02440
C SET VSET FOR NO CARD I/O IN XSCALE                                           CWC02450
      VSET=89.                                                                 CWC02460
      IF((AST.GT.0.1 .OR. AST.LT.-0.1).AND.IFOG.LT.5)                          CWC02470
     *CALL XSCALE(WAVE(1),VSET,EX55,XSTRN,IERR,ISLANT(1),IFOG,H3,AST)          CWC02480
      IF (XSTRN.LE.0.) XSTRN=.0001                                             CWC02500
      IF (XSTRN.GT.1.) XSTRN=1.                                                CWC02510
      T(1)=T(1)/XSTRN                                                          CWC02520
      IF (IFOG.EQ.0) GOTO 100                                                  CWC02530
C CORRECT NON-VISIBLE FOR SEEABILITY.                                          CWC02540
94    CONTINUE                                                                 CWC02550
      DO 96 I=2,4                                                              CWC02560
      ISLNT=ISLANT(I)                                                          CWC02570
      IF (AST.EQ.0.) ISLNT=0                                                   CWC02580
      IF (IFOG.GE.5) ISLNT=0
      CALL XSCALE(WAVE(I),VSET,EX55,XSTRN,IERR,ISLNT,IFOG,H3,AST)              CWC02590
      IF (XSTRN.LE.0.) XSTRN=.0001                                             CWC02610
      IF (XSTRN.GT.1.) XSTRN=1.                                                CWC02620
      T(I)=T(I)/XSTRN                                                          CWC02630
96    CONTINUE                                                                 CWC02640
C*** COMPUTE CL FOR THRESHOLD TRANSMISSION REQUIRED OF SMOKE.                  CWC02650
100   DO 108 I=1,4                                                            CWC02660
      IF (T(I).GT.1.) T(I)=1.                                                  CWC02670
      IF (T(I).LE.0.) T(I)=.00001                                              CWC02680
      DO 105 J=1,2                                                             CWC02690
      C(I,J)=ALOG(T(I))/(-CS(I,J))                                             CWC02700
105   CONTINUE                                                                 CWC02710
108   CONTINUE                                                                 CWC02720
C*** ATMOSPHERIC DIFFUSION                                                     CWC02730
      CALL CWIC3 (ARE,DLS,C2,D0,D2,H,IP0,R0,R2,U,V,Y1)                         CWC02740
C*** MUNITIONS EXPENDITURES                                                    CWC02750
      CALL CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0)              CWC02760
C** END MAIN CWIC COMPUTATIONS,  FINAL OUTPUT.                                 CWC02770
      IF(ICITAM.GT.0) WRITE(IOOUT,10000)                                       CWC02780
      WRITE (IOOUT,10200)                                                      CWC02790
      WRITE(IOOUT,10800)                                                       CWC02800
```

```
      WRITE (IOOUT,10140)                                                    CWC02810
      WRITE(IOOUT,10300)                                                     CWC02820
      WRITE(IOOUT,10400)                                                     CWC02830
      WRITE(IOOUT,10200)                                                     CWC02840
      WRITE (IOOUT,14000) H3                                                 CWC02850
      WRITE (IOOUT,14100) AST                                                CWC02860
      WRITE (IOOUT,14200) DLS                                                CWC02870
      WRITE (IOOUT,11900) ARE                                                CWC02880
      WRITE (IOOUT,11400)                                                    CWC02890
      IAD=1                                                                  CWC02900
      IF (IFOG.GE.0) IAD=3                                                   CWC02910
      WRITE (IOOUT,14300) JANS(IAD),JANS(IAD+1)                              CWC02920
      IAD=1                                                                  CWC02930
      IF (IFOG.EQ.1) IAD=3                                                   CWC02940
      WRITE (IOOUT,14400) JANS(IAD),JANS(IAD+1)                              CWC02950
      IAD=1                                                                  CWC02960
      IF (IFOG.EQ.2) IAD=3                                                   CWC02970
      WRITE (IOOUT,14500) JANS(IAD),JANS(IAD+1)                              CWC02980
      IAD=1                                                                  CWC02990
      IF (IFOG.EQ.3) IAD=3                                                   CWC03000
      WRITE (IOOUT,14600) JANS(IAD),JANS(IAD+1)                              CWC03010
      IAD=1                                                                  CWC03020
      IF (IFOG.EQ.4) IAD=3                                                   CWC03030
      WRITE (IOOUT,14700) JANS(IAD),JANS(IAD+1)                              CWC03040
      IAD=1                                                                  CWC03050
      IF (IFOG.EQ.5) IAD=3                                                   CWC03060
      WRITE (IOOUT,14800) JANS(IAD),JANS(IAD+1)                              CWC03070
      WRITE (IOOUT,15000)                                                    CWC03080
      WRITE (IOOUT,11800) S3                                                 CWC03090
      WRITE (IOOUT,11700) D0                                                 CWC03100
      JP=JPASCT(IP0)                                                         CWC03110
      WRITE (IOOUT,12000) JP                                                 CWC03120
      WRITE (IOOUT,11300) VS                                                 CWC03130
      WRITE (IOOUT,12100) R0                                                 CWC03140
      IF (IRNOT.EQ.1) WRITE (IOOUT,11500) T0                                 CWC03150
      IF (IRNOT.EQ.1) WRITE (IOOUT,11600) T1                                 CWC03160
      IF (IPNOT.EQ.0) GOTO 120                                               CWC03170
      WRITE (IOOUT,15100)                                                    CWC03180
      IF (SLAT.GE.0.) WRITE(IOOUT,10601) SLAT                                CWC03190
      SLATI=-SLAT                                                            CWC03200
      IF (SLAT.LT.0.) WRITE(IOOUT,10602) SLATI                               CWC03210
      IF (SLONG.GE.0.) WRITE(IOOUT,10701) SLONG                              CWC03220
      SLONGI=-SLONG                                                          CWC03230
      IF (SLONG.LT.0.) WRITE(IOOUT,10702) SLONGI                             CWC03240
      WRITE(IOOUT,10900) SJDATE                                              CWC03250
      WRITE(IOOUT,11000) SZHOUR                                              CWC03260
      WRITE(IOOUT,11100) C0                                                  CWC03270
      WRITE(IOOUT,11200) C1                                                  CWC03280
120   CONTINUE                                                               CWC03290
      WRITE (IOOUT,15200)                                                    CWC03300
      DO 150 I=1,4                                                           CWC03310
      IWL=4*(I-1)                                                            CWC03320
      WRITE (IOOUT,15300) (LNGTH(IWL+J),J=1,4),TR(I),T(I)                    CWC03330
150   CONTINUE                                                               CWC03340
      WRITE(IOOUT,10000)                                                     CWC03350
C* PRINT MUNITION EXPENDITURES                                               CWC03360
      WRITE(IOOUT,10130)                                                     CWC03370
      WRITE(IOOUT,16200) (LNGTH(J),J=1,8)                                    CWC03380
      WRITE(IOOUT,10100)                                                     CWC03390
      WRITE(IOOUT,16300)                                                     CWC03400
      WRITE(IOOUT,16400) X0,TIME,X0,TIME                                     CWC03410
      WRITE(IOOUT,10200)                                                     CWC03420
      WRITE(IOOUT,16500) ISMOKE(1),ISMOKE(1)                                 CWC03430
      WRITE(IOOUT,10100)                                                     CWC03440
      WRITE(IOOUT,16600) IGUN(1),IGUN(2),IGUN(1),IGUN(2)                     CWC03450
      WRITE(IOOUT,10100)                                                     CWC03460
      WRITE(IOOUT,16700)                                                     CWC03470
      WRITE(IOOUT,16800)                                                     CWC03480
      WRITE(IOOUT,16900) Q1(1,1,1),Y(1,1,1),Q1(2,1,1),Y(2,1,1)               CWC03490
      WRITE(IOOUT,17000) Q1(1,2,1),R(1,1),Y(1,2,1),P(1,1,1),Q1(2,2,1),       CWC03500
```

```
     *R(2,1),Y(2,2,1),P(2,1,1)                                                  CWC03510
      WRITE(IOOUT,10200)                                                        CWC03520
      WRITE(IOOUT,16600) IGUN(3),IGUN(4),IGUN(3),IGUN(4)                        CWC03530
      WRITE(IOOUT,10100)                                                        CWC03540
      WRITE(IOOUT,16700)                                                        CWC03550
      WRITE(IOOUT,16800)                                                        CWC03560
      WRITE(IOOUT,16900) Q1(1,1,2),Y(1,1,2),Q1(2,1,2),Y(2,1,2)                  CWC03570
      WRITE(IOOUT,17000) Q1(1,2,2),R(1,1),Y(1,2,2),P(1,2,1),Q1(2,2,2),          CWC03580
     *R(2,1),Y(2,2,2),P(2,2,1)                                                  CWC03590
      WRITE(IOOUT,10200)                                                        CWC03600
      WRITE(IOOUT,16500) ISMOKE(2),ISMOKE(2)                                    CWC03610
      WRITE(IOOUT,10100)                                                        CWC03620
      WRITE(IOOUT,16600) IGUN(1),IGUN(2),IGUN(1),IGUN(2)                        CWC03630
      WRITE(IOOUT,10100)                                                        CWC03640
      WRITE(IOOUT,16700)                                                        CWC03650
      WRITE(IOOUT,16800)                                                        CWC03660
      WRITE(IOOUT,16900) Q2(1,1,1),Z(1,1,1),Q2(2,1,1),Z(2,1,1)                  CWC03670
      IF(IP0 .GT. 4) R(1,2)=2.                                                  CWC03680
      IF(IP0 .GT. 4) R(2,2)=2.                                                  CWC03690
      WRITE(IOOUT,17000) Q2(1,2,1),R(1,2),Z(1,1,1),P(1,1,2),Q2(2,2,1),          CWC03700
     *R(2,2),Z(2,1,1),P(2,1,2)                                                  CWC03710
      WRITE(IOOUT,10200)                                                        CWC03720
      WRITE(IOOUT,16600) IGUN(3),IGUN(4),IGUN(3),IGUN(4)                        CWC03730
      WRITE(IOOUT,10100)                                                        CWC03740
      WRITE(IOOUT,16700)                                                        CWC03750
      WRITE(IOOUT,16800)                                                        CWC03760
      WRITE(IOOUT,16900) Q2(1,1,2),Z(1,2,2),Q2(2,1,2),Z(2,2,2)                  CWC03770
      IF(IP0 .GT. 4) R(1,2)=1.                                                  CWC03780
      IF(IP0 .GT. 4) R(2,2)=1.                                                  CWC03790
      WRITE(IOOUT,17000) Q2(1,2,2),R(1,2),Z(1,2,2),P(1,2,2),Q2(1,2,2),          CWC03800
     *R(2,2),Z(2,2,2),P(2,2,2)                                                  CWC03810
      WRITE(IOOUT,10000)                                                        CWC03820
      WRITE (IOOUT,10130)                                                       CWC03830
      DO 7100 I=3,4                                                             CWC03840
      IWL=4*(I-1)                                                               CWC03850
      J1=IWL+1                                                                  CWC03860
      J2=IWL+4                                                                  CWC03870
      WRITE (IOOUT,12200) (LNGTH(J),J=J1,J2)                                    CWC03880
      WRITE (IOOUT,10100)                                                       CWC03890
      WRITE (IOOUT,12300)                                                       CWC03900
      WRITE (IOOUT,12400) X0,TIME                                               CWC03910
      WRITE (IOOUT,10200)                                                       CWC03920
      WRITE(IOOUT,12500) ISMOKE(2)                                              CWC03930
      WRITE(IOOUT,10100)                                                        CWC03940
      WRITE(IOOUT,13100)                                                        CWC03950
      WRITE(IOOUT,13200)                                                        CWC03960
      WRITE(IOOUT,13300) IGUN(1),IGUN(2),Q2(I,2,1),R(I,2),P(I,1,2)              CWC03970
      WRITE(IOOUT,13300) IGUN(3),IGUN(4),Q2(I,2,2),R(I,2),P(I,2,2)              CWC03980
      IF (I.EQ.3) WRITE(IOOUT,10140)                                            CWC03990
      IF (I.EQ.4) WRITE(IOOUT,10000)                                            CWC04000
7100  CONTINUE                                                                  CWC04010
      RETURN                                                                    CWC04020
C* FORMAT STATEMENTS.                                                          CWC04030
10000 FORMAT(1H1)                                                               CWC04040
10100 FORMAT(1H )                                                               CWC04050
10200 FORMAT(1H0)                                                               CWC04060
10130 FORMAT(///)                                                               CWC04070
10140 FORMAT(////)                                                              CWC04080
10300 FORMAT(55X,21HMUNITION EXPENDITURES)                                      CWC04090
10400 FORMAT(56X,19HFOR HC AND WP SMOKE)                                        CWC04100
10601 FORMAT(45X,35HLATITUDE                - DEG       =,F7.2,6H NORTH)      CWC04110
10602 FORMAT(45X,35HLATITUDE                - DEG       =,F7.2,6H SOUTH)      CWC04120
10701 FORMAT(45X,35HLONGITUDE               - DEG       =,F7.2,6H EAST )      CWC04130
10702 FORMAT(45X,35HLONGITUDE               - DEG       =,F7.2,6H WEST )      CWC04140
10800 FORMAT(1H0,47X,36(1H*)/48X,1H*,34X,1H*/48X,1H*,4X,                        CWC04150
     *26HCWIC MUNITION EXPENDITURES,4X,1H*/48X,1H*,34X,1H*/                     CWC04160
     *48X,36(1H*))                                                              CWC04170
10900 FORMAT(45X,35HJULIAN DATE             - DAY       =,F7.0)               CWC04180
11000 FORMAT(45X,35HGMT TIME                - HOUR      =,F7.2)               CWC04190
11100 FORMAT(45X,35HCEILING                 - METERS    =,F7.1)               CWC04200
```

```
11200 FORMAT(45X,35HCLOUD COVER                - PERCENT  =,F7.1)                CWC04210
11300 FORMAT(45X,35HVISIBILITY                 - KM       =,F7.3)                CWC04220
11400 FORMAT(1H0,44X,35HATMOSPHERIC EXTINCTION CORRECTIONS )                     CWC04230
11500 FORMAT(45X,35HTEMPERATURE                - DEG C    =,F7.1)                CWC04240
11600 FORMAT(45X,35HDEW POINT                  - DEG C    =,F7.1)                CWC04250
11700 FORMAT(45X,35HWIND DIRECTION             - DEG      =,F7.2)                CWC04260
11800 FORMAT(45X,35HWINDSPEED                  - M/SEC    =,F7.2)                CWC04270
11900 FORMAT(45X,35HAVG ROUGHNESS ELEMENT      - CM       =,F7.1)                CWC04280
12000 FORMAT(45X,35HPASQUILL CATEGORY          -          =,5X,A2)               CWC04290
12100 FORMAT(45X,35HRELATIVE HUMIDITY          - PERCENT  =,F7.1)                CWC04300
12200 FORMAT(60X,11(1H-)/60X,1H-,1X,4A2,1H-/60X,11(1H-))                         CWC04310
12300 FORMAT(62X,6HLENGTH,8X,8HDURATION/62X,6HMETERS,9X,7HMINUTES)               CWC04320
12400 FORMAT(47X,6HSCREEN,8X,F7.0,9X,F7.2)                                       CWC04330
12500 FORMAT(59X,A2,13H SMOKE SCREEN/59X,15(1H-))                                CWC04340
12600 FORMAT(59X,A2,A1,11HMM HOWITZER)                                           CWC04350
12700 FORMAT(47X,37HVOLLEY       GUNS RATE  SPACING ROUNDS)                      CWC04360
12800 FORMAT(47X,30H                 /MIN   METERS)                              CWC04370
12900 FORMAT(47X,11HINITIAL:    ,F5.0,6X,F8.0)                                   CWC04380
13000 FORMAT(47X,11HSUSTAINING:,F5.0,F5.1,F9.0,F7.0)                             CWC04390
13100 FORMAT(47X,31H         ROUNDS/   RATE/   TOTAL)                           CWC04400
13200 FORMAT(47X,32H       60 METERS  MINUTE  ROUNDS)                          CWC04410
13300 FORMAT(47X,A2,A1,5HMM:   ,F5.0,5X,F4.0,3X,F7.0)                            CWC04420
14000 FORMAT(45X,35HSLANT RANGE OBS-TGT    - KM       =,F7.3)                    CWC04430
14100 FORMAT(45X,35HELEVATION OF TARGET    - DEG      =,F7.2)                    CWC04440
14200 FORMAT(45X,35HAZIMUTH OF TARGET      - DEG      =,F7.2)                    CWC04450
14300 FORMAT(45X,35HCORRECTED FOR VISIBILITY   -        ,4X,2A2)                CWC04460
14400 FORMAT(45X,35HMARITIME ARCTIC AIR MASS   -        ,4X,2A2)                CWC04470
14500 FORMAT(45X,35HMARITIME POLAR AIR MASS    -        ,4X,2A2)                CWC04480
14600 FORMAT(45X,35HCONTINENTAL POLAR AIR MASS -        ,4X,2A2)                CWC04490
14700 FORMAT(45X,35HCORRECTED FOR RAIN         -        ,4X,2A2)                CWC04500
14800 FORMAT(45X,35HCORRECTED FOR SNOW         -        ,4X,2A2)                CWC04510
15000 FORMAT(1H0,44X,35HMETEOROLOGICAL INPUTS                )                   CWC04520
15100 FORMAT(1H0,44X,35HINPUTS FOR PASQUILL CATEGORY         )                   CWC04530
15200 FORMAT(1H0,44X,42HTRANSMISSION THRESHOLDS   TOTAL       SMOKE)             CWC04540
15300 FORMAT(45X,4A2,18X,F5.3,6X,F5.3)                                           CWC04550
16200 FORMAT(33X,12(1H-),44X,12(1H-)/33X,1H-,1X,4A2,1X,1H-,44X,1H-,1X,           CWC04560
     *4A2,1X,1H-/33X,12(1H-),44X,12(1H-))                                        CWC04570
16300 FORMAT(1X,2(34X,6HLENGTH,8X,8HDURATION)/1X,2(34X,6HMETERS,9X,              CWC04580
     *7HMINUTES))                                                                CWC04590
16400 FORMAT(1X,2(19X,6HSCREEN,8X,F7.0,9X,F7.2))                                 CWC04600
16500 FORMAT(32X,A2,13H SMOKE SCREEN,41X,A2,13H SMOKE SCREEN/32X,                CWC04610
     *15(1H-),41X,15(1H-))                                                       CWC04620
16600 FORMAT(32X,A2,A1,11HMM HOWITZER,42X,A2,A1,11HMM HOWITZER)                  CWC04630
16700 FORMAT(1X,2(19X,6HVOLLEY,6X,25HGUNS RATE  SPACING ROUNDS))                 CWC04640
16800 FORMAT(37X,4H/MIN,3X,6HMETERS,43X,4H/MIN,3X,6HMETERS)                      CWC04650
16900 FORMAT(20X,11HINITIAL:    ,F5.0,6X,F8.0,26X,11HINITIAL:    ,               CWC04660
     *F5.0,6X,F8.0)                                                              CWC04670
17000 FORMAT(1X,2(19X,11HSUSTAINING:,F5.0,F5.1,F9.0,F7.0))                       CWC04680
      END                                                                        CWC04690
```

```
      SUBROUTINE CWIC1 (IP0,C0,C1,SLAT,SLONG,SJDATE,SZHOUR,S0)          CWX00010
      DIMENSION ITAB(7,9)                                               CWX00020
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                    CWX00030
      DATA ITAB /                                                       CWX00040
     1        1,1,2,3,4,6,6,                                           CWX00050
     2        1,2,2,3,4,6,6,                                           CWX00060
     3        1,2,3,4,4,5,6,                                           CWX00070
     4        2,2,3,4,4,5,6,                                           CWX00080
     5        2,2,3,4,4,4,5,                                           CWX00090
     6        2,3,3,4,4,4,5,                                           CWX00100
     7        3,3,4,4,4,4,5,                                           CWX00110
     8        3,3,4,4,4,4,4,                                           CWX00120
     9        3,4,4,4,4,4,4/                                           CWX00130
      ASIN(ARG)=ATAN2(ARG,SQRT(1.-ARG**2))
C*METEOROLOGICAL CALCULATIONS.                                          CWX00140
C                                                                       CWX00150
      IF(C1 .NE. 100.) GO TO 1000                                       CWX00160
      IF(C0 .GT. 2133.6042) GO TO 1000                                  CWX00170
      I1=0                                                              CWX00180
      I2=0                                                              CWX00190
      GO TO 2300                                                        CWX00200
1000  CONTINUE                                                          CWX00210
C CALCULATE ANGULAR FRACTION OF A YEAR FOR A GIVEN JULIAN DATE.         CWX00220
      R9=PIRAD                                                          CWX00230
      D9=1./R9                                                          CWX00240
      SLAT1=SLAT*R9                                                     CWX00250
      A0=((SJDATE-1.)*360.)/365.242                                     CWX00260
C CALCULATE SOLAR DECLINATION ANGLE (A4).                               CWX00270
      A1=A0*R9                                                          CWX00280
      A2=279.9348+A0                                                    CWX00290
      A2=A2+(1.914827*SIN(A1))-(0.079525*COS(A1))                       CWX00300
      A2=A2+(0.019938*SIN(2*A1))-(0.00162*COS(2*A1))                    CWX00310
      A2=A2*R9                                                          CWX00320
      A3=23.4438*R9                                                     CWX00330
      A4=SIN(A3)*SIN(A2)                                                CWX00340
      A4=ASIN(A4)                                                       CWX00350
C CALCULATE THE TIME OF MERIDIAN PASSAGE - TRUE SOLAR NOON (A5).        CWX00360
      A5= 12.+(0.12357*SIN(A1))-(0.004289*COS(A1))                      CWX00370
      A5=A5+(0.153809*SIN(2*A1))+(0.060783*COS(2*A1))                   CWX00380
C CALCULATE SOLAR HOUR ANGLE (A6)*** NOTE THIS VERSION USES + SIGN      CWX00390
C ON SLONG DUE TO EAST-LONGITUDE POSITIVE CONVENTION.                   CWX00400
      A6=15.*(SZHOUR-A5)+SLONG                                          CWX00410
      A6=A6*R9                                                          CWX00420
C CALCULATE SOLAR ALTITUDE (A7).                                        CWX00430
      A7=SIN(SLAT1)*SIN(A4)+COS(SLAT1)*COS(A4)*COS(A6)                  CWX00440
      A7=ASIN(A7)                                                       CWX00450
1100  CONTINUE                                                          CWX00460
      A7=A7*D9                                                          CWX00470
C CALCULATE INSOLATION CLASS NUMBER.                                    CWX00480
      I2=0                                                              CWX00490
      IF(A7 .LE. 60.) GO TO 1200                                        CWX00500
      I2=4                                                              CWX00510
      GO TO 1500                                                        CWX00520
1200  CONTINUE                                                          CWX00530
      IF(A7 .LE. 35.) GO TO 1300                                        CWX00540
      I2=3                                                              CWX00550
      GO TO 1500                                                        CWX00560
1300  CONTINUE                                                          CWX00570
      IF(A7 .LE. 15.) GO TO 1400                                        CWX00580
      I2=2                                                              CWX00590
      GO TO 1500                                                        CWX00600
1400  CONTINUE                                                          CWX00610
      IF(A7 .LE. 0.) GO TO 2200                                         CWX00620
      I2=1                                                              CWX00630
C CALCULATE NET RADIATION INDEX FOR DAYTIME.                            CWX00640
1500  CONTINUE                                                          CWX00650
      I3=0                                                              CWX00660
      IF(C1 .GT. 50.) GO TO 1600                                        CWX00670
      I3=I2                                                             CWX00680
      GO TO 1900                                                        CWX00690
```

```
1600  CONTINUE                                                          CWX00700
      IF(C0 .GT. 2133.6042) GO TO 1700                                  CWX00710
      I3=I2-2                                                           CWX00720
      GO TO 1900                                                        CWX00730
1700  CONTINUE                                                          CWX00740
      IF(C0 .GE. 4876.8096) GO TO 1800                                  CWX00750
      I3=I2-1                                                           CWX00760
      GO TO 1900                                                        CWX00770
1800  CONTINUE                                                          CWX00780
      IF(C1 .NE. 100.) GO TO 1900                                       CWX00790
      I3=I2-1                                                           CWX00800
1900  CONTINUE                                                          CWX00810
      IF(I3 .NE. 0) GO TO 2000                                          CWX00820
      I3=I2                                                             CWX00830
2000  CONTINUE                                                          CWX00840
      IF(I3 .GT. 1) GO TO 2100                                          CWX00850
      I3=1                                                              CWX00860
2100  CONTINUE                                                          CWX00870
      I1=I3                                                             CWX00880
      GO TO 2300                                                        CWX00890
C COMPUTE NET RADIATION INDEX FOR NIGHTTIME.                            CWX00900
2200  CONTINUE                                                          CWX00910
      IF(C1 .GT. 40.) GO TO 2250                                        CWX00920
      I1=-2                                                             CWX00930
      GO TO 2300                                                        CWX00940
2250  CONTINUE                                                          CWX00950
      I1=-1                                                             CWX00960
C CALCULATE PASQUILL STABILITY CATAGORY.                                CWX00970
2300  CONTINUE                                                          CWX00980
      I4=0                                                              CWX00990
      I5=0                                                              CWX01000
      IF(I1 .NE. 4) GO TO 2400                                          CWX01010
      I4=1                                                              CWX01020
2400  CONTINUE                                                          CWX01030
      IF(I1 .NE. 3) GO TO 2420                                          CWX01040
      I4=2                                                              CWX01050
2420  CONTINUE                                                          CWX01060
      IF(I1 .NE. 2 ) GO TO 2440                                         CWX01070
      I4=3                                                              CWX01080
2440  CONTINUE                                                          CWX01090
      IF(I1 .NE. 1) GO TO 2460                                          CWX01100
      I4=4                                                              CWX01110
2460  CONTINUE                                                          CWX01120
      IF(I1 .NE. 0) GO TO 2480                                          CWX01130
      I4=5                                                              CWX01140
2480  CONTINUE                                                          CWX01150
      IF(I1 .NE. -1) GO TO 2500                                         CWX01160
      I4=6                                                              CWX01170
2500  CONTINUE                                                          CWX01180
      IF(I1 .NE. -2) GO TO 2520                                         CWX01190
      I4=7                                                              CWX01200
2520  CONTINUE                                                          CWX01210
      IF(S0 .GE. 2.) GO TO 2540                                         CWX01220
      I5=1                                                              CWX01230
      GO TO 2700                                                        CWX01240
2540  CONTINUE                                                          CWX01250
      IF(S0 .GE. 4.) GO TO 2560                                         CWX01260
      I5=2                                                              CWX01270
      GO TO 2700                                                        CWX01280
2560  CONTINUE                                                          CWX01290
      IF(S0 .GE. 6.) GO TO 2580                                         CWX01300
      I5=3                                                              CWX01310
      GO TO 2700                                                        CWX01320
2580  CONTINUE                                                          CWX01330
      IF(S0 .GE.  7.) GO TO 2600                                        CWX01340
      I5=4                                                              CWX01350
      GO TO 2700                                                        CWX01360
2600  CONTINUE                                                          CWX01370
      IF(S0 .GE.  8.) GO TO 2620                                        CWX01380
      I5=5                                                              CWX01390
```

```
      GO TO 2700                                                        CWX01400
2620  CONTINUE                                                          CWX01410
      IF(S0 .GE. 10.) GO TO 2640                                        CWX01420
      I5=6                                                              CWX01430
      GO TO 2700                                                        CWX01440
2640  CONTINUE                                                          CWX01450
      IF(S0 .GE. 11.) GO TO 2660                                        CWX01460
      I5=7                                                              CWX01470
      GO TO 2700                                                        CWX01480
2660  CONTINUE                                                          CWX01490
      IF(S0 .GE. 12.) GO TO 2680                                        CWX01500
      I5=8                                                              CWX01510
      GO TO 2700                                                        CWX01520
2680  CONTINUE                                                          CWX01530
      I5=9                                                              CWX01540
2700  CONTINUE                                                          CWX01550
      IP0=ITAB(I4,I5)                                                   CWX01560
      RETURN                                                            CWX01570
      END                                                               CWX01580
```

```
      SUBROUTINE CWIC3 (ARE,DLS,C2,D0,D2,H,IP0,R0,R2,U V,Y1)                CWY00010
      DIMENSION A(6),S(6,3),D(6,3),H(2,2),U(2,2),V(2)                      CWY00020
      DIMENSION W(6)                                                       CWY00030
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                       CWY00040
      DATA A/0.4,0.32,0.22,0.144,0.102,0.076/                              CWY00050
      DATA S /                                                             CWY00060
     *.139085297,.122097643,.110104377,.097649832,.070772166,.055487093,CWY00070
     *.015017284,.010970370,.010962963,.010418519,7.27284E-3,6.55309E-3,CWY00080
     *-1.02581E-4,-6.80135E-5,-6.73401E-5,-6.83502E-5,-4.50056E-5,        CWY00090
     *-4.01796E-5/                                                        CWY00100
      DATA D /                                                            CWY00110
     *.944814815,.894803591,.854792368,.816026936,.786026936,.726015713,CWY00120
     *-4.85185E-3,-4.83951E-3,-4.82716E-3,-6.07407E-3,-6.07407E-3,        CWY00130
     *-6.06173E-3 ,                                                       CWY00140
     *3.7037E-5,3.59147E-5,3.47924E-5,4.7138E-5,4.7138E-5,4.60157E-5/      CWY00150
      DATA W/0.016,0.016,0.016,0.016,0.016,0.016/                         CWY00160
C*ATMOSPHERIC DIFFUSION CALCULATIONS.                                      CWY00170
      A1=-1.24+1.19*ALOG10(ARE)                                           CWY00180
      Z1=10.**A1                                                          CWY00190
      A2=ABS(DLS-D0)*(PI/180.)                                            CWY00200
      R2=SQRT(13.69/(13.69*SIN(A2)*SIN(A2)+COS(A2)*COS(A2)))              CWY00210
      Y1=1.09521547+(0.02906894*R0)-(4.9575E-04*R0*R0)+                   CWY00220
     2               (4.82E-06*R0*R0*R0)                                  CWY00230
      Y2=3.364059144+(0.06050271*R0)-(1.15301E-03*R0*R0)+                 CWY00240
     2               (1.33942E-05*R0*R0*R0)                               CWY00250
      C2=S(IP0,1)+S(IP0,2)*Z1+S(IP0,3)*Z1**2                              CWY00260
      D1=D(IP0,1)+D(IP0,2)*Z1+D(IP0,3)*Z1**2                              CWY00270
      D2=1/D1                                                             CWY00280
      DO 5400 I=1,4                                                       CWY00290
C*CALCULATE CROSSWIND INTEGRATED CONCENTRATION FOR WP SMOKE.              CWY00300
      DO 5300 K=1,2                                                       CWY00310
      IF (I .LT. 3 .AND. IP0 .GT. 4) GOTO 5300                            CWY00320
      S1=U(K,1)+0.74*A(IP0)*100.**0.9                                     CWY00330
      S2=U(K,2)+0.667*C2*100.**D1                                         CWY00340
      V(K)=(W(IP0)*Y2*H(K,2))/(PI*S1*S2)                                  CWY00350
                                                                          CWY00360
5300  CONTINUE                                                            CWY00370
5400  CONTINUE                                                            CWY00380
      RETURN                                                              CWY00390
      END
```

```
      SUBROUTINE CWIC4 (C,C2,D2,H,P,Q1,Q2,R,R2,S3,TIME,V,X0,Y,Y1,Z,IP0) CWZ00010
      REAL ME(2)                                                       CWZ00020
      DIMENSION C(4,2),H(2,2),P(4,2,2),Q1(4,2,2),Q2(4,2,2),R(4,2),V(2), CWZ00030
     *Y(4,2,2),Z(4,2,2)                                                CWZ00040
      DATA ME/0.4,0.4/                                                 CWZ00050
C*MUNITIONS EXPENDITURES.                                              CWZ00060
      DO 6900 I=1,4                                                    CWZ00070
C* CALCULATE INITIAL SHELL SPACING FOR HC SMOKE                        CWZ00080
      DO 6800 K=1,2                                                    CWZ00090
      IF(I .GT. 2) GO TO 6100                                          CWZ00100
      Y(I,1,K)=45.*S3                                                  CWZ00110
C* CALCULATE SUSTAINING SHELL SPACING FOR HC SMOKE                     CWZ00120
      IF(C(I,1) .NE. 0.) GO TO 5500                                    CWZ00130
      Y(I,2,K)=0.                                                      CWZ00140
      GO TO 5600                                                       CWZ00150
5500  CONTINUE                                                         CWZ00160
      Y(I,2,K)=1/R2*((0.731*ME(K)*Y1*H(K,1))/(C2*S3*C(I,1)))**D2       CWZ00170
      IF(Y(I,2,K) .GT. X0) Y(I,2,K)=X0                                 CWZ00180
5600  CONTINUE                                                         CWZ00190
      IF(Y(I,2,K) .NE. 0.) GO TO 5700                                  CWZ00200
      Q1(I,1,K)=1.                                                     CWZ00210
      Q1(I,2,K)=1.                                                     CWZ00220
      GO TO 5900                                                       CWZ00230
5700  CONTINUE                                                         CWZ00240
C* CALCULATE INITIAL VOLLEY FOR HC SMOKE                              CWZ00250
      IF(Y(I,1,K) .GT. Y(I,2,K)) Y(I,1,K)=Y(I,2,K)                     CWZ00260
      Q1(I,1,K)=X0/Y(I,1,K)                                            CWZ00270
      Q5=AINT(Q1(I,1,K))                                               CWZ00280
      Q6=Q1(I,1,K)-Q5                                                  CWZ00290
      IF(Q6 .EQ. 0.) GO TO 5800                                        CWZ00300
      Q1(I,1,K)=Q5+1.                                                  CWZ00310
5800  CONTINUE                                                         CWZ00320
C* CALCULATE NUMBER OF GUNS FOR SUSTAINING VOLLEYS (HC)               CWZ00330
      Q1(I,2,K)=X0/Y(I,2,K)                                            CWZ00340
      Q5=AINT(Q1(I,2,K))                                               CWZ00350
      Q6=Q1(I,2,K)-Q5                                                  CWZ00360
      IF(Q6 .EQ. 0.) GO TO 5900                                        CWZ00370
      Q1(I,2,K)=Q5+1.                                                  CWZ00380
5900  CONTINUE                                                         CWZ00390
C* CALCULATE RATE OF FIRE FOR HC SMOKE                                 CWZ00400
      R(I,1)=0.5                                                       CWZ00410
      IF(C(I,1) .NE. 0.) GO TO 6000                                    CWZ00420
      R(I,1)=0.                                                        CWZ00430
6000  CONTINUE                                                         CWZ00440
C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (HC SMOKE)                CWZ00450
      P(I,K,1)=Q1(I,1,K)+(0.5*Q1(I,2,K)*(TIME-2.))                     CWZ00460
      Q5=AINT(P(I,K,1))                                                CWZ00470
      Q6=P(I,K,1)-Q5                                                   CWZ00480
      IF(Q6 .EQ. 0.) GO TO 6100                                        CWZ00490
      P(I,K,1)=Q5+1.                                                   CWZ00500
6100  CONTINUE                                                         CWZ00510
      IF(I .LT. 3 .AND. IP0 .GT.4) GO TO 6775                          CWZ00520
C* SHELL SPACING (Z( )) AND VOLLEYS (Q( )) - WP SMOKE                 CWZ00530
      IF(C(I,2) .NE. 0.) GO TO 6200                                    CWZ00540
      Z(I,1,K)=0.                                                      CWZ00550
      Z(I,2,K)=0.                                                      CWZ00560
      Q2(I,1,K)=0.                                                     CWZ00570
      Q2(I,2,K)=0.                                                     CWZ00580
      GO TO 6400                                                       CWZ00590
6200  CONTINUE                                                         CWZ00600
      IF(I .LT. 3) GO TO 6250                                          CWZ00610
      IF (I .GT. 2) Q2(I,1,K)=0.6*C(I,2)/V(K)                          CWZ00620
      GO TO 6300                                                       CWZ00630
 6250 CONTINUE                                                         CWZ00640
      Z(I,1,K)=V(K)/C(I,2)*100.                                        CWZ00650
      Z(I,2,K)=Z(I,1,K)                                                CWZ00660
      Q2(I,1,K)=X0/Z(I,2,K)+1.                                         CWZ00670
6300  CONTINUE                                                         CWZ00680
      Q5=AINT(Q2(I,1,K))                                               CWZ00690
      Q6=Q2(I,1,K)-Q5                                                  CWZ00700
```

```
      IF (Q6 .EQ. 0.) GO TO 6350                                        CWZ00710
      Q2(I,1,K)=Q5+1.                                                   CWZ00720
6350  CONTINUE                                                          CWZ00730
      Q2(I,2,K)=Q2(I,1,K)                                               CWZ00740
6400  CONTINUE                                                          CWZ00750
C* RATE OF FIRE FOR WP SMOKE                                           CWZ00760
      IF(C(I,2) .NE. 0.) GO TO 6425                                     CWZ00770
      R(I,2)=0.                                                         CWZ00780
      GO TO 6600                                                        CWZ00790
6425  CONTINUE                                                          CWZ00800
      IF (I .GT. 2) GOTO 6450                                           CWZ00810
      R(I,2)=(Z(I,2,K)+60.)/S3                                          CWZ00820
      GO TO 6475                                                        CWZ00830
6450  CONTINUE                                                          CWZ00840
      R(I,2)=120./S3                                                    CWZ00850
6475  CONTINUE                                                          CWZ00860
      R(I,2)=R(I,2)/20.                                                 CWZ00870
      R5=AINT(R(I,2))                                                   CWZ00880
      R6=R(I,2)-R5                                                      CWZ00890
      IF(R6 .LT. 0.5) GO TO 6500                                        CWZ00900
      R5=R5+1.                                                          CWZ00910
6500  CONTINUE                                                          CWZ00920
      IF(R5 .NE. 0.) GO TO 6550                                         CWZ00930
      R5=1.                                                             CWZ00940
6550  CONTINUE                                                          CWZ00950
      R(I,2)=R5*20./60.                                                 CWZ00960
6600  CONTINUE                                                          CWZ00970
C* CALCULATE TOTAL NUMBER OF ROUNDS REQUIRED (WP)                       CWZ00980
      IF(C(I,2) .NE. 0.) GO TO 6650                                     CWZ00990
      P(I,K,2)=0.                                                       CWZ01000
      GO TO 6700                                                        CWZ01010
6650  CONTINUE                                                          CWZ01020
      IF(I .GT. 2) GO TO 6700                                           CWZ01030
      P(I,K,2)=Q2(I,2,K)*(1./R(I,2))*(TIME-R(I,2))                      CWZ01040
      GO TO 6750                                                        CWZ01050
6700  CONTINUE                                                          CWZ01060
      P(I,K,2)=Q2(I,1,K)*(X0/60.+1.)*(1./R(I,2))*(TIME-R(I,2))          CWZ01070
6750  CONTINUE                                                          CWZ01080
      Q5=AINT(P(I,K,2))                                                 CWZ01090
      Q6=P(I,K,2)-Q5                                                    CWZ01100
      IF(Q6 .EQ. 0.) GO TO 6800                                         CWZ01110
      P(I,K,2)=Q5+1.                                                    CWZ01120
      GO TO 6800                                                        CWZ01130
6775  CONTINUE                                                          CWZ01140
C* CALCULATIONS FOR E AND F STABILIY CAT (STABLE FLOW)                  CWZ01150
C* INITIAL SHELL SPACING FOR WP SMOKE                                   CWZ01160
      IF(I .EQ. 1 .AND. K .EQ. 1) Z(I,1,K)=100.                         CWZ01170
      IF(I .EQ. 2 .AND. K .EQ. 1) Z(I,1,K)=50.                          CWZ01180
      IF(I .LT. 3 .AND. K .EQ. 2) Z(I,1,K)=100.                         CWZ01190
C* SUSTAINING SHELL SPACING FOR WP SMOKE                                CWZ01200
      IF(I .EQ. 1 .AND. K .EQ. 1) Z(I,2,K)=100.                         CWZ01210
      IF(I .EQ. 1 .AND. K .EQ. 2) Z(I,2,K)=200.                         CWZ01220
      IF(I .EQ. 2 .AND. K .EQ. 1) Z(I,2,K)=50.                          CWZ01230
      IF(I .EQ. 2 .AND. K .EQ. 2) Z(I,2,K)=100.                         CWZ01240
C* INITIAL VOLLEY FOR WP SMOKE                                          CWZ01250
      Q2(I,1,K)=X0/Z(I,1,K)+1.                                          CWZ01260
C* SUSTAINING VOLLEY FOR WP SMOKE                                       CWZ01270
      Q2(I,2,K)=X0/Z(I,2,K)+1.                                          CWZ01280
C* RATE OF FIRE FOR WP SMOKE                                            CWZ01290
      IF(K .EQ. 1) R(I,2)=.5                                            CWZ01300
      IF(K .EQ. 2) R(I,2)=1.                                            CWZ01310
C* TOTAL NUMBER OF ROUNDS REQUIRED (WP)                                 CWZ01320
      P(I,K,2)=Q2(I,1,K)+Q2(I,2,K)*1./R(I,2)*(TIME-R(I,2))              CWZ01330
6800  CONTINUE                                                          CWZ01340
      R(I,2)=1./R(I,2)                                                  CWZ01350
6900  CONTINUE                                                          CWZ01360
      RETURN                                                            CWZ01370
      END                                                               CWZ01380
```

```
      SUBROUTINE ITAM(IERR,TFNL)                                        ITA00010
C                                                                       ITA00020
C     THIS ROUTINE IS AN INVERSION OF THE NV&EOL TARGET ACQUISITION     ITA00030
C        MODEL, WITH EMPHASIS ON THE DEGRADATION (TRANSMITTANCE)        ITA00040
C        REQUIRED TO PREVENT DETECTION ABOVE A GIVEN PROBABILITY.       ITA00050
C                                                                       ITA00060
C                                                                       ITA00070
C     INPUTS ARE ON STANDARDIZED RECORDS:                               ITA00080
C                                                                       ITA00090
C        KEY WORD - COLUMNS 1-4, FROM AMONG TARV,SENS,GO,PAGE AND DONE. ITA00100
C        DATA FIELDS, ALL REAL - COLS 11-20, 21-30, 31-40,...,71-80.    ITA00110
C                                                                       ITA00120
C        INPUT CARDS ARE ORDER INDEPENDENT, WITH ARBITRARY OR SYSTEM    ITA00130
C        DEFAULTS CHOSEN IF CARDS ARE LEFT OUT, OR IF VALUES ARE INPUT 0. ITA00140
C        THE EXCEPTION TO ORDER INDEPENDENCE IS THAT AFTER EACH SET OF  ITA00150
C        INPUTS A (GO) CARD MUST BE PLACED TO INITIATE EXECUTION OF     ITA00160
C        ONE LOOP THROUGH THE PROGRAM.  FOLLOWING, OR IN PLACE OF       ITA00170
C        THE LAST INPUT-SET GO CARD, A (DONE) CARD MUST BE PROVIDED TO  ITA00180
C        CAUSE CONTROL TO EXIT ITAM AND RETURN TO THE SCREEN EXEC MODULE. ITA00190
C                                                                       ITA00200
C        IN SUBSEQUENT RUNS (DELINEATED BY GO CARDS) TABLES MAY BE      ITA00210
C        PRODUCED LINE-BY-LINE, AND ANY INPUT WHICH HAS NOT             ITA00220
C        BEEN CHANGED (IE INPUT AS 0.) WILL USE THE VALUE GIVEN ON THE  ITA00230
C        PREVIOUS RUN. EXCEPTIONS ARE THE FOV AND AMAG VALUES WHICH MUST ITA00240
C        BE SPECIFIED WHENEVER A NEW DEVICE NUMBER (LSC) IS INPUT, OR   ITA00250
C        THESE VALUES ASSUME DEFAULTS FOR THE DEVICE.                   ITA00260
C                                                                       ITA00270
C  KEY WORD     COLS.     VARIABLE          DESCRIPTION                 ITA00280
C  --------     -----     --------  ------------------------------      ITA00290
C     TARV        1-4                TARGET/SCENARIO DESCRIPTION        ITA00300
C                11-20      ACON    -  INTRINSIC CONTRAST (DIMENSION-   ITA00310
C                                      LESS) OR TEMPERATURE DIFFERENCE  ITA00320
C                                      OF TARGET/BACKGROUND (DEG K)     ITA00330
C                                      FOR THERMAL DEVICES.             ITA00340
C                21-30      SOG     -  SKY/GROUND RATIO, (DIMENSION-    ITA00350
C                                      LESS), USED FOR NON-THERMAL      ITA00360
C                                      DEVICES. SOG SHOULD INCLUDE      ITA00370
C                                      A FACTOR FOR CLOUD REFLECTANCE   ITA00380
C                31-40      R       -  RANGE TO TARGET (KM)            ITA00390
C                41-50      TARSZ   -  TARGET MINIMUM DIMENSION (M).   ITA00400
C                51-60      ZONE    -  SEARCH ZONE (DEG.**2)           ITA00410
C                61-70      ALFLG   -  AMBIENT ILLUM. CATEGORY (SEE    ITA00420
C                                      LIST BELOW) CONVERTED TO NAL,    ITA00430
C                                      AN INTEGER, IN PROGRAM. USED     ITA00440
C                                      FOR NON-THERMAL DEVICES.         ITA00450
C                71-80      ALIGHT  -  IF ALFLG=0., THE USER MUST      ITA00460
C                                      PROVIDE AN AMBIENT ILLUM. HERE   ITA00470
C                                      (FT. CDLS.)                      ITA00480
C                                                                       ITA00490
C     SENS        1-4                SENSOR DESCRIPTION                 ITA00500
C                11-20      PS      -  INPUT PROBABILITY OF DETECTION.  ITA00510
C                                      (DIMENSIONLESS)                  ITA00520
C                21-30      DVNUM   -  DEVICE NUMBER. CONVERTED TO LSC, ITA00530
C                                      AN INTEGER, IN PROGRAM.          ITA00540
C                31-40      DMODE   -  OPERATIONAL MODE (1. = WIDE FOV, ITA00550
C                                      2. = NARROW FOV). CONVERTED TO   ITA00560
C                                      MODE, AN INTEGER, IN PROGRAM.    ITA00570
C                41-50      FOV     -  FIELD OF VIEW (DEG.)            ITA00580
C                51-60      AMAG    -  MAGNIFICATION (FOR VISIBLE ONLY) ITA00590
C                61-70      AJOB    -  LEVEL OF ACQUISITION (MEDIAN    ITA00600
C                                      RESOLVABLE NORMALIZED CYCLES,    ITA00610
C                                      USUALLY 1. FOR DETECTION)        ITA00620
C                                                                       ITA00630
C     PAGE        1-4                (OPTIONAL) - FORCE PAGE EJECT,     ITA00640
C                                    WRITE NEW HEADER. (USEFUL FOR TABLE ITA00650
C                                    GENERATION.)                       ITA00660
C                                                                       ITA00670
C     GO          1-4                EXECUTE ONE LOOP WITH GIVEN INPUTS ITA00680
C                                                                       ITA00690
C     DONE        1-4                END COMPUTATIONS AND EXIT THE ITAM ITA00700
```

```
C                                       ROUTINE. (MAY BE USED IN PLACE OF   ITA00710
C                                       OR FOLLOWING THE LAST GO CARD)      ITA00720
C                                                                           ITA00730
C     OUTPUTS:  TFNL, THE TOTAL PATH TRANSMITTANCE (DIMENSIONLESS)          ITA00740
C               REQUIRED TO KEEP THE PROBABILITY OF DETECTION LESS          ITA00750
C               THAN OR EQUAL TO THAT INPUT, AS COMPUTED FROM THE           ITA00760
C               FINAL SET OF INPUTS PRECEDING THE DONE CARD.                ITA00770
C                                                                           ITA00780
C     PRINT OUTPUTS: ALL INPUT PARAMETERS AND                               ITA00790
C          C     = CONTRAST OR TEMPERATURE DIFFERENCE AT DEVICE             ITA00800
C                  REQUIRED FOR GIVEN LEVEL OF DETECTION PROBABILITY        ITA00810
C          RC    = RESOLVABLE CYCLES REQUIRED AT TARGET FOR GIVEN           ITA00820
C                  PROBABILITY OF DETECTION.                                ITA00830
C          TTOT  = TOTAL PATH TRANSMITTANCE TO REDUCE ACON AT TARGET        ITA00840
C                  TO C AT DEVICE.                                          ITA00850
C                                                                           ITA00860
C    SUBROUTINES:  ITAM CALLED BY SCREEN. ITAM CALLS CYCLE, CINV, TREQ.     ITA00870
C                                                                           ITA00880
C                                                                           ITA00890
C     AMBIENT ILLUMINATION CATEGORIES:                                      ITA00900
C                                                                           ITA00910
C          ALFLG OR NAL     FT. CDLS.        APPROX. CONDITIONS             ITA00920
C          ------------     ---------        ------------------             ITA00930
C               0           USER SPEC.                                      ITA00940
C               1           100.             CLEAR TO SLIGHT OVERCAST       ITA00950
C               2            10.             HEAVILY OVERCAST DAY           ITA00960
C               3             1.             SUNSET                         ITA00970
C               4             0.1            1/4 HR. AFTER SUNSET           ITA00980
C               5             0.01           1/2 HR. AFTER SUNSET           ITA00990
C               6             0.001          MOONLIT, CLEAR                 ITA01000
C               7             0.0001         MOONLIT, OVERCAST              ITA01010
C               8             0.00001        STARLIGHT ONLY                 ITA01020
C               9             0.000001       MINIMAL STARLIGHT              ITA01030
C                                                                           ITA01040
      DIMENSION RV(7),IT(2),AMAGW(14),AMAGN(14),FOVW(14),FOVN(14),          ITA01050
     *ALFA(9),IR(10),BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)                  ITA01060
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU    ITA01070
      COMMON /GEOMET/PTS(15),IGEOSW                                         ITA01080
      DATA IR/2HDO,2HNE,2HGO,2H   ,2HTA,2HRV,2HSE,2HNS,2HPA,2HGE/           ITA01090
      DATA AMAGW /1.0,7.0,12*0./,                                           ITA01100
     *     AMAGN /1.0,7.0,12*0./,                                         ITA01110
     *     FOVW /24.5,8.0,15.0,9.0,6.0,4.67,4.67,10.62,9.24,8.0,4*0./,    ITA01120
     *     FOVN /24.5,8.0,3*0.,1.56,0.,3.54,2.31,1.0,4*0./                ITA01130
      DATA TGT,SGR,FV,RN,ALV,DMG,AJB,PD,DVN,DMD,ACN /2.3,                   ITA01140
     *     3.,7.,1.,1.,1.,1.,1.,2.,1.,1./                                 ITA01150
      DATA ALFA /1.089005,.009730,1.8528,-0.0002197,.139,0.,0.,0.,0./       ITA01160
      DATA BETA/-1.801654,.03536,-3.13169,.0514,0.,.0651,.1124,.0968,       ITA01170
     *.12/                                                                  ITA01180
      DATA GAMA/4.217775,2.3575,5.17227,2.32178,2.7933,1.6149,1.3364,       ITA01190
     *.56867,.29/                                                           ITA01200
      DATA AMRAL/.05850156,.1022697,.2444688,.7264854,3.736508,             ITA01210
     *-16.87941, .06709132,.1137755,.3177794,.7642010,6.549327,             ITA01220
     *-16.18440, .04109820,.0715830,.1524969,.2319469,1.494814,             ITA01230
     *5.731956/                                                             ITA01240
      DATA AMRGM/2.94273,2.43789,2.11113,2.05233,3.33957,-4.98529,          ITA01250
     *           5.22111,4.28211,4.13668,3.44685,9.16003,-7.70891,          ITA01260
     *           2.80725,2.70329,2.48447,1.63044,2.73650,3.47838 /          ITA01270
C ***** INITIALIZE VALUES (FIRST EXECUTION)                                 ITA01280
      PB=.632                                                               ITA01290
      ACON=0.                                                               ITA01300
      SOG=0.                                                                ITA01310
      RNG=0.                                                                ITA01320
      TARSZ=0.                                                              ITA01330
      UZONE=0.                                                              ITA01340
      ALFLG=0.                                                              ITA01350
      ALIGHT=0.                                                             ITA01360
      PDET=0.                                                               ITA01370
      DVNUM=0.                                                              ITA01380
      DMODE=0.                                                              ITA01390
      DFOV=0.                                                               ITA01400
```

```
      DMAG=0.                                                                   ITA01410
      AJOB=0.                                                                   ITA01420
      TBAR=0.                                                                   ITA01430
      NAL=0                                                                     ITA01440
      TFNL=1.                                                                   ITA01450
      ILAST=0                                                                   ITA01460
      IFLGFV=0                                                                  ITA01470
      IFLGMG=0                                                                  ITA01480
C ***** OUTPUT LINE COUNTER                                                     ITA01490
      ILINE=0                                                                   ITA01500
C ***** INITIALIZE VALUES (REPEATED EXECUTIONS)                                 ITA01510
C ***** INPUT CARD COUNTER                                                      ITA01520
4     ICOV=0                                                                    ITA01530
      IF (ILAST.NE.0) GOTO 6                                                    ITA01540
C ***** FLAG TO WRITE TITLE                                                     ITA01550
      ITITL=0                                                                   ITA01560
      TTOT=1.                                                                   ITA01570
      RC=0.                                                                     ITA01580
      C=0.                                                                      ITA01590
      ICM=0                                                                     ITA01600
      INEWD=0                                                                   ITA01610
C ***** READ A CARD                                                             ITA01620
5     READ(IOIN,900) IT(1),IT(2),(RV(I),I=1,7)                                  ITA01630
900   FORMAT(2A2,6X,7F10.2)                                                     ITA01640
      ICOV=ICOV+1                                                               ITA01650
      IF(IT(1).EQ.IR(1).AND.IT(2).EQ.IR(2)) GO TO 6                             ITA01660
      IF(IT(1).EQ.IR(3).AND.IT(2).EQ.IR(4)) GO TO 9                             ITA01670
      IF(IT(1).EQ.IR(5).AND.IT(2).EQ.IR(6)) GO TO 7                             ITA01680
      IF(IT(1).EQ.IR(7).AND.IT(2).EQ.IR(8)) GO TO 8                             ITA01690
      IF (IT(1).EQ.IR(9).AND.IT(2).EQ.IR(10)) GOTO 2                            ITA01700
      WRITE(IOOUT,901) IT(1),IT(2),(RV(I),I=1,7)                                ITA01710
901   FORMAT(1X,61HTHE FOLLOWING CARD DOES NOT CONFORM TO ITAM INPUT CONITA01720
     *VENTIONS/1X,2A2,6X,7E10.3)                                                ITA01730
      IF(ICOV.LE.5) GO TO 5                                                     ITA01740
      IERR=1                                                                    ITA01750
      GOTO 1                                                                    ITA01760
C ***** ALL DONE                                                                ITA01770
6     IF (ICOV.GT.1) GOTO 3                                                     ITA01780
1     WRITE(IOOUT,902) TFNL                                                     ITA01790
902   FORMAT(1H0,5X,41H*** FINAL TOTAL TRANSMISSION FROM ITAM = ,F5.3)          ITA01800
      RETURN                                                                    ITA01810
C ***** TARGET CARD PROCESSING (TARV)                                           ITA01820
7     RCHK=ACON                                                                 ITA01830
      IF(RV(1).NE.0.) ACON=RV(1)                                                ITA01840
      IF(ACON.NE.RCHK) ITITL=1                                                  ITA01850
      IF(RV(2).NE.0.) SOG=RV(2)                                                 ITA01860
      IF(RV(3).NE.0.) RNG=RV(3)                                                 ITA01870
      IF(IGEOSW.NE.1) GO TO 477                                                 ITA01880
      RNG=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+(PTS(3)-PTS(6))**2)ITA01890
  477 CONTINUE                                                                  ITA01900
      RCHK=TARSZ                                                                ITA01910
      IF(RV(4).NE.0.) TARSZ=RV(4)                                               ITA01920
      IF(TARSZ.NE.RCHK) ITITL=1                                                 ITA01930
      RCHK=UZONE                                                                ITA01940
      IF(RV(5).NE.0.) UZONE=RV(5)                                               ITA01950
      IF(RCHK.NE.UZONE) ITITL=1                                                 ITA01960
      IF(RV(6).NE.0.) ALFLG=RV(6)                                               ITA01970
      IF(RV(7).NE.0.) ALIGHT=RV(7)                                              ITA01980
      GO TO 5                                                                   ITA01990
C ***** SENSOR CARD PROCESSING (SENS)                                           ITA02000
8     IF(RV(1).GT.0.) PDET=RV(1)                                                ITA02010
      RCHK=DVNUM                                                                ITA02020
      IF(RV(2).NE.0.) DVNUM=RV(2)                                               ITA02030
      IF(DVNUM.NE.RCHK) ITITL=1                                                 ITA02040
      IF (DVNUM.NE.RCHK) INEWD=1                                                ITA02050
      RCHK=DMODE                                                                ITA02060
      IF(RV(3).NE.0.) DMODE=RV(3)                                               ITA02070
      IF(DMODE.NE.RCHK) ITITL=1                                                 ITA02080
      IF (INEWD.EQ.1) DFOV=0.                                                   ITA02090
      IF (INEWD.EQ.1) IFLGFV=0                                                  ITA02100
```

```
      RCHK=DFOV                                                          ITA02110
      IF(RV(4).NE.0.) DFOV=RV(4)                                         ITA02120
      IF (RV(4).NE.0..AND.IFLGFV.EQ.1) IFLGFV=0                          ITA02130
      IF(DFOV.NE.RCHK) ITITL=1                                           ITA02140
      IF (INEWD.EQ.1) DMAG=0.                                            ITA02150
      IF (INEWD.EQ.1) IFLGMG=0                                           ITA02160
      RCHK=DMAG                                                          ITA02170
      IF(RV(5).NE.0.) DMAG=RV(5)                                         ITA02180
      IF (RV(5).NE.0..AND.IFLGMG.EQ.1) IFLGMG=0                          ITA02190
      IF(DMAG.NE.RCHK) ITITL=1                                           ITA02200
      RCHK=AJOB                                                          ITA02210
      IF(RV(6).NE.0.) AJOB=RV(6)                                         ITA02220
      IF(AJOB.NE.RCHK) ITITL=1                                           ITA02230
      GO TO 5                                                            ITA02240
C ***** PAGE TO NEW HEADER                                               ITA02250
2     ITITL=1                                                            ITA02260
      GOTO 5                                                             ITA02270
C ***** BEGIN PROCESSING (GO)                                            ITA02280
C ***** FIRST CHECK IF DEFAULTS NEEDED, AND SET INTEGER VALUES, LIGHT CA ITA02290
3     ILAST=1                                                            ITA02300
5     RCHK=DVNUM                                                         ITA02310
      IF(DVNUM.EQ.0.) DVNUM=DVN                                          ITA02320
      IF(RCHK.NE.DVNUM) ITITL=1                                          ITA02330
      LSC=IFIX(DVNUM+.0001)                                              ITA02340
      IF (LSC.LT.1) LSC=1                                                ITA02350
      IF (LSC.GT.14) LSC=14                                              ITA02360
      RCHK=DMODE                                                         ITA02370
      IF(DMODE.EQ.0.) DMODE=DMD                                          ITA02380
      IF(RCHK.NE.DMODE) ITITL=1                                          ITA02390
      MODE=IFIX(DMODE+0.0001)                                            ITA02400
      IF(MODE.LT.1) MODE=1                                               ITA02410
      IF(MODE.GT.2) MODE=2                                               ITA02420
      IF(PDET.EQ.0.) PDET=PD                                             ITA02430
      PS=PDET                                                            ITA02440
      IF(PS.GT.1.) PS=1.                                                 ITA02450
      IF(PS.LT.0.) PS=0.                                                 ITA02460
      RCHK=DFOV                                                          ITA02470
      IF(DFOV.LE.0.0.AND.MODE.EQ.1) DFOV=FOVW(LSC)                       ITA02480
      IF(DFOV.LE.0.0.AND.MODE.EQ.2) DFOV=FOVN(LSC)                       ITA02490
      IF(DFOV.LE.0.0) IFLGFV=1                                           ITA02500
      IF(DFOV.LE.0..AND.MODE.EQ.1) DFOV=FV                               ITA02510
      IF(DFOV.LE.0..AND.MODE.EQ.2) DFOV=FV/2.                            ITA02520
      IF(RCHK.NE.DFOV)  ITITL=1                                          ITA02530
      FOV=DFOV                                                           ITA02540
      RCHK=DMAG                                                          ITA02550
      IF(DMAG.LE.0.0.AND.MODE.EQ.1) DMAG=AMAGW(LSC)                      ITA02560
      IF(DMAG.LE.0.0.AND.MODE.EQ.2) DMAG=AMAGN(LSC)                      ITA02570
      IF(DMAG.LE.0.0) IFLGMG=1                                           ITA02580
      IF(DMAG.LE.0.0) DMAG=DMG                                           ITA02590
      IF(RCHK.NE.DMAG) ITITL=1                                           ITA02600
      AMAG=DMAG                                                          ITA02610
      RCHK=AJOB                                                          ITA02620
      IF(AJOB.EQ.0.0) AJOB=AJB                                           ITA02630
      IF(RCHK.NE.AJOB) ITITL=1                                           ITA02640
      RCHK=ACON                                                          ITA02650
      IF(ACON.EQ.0.0) ACON=ACN                                           ITA02660
      IF(RCHK.NE.ACON) ITITL=1                                           ITA02670
      C=ACON                                                             ITA02680
      IF(SOG.EQ.0.0) SOG=SGR                                             ITA02690
      IF(RNG.LE.0.0) RNG=RN                                              ITA02700
      R=RNG                                                              ITA02710
      RCHK=TARSZ                                                         ITA02720
      IF(TARSZ.LE.0.0) TARSZ=TGT                                         ITA02730
      IF(RCHK.NE.TARSZ) ITITL=1                                          ITA02740
      DIM=TARSZ                                                          ITA02750
      RCHK=UZONE                                                         ITA02760
      IF(UZONE.LE.0.0) UZONE=FOV**2                                      ITA02770
      IF(RCHK.NE.UZONE) ITITL=1                                          ITA02780
      ZONE=UZONE                                                         ITA02790
      IF (ALFLG.LT.0.) ALFLG=0.                                          ITA02800
```

```
      IF (ALFLG.GT.9.) ALFLG=9.                                          ITA02810
      IAL=IFIX(ALFLG+0.0001)                                             ITA02820
      IF(IAL.GT.0) AL=10.**(3-IAL)+1.E-10                                ITA02830
      IF(IAL.EQ.0.AND.ALIGHT.EQ.0.0) ALIGHT=ALV                          ITA02840
      IF(IAL.EQ.0) AL=ALIGHT                                             ITA02850
C ***** CHECK AND WRITE PAGE TITLE HEADING                              ITA02860
10    ILINE=ILINE+1                                                      ITA02870
      IF(ITITL.EQ.0.AND.ILINE.LE.48) GO TO 11                            ITA02880
      IF (ITITL.EQ.0.AND.ILINE.GT.49) WRITE (IOOUT,905)                  ITA02890
905   FORMAT(1H ,35X,28H** CONTINUED ON NEXT PAGE **)                    ITA02900
      WRITE(IOOUT,910)                                                   ITA02910
910   FORMAT(1H0/    41X,43(1H*)/41X,1H*,41X,1H*/41X,1H*,2X,37HINVERSE SITA02920
     *TATIC TARGET DETECTION MODEL,2X,1H*/41X,1H*,41X,1H*/41X,43(1H*))   ITA02930
      WRITE(IOOUT,911)                                                   ITA02940
911   FORMAT(1H0,64X,28HTARGET INTRINSIC CONTRAST OR)                    ITA02950
      WRITE(IOOUT,912) LSC,ACON                                          ITA02960
912   FORMAT(20X,13HDEVICE NUMBER,4X,I2,32X,22HTEMPERATURE DIFFERENCE,   ITA02970
     *       2X,F7.3)                                                    ITA02980
      IF(MODE.EQ.2) WRITE(IOOUT,913) DIM                                 ITA02990
913   FORMAT(1H0,19X,11HFOV TYPE - ,4X,6HNARROW,24X,28HMINIMUM TARGET DIITA03000
     *MENSION (M),2X,F7.3)                                              ITA03010
      IF(MODE.EQ.1)WRITE(IOOUT,914)DIM                                   ITA03020
914   FORMAT(1H0,19X,11HFOV TYPE - ,6X,4HWIDE,24X,28HMINIMUM TARGET DIMEITA03030
     *NSION (M),2X,F7.3)                                                ITA03040
      IF(IFLGFV.EQ.0) WRITE(IOOUT,915) FOV,AJOB                          ITA03050
915   FORMAT(1H0,19X,9HFOV (DEG),5X,F7.3,24X,27HACQUISITION LEVEL (50 PCITA03060
     *NT),3X,F7.3)                                                      ITA03070
      IF(IFLGFV.EQ.1) WRITE(IOOUT,916) FOV,AJOB                          ITA03080
916   FORMAT(1H0,19X,9HFOV (DEG),5X,F7.3,1X,19H(ARBITRARY DEFAULT),      ITA03090
     *       4X,27HACQUISITION LEVEL (50 PCNT),3X,F7.3)                  ITA03100
      IF(IFLGMG.EQ.0) WRITE(IOOUT,917) AMAG,ZONE                         ITA03110
917   FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,24X,24HSEARCH ZONE (DEGREEITA03120
     *S**2),6X,F7.3)                                                    ITA03130
      IF(IFLGMG.EQ.1) WRITE(IOOUT,918) AMAG,ZONE                         ITA03140
918   FORMAT(1H0,19X,13HMAGNIFICATION,2X,F6.3,1X,19H(ARBITRARY DEFAULT),ITA03150
     *       4X,24HSEARCH ZONE (DEGREES**2),6X,F7.3)                     ITA03160
      WRITE (IOOUT,920)                                                  ITA03170
920   FORMAT(1H0,52X,35HREQUIRES (TO DEFEAT DEVICE) AT MOST/             ITA03180
     *             4X,14HFOR NO GREATER,6X,5HUNDER,8X,3HAND,7X,2HAT,4X,  ITA03190
     *35(1H-))                                                          ITA03200
      WRITE(IOOUT,919)                                                   ITA03210
919   FORMAT(8X,5HINPUT,11X,5HINPUT,7X,5HINPUT,5X,5HINPUT,3X,            ITA03220
     *       8HCOMPUTED,2X,11HCONTRAST OR,3X,8HCOMPUTED/6X,             ITA03230
     *       9HDETECTION,6X,11HAMB. ILLUM.,2X,10HSKY/GROUND,2X,         ITA03240
     *       5HRANGE,2X,10HRESOLVABLE,1X,11HTEMP. DIFF.,2X,10HTOTAL PATHITA03250
     *       /3X,5HPROB.,2X,9HTIME(SEC),3X,9H(FT CDL9),5X,5HRATIO,5X,   ITA03260
     *       4H(KM),3X,10HCYCLES, RC,2X,9HAT DEVICE,2X,13HTRANSMITTANCE,ITA03270
     *       13X,8HCOMMENTS/3X,5(1H-),2X,9(1H-),2X,11(1H-),2X,10(1H-),  ITA03280
     *       1X,6(1H-),2X,10(1H-),2X,9(1H-),2X,13(1H-),2X,30(1H-))      ITA03290
      ILINE=23                                                           ITA03300
C ***** BEGIN COMPUTATIONS   FOV,RC,TBAR                                ITA03310
11    S=DIM/R                                                            ITA03320
      IF (FOV.LE.0.) FOV=.0001                                           ITA03330
      TS=1.7*ZONE/FOV**2                                                 ITA03340
      IF(ZONE.GT.9.0.AND.FOV.GT.5.) TS=(1.7*ZONE)/(5.0*FOV)              ITA03350
      IF (TS.LT..5) TS=.5                                                ITA03360
      CALL CYCLE(PS,PB,AJOB,RC)                                          ITA03370
      IF (AL.LE.0.) AL=1.E-7                                             ITA03380
      ALPRNT=AL                                                          ITA03390
      IF (LSC.EQ.13) GOTO 13                                             ITA03400
      TBAR=0.                                                            ITA03410
      IF (RC.LT.0.1) GOTO 14                                             ITA03420
      IF((LSC.GT.5.AND.LSC.LT.10).OR.LSC.EQ.11.OR.LSC.EQ.14) GO TO 12    ITA03430
      RCS=RC                                                             ITA03440
      PINF=1.-EXP(-1.7*RCS/6.8)                                          ITA03450
      IF (PINF.LE.0.) PINF=.0001                                         ITA03460
      TBAR=0.5*TS*(2.-PINF)/PINF                                         ITA03470
      GO TO 14                                                           ITA03480
12    TBAR=0.5*TS*(2.0-PS)/PS                                            ITA03490
      GO TO 14                                                           ITA03500
```

```
C *****  DEVICE #13                                                          ITA03510
13      NAL=1                                                               ITA03520
        ICM=0                                                               ITA03530
        PS=.99                                                              ITA03540
        TBAR=1.8                                                            ITA03550
        C=ACON                                                              ITA03560
        TTOT=1.                                                             ITA03570
        TFNL=TTOT                                                           ITA03580
        GO TO 22                                                            ITA03590
C *****  COMPUTE C BASED ON AMBIENT ILLUMINATION.  INTERPOLATE IF NECESSA   ITA03600
14      IF (S.LE.0.) S=.0001                                                ITA03610
        IF (RC.LT.0.) RC=0.                                                 ITA03620
        RC=RC/S                                                             ITA03630
        IF (AMAG.LE.0.) AMAG=.0001                                          ITA03640
        IF(LSC.EQ.1.OR.LSC.EQ.2) RC=RC/AMAG                                 ITA03650
        IF(LSC.EQ.2) AL=0.7*AL                                              ITA03660
C *****  CORRECT FOR FIELD OF VIEW                                          ITA03670
        IF (LSC.LT.6.OR.LSC.EQ.7) GOTO 140                                  ITA03680
        IF ((LSC.EQ.6).AND.(MODE.EQ.2)) RC=RC/3.                            ITA03690
        IF ((LSC.EQ.8).AND.(MODE.EQ.1)) RC=RC*3.                            ITA03700
        IF ((LSC.EQ.9).AND.(MODE.EQ.1)) RC=RC*4.                            ITA03710
        IF ((LSC.EQ.11).AND.(MODE.EQ.1)) RC=RC*3.                           ITA03720
        IF ((LSC.EQ.14).AND.(MODE.EQ.1)) RC=RC*3.                           ITA03730
        IF ((LSC.EQ.10).AND.(MODE.EQ.1)) RC=RC*8.                           ITA03740
        IF ((LSC.EQ.12).AND.(MODE.EQ.2)) RC=RC/4.                           ITA03750
140     CONTINUE                                                            ITA03760
        IF(LSC.EQ.2) IAL=0                                                  ITA03770
        IF(IAL.GT.0) GO TO 20                                               ITA03780
        AV=100.                                                             ITA03790
        IF(AL.GT.100.) AL=100.                                              ITA03800
        DO 15 I=1,9                                                         ITA03810
        IF(AL.GT.(0.999*AV)) GO TO 16                                       ITA03820
        AV=AV/10.                                                           ITA03830
15      CONTINUE                                                            ITA03840
        I=2                                                                 ITA03850
        IF (LSC.GT.5) GOTO 160                                              ITA03860
        NAL=0                                                               ITA03870
        ICM=0                                                               ITA03880
        GO TO 22                                                            ITA03890
C *****  CHECK IF INTERPOLATION NEEDED, IF NOT, GO TO 20                    ITA03900
16      IF (LSC.GT.5) GOTO 160                                              ITA03910
        IF(AL.GT.(1.001*AV)) GO TO 17                                       ITA03920
160     IAL=I                                                               ITA03930
        GO TO 20                                                            ITA03940
C *****  INTERPOLATE                                                        ITA03950
17      NAL1=I-1                                                            ITA03960
        NAL2=I                                                              ITA03970
        ICM1=0                                                              ITA03980
        ICM2=0                                                              ITA03990
        CALL INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA,AMRAL,    ITA04000
       *AMRGM)                                                              ITA04010
        NAL=0                                                               ITA04020
        IF(NAL1.EQ.0.AND.NAL2.EQ.0) GO TO 22                                ITA04030
        NAL=NAL2                                                            ITA04050
        IF(NAL1.GT.0) GO TO 18                                              ITA04040
        ICM=ICM2                                                            ITA04060
        GO TO 21                                                            ITA04070
18      IF(NAL2.GT.0) GO TO 19                                              ITA04080
        NAL=NAL1                                                            ITA04090
        ICM=ICM1                                                            ITA04100
        GO TO 21                                                            ITA04110
19      IF(ICM1.GT.0.AND.ICM2.GT.0) ICM=1                                   ITA04120
        IF(ICM1.LT.0.AND.ICM2.LT.0) ICM=-1                                  ITA04130
        GO TO 21                                                            ITA04140
C *****  NO INTERPOLATION NEEDED                                            ITA04150
20      NAL=IAL                                                             ITA04160
        CALL CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM)              ITA04170
21      IF(ICM.EQ.-1) C=.99*C                                               ITA04180
        CALL TREQ(ACON,SOC,C,LSC,TTOT)                                      ITA04190
        IF (ICM.EQ.1) TTOT=1.                                               ITA04200
```

```
      TFNL=TTOT                                                          ITA04210
C ****** PRINT RESULTS                                                   ITA04220
22    IF(NAL.GT.0) GO TO 23                                              ITA04230
      TFNL=1.                                                            ITA04240
      WRITE(IOOUT,930) PS,TBAR,ALPRNT,SOG,R,TFNL                         ITA04250
930   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,4X,6(1H-),7X,5(1H-),5X,       ITA04260
     *       F8.3,6X,28HNOTE- AMBIENT ILLUM. OUTSIDE/91X,                ITA04270
     *       29HDEVICE OPERATIONAL LIMITS. NO/91X,                       ITA04280
     *       21HOBSCURATION REQUIRED./1X)                                ITA04290
      ILINE=ILINE+3                                                      ITA04300
      GO TO 4                                                            ITA04310
23    IF (TTOT.GE.1..AND.ICM.EQ.0) GOTO 24                               ITA04320
      IF(ICM.EQ.0) WRITE(IOOUT,931) PS,TBAR,ALPRNT,SOG,R,RC,C,TTOT       ITA04330
      IF(ICM.EQ.1) WRITE(IOOUT,932) PS,TBAR,ALPRNT,SOG,R,RC,TTOT,C       ITA04340
      IF(ICM.EQ.1) ILINE=ILINE+6                                         ITA04350
      IF(ICM.EQ.-1) WRITE(IOOUT,933) PS,TBAR,ALPRNT,SOG,R,RC,C,TTOT      ITA04360
      IF(ICM.EQ.-1) ILINE=ILINE+7                                        ITA04370
      GO TO 4                                                            ITA04380
24    WRITE (IOOUT,934  ) PS,TBAR,ALPRNT,SOG,R,RC,C,TTOT                 ITA04390
934   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,       ITA04400
     *       31HNOTE- CONTRAST (OR TEMP. DIFF.)/91X,                     ITA04410
     *       31HREQUIRED WOULD EXCEED INTRINSIC/91X,                     ITA04420
     *       31HCONTRAST (TEMP. DIFF.). NO OBS-/91X,                     ITA04430
     *       17HCURATION REQUIRED/1X)                                    ITA04440
      ILINE=ILINE+3                                                      ITA04450
      GOTO 4                                                             ITA04460
931   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3)          ITA04470
932   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,7X,5(1H-),5X,F8.3,6X,   ITA04480
     *       29HNOTE- DETECTION PROBABILITY  /91X,                       ITA04490
     *       30HREQUIRES CONTRAST (OR TEMP.   /91X,                      ITA04500
     *       29HDIFF.) AND RESOLVABLE CYCLES /91X,                       ITA04510
     *       27HABOVE LIMIT FOR DEVICE. NO /91X,                         ITA04520
     *       26HOBSCURANT REQUIRED. DEVICE/91X,                          ITA04530
     *       18HUPPER LIMIT IS C= ,F8.3/1X)                              ITA04540
933   FORMAT(3X,F5.3,F9.2,F14.6,F10.3,F9.2,F10.3,F12.3,5X,F8.3,6X,       ITA04550
     *       30HNOTE- INPUT DETECTION PROBAB- /91X,                      ITA04560
     *       30HILITY REQUIRES CONTRAST (OR   /91X,                      ITA04570
     *       30HTEMP. DIFF.) BELOW THRESHOLD. /91X,                      ITA04580
     *       29HVALUES ASSUMED ARE 99 PERCENT/91X,                       ITA04590
     *       29HOF THRESHOLD. ADDITIONAL OBS-/91X,                       ITA04600
     *       29HCURANT WILL NOT DECREASE     /91X,                       ITA04610
     *       30HDETECTION PROBABILITY.        /1X)                       ITA04620
      END                                                                ITA04630
```

```
      SUBROUTINE CINV(LSC,RC,NAL,C,ICM,ALFA,BETA,GAMA,AMRAL,AMRGM)       CIN00010
      DIMENSION ALFA(9),BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)            CIN00020
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCIN00030
      DATA IPOS,INEG /1,-1/                                              CIN00040
C   THIS ROUTINE COMPUTES THE REQUIRED CONTRAST AT THE DEVICE FOR A      CIN00050
C     GIVEN NUMBER OF RESOLVABLE CYCLES AT THE DEVICE.  EQUATIONS        CIN00060
C     ARE INVERSIONS OF THOSE OF THE NV&EOL TARGET ACQUISITION           CIN00070
C     (FOR STATIC DETECTION) MODEL.                                      CIN00080
C                                                                        CIN00090
C    INPUTS:  LSC    = DEVICE NUMBER (1-14)                             CIN00100
C             RC     = RESOLVABLE CYCLES (DECIMAL)                      CIN00110
C             NAL    = AMBIENT ILLUMINATION CATEGORY (1-9)              CIN00120
C                                                                        CIN00130
C    OUTPUTS: C      = CONTRAST OR TEMPERATURE DIFFERENCE(DEG K) REQUIRED CIN00140
C                       BY DEVICE FOR GIVEN RC.                          CIN00150
C             ICM    = OPERATIONAL LIMITS FLAG (+1 EXCEEDS LIMIT, -1    CIN00160
C                       BELOW LIMIT, 0 WITHIN RANGE)                     CIN00170
C             NAL    = SET TO 0 IF TOO MUCH OR NOT ENOUGH AMBIENT ILLUM. CIN00180
C                                                                        CIN00190
      CFUN(RX,ALF,BET,GAM)=(BET+ALF*RX)/(GAM-RX)                         CIN00200
      ICM=0                                                              CIN00210
      IF(NAL.EQ.0) GO TO 9020                                            CIN00220
C ***** BRANCH TO LSC                                                    CIN00230
5     GO TO (10,10,30,40,50,60,60,80,90,100,1111,1200,9060,1400), LSC    CIN00240
10    GO TO (110,120,130,140,150,160,170,9060,9060), NAL                 CIN00250
C ***** DEVICE #1      NAL #1                                            CIN00260
110   IF(2.74.LT.RC) GO TO 9030                                          CIN00270
      IF((2.133.LT.RC).AND.(RC.LE.2.74)) GO TO 112                       CIN00280
      IF((.26795.LT.RC).AND.(RC.LE.2.133)) GO TO 113                     CIN00290
      IF(RC.LE..26795) GO TO 114                                         CIN00300
      GO TO 9000                                                         CIN00310
112   IEQ=1                                                              CIN00320
      GO TO 9050                                                         CIN00330
113   IEQ=2                                                              CIN00340
      GO TO 9050                                                         CIN00350
114   C=0.015                                                            CIN00360
      GO TO 9015                                                         CIN00370
C ***** DEVICE #1 NAL #2                                                 CIN00380
120   IF(2.74.LT.RC) GO TO 9030                                          CIN00390
      IF((2.133.LT.RC).AND.(RC.LE.2.74)) GO TO 122                       CIN00400
      IF((.0001.LE.RC).AND.(RC.LE.2.133)) GO TO 123                      CIN00410
      IF(RC.LT..0001) GO TO 124                                          CIN00420
      GO TO 9000                                                         CIN00430
122   IEQ=3                                                              CIN00440
      GO TO 9050                                                         CIN00450
123   IEQ=4                                                              CIN00460
      GO TO 9050                                                         CIN00470
124   C=0.025                                                            CIN00480
      GO TO 9015                                                         CIN00490
C ***** DEVICE #1    NAL #3                                              CIN00500
130   IF(2.29.LT.RC) GO TO 131                                           CIN00510
      IF((0.49585.LE.RC).AND.(RC.LE.2.29)) GO TO 132                     CIN00520
      IF(RC.LT.0.49585) GO TO 133                                        CIN00530
      GO TO 9000                                                         CIN00540
131   C=0.6324                                                           CIN00550
      GO TO 9035                                                         CIN00560
132   IEQ=5                                                              CIN00570
      GO TO 9050                                                         CIN00580
133   C=0.030                                                            CIN00590
      GO TO 9015                                                         CIN00600
C ***** DEVICE #1     NAL #4                                             CIN00610
140   IF(1.5219.LT.RC) GO TO 141                                         CIN00620
      IF((0.313.LE.RC).AND.(RC.LE.1.5219)) GO TO 142                     CIN00630
      IF(RC.LT.0.313) GO TO 143                                          CIN00640
      GO TO 9000                                                         CIN00650
141   C=0.70                                                             CIN00660
      GO TO 9035                                                         CIN00670
142   IEQ=6                                                              CIN00680
      GO TO 9050                                                         CIN00690
143   C=0.05                                                             CIN00700
```

```
      GO TO 9015                                                        CIN00710
C ***** DEVICE #1     NAL #5                                            CIN00720
150   IF(1.1959.LT.RC) GO TO 9030                                       CIN00730
      IF((0.0.LE.RC).AND.(RC.LE.1.1959)) GO TO 152                      CIN00740
      IF(RC.LT.0.0) GO TO 153                                           CIN00750
      GO TO 9000                                                        CIN00760
152   IEQ=7                                                             CIN00770
      GO TO 9050                                                        CIN00780
153   C=0.0841                                                          CIN00790
      GO TO 9015                                                        CIN00800
C ***** DEVICE #1     NAL #6                                            CIN00810
160   IF(0.44767.LT.RC) GO TO 9030                                      CIN00820
      IF((0.0309.LE.RC).AND.(RC.LE.0.44767)) GO TO 162                  CIN00830
      IF(RC.LT.0.0309) GO TO 163                                        CIN00840
      GO TO 9000                                                        CIN00850
162   IEQ=8                                                             CIN00860
      GO TO 9050                                                        CIN00870
163   C=0.18                                                            CIN00880
      GO TO 9015                                                        CIN00890
C ***** DEVICE #1     NAL #7                                            CIN00900
170   IF(0.14.LT.RC) GO TO 9030                                         CIN00910
      IF((0.05.LE.RC).AND.(RC.LE.0.14)) GO TO 172                       CIN00920
      IF(RC.LT.0.05) GO TO 173                                          CIN00930
      GO TO 9000                                                        CIN00940
172   IEQ=9                                                             CIN00950
      GO TO 9050                                                        CIN00960
173   C=0.50                                                            CIN00970
      GO TO 9015                                                        CIN00980
C ***** DEVICE #3                                                       CIN00990
30    GO TO (9060,9060,9060,340,350,360,370,380,390), NAL               CIN01000
C ***** DEVICE #3     NAL #4                                            CIN01010
340   IF(0.7497.LE.RC.AND.RC.LT.2.941) GO TO 342                        CIN01020
      IF (RC.GE.2.941) GOTO 9075                                        CIN01030
      IF(RC.LT.0.7497) GO TO 9040                                       CIN01040
      GO TO 9000                                                        CIN01050
342   IEQ=1                                                             CIN01060
      GO TO 9080                                                        CIN01070
C ***** DEVICE #3     NAL #5                                            CIN01080
350   IF (0.3988.LE.RC.AND.RC.LT.2.4350) GO TO 352                      CIN01090
      IF (RC.GE.2.435) GOTO 9075                                        CIN01100
      IF(RC.LT.0.3988) GO TO 9040                                       CIN01110
      GO TO 9000                                                        CIN01120
352   IEQ=2                                                             CIN01130
      GO TO 9080                                                        CIN01140
C ***** DEVICE #3     NAL #6                                            CIN01150
360   IF (0.15965.LE.RC.AND.RC.LT.2.1060) GO TO 362                     CIN01160
      IF (RC.GE.2.106) GOTO 9075                                        CIN01170
      IF(RC.LT.0.15965) GO TO 9040                                      CIN01180
      GO TO 9000                                                        CIN01190
362   IEQ=3                                                             CIN01200
      GO TO 9080                                                        CIN01210
C ***** DEVICE #3     NAL #7                                            CIN01220
370   IF (0.05498.LE.RC.AND.RC.LT.2.0375) GO TO 372                     CIN01230
      IF (RC.GE.2.0375) GOTO 9075                                       CIN01240
      IF(RC.LT.0.05498) GO TO 9040                                      CIN01250
      GO TO 9000                                                        CIN01260
372   IEQ=4                                                             CIN01270
      GO TO 9080                                                        CIN01280
C ***** DEVICE #3     NAL #8                                            CIN01290
380   IF (0.442.LE.RC.AND.RC.LT.3.2190) GO TO 382                       CIN01300
      IF (RC.GE.3.219) GOTO 9075                                        CIN01310
      IF((RC.LT.0.442).AND.(RC.GT.0.)) GO TO 383                        CIN01320
      IF (RC.LE.0.) GOTO 384                                            CIN01330
      GO TO 9000                                                        CIN01340
382   IEQ=5                                                             CIN01350
      GO TO 9080                                                        CIN01360
383   C=0.57                                                            CIN01370
      GO TO 9015                                                        CIN01380
384   C=0.33                                                            CIN01390
      GOTO 9015                                                         CIN01400
```

```
C ***** DEVICE #3      NAL #9                                                  CIN01410
390     IF (0.172.LE.RC) GO TO 392                                            CIN01420
        IF(RC.LT.0.172.AND.RC.GT.0.) GO TO 393                                CIN01430
        IF (RC.LE.0.) GOTO 394                                                CIN01440
        GO TO 9000                                                            CIN01450
392     IEQ=6                                                                 CIN01460
        GO TO 9080                                                            CIN01470
393     C=0.57                                                                CIN01480
        GO TO 9015                                                            CIN01490
394     C=.33                                                                 CIN01500
        GOTO 9015                                                             CIN01510
C ***** DEVICE #4                                                             CIN01520
40      GO TO (9060,9060,9060,440,450,460,470,480,490), NAL                   CIN01530
C ***** DEVICE #4      NAL #4                                                 CIN01540
440     IF (1.199.LE.RC.AND.RC.LT.5.2180) GO TO 442                           CIN01550
        IF (RC.GE.5.218) GOTO 9075                                            CIN01560
        IF(RC.LT.1.199) GO TO 9040                                            CIN01570
        GO TO 9000                                                            CIN01580
442     IEQ=1                                                                 CIN01590
        GO TO 9080                                                            CIN01600
C ***** DEVICE #4      NAL #5                                                 CIN01610
450     IF (0.6402.LE.RC.AND.RC.LT.4.2770) GO TO 452                          CIN01620
        IF (RC.GE.4.277) GOTO 9075                                            CIN01630
        IF(RC.LT.0.6402) GO TO 9040                                           CIN01640
        GO TO 9000                                                            CIN01650
452     IEQ=2                                                                 CIN01660
        GO TO 9080                                                            CIN01670
C ***** DEVICE #4      NAL #6                                                 CIN01680
460     IF (0.2449.LE.RC.AND.RC.LT.4.1240) GO TO 462                          CIN01690
        IF (RC.GE.4.124) GOTO 9075                                            CIN01700
        IF(RC.LT.0.2449) GO TO 9040                                           CIN01710
        GO TO 9000                                                            CIN01720
462     IEQ=3                                                                 CIN01730
        GO TO 9080                                                            CIN01740
C ***** DEVICE #4      NAL #7                                                 CIN01750
470     IF (0.08791.LE.RC.AND.RC.LT.3.4200) GO TO 472                         CIN01760
        IF (RC.GE.3.420) GOTO 9075                                            CIN01770
        IF(RC.LT.0.08791) GO TO 9040                                          CIN01780
        GO TO 9000                                                            CIN01790
472     IEQ=4                                                                 CIN01800
        GO TO 9080                                                            CIN01810
C ***** DEVICE #4      NAL #8                                                 CIN01820
480     IF (0.4394.LE.RC.AND.RC.LT.8.5970) GO TO 482                          CIN01830
        IF (RC.GE.8.597) GOTO 9075                                            CIN01840
        IF((RC.LT.0.4394).AND.(RC.GT.0.)) GO TO 483                           CIN01850
        IF (RC.LE.0.) GOTO 484                                                CIN01860
        GO TO 9000                                                            CIN01870
482     IEQ=5                                                                 CIN01880
        GO TO 9080                                                            CIN01890
483     C=0.33                                                                CIN01900
        GO TO 9015                                                            CIN01910
484     C=0.07                                                                CIN01920
        GOTO 9015                                                             CIN01930
C ***** DEVICE #1      NAL #9                                                 CIN01940
490     IF (0.1605.LE.RC) GO TO 492                                           CIN01950
        IF(RC.LT.0.1605.AND.RC.GT.0.) GO TO 493                               CIN01960
        IF (RC.LE.0.) GOTO 494                                                CIN01970
        GO TO 9000                                                            CIN01980
492     IEQ=6                                                                 CIN01990
        GO TO 9080                                                            CIN02000
493     C=0.33                                                                CIN02010
        GO TO 9015                                                            CIN02020
494     C=0.07                                                                CIN02030
        GOTO 9015                                                             CIN02040
C ***** DEVICE # 5                                                            CIN02050
50      GO TO (9060,9060,9060,540,550,560,570,580,590), NAL                   CIN02060
540     IF (0.9189.LE.RC.AND.RC.LT.2.8060) GO TO 542                          CIN02070
        IF (RC.GE.2.806) GOTO 9075                                            CIN02080
        IF(RC.LT.0.9189) GO TO 9040                                           CIN02090
        GO TO 9000                                                            CIN02100
```

```
542   IEQ=1                                                                  CIN02110
      GO TO 9080                                                             CIN02120
C ***** DEVICE #5      NAL #5                                                CIN02130
550   IF (0.59035.LE.RC.AND.RC.LT.2.7010) GO TO 552                          CIN02140
      IF (RC.GE.2.7010) GOTO 9075                                            CIN02150
      IF(RC.LT.0.59035) GO TO 9040                                           CIN02160
      GO TO 9000                                                             CIN02170
552   IEQ=2                                                                  CIN02180
      GO TO 9080                                                             CIN02190
C ***** DEVICE #5      NAL #6                                                CIN02200
560   IF( 0.2881.LE.RC.AND.RC.LT.2.4800) GO TO 562                           CIN02210
      IF (RC .GE.2.480) GOTO 9075                                            CIN02220
      IF(RC.LT.0.2881) GO TO 9040                                            CIN02230
      GO TO 9000                                                             CIN02240
562   IEQ=3                                                                  CIN02250
      GO TO 9080                                                             CIN02260
C ***** DEVICE #5      NAL #7                                                CIN02270
570   IF (0.12943.LE.RC.AND.RC.LT.1.6270) GO TO 572                          CIN02280
      IF (RC.GE.1.627) GOTO 9075                                             CIN02290
      IF(RC.LT.0.12943) GO TO 9040                                           CIN02300
      GO TO 9000                                                             CIN02310
572   IEQ=4                                                                  CIN02320
      GO TO 9080                                                             CIN02330
C ***** DEVICE #5      NAL #8                                                CIN02340
580   IF (0.4949.LE.RC.AND.RC.LT.2.6960) GO TO 582                           CIN02350
      IF (RC.GE.2.696) GOTO 9075                                             CIN02360
      IF(RC.LT.0.4949.AND.RC.GT.0.) GO TO 583                                CIN02370
      IF (RC.LE.0.) GOTO 584                                                 CIN02380
      GO TO 9000                                                             CIN02390
582   IEQ=5                                                                  CIN02400
      GO TO 9080                                                             CIN02410
583   C=0.33                                                                 CIN02420
      GO TO 9015                                                             CIN02430
584   C=0.07                                                                 CIN02440
      GOTO 9015                                                              CIN02450
C ***** DEVICE #5      NAL #9                                                CIN02460
590   IF (0.1894.LE.RC.AND.RC.LT.3.2898) GO TO 592                           CIN02470
      IF (RC.GE.3.2898) GOTO 9075                                            CIN02480
      IF(RC.LT.0.1894.AND.RC.GT.0.) GO TO 593                                CIN02490
      IF (RC.LE.0.) GOTO 594                                                 CIN02500
      GO TO 9000                                                             CIN02510
592   IEQ=6                                                                  CIN02520
      GO TO 9080                                                             CIN02530
593   C=0.33                                                                 CIN02540
      GO TO 9015                                                             CIN02550
594   C=0.07                                                                 CIN02560
      GOTO 9015                                                              CIN02570
C ***** DEVICES #6 AND #7                                                    CIN02580
60    IF(1.7934.LT.RC) GO TO 601                                             CIN02590
      IF((0.92376.LT.RC).AND.(RC.LE.1.7934)) GO TO 602                       CIN02600
      IF((0.11022.LE.RC).AND.(RC.LE.0.92376)) GO TO 603                      CIN02610
      IF(RC.LT.0.11022) GO TO 9070                                           CIN02620
      GO TO 9000                                                             CIN02630
601   C=8.38                                                                 CIN02640
      GO TO 9035                                                             CIN02650
602   C=(.172363*RC-0.0392775)/(1.82560-RC)                                  CIN02660
      GO TO 9020                                                             CIN02670
603   C=(0.09772*RC)/(1.-0.34779*RC)                                         CIN02680
      GO TO 9020                                                             CIN02690
C ***** DEVICE #8                                                            CIN02700
80    IF(4.702.LT.RC) GO TO 801                                              CIN02710
      IF((0.996.LE.RC).AND.(RC.LE.4.702)) GO TO 802                          CIN02720
      IF(RC.LT.0.996) GO TO 803                                              CIN02730
      GO TO 9000                                                             CIN02740
801   C=2.18                                                                 CIN02750
      GO TO 9035                                                             CIN02760
802   C=(0.0298*RC)/(1.-0.199*RC)                                            CIN02770
      GO TO 9020                                                             CIN02780
803   C=0.037                                                                CIN02790
      GO TO 9015                                                             CIN02800
```

```
C *****  DEVICE #9                                                        CIN02810
90      IF(0.37181.LE.RC.AND.RC.LT.9.14027) GO TO 901                    CIN02820
        IF (RC.GE.9.14027) GOTO 9075                                     CIN02830
        IF(RC.LT.0.37181) GO TO 9070                                     CIN02840
        GO TO 9000                                                       CIN02850
901     C=(0.0289*RC)/(1.-0.1092*RC)                                     CIN02860
        GO TO 9020                                                       CIN02870
C *****  DEVICE #10                                                      CIN02880
100     IF(NAL.GT.2) GO TO 9060                                          CIN02890
        IF(4.227.LE.RC.AND.RC.LT.91.05956) GO TO 1010                    CIN02900
        IF (RC.GE.91.05956) GOTO 9075                                    CIN02910
        C=0.01                                                           CIN02920
C   NEXT STATEMENT  IS LIMINAL CONTRAST ADJUSTMENT                       CIN02930
        C=C/1.63                                                         CIN02940
        GO TO 9015                                                       CIN02950
1010    C=(0.0303*RC)/(13.43-0.1473*RC)                                  CIN02960
C   NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT                        CIN02970
        C=C/1.63                                                         CIN02980
        GO TO 9020                                                       CIN02990
C *****  DEVICE # 11                                                     CIN03000
1111    IF(0.5057.LE.RC.AND.RC.LT.7.73353) GO TO 1110                    CIN03010
        IF (RC.GE.7.73353) GOTO 9075                                     CIN03020
        GO TO 9070                                                       CIN03030
1110    C=(0.0207*RC)/(1.-0.1291*RC)                                     CIN03040
        GO TO 9020                                                       CIN03050
C *****  DEVICE # 12                                                     CIN03060
1200    IF(NAL.GT.2) GO TO 9060                                          CIN03070
        IF(3.48.LE.RC.AND.RC.LT.31.48113) GO TO 1210                     CIN03080
        IF (RC.GE.31.41883) GOTO 9075                                    CIN03090
        C=0.01                                                           CIN03100
C   NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT                        CIN03110
        C=C/1.63                                                         CIN03120
        GO TO 9015                                                       CIN03130
1210    C=(0.0206*RC)/(8.06-0.2559*RC)                                   CIN03140
C   NEXT STATEMENT IS LIMINAL CONTRAST ADJUSTMENT                        CIN03150
        C=C/1.63                                                         CIN03160
        GO TO 9020                                                       CIN03170
C *****  DEVICE # 14                                                     CIN03180
1400    IF(0.9098.LE.RC.AND.RC.LT.4.57711) GO TO 1410                    CIN03190
        IF (RC.GE.4.57711) GOTO 9075                                     CIN03200
        C=0.037                                                          CIN03210
        GO TO 9015                                                       CIN03220
1410    C=(0.0297567*RC)/(0.91287-0.1991449*RC)                          CIN03230
        GO TO 9020                                                       CIN03240
9000    WRITE(IOOUT,9010) LSC,NAL,RC                                     CIN03250
9010    FORMAT(5X,46HBAD PARAMETER PASSED TO SUBRTN CINV **** LSC= ,I3,  CIN03260
     *         6H NAL= ,I3,5H RC= ,F10.4)                                CIN03270
9015    ICM=INEG                                                         CIN03280
9020    RETURN                                                           CIN03290
9030    C=0.80                                                           CIN03300
9035    ICM=IPOS                                                         CIN03310
        GO TO 9020                                                       CIN03320
9040    C=0.02                                                           CIN03330
        GO TO 9015                                                       CIN03340
9050    C=CFUN(RC,ALFA(IEQ),BETA(IEQ),GAMA(IEQ))                         CIN03350
        GO TO 9020                                                       CIN03360
9060    NAL=0                                                            CIN03370
        GO TO 9020                                                       CIN03380
9070    C=0.0112                                                         CIN03390
        GO TO 9015                                                       CIN03400
9075    C=100.                                                           CIN03410
        GOTO 9035                                                        CIN03420
9080    L=LSC-2                                                          CIN03430
        C=CFUN(RC,AMRAL(IEQ,L),0.,AMRGM(IEQ,L))                          CIN03440
        GOTO 9020                                                        CIN03450
        END                                                              CIN03460
```

```
      SUBROUTINE INTAL(LSC,RC,C,AL,NAL1,NAL2,ICM1,ICM2,ALFA,BETA,GAMA,  INT00010
     *AMRAL,AMRGM)                                                      INT00020
C                                                                       INT00030
C  INTERPOLATION ON THE SURFACE IN C, RC, AMBIENT ILLUMINATION SPACE.   INT00040
C     THE USER INPUTS A VALUE OF RC, AND AN AMBIENT ILLUMINATION THAT   INT00050
C     LIES BETWEEN ONE PAIR OF MODELED RC VS C CURVES AT TWO DISCRETELY INT00060
C     MODELED AMBIENT ILLUMINATIONS. THE TECHNIQUE IS COMPLICATED BY A  INT00070
C     REQUIREMENT TO REPRODUCE ALMOST EXACTLY THE INTERPOLATED VALUE    INT00080
C     FROM RC VS C OF THE NVL TARGET ACQUISITION MODEL.  THUS, WHILE    INT00090
C     RC IS INPUT IN THIS INVERSION MODEL, THE INTERPOLATION IS BETWEEN INT00100
C     RC VALUES AT CONSTANT C OVER TWO AMB. ILLUM. REGIONS. THE         INT00110
C     THRESHOLDS OR OPERATIONAL LIMITS OF THE DEVICE ARE TREATED SEPAR- INT00120
C     ATELY.  IN ALL, FOUR TYPES OF INTERPOLATION SITUATIONS ARE        INT00130
C     UTILIZED.                                                         INT00140
C                                                                       INT00150
      DIMENSION IRGNS(21),INDEX(21),RCL(25),RCU(25),CU(25),CL(25),      INT00160
     *IEQ1(36),IEQ2(36),RLW(36),RUP(36),CLW(36),CUP(36),ALFA(9),        INT00170
     *BETA(9),GAMA(9),AMRAL(6,3),AMRGM(6,3)                             INT00180
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUINT00190
C** NUMBER OF C REGIONS FOR INTERPOLATION CONSIDERATION, DEV. 1-5      INT00200
      DATA IRGNS /4,4,3,3,2,2,1,1,1,2,1,1,1,1,2,1,1,1,1,2,1/            INT00210
C** STARTING INDEX FOR EACH DEVICE, AMB.ILLUM. PAIR.                  INT00220
      DATA INDEX /1,5,9,12,15,17,19,20,21,22,24,25,26,27,28,30,31,      INT00230
     *32,33,34,36/                                                      INT00240
C** LOWER LIGHT LEVEL THRESHOLDS OF RC.                               INT00250
      DATA RCL  /                                                       INT00260
     *.0001 ,.26795,.49585,.313  ,0.    ,.0309 ,.05   ,.7497 ,.3988 ,  INT00270
     *.15965,.055  ,.442  ,.1742 ,1.199 ,.6402 ,.2449 ,.0879 ,.4394 ,  INT00280
     *.1605 ,.9189 ,.5903 ,.288  ,.1294 ,.4949 ,.1894 /                 INT00290
C** UPPER LIGHT LEVEL THRESHOLDS OF RC                                INT00300
      DATA RCU  /                                                       INT00310
     *2.74  ,2.74  ,2.29  ,1.5219,1.1959,.44767,.14   ,2.941 ,2.435 ,  INT00320
     *2.106 ,2.0375,3.219 ,3.219 ,5.218 ,4.277 ,4.124 ,3.42  ,8.579 ,  INT00330
     *8.597 ,2.806 ,2.701 ,2.48  ,1.627 ,2.696 ,3.2898/                 INT00340
C** VALUE OF C AT LOWER THRESHOLDS.                                   INT00350
      DATA CL   /                                                       INT00360
     *.015 ,.025 ,.03  ,.05  ,.0841,.18  ,.50  ,.02  ,.02  ,.02  ,     INT00370
     *.02  ,.57  ,.57  ,.02  ,.02  ,.02  ,.02  ,.33  ,.33  ,.02  ,     INT00380
     *.02  ,.02  ,.02  ,.33  ,.33  /                                    INT00390
C** VALUE OF C AT UPPER THRESHOLD                                     INT00400
      DATA CU   /                                                       INT00410
     *.80  ,.80  ,.6324,.70  ,.80  ,.80  ,.80  ,100. ,100. ,100. ,     INT00420
     *100. ,100. ,16.88,100. ,100. ,100. ,100. ,100. ,16.17,100. ,     INT00430
     *100. ,100. ,100. ,100. ,100. /                                    INT00440
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR UPPER AMB. ILL.    INT00450
      DATA IEQ1 /2,2,1,1,4,4,3,3,5,5,0,6,6,0,7,7,8,8,1,2,3,4,4,5,1,2,3, INT00460
     *          4,4,5,1,2,3,4,4,5/                                     INT00470
C** INDEX OF MODEL EQUATION COEFFICIENT INDICES FOR LOWER AMB ILL.     INT00480
      DATA IEQ2 /-1,4,4,3,-1,5,5, 0,-1,6,6,-1,7,7,-1,8,-1,9,2,3,4,-1,5, INT00490
     *          6,2,3,4,-1,5,6,2,3,4,-1,5,6/                           INT00500
C** LOWEST RC VALUE IN EACH REGION.                                    INT00510
      DATA RLW  /                                                       INT00520
     *0.    ,.26794,2.118 ,2.133 ,.26794,.4958 ,1.8439,2.29  ,.312  ,  INT00530
     *.312  ,1.5119,0.    ,0.    ,1.1758,0.    ,.03089,.03089,.05  ,  INT00540
     *.3987 ,.15964,.054  ,.054  ,.442  ,.174  ,.64   ,.244  ,.087 ,  INT00550
     *.087  ,.4394 ,.1605 ,.59   ,.287  ,.129  ,.129  ,.4948 ,.1894 /  INT00560
C** LARGEST RC VALUE IN EACH REGION.                                   INT00570
      DATA RUP  /                                                       INT00580
     *.67890,2.1335,2.1637,2.74  ,.61280,2.133 ,2.5764,2.74  ,.739  ,  INT00590
     *2.29  ,2.29  ,.84083,1.5219,1.5219,.712  ,1.1959,.3751 ,.4477 ,  INT00600
     *2.942 ,2.435 ,2.106 ,.9023 ,3.219 ,3.219 ,5.218 ,4.277 ,4.124 ,  INT00610
     *1.0396,8.597 ,8.597 ,2.81  ,2.71  ,2.49  ,.9575 ,2.696 ,3.29 /   INT00620
C** LOWEST VALUE OF C OVER EACH REGION.                               INT00630
      DATA CLW /                                                        INT00640
     *.015  ,.025  ,.250  ,.270  ,.025  ,.030  ,.270  ,.6324,.030  ,  INT00650
     *.050  ,.6324 ,.050  ,.084107,.70  ,.0841 ,.180  ,.180  ,.500  ,  INT00660
     *.020  ,.020  ,.020  ,.020  ,.570  ,.570  ,.020  ,.020  ,.020  ,  INT00670
     *.020  ,.330  ,.330  ,.020  ,.020  ,.020  ,.020  ,.330  ,.330  /  INT00680
C** LARGEST VALUE OF C OVER EACH REGION.                              INT00690
      DATA CUP  /                                                       INT00700
```

```
     *.025  ,.250  ,.270  ,.800  ,.030  ,.270  ,.6324 ,.800  ,.050  ,    INT00710
     *.6324 ,.700  ,.084107,.70  ,.800  ,.180  ,.800  ,.500  ,.800  ,    INT00720
     *100.  ,100.  ,100.  ,.570  ,100.  ,16.88 ,100.  ,100.  ,100.  ,    INT00730
     *.330  ,100.  ,16.17 ,100.  ,100.  ,100.  ,.330  ,100.  ,100.  /    INT00740
C** SET LIGHT LEVELS, FRACTIONAL INTERPOLANT.                              INT00750
      ICM1=0                                                               INT00760
      ICM2=0                                                               INT00770
      IF (NAL1.EQ.NAL2) GOTO 10                                            INT00780
      IF (LSC.GT.5) GOTO 10                                                INT00790
      AL1=10.**(3-NAL1)                                                    INT00800
      AL2=10.**(3-NAL2)                                                    INT00810
      FACTR=(AL-AL2)/(AL1-AL2)                                             INT00820
      DFAC=1.-FACTR                                                        INT00830
      GOTO (100,100,200,200,200),LSC                                       INT00840
10    CALL CINV(LSC,RC,NAL1,C,ICM1,ALFA,BETA,GAMA,AMRAL,AMRGM)             INT00850
20    NAL2=0                                                               INT00860
      RETURN                                                               INT00870
30    CALL CINV(LSC,RC,NAL2,C,ICM2,ALFA,BETA,GAMA,AMRAL,AMRGM)             INT00880
40    NAL1=0                                                               INT00890
      RETURN                                                               INT00900
C** DEVICES 1,2                                                            INT00910
100   IF (NAL2.LE.7) GOTO 105                                              INT00920
      IF (NAL1.EQ.7.AND.FACTR.GT.0.75) GOTO 10                             INT00930
      NAL1=0                                                               INT00940
      GOTO 20                                                              INT00950
105   NRG=IRGNS(NAL1)                                                      INT00960
      IDX=INDEX(NAL1)                                                      INT00970
      LIM1=NAL1                                                            INT00980
      LIM2=NAL2                                                            INT00990
      GOTO 500                                                             INT01000
C** DEVICES 3,4,5                                                          INT01010
200   IF (NAL1.GE.4) GOTO 205                                              INT01020
      IF (NAL2.EQ.4.AND.FACTR.LT.0.3) GOTO 30                              INT01030
      NAL2=0                                                               INT01040
      GOTO 40                                                              INT01050
205   IS=6+(LSC-3)*5+(NAL1-3)                                              INT01060
      NRG=IRGNS(IS)                                                        INT01070
      IDX=INDEX(IS)                                                        INT01080
      LIM1=7+(LSC-3)*6+(NAL1-3)                                            INT01090
      LIM2=LIM1+1                                                          INT01100
      BVAL=0.                                                              INT01110
      IDV=LSC-2                                                            INT01120
C** FIRST, CHECK LIMIT.                                                    INT01130
500   IF (RC.GE.RCL(LIM1).AND.RC.GE.RCL(LIM2)) GOTO 510                    INT01140
      CALL CASE3(FACTR,CNOT,RCN,RCL(LIM1),RCL(LIM2),CL(LIM1),CL(LIM2))     INT01150
      IF (RC.GT.RCN) GOTO 520                                              INT01160
      CV=CNOT                                                              INT01170
      ICM1=-1                                                              INT01180
      ICM2=-1                                                              INT01190
      GOTO 800                                                             INT01200
510   IF (RC.LE.RCU(LIM1).AND.RC.LE.RCU(LIM2)) GOTO 520                    INT01210
      CALL CASE3(FACTR,CNOT,RCN,RCU(LIM1),RCU(LIM2),CU(LIM1),CU(LIM2))     INT01220
      IF (RC.LT.RCN) GOTO 520                                              INT01230
      CV=CNOT                                                              INT01240
      ICM1=1                                                               INT01250
      ICM2=1                                                               INT01260
      GOTO 800                                                             INT01270
C** RC IS IN BOUNDS OF OPERATIONAL LIMITS, BUT NOT NECESSARILLY            INT01280
C     BETWEEN CURVES.  TEST SUB-REGIONS IN TURN.                           INT01290
520   DO 600 I=1,NRG                                                       INT01300
      J=IDX+I-1                                                            INT01310
      IF (RC.LT.RLW(J).OR.RC.GT.RUP(J)) GOTO 600                           INT01320
C** RC IS IN SUB-REGION. NOW TEST C                                        INT01330
      CV=0.                                                                INT01340
      CR1=0.                                                               INT01350
      CR2=0.                                                               INT01360
      IQ1=IEQ1(J)                                                          INT01370
      IQ2=IEQ2(J)                                                          INT01380
      IF (IQ1.GT.0) GOTO 530                                               INT01390
      RCN=RCU(LIM1)                                                        INT01400
```

```
      CNOT=CU(LIM1)                                                          INT01410
      IF (IQ1.LT.0) RCN=RCL(LIM1)                                            INT01420
      IF (IQ1.LT.0) CNOT=CL(LIM1)                                            INT01430
      IF (LSC.LE.2) CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ2),       INT01440
     *BETA(IQ2),GAMA(IQ2))                                                   INT01450
      IF (LSC.GT.2)CALL CASE2(FACTR,RC,CV,IFLG,CP,RCN,CNOT,                  INT01460
     *AMRAL(IQ2,IDV),BVAL,AMRGM(IQ2,IDV))                                    INT01470
      IF (IFLG.NE.0) GOTO 600                                                INT01480
      IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600                             INT01490
      IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600                             INT01500
      IF (RCN.EQ.RCU(LIM1)) ICM1=1                                           INT01510
      IF (RCN.EQ.RCL(LIM1)) ICM1=-1                                          INT01520
      GOTO 800                                                               INT01530
530   IF (IQ2.GT.0) GOTO 540                                                 INT01540
      RCN=RCU(LIM2)                                                          INT01550
      CNOT=CU(LIM2)                                                          INT01560
      IF (IQ2.LT.0) RCN=RCL(LIM2)                                            INT01570
      IF (IQ2.LT.0) CNOT=CL(LIM2)                                            INT01580
      IF (LSC.LE.2) CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,ALFA(IQ1),        INT01590
     *BETA(IQ1),GAMA(IQ1))                                                   INT01600
      IF (LSC.GT.2)CALL CASE2(DFAC,RC,CV,IFLG,CP,RCN,CNOT,                   INT01610
     *AMRAL(IQ1,IDV),BVAL,AMRGM(IQ1,IDV))                                    INT01620
      IF (IFLG.NE.0) GOTO 600                                                INT01630
      IF (CV.GT.CUP(J).OR.CV.LT.CLW(J)) GOTO 600                             INT01640
      IF (CP.GT.CUP(J).OR.CP.LT.CLW(J)) GOTO 600                             INT01650
      IF (RCN.EQ.RCU(LIM2)) ICM2=1                                           INT01660
      IF (RCN.EQ.RCL(LIM2)) ICM2=-1                                          INT01670
      GOTO 800                                                               INT01680
540   IF(LSC.LE.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,ALFA(IQ1),ALFA(IQ2),      INT01690
     *BETA(IQ1),BETA(IQ2),GAMA(IQ1),GAMA(IQ2))                               INT01700
      IF(LSC.GT.2)CALL CASE1(FACTR,RC,CR1,CR2,IFLG,AMRAL(IQ1,IDV),           INT01710
     *AMRAL(IQ2,IDV),BVAL,BVAL,AMRGM(IQ1,IDV),AMRGM(IQ2,IDV))                INT01720
      IF (IFLG.GE.2) GOTO 600                                                INT01730
      CV=CR1                                                                 INT01740
      IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800                            INT01750
      IF (IFLG.EQ.1) GOTO 600                                                INT01760
      CV=CR2                                                                 INT01770
      IF (CV.LE.CUP(J).AND.CV.GE.CLW(J)) GOTO 800                            INT01780
600   CONTINUE                                                               INT01790
C**  NO VALUES MET INTERPOLATION CRITERIA.                                   INT01800
C**  FINAL TEST IS IN THE LIMITING REGIONS                                   INT01810
640   CONTINUE                                                               INT01820
      WRITE (IOOUT,1000) LSC,AL1,AL2                                         INT01830
1000  FORMAT(1X,40H*** IN ITAM INTAL ROUTINE, DEVICE LSC = ,I2,              INT01840
     */1X,48H*** COULD NOT INTERPOLATE BETWEEN AMBIENT ILLUM ,F11.6,         INT01850
     *5H AND ,F11.6/1X,39H*** UPPER AMBIENT ILLUM. VALUE ASSUMED.)           INT01860
      GOTO 10                                                                INT01870
800   C=CV                                                                   INT01880
      RETURN                                                                 INT01890
      END                                                                    INT01900
```

```
      SUBROUTINE CASE1(FACTR,RC,CR1,CR2,IFLG,ALP1,ALP2,BET1,BET2,
     *GAM1,GAM2)
C
C FINDS CONTRAST C INTERPOLATING ON RC VS C.  RETURNS AT MOST TWO
C  ROOTS CR1, CR2. IFLG=1 IF ROOTS ARE IDENTICAL, IFLG=2 IF ROOTS
C  ARE COMPLEX.
C
      DFAC=1.-FACTR
      A1=RC-GAM1*FACTR-GAM2*DFAC
      A2=RC*(ALP1+ALP2)+(BET1-ALP2*GAM1)*FACTR+(BET2-ALP1*GAM2)*DFAC
      A3=ALP1*ALP2*RC+ALP2*BET1*FACTR+ALP1*BET2*DFAC
      IFLG=2
      IF (A1.LT.-1.E-10.OR.A1.GT.1.E-10) GOTO 10
      IF (A2.EQ.0.) RETURN
      IFLG=1
      CR1=-A3/A2
      CR2=CR1
      RETURN
10    DISCR=A2*A2-4.*A1*A3
      IF (DISCR.LT.0.) RETURN
      IF (DISCR.EQ.0.) IFLG=1
      IF (DISCR.GT.0.) IFLG=0
      VA=SQRT(DISCR)/(2.*A1)
      VB=-A2/(2.*A1)
      SG=-1.
      IF (A1.LT.0.) SG=1.
      CR1=VB+SG*VA
      CR2=VB-SG*VA
      RETURN
      END
```

```
      SUBROUTINE CASE2(FACTR,RC,C,IFLG,CP,RNOT,CNOT,ALP,BET,GAM)          CSB00010
C                                                                         CSB00020
C FINDS INTERPOLATED CONTRAST C FOR CASE OF SINGLE FIXED RC AT            CSB00030
C SOME LIMIT.                                                             CSB00040
C                                                                         CSB00050
      IFLG=1                                                              CSB00060
      IF (FACTR.NE.1.) GOTO 10                                            CSB00070
      IF (RC.NE.RNOT) RETURN                                              CSB00080
      GOTO 90                                                             CSB00090
10    IF (RC.EQ.RNOT) GOTO 25                                             CSB00100
      DIFF=RC-RNOT*FACTR                                                  CSB00110
      IF (DIFF.NE.0.) GOTO 20                                             CSB00120
      IF (GAM.EQ.0.) RETURN                                               CSB00130
      CP=BET/GAM                                                          CSB00140
      GOTO 50                                                             CSB00150
20    VA=BET*(1.-FACTR)+ALP*DIFF                                          CSB00160
      VB=GAM*(1.-FACTR)-DIFF                                              CSB00170
      IF (VB.EQ.0.) GOTO 90                                               CSB00180
      CP=VA/VB                                                            CSB00190
      GOTO 50                                                             CSB00200
25    IF (GAM.EQ.RC) RETURN                                               CSB00210
      CP=(ALP*RC+BET)/(GAM-RC)                                            CSB00220
      GOTO 60                                                             CSB00230
50    RP=(GAM*CP-BET)/(ALP+CP)                                            CSB00240
      C=CNOT+(CP-CNOT)*(RC-RNOT)/(RP-RNOT)                                CSB00250
      GOTO 100                                                            CSB00260
60    C=CNOT*FACTR+CP*(1.-FACTR)                                          CSB00270
      GOTO 100                                                            CSB00280
90    C=CNOT                                                              CSB00290
      CP=CNOT                                                             CSB00300
100   IFLG=0                                                              CSB00310
      RETURN                                                              CSB00320
      END                                                                 CSB00330
```

```
      SUBROUTINE CASE3(FACTR,C,RCN,RNOT1,RNOT2,CNOT1,CNOT2)                CSC00010
C                                                                          CSC00020
C FINDS INTERPOLATED C FOR CASE OF BOTH RC VALUES FIXED AT                 CSC00030
C LIMITS.                                                                  CSC00040
C                                                                          CSC00050
      RCN=RNOT1*FACTR+RNOT2*(1.-FACTR)                                     CSC00060
      IF (RNOT1.EQ.RNOT2) GOTO 10                                          CSC00070
      C=CNOT1+(CNOT2-CNOT1)*(RCN-RNOT1)/(RNOT2-RNOT1)                      CSC00080
      RETURN                                                               CSC00090
10    C=CNOT1*FACTR+CNOT2*(1.-FACTR)                                       CSC00100
      RETURN                                                               CSC00110
      END                                                                  CSC00120
```

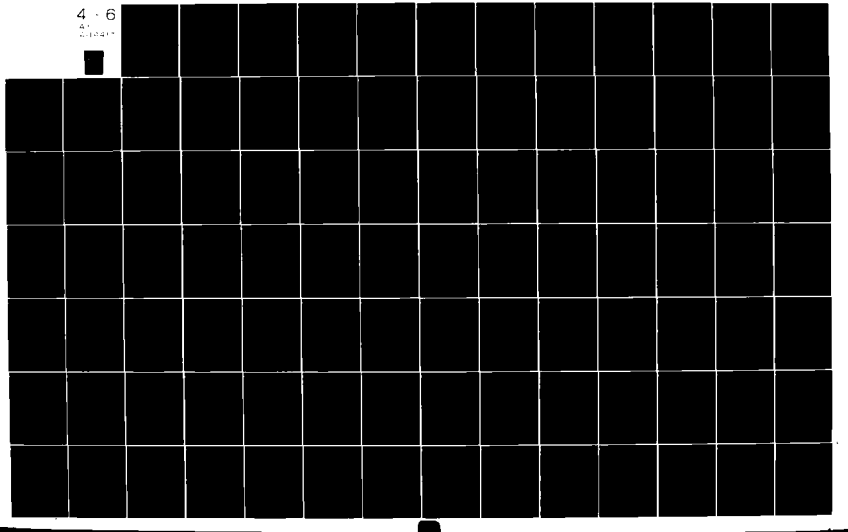
```
      SUBROUTINE CYCLE(PS,PB,AJOB,RC)                                   CYC00010
C                                                                       CYC00020
C  THIS ROUTINE COMPUTES THE NORMALIZED NUMBER OF RESOLVABLE CYCLES     CYC00030
C     FOR GIVEN CUMULATIVE NORMAL DISTRIBUTION OF PROBABILITY OF        CYC00040
C     DETECTION.                                                        CYC00050
C                                                                       CYC00060
C  INPUTS:  PS   = CUMULATIVE PROBABILITY OF DETECTION.                 CYC00070
C           PB   = STANDARD DEVIATION FOR PS (.632 )                    CYC00080
C           AJOB = MEAN RESOLVABLE CYCLES FOR 50 PERCENT PROBABILITY    CYC00090
C                  OF DETECTION.                                        CYC00100
C  OUTPUTS: RC   = RESOLVABLE CYCLES REQUIRED FOR PS                    CYC00110
C  SUBROUTINE: CYCLE CALLED BY ITAM.  CALLS NONE.                       CYC00120
C                                                                       CYC00130
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCYC00140
      DATA A0,A1,A2 /2.515517,0.802853,0.010328/                        CYC00150
      DATA B1,B2,B3 /1.432788,0.189269,0.001308/                        CYC00160
      ICASE=0                                                           CYC00170
      IF((0.0,LE.PS).AND.(PS.LE.0.0000003)) GO TO 10                    CYC00180
      IF((.9999997.LE.PS).AND.(PS.LE.1.)) GO TO 20                      CYC00190
      IF((.0000003.LT.PS).AND.(PS.LE.0.5)) GO TO 30                     CYC00200
      IF((.5.LT.PS).AND.(PS.LT.0.9999997)) GO TO 40                     CYC00210
C *****  ERROR ROUTINE                                                  CYC00220
      WRITE(IOOUT,900) PS                                               CYC00230
      RETURN                                                            CYC00240
C *****  SPECIAL CASE I                                                 CYC00250
10    X=-5.                                                             CYC00260
      IF (PS.EQ.0.) X=-45.                                              CYC00270
      GO TO 60                                                          CYC00280
C *****  SPECIAL CASE II                                                CYC00290
20    X=5.                                                              CYC00300
      IF (PS.EQ.1.) X=45.                                               CYC00310
      GO TO 60                                                          CYC00320
C *****  CASE I                                                         CYC00330
30    XX=SQRT(ALOG(1/(PS*PS)))                                          CYC00340
      ICASE=1                                                           CYC00350
      GO TO 50                                                          CYC00360
C *****  CASE II                                                        CYC00370
40    XX=SQRT(ALOG(1./(1.-PS)**2))                                      CYC00380
50    X=XX-(A0+A1*XX+A2*(XX*XX))/(1.+B1*XX+B2*(XX*XX)+B3*(XX*XX*XX))    CYC00390
      IF(ICASE.EQ.1) X=X*(-1.)                                          CYC00400
60    RC=AJOB+AJOB*PB*X                                                 CYC00410
      RETURN                                                            CYC00420
900   FORMAT(2X,39HERROR - INCORRECT PS VALUE PASSED  PS =  ,F10.4)     CYC00430
      END                                                               CYC00440
```

```
      SUBROUTINE TREQ(ACON,SOG,C,LSC,TRQ)                                TRQ00010
C                                                                        TRQ00020
C   THIS ROUTINE COMPUTES THE TRANSMISSION REQUIRED TO MATCH GIVEN       TRQ00030
C     TARGET AND DETECTOR CONTRASTS, WITH MODIFICATION BY SKY/           TRQ00040
C     GROUND RATIO FOR NON-THERMAL DEVICES.                              TRQ00050
C                                                                        TRQ00060
C   INPUTS:  ACON  = INTRINSIC CONTRAST OF TARGET, OR TEMP. DIFF.        TRQ00070
C                    (DEG K) FOR THERMAL.                                TRQ00080
C            SOG   = SKY/GROUND RATIO.                                   TRQ00090
C            C     = CONTRAST OR TEMP. DIFF. SEEN AT DEVICE              TRQ00100
C            LSC   = DEVICE NUMBER.                                      TRQ00110
C   OUTPUTS: TRQ   = REQUIRED TRANSMISSION TO REDUCE ACON TO C.          TRQ00120
C   SUBROUTINE: TREQ CALLED BY ITAM.  CALLS NONE.                        TRQ00130
C                                                                        TRQ00140
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUTRQ00150
      IF((LSC.LT.1).OR.(LSC.GT.14)) GO TO 30                             TRQ00160
      IF(LSC.GT.5) GO TO 20                                              TRQ00170
C ***** NON-THERMAL                                                      TRQ00180
10    IF (SOG.EQ.0. .OR. C.EQ.0.) GOTO 45                                TRQ00190
      TRQ=1./(1.+((ACON-C)/(C*SOG)))                                     TRQ00200
      GOTO 50                                                            TRQ00210
20    IF((LSC.EQ.10).OR.(LSC.EQ.12)) GO TO 10                            TRQ00220
      IF(LSC.EQ.13) GO TO 30                                             TRQ00230
C ***** THERMAL                                                          TRQ00240
      IF (ACON.EQ.0.) GOTO 25                                            TRQ00250
      TRQ=C/ACON                                                         TRQ00260
      GOTO 50                                                            TRQ00270
25    TRQ=1.                                                             TRQ00280
      GOTO 50                                                            TRQ00290
C ***** ERROR ROUTINE                                                    TRQ00300
30    WRITE(IOOUT,100) LSC                                               TRQ00310
      TRQ=1.                                                             TRQ00320
100   FORMAT(5X,19HINPUT ERROR  LSC = ,I2)                               TRQ00330
      RETURN                                                             TRQ00340
45    TRQ=0.                                                             TRQ00350
50    IF (TRQ.LT.0.) TRQ=0.                                              TRQ00360
      IF (TRQ.GT.1.) TRQ=1.                                              TRQ00370
      RETURN                                                             TRQ00380
      END                                                                TRQ00390
```

```
      SUBROUTINE FCLOUD(LAMBDA,TRANS,IERR)                                  FCL00010
      REAL LAMBDA,KAPPA,LBO,LP                                              FCL00020
      REAL L,M,N,L0,M0,N0,L0S,M0S,N0S,L0M,L0N,N0M,MXLY,NXLZ,MZNY            FCL00030
      DIMENSION IALPH(10)                                                   FCL00040
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUFCL00050
      COMMON /FGEOM/XC,YC,ZC,AE,BE,CE,XR,YR,ZR,XS,YS,ZS                     FCL00060
      COMMON /OPT/ INDEXP,ETA,KAPPA,W0,THETA0,PHI0,LD,TAUBAR,               FCL00070
     +RHO,LB0,TMPA,TMPC                                                     FCL00080
      COMMON /GEOMET/PTS(15),IGEOSW                                         FCL00090
      DATA IZERO/0/                                                         FCL00100
      DATA IALPH/2HCP   ,2HRP   ,2HSP   ,2HAX   ,2HCL   ,2HAT   ,2HBK   ,  FCL00110
     +2HSA   ,2HLU   ,2HGO   /                                              FCL00120
      ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)
C*****************************************************************************FCL00130
C                                                                             FCL00140
C  SUBROUTINE FCLOUD COMPUTES THE FOLLOWING QUANTITIES FOR A FINITE           FCL00150
C  ELLIPSOIDAL CLOUD:                                                         FCL00160
C                                                                             FCL00170
C     S = LENGTH OF OPTICAL PATH IN CLOUD                                     FCL00180
C     TRANS = BEAM TRANSMITTANCE THROUGH CLOUD                                FCL00190
C     LP = PATH RADIANCE                                                      FCL00200
C     TC = CONTRAST TRANSMITTANCE.                                            FCL00210
C                                                                             FCL00220
C  INPUT OF COMPUTATIONAL PARAMETERS TAKES PLACE THROUGH AN ORDER             FCL00230
C  INDEPENDENT READ OF A GROUP OF RECORDS IDENTIFIED BY THE FOLLOWING         FCL00240
C  MNEMONICS (EACH RECORD IS FORMAT (A4,1X,5(E10.4,1X)) ) :                   FCL00250
C                                                                             FCL00260
C  MNEUMONIC       VARIABLES READ          DESCRIPTION                       FCL00270
C  ---------       --------------          -----------                       FCL00280
C                                                                             FCL00290
C    CPOS            XC,YC,ZC            CLOUD CENTER POSITION               FCL00300
C    RPOS            XR,YR,ZR            RECEIVER POSITION                   FCL00310
C    SPOS            XS,YS,ZS            SOURCE POSITION                     FCL00320
C    AXES            AE,BE,CE            SEMI-AXES OF CLOUD ELLIPSOID        FCL00330
C                                                                             FCL00340
C    CLDS   INDEXP,ETA,KAPPA,W0,TMPC     CLOUD AEROSOL PARAMETERS            FCL00350
C    ATMO            TAUBAR,TMPA         ATMOSPHERIC PARAMETERS              FCL00360
C    BKGR            RHO,LB0             BACKGROUND PARAMETERS               FCL00370
C    SANG            THETA0,PHI0         SOLAR ANGLES                        FCL00380
C    LUND            LD                  LUNAR DAY                           FCL00390
C                                                                             FCL00400
C    GO                                  TERMINATES READ                     FCL00410
C                                                                             FCL00420
C  ** NOTE : THE GO SENTINEL CARD MUST BE THE LAST CARD READ                  FCL00430
C                                                                             FCL00440
C  THIS INPUT DATA IS STORED FOR LATER USE IN COMMON                          FCL00450
C  BLOCKS /FGEOM/ AND /OPT/.                                                  FCL00460
C                                                                             FCL00470
C  /FGEOM/ INPUT PARAMETERS:                                                  FCL00480
C                                                                             FCL00490
C     (XC,YC,ZC) = CENTER OF ELLIPSOID                                        FCL00500
C     (AE,BE,CE) = SEMI-AXES OF ELLIPSOID                                     FCL00510
C     (XR,YR,ZR) = COORDINATES OF RECEIVER LOCATION                           FCL00520
C     (XS,YS,ZS) = COORDINATES OF SOURCE LOCATION.                            FCL00530
C                                                                             FCL00540
C  /OPT/ INPUT PARAMETERS:                                                    FCL00550
C                                                                             FCL00560
C     INDEXP = PHASE FUNCTION IDENTIFIER                                      FCL00570
C                  =0,  USER SUPPLIED                                         FCL00580
C                  =1,  MARITIME ARCTIC, VIS=0.1 TO 2.0 KM                    FCL00590
C                  =2,  MARITIME POLAR, VIS=0.2 KM                            FCL00600
C                  =3,  MARITIME POLAR, VIS=02. KM                            FCL00610
C                  =4,  CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM                 FCL00620
C                  =5,  WHITE PHOSPHORUS                                      FCL00630
C                  =6,  HEXACHLOROETHANE                                      FCL00640
C                  =7,  FOG OIL                                               FCL00650
C                  =8,  DUST (MODERATE AEROSOL LOADING)                       FCL00660
C                  =9,  DUST (HEAVY AEROSOL LOADING)                          FCL00670
C                  =10, MARITIME MODEL B, VIS=5KM, RH=95%                     FCL00680
C                  =11, MARITIME MODEL B, VIS=10KM,RH=90%                     FCL00690
```

```
C                    =12, MARITIME MODEL B, VIS=50KM,RH=50%                    FCL00700
C     ETA = FORWARD SCATTERING PARAMETER                                       FCL00710
C           ETA MAY ALSO BE CALCULATED BY DEFAULT (I.E. INPUT ETA =0.0),
C           IN THIS CASE INDEXP SHOULD BE THE NEGATIVE OF THE PHASE
C           FUNCTION FOR WHICH ETA IS DESIRED.
C     KAPPA = VOLUME EXTINCTION COEFFICIENT (KM-1)                             FCL00720
C     W0 = SINGLE SCATTERING ALBEDO                                            FCL00730
C     (THETA0,PHI0) = SOLAR ANGLES (DEGREES)                                   FCL00740
C     LD = LUNAR DAY (INTEGER BETWEEN 0 AND 28)                                FCL00750
C     TAUBAR = ATMOSPHERIC OPTICAL THICKNESS ABOVE CLOUD                       FCL00760
C     RHO = BACKGROUND SURFACE ALBEDO                                          FCL00770
C     LB0 = BACKGROUND RADIANCE                                                FCL00780
C     TMPA = TEMPERATURE OF ATMOSPHERE (DEG.C)                                 FCL00790
C     TMPC = TEMPERATURE OF CLOUD (DEG.C).                                     FCL00800
C                                                                              FCL00810
C  ALL LENGTH UNITS ARE KM.  PROGRAM FLOW IS CONTROLLED BY THE                 FCL00820
C  VARIABLES TMPA, INDEXP, AND, IMPLICITLY, BY ISW .                           FCL00830
C  IF TMPA >= -99.0, THIS SPECIFIES                                            FCL00840
C  A THERMAL COMPUTATION, WHICH IS PERFORMED IN SUBROUTINE THRMCL:             FCL00850
C  IF TMPA < -99.0 THEN A SCATTERING COMPUTATION IS PERFORMED.                 FCL00860
C  IF INDEXP < 0,  THIS IS A MULTIPLE SCATTERING COMPUTATION, WHICH            FCL00870
C  IS DONE IN SUBROUTINE MSCLD; IF ETA HAS BEEN INPUT AS ZERO, THEN            FCL00880
C     INDEXP SHOULD BE THE NEGATIVE OF THE PHASE FUNCTION INDENTIFIER,
C     SO THAT ETA WILL BE FOUND FROM THE PROPER PHASE FUNCTION.
C     IF INDEXP > 0 A SINGLE SCATTERING
C  COMPUTATION IS CARRIED OUT IN SUBROUTINE SSCLD.  IN THIS CASE               FCL00890
C  INDEXP ALSO SPECIFIES THE PHASE FUNCTION TO BE USED, WITH                   FCL00900
C  INDEXP = I SELECTING THE I-TH PHASE FUNCTION.  THE VALUE OF                 FCL00910
C  ISW OCCURRING IN THE SUBROUTINE PARAMETER LIST INDICATES WHETHER            FCL00920
C  CERTAIN PARAMETERS ARE THE SAME AS IN THE PREVIOUS CALL TO FCLOUD,          FCL00930
C  AS FOLLOWS.                                                                 FCL00940
C                                                                              FCL00950
C  ISW IS SET TO 0 WHEN THE FOLLOWING CONDITIONS ARE ENCOUNTERED :             FCL00960
C      A) ALL 9 DATA CARDS ARE READ                                            FCL00970
C      B) ANY ONE OF CARDS 1-4 AND ANY ONE OF CARDS 5-9                        FCL00980
C         (AS LISTED IN THE ORDER ABOVE) ARE READ                              FCL00990
C  ISW IS SET TO 2 IF NONE OF CARDS 1-4 IS READ.                               FCL01000
C  ISW IS SET TO 1 IF NONE OF CARDS 5-9 IS READ.                               FCL01010
C  ISW DEFAULTS TO 2 IF NOTHING IS READ (OTHER THAN THE GO CARD).              FCL01020
C                                                                              FCL01030
C      ISW = 2 => PARAMETERS IN COMMON /FGEOM/ ARE THE SAME AS IN              FCL01040
C                 PREVIOUS CALL: SKIP PRELIMINARY GEOMETRICAL                  FCL01050
C                 CALCULATIONS                                                 FCL01060
C      ISW = 1 => PARAMETERS IN COMMON /OPT/ ARE THE SAME AS IN                FCL01070
C                 PREVIOUS CALL: SKIP COMPUTATIONS INVOLVING                   FCL01080
C                 ONLY THESE PARAMETERS                                        FCL01090
C      ISW = 0 => NEW PARAMETERS IN BOTH /FGEOM/ AND /OPT/:                    FCL01100
C                 NO CALCULATIONS ARE SKIPPED IN THIS CASE.                    FCL01110
C                                                                              FCL01120
C  SUBROUTINES CALLED FROM FCLOUD:                                             FCL01130
C                                                                              FCL01140
C     THRMCL           FOR THERMAL CALCULATION OF PATH RADIANCE                FCL01150
C     MSCLD            FOR MULTIPLE SCATTERING CALCULATION OF                  FCL01160
C                      PATH RADIANCE                                           FCL01170
C     SSCLD            FOR SINGLE SCATTERING CALCULATION OF PATH               FCL01180
C                      RADIANCE                                                FCL01190
C     ILLUM            COMPUTES THE EXTRATERRESTRIAL SOLAR OR                  FCL01200
C                      LUNAR IRRADIANCE.  NOT NEEDED FOR THERMAL               FCL01210
C                      CALCULATIONS                                            FCL01220
C     PFN              RETURNS A PHASE FUNCTION VALUE FOR USE BY               FCL01230
C                      SSCLD.                                                  FCL01240
C                                                                              FCL01250
C  GROUND RADIANCE CAN BE SPECIFIED IN ONE OF TWO WAYS:                        FCL01260
C     1.)  BY GIVING LB0 A NONZERO VALUE, OR                                   FCL01270
C     2.)  AS REFLECTED SOLAR IRRADIANCE, USING THE VALUE OF                   FCL01280
C          RHO INPUT TO THE PROGRAM.                                           FCL01290
C                                                                              FCL01300
C  THE VALUE OF LD SPECIFIES WHETHER IT IS DAY OR NIGHT, I. E.                 FCL01310
C  WHETHER SOLAR OR LUNAR EXTRATERRESTRIAL IRRADIANCE IS TO BE                 FCL01320
C  USED.  IF LD = 0 THEN SOLAR IRRADIANCE IS CALCULATED, AND IF                FCL01330
```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF
UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL



```
C   1 < LD < 28 THE IRRADIANCE VALUE IS FOR LUNAR DAY LD.                    FCL01340
C                                                                            FCL01350
C*************************************************************************FCL01360
C                                                                            FCL01370
C   INITIALIZATION OF INPUT DATA.                                            FCL01380
C                                                                            FCL01390
      IF (IZERO.NE.0) WRITE (IOOUT,99)
99    FORMAT(1H1)
      IF(IZERO.NE.0) GO TO 477                                               FCL01400
      XC=0.0                                                                 FCL01410
      YC=0.0                                                                 FCL01420
      ZC=0.0                                                                 FCL01430
      XR=0.0                                                                 FCL01440
      YR=0.0                                                                 FCL01450
      ZR=0.0                                                                 FCL01460
      XS=0.0                                                                 FCL01470
      YS=0.0                                                                 FCL01480
      ZS=0.0                                                                 FCL01490
      AE=0.0                                                                 FCL01500
      BE=0.0                                                                 FCL01510
      CE=0.0                                                                 FCL01520
      INDEXP=0                                                               FCL01530
      ETA=0.0                                                                FCL01540
      KAPPA=0.0                                                              FCL01550
      W0=0.0                                                                 FCL01560
      TMPC=0.0                                                               FCL01570
      TAUBAR=0.0                                                             FCL01580
      TMPA=0.0                                                               FCL01590
      RHO=0.0                                                                FCL01600
      LB0=0.0                                                                FCL01610
      THETA0=0.0                                                             FCL01620
      PHI0=0.0                                                               FCL01630
      LD=0                                                                   FCL01640
      ISW=0                                                                  FCL01650
      IZERO=1                                                                FCL01660
  477 CONTINUE                                                               FCL01670
      IFLG=2                                                                 FCL01680
      IFLO=1                                                                 FCL01690
      DO 360 K=1,10                                                          FCL01700
      READ(IOIN,334)IA,IA2,R1,R2,R3,R4,R5                                      FCL0
  334 FORMAT(2A2,1X,5(E10.4,1X))                                             FCL0172
      DO 333 I=1,11                                                          FCL01730
      IF(IA.NE.IALPH(I)) GO TO 333                                           FCL01740
      IND=I                                                                  FCL01750
      IF(IND.EQ.10) GO TO 361                                                FCL01760
  333 CONTINUE                                                               FCL01770
      IF(K.EQ.10.AND.IND.NE.10) GO TO 358                                    FCL01780
      IF(IND.LT.5)IFLG=0                                                     FCL01790
      IF(IND.GE.5.AND.IND.LE.9)IFLO=0                                        FCL01800
      IF(IND.EQ.11) GO TO 355                                                FCL01810
      IF(IND.LT.5) GO TO (341,342,343,344),IND                               FCL01820
      INDM4=IND-4                                                            FCL01830
      GO TO (345,346,347,348,349),INDM4                                      FCL01840
  341 XC=R1                                                                  FCL01850
      YC=R2                                                                  FCL01860
      ZC=R3                                                                  FCL01870
      GO TO 360                                                              FCL01880
  342 XR=R1                                                                  FCL01890
      YR=R2                                                                  FCL01900
      ZR=R3                                                                  FCL01910
      GO TO 360                                                              FCL01920
  343 XS=R1                                                                  FCL01930
      YS=R2                                                                  FCL01940
      ZS=R3                                                                  FCL01950
      GO TO 360                                                              FCL01960
  344 AE=R1                                                                  FCL01970
      BE=R2                                                                  FCL01980
      CE=R3                                                                  FCL01990
      GO TO 360                                                              FCL02000
  345 INDEXP=IFIX(R1)                                                        FCL02010
```

```
      IF (R2.GT..001) ETA=R2                                              FCL02020
      KAPPA=R3                                                            FCL02030
      W0=R4                                                               FCL02040
      TMPC=R5                                                             FCL02050
      GO TO 360                                                           FCL02060
  346 TAUBAR=R1                                                           FCL02070
      TMPA=R2                                                             FCL02080
      GO TO 360                                                           FCL02090
  347 RHO=R1                                                              FCL02100
      LB0=R2                                                              FCL02110
      GO TO 360                                                           FCL02120
  348 THETA0=R1                                                           FCL02130
      PHI0=R2                                                             FCL02140
      GO TO 360                                                           FCL02150
  349 LD=IFIX(R1)                                                         FCL02160
      GO TO 360                                                           FCL02170
  355 WRITE(IOOUT,357)                                                    FCL02180
  357 FORMAT(1H0,20X,46H***FCLOUD ERROR*** INPUT RECORD DETECTED WHICH,   FCL02190
     + 44HDOES NOT CORRESPOND TO CORRECT INPUT FORMAT,/)                   FCL02200
      GO TO 360                                                           FCL02210
  358 WRITE(IOOUT,359)                                                    FCL02220
  359 FORMAT(1H0,35X,45H***FCLOUD ERROR*** TOO MANY INPUT CARDS OR GO,    FCL02230
     + 16H SENTINEL ABSENT,/)                                             FCL02240
      IERR=1                                                              FCL02250
      GO TO 200                                                           FCL02260
  360 CONTINUE                                                            FCL02270
  361 CONTINUE                                                            FCL02280
      ISW=IFLG+IFLO                                                       FCL02290
      IF(ISW.EQ.3) ISW=2                                                  FCL02300
      IF(IGEOSW.NE.1) GO TO 222                                           FCL02310
      XC=PTS(13)                                                          FCL02320
      YC=PTS(14)                                                          FCL02330
      ZC=PTS(15)                                                          FCL02340
      XR=PTS(4)                                                           FCL02350
      YR=PTS(5)                                                           FCL02360
      ZR=PTS(6)                                                           FCL02370
      XS=PTS(1)                                                           FCL02380
      YS=PTS(2)                                                           FCL02390
      ZS=PTS(3)                                                           FCL02400
  222 CONTINUE                                                            FCL02410
C                                                                         FCL02420
C  ECHO INPUT                                                             FCL02430
C                                                                         FCL02440
      WRITE(IOOUT,1000) XC,YC,ZC,AE,BE,CE,XR,YR,ZR,XS,YS,ZS               FCL02450
      IF (ETA.LT.1.E-20) CALL PFNN(LAMBDA,0.,INDEXP,PFN,ETA)              FCL02460
      WRITE(IOOUT,1100) INDEXP,ETA,LAMBDA,KAPPA,W0,TAUBAR,THETA0,PHI0,    FCL02470
     1RHO,LB0,TMPA,TMPC,LD                                                FCL02480
      IF(ISW.EQ.2) GO TO 15                                               FCL02490
C                                                                         FCL02500
C  ISW .NE. 2 INDICATES PRELIMINARY GEOMETRICAL CALCULATIONS TO           FCL02510
C  BE PERFORMED: COMPUTE INTERSECTIONS (XM,YM,ZM) AND (XN,YN,ZN) OF       FCL02520
C  LINE OF SIGHT WITH CLOUD, SBAR = LENGTH OF PATH FROM SOURCE TO         FCL02530
C  RECEIVER, S = LENGTH OF PATH IN CLOUD, AND TRANS = TRANSMITTANCE       FCL02540
C  THROUGH CLOUD                                                          FCL02550
C                                                                         FCL02560
      L=XS-XR                                                             FCL02570
      M=YS-YR                                                             FCL02580
      N=ZS-ZR                                                             FCL02590
      SBAR=SQRT(L*L+M*M+N*N)                                              FCL02600
      L=L/SBAR                                                            FCL02610
      M=M/SBAR                                                            FCL02620
      N=N/SBAR                                                            FCL02630
      DX=XS-XC                                                            FCL02640
      DY=YS-YC                                                            FCL02650
      DZ=ZS-ZC                                                            FCL02660
      ASQ=AE*AE                                                           FCL02670
      BSQ=BE*BE                                                           FCL02680
      CSQ=CE*CE                                                           FCL02690
      ABSQ=ASQ*BSQ                                                        FCL02700
      ACSQ=ASQ*CSQ                                                        FCL02710
```

```
      BCSQ=BSQ*CSQ                                                      FCL02710
      ABCSQ=ASQ*BSQ*CSQ                                                 FCL02720
      A=BCSQ*L*L+ACSQ*M*M+ABSQ*N*N                                      FCL02730
      B=2.0*(L*DX*BCSQ+M*DY*ACSQ+N*DZ*ABSQ)                             FCL02740
      C=BCSQ*DX*DX+ACSQ*DY*DY+ABSQ*DZ*DZ-ABCSQ                          FCL02750
      DISCRM=B*B-4.0*A*C                                                FCL02760
      IF(DISCRM.GE.0.0) GO TO 10                                        FCL02770
      WRITE(IOOUT,1200)                                                 FCL02780
      S=0.0                                                             FCL02790
      TRANS=1.0                                                         FCL02800
      LP=0.0                                                            FCL02810
      GO TO 90                                                          FCL02820
   10 SRTDSC=SQRT(DISCRM)                                               FCL02830
      TWOA=2.0*A                                                        FCL02840
      VPLUS=(-B+SRTDSC)/TWOA                                            FCL02850
      VMINUS=(-B-SRTDSC)/TWOA                                           FCL02860
      XM=XS+VPLUS*L                                                     FCL02870
      YM=YS+VPLUS*M                                                     FCL02880
      ZM=ZS+VPLUS*N                                                     FCL02890
      XN=XS+VMINUS*L                                                    FCL02900
      YN=YS+VMINUS*M                                                    FCL02910
      ZN=ZS+VMINUS*N                                                    FCL02920
      S=SQRT((XM-XN)*(XM-XN)+(YM-YN)*(YM-YN)+(ZM-ZN)*(ZM-ZN))           FCL02930
      IF(S.LT.1.E-4) S=0.0                                              FCL02940
   15 TRANS=EXP(-KAPPA*S)                                               FCL02950
      IF(TMPA.LT.-99.0) GO TO 20                                        FCL02960
C                                                                       FCL02970
C  TMPA >= -99.0 SPECIFIES THERMAL CALCULATION                          FCL02980
C                                                                       FCL02990
      RBAR=(AE+BE+CE)/3.0                                               FCL03000
      CALL THRMCL(RBAR,W0,TMPA,TMPC,LAMBDA,KAPPA,TRANS,LP)              FCL03010
      GO TO 90                                                          FCL03020
C                                                                       FCL03030
C  ISW .NE. 1 SPECIFIES NEW PARAMETERS IN COMMON /OPT/;                 FCL03040
C  ASSOCIATED PRELIMINARY COMPUTATIONS YIELD EXTRATERRESTRIAL            FCL03050
C  IRRADIANCE E0, COORDINATES (L0,M0,N0) OF UNIT VECTOR                 FCL03060
C  POINTING TO SUN, AND TATM = TRANSMITTANCE OF ATMOSPHERE              FCL03070
C  ABOVE CLOUD                                                          FCL03080
C                                                                       FCL03090
   20 IF(ISW.EQ.1) GO TO 30                                             FCL03100
      CALL ILLUM(LAMBDA,LD,E0)                                          FCL03110
      THT0=THETA0/57.2958                                               FCL03120
      PH0=PHI0/57.2958                                                  FCL03130
      L0=SIN(THT0)*COS(PH0)                                             FCL03140
      M0=SIN(THT0)*SIN(PH0)                                             FCL03150
      N0=COS(THT0)                                                      FCL03160
      TATM=EXP(-TAUBAR/N0)                                              FCL03170
   30 WRITE(IOOUT,1600) L,M,N,L0,M0,N0                                  FCL03180
      IF(INDEXP.GT.-1) GO TO 40                                         FCL03190
C                                                                       FCL03200
C  INDEXP <  0 SPECIFIES MULTIPLE SCATTERING COMPUTATION;               FCL03210
C  COMPUTE TAU AND TAU0 = CLOUD OPTICAL DEPTH AND THICKNESS,            FCL03220
C  AND CALL MSCLD FOR VALUE OF LP = PATH RADIANCE                       FCL03230
C                                                                       FCL03240
      XG=(XM+XN)/2.0                                                    FCL03250
      YG=(YM+YN)/2.0                                                    FCL03260
      ZG=(ZM+ZN)/2.0                                                    FCL03270
      H=CE*SQRT(1.0-(XG-XC)*(XG-XC)/(AE*AE)-(YG-YC)*(YG-YC)/(BE*BE))    FCL03280
      TAU0=2.0*H*KAPPA                                                  FCL03290
      H=ZC-ZG+H                                                         FCL03300
      TAU=KAPPA*H                                                       FCL03310
      CALL MSCLD(TAU,TAU0,TRANS,TATM,E0,W0,ETA,RHO,LP)                  FCL03320
      GO TO 90                                                          FCL03330
C                                                                       FCL03340
C  INDEXP > -1 SPECIFIES SINGLE SCATTERING COMPUTATION;                 FCL03350
C  COMPUTE SCATTERING ANGLE CHI,PHASE FUNCTION VALUE P =                FCL03360
C  PFN(LAMBDA,CHI), GEOMETRICAL PARAMETERS ALPHA, BETA, GAMMA,          FCL03370
C  DELTA, AND EPS, RGRND = GROUND RADIANCE, AND CALL SSCLD              FCL03380
C  TO COMPUTE PATH RADIANCE                                             FCL03390
C                                                                       FCL03400
```

```
   40 THETA= ACOS(N)                                                        FCL03410
      PHI=0.0                                                               FCL03420
      IF(M.NE.0.0.OR.L.NE.0.0) PHI=ATAN2(M,L)                               FCL03430
      COSCHI=N*N0+SIN(THETA)*SIN(THT0)*COS(PHI-PH0)                         FCL03440
      CHI= ACOS(COSCHI)*57.2958                                             FCL03450
      CALL PFNN(LAMBDA,CHI,INDEXP,PFN,ETA)                                  FCL03460
      P=PFN                                                                 FCL03470
      RGRND=E0*TATM*N0                                                      FCL03480
      IF(LB0.LT.1.E-10) LB0=RHO*RGRND                                       FCL03490
      DX=XM-XC                                                              FCL03500
      DY=YM-YC                                                              FCL03510
      DZ=ZM-ZC                                                              FCL03520
      L0S=L0*L0                                                             FCL03530
      M0S=M0*M0                                                             FCL03540
      N0S=N0*N0                                                             FCL03550
      L0M=L0*M-M0*L                                                         FCL03560
      L0N=L0*N-N0*L                                                         FCL03570
      N0M=N0*M-M0*N                                                         FCL03580
      MXLY=M0*DX-L0*DY                                                      FCL03590
      NXLZ=N0*DX-L0*DZ                                                      FCL03600
      MZNY=M0*DZ-N0*DY                                                      FCL03610
      DENOM=BCSQ*L0S+ACSQ*M0S+ABSQ*N0S                                      FCL03620
      ALPHA=(BCSQ*L*L0+ACSQ*M*M0+ABSQ*N*N0)/DENOM                           FCL03630
      BETA=-(BCSQ*L0*DX+ACSQ*M0*DY+ABSQ*N0*DZ)/DENOM                        FCL03640
      GAMMA=-ABCSQ*(CSQ*L0M*L0M+BSQ*L0N*L0N+ASQ*N0M*N0M)/(DENOM*DENOM)      FCL03650
      DELTA=CSQ*MXLY*L0M+BSQ*NXLZ*L0N+ASQ*MZNY*N0M                          FCL03660
      DELTA=-2.0*ABCSQ*DELTA/(DENOM*DENOM)                                  FCL03670
      EPS=BCSQ*L0S+ACSQ*M0S+ABSQ*N0S-CSQ*MXLY*MXLY-BSQ*NXLZ*NXLZ            FCL03680
     1-ASQ*MZNY*MZNY                                                        FCL03690
      EPS=ABCSQ*EPS/(DENOM*DENOM)                                           FCL03700
      IF(EPS.GE.0.0) GO TO 50                                               FCL03710
      WRITE(IOOUT,1300) EPS                                                 FCL03720
      EPS=0.0                                                               FCL03730
   50 IF(GAMMA.LT.0.0) GO TO 60                                             FCL03740
      WRITE(IOOUT,1400) GAMMA                                               FCL03750
      GO TO 200                                                             FCL03760
   60 CALL SSCLD(ALPHA,BETA,GAMMA,DELTA,EPS,S,KAPPA,W0,RGRND,P,LP)          FCL03770
C                                                                           FCL03780
C  WRITE RESULTS                                                            FCL03790
C                                                                           FCL03800
   90 F=LP/(LB0*TRANS)                                                      FCL03810
      TC=1.0/(1.0+F)                                                        FCL03820
      WRITE(IOOUT,1500) S,TRANS,LP,TC                                       FCL03830
  200 RETURN                                                                FCL03840
 1000 FORMAT(1H0,43X,45H-- RADIATIVE TRANSFER THROUGH FINITE CLOUD --/      FCL03850
     1 1H0/45X,14H(XC,YC,ZC) = (,2(F8.4,1H,),F8.4,12H) KILOMETERS/          FCL03860
     2 45X,14H(AE,BE,CE) = (,2(F8.4,1H,),F8.4,1H)/                         FCL03870
     3 45X,14H(XR,YR,ZR) = (,2(F8.4,1H,),F8.4,1H)/                         FCL03880
     4 45X,14H(XS,YS,ZS) = (,2(F8.4,1H,),F8.4,1H))                         FCL03890
 1100 FORMAT(1H0,45X,8HINDEXP =,I9,10X,8HETA    =,F11.3/                   FCL03900
     1 45X,8HLAMBDA =,F9.3,8H (MU)    ,2X,9HKAPPA  = ,1PE10.4,7H (KM-1)/   FCL03910
     2 45X,8HOMEGA 0=,0PF9.3,10X,8HTAUBAR =,1X,1PE10.4/                    FCL03920
     3 45X,8HTHETA0 =,0PF9.1,10X,8HPHI0   =,F11.1,10H (DEGREES)/           FCL03930
     4 45X,8HRHO    =,0PF9.3,10X,8HLB0    =,F11.3,13H (W/M2-SR-MU)/       FCL03940
     5 45X,8HTMPA   =,F9.1,8H (DEG.C),2X,8HTMPC   =,F11.1/                 FCL03950
     6 45X,8HLD     =,I9)                                                  FCL03960
 1200 FORMAT(1H0,44X,43H**LINE-OF-SIGHT MISSES CLOUD.  S SET TO 0.0)        FCL03970
 1300 FORMAT(1H0,44X,4HEPS=,E10.4,24H LT 0.0.  EPS SET TO 0.0)              FCL03980
 1400 FORMAT(1H0,44X,6HGAMMA=,E10.4,29H  GE 0.0.  SKIP TO NEXT CASE.)       FCL03990
 1500 FORMAT(1H0/1H0,37X,11HPATH LENGTH,3X,15HTRANSMITTANCE  ,              FCL04000
     1 13HPATH RADIANCE,4X,8HCONTRAST/                                      FCL04010
     2 38X,11H(IN CLOUD) ,19X,12H(W/M2-SR-MU),15H  TRANSMITTANCE/           FCL04020
     3 36X,4(15H+-------------+)/                                           FCL04030
     4 1H0,40X,F6.3,7X,1PE9.3,5X,1PE9.3,5X,1PE9.3)                          FCL04040
 1600 FORMAT(1H0,45X,28HUNIT SOURCE VECTOR L, M, N =,3(1X,F7.4)/            FCL04050
     1 1H0,45X,23HSOLAR VECTOR L0,M0,N0 =,3(1X,F7.4))                       FCL04060
      END                                                                   FCL04070
```

```
      SUBROUTINE THRMCL(RBAR,W0,TMPA,TMPC,LAMBDA,KAPPA,TRANS,LP)          THR00010
      REAL LAMBDA,LP,LI,KAPPA                                            THR00020
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUTHR00030
      BB(X,T)=1.19106E8/(X**5*(EXP(1.4388E4/(X*T))-1.0))                 THR00040
C*****************************************************************     THR00050
C                                                                        THR00060
C   SUBROUTINE PERFORMS THERMAL RADIATION CALCULATIONS FOR FINITE        THR00070
C   CLOUD, RETURNING THE VALUE OF LP = PATH RADIANCE.                    THR00080
C   INPUTS ARE:                                                          THR00090
C                                                                        THR00100
C      RBAR = AVERAGE HALF-LENGTH OF PATH THROUGH CLOUD                  THR00110
C      W0 = SINGLE SCATTERING ALBEDO                                     THR00120
C      TMPA = TEMPERATURE OF THE ATMOSPHERE                              THR00130
C      TMPC = TEMPERATURE OF THE CLOUD                                   THR00140
C      LAMBDA = WAVELENGTH                                               THR00150
C      KAPPA = VOLUME EXTINCTION COEFFICIENT                             THR00160
C      TRANS = TRANSMITTANCE THROUGH CLOUD.                              THR00170
C                                                                        THR00180
C*****************************************************************     THR00190
      G=1.0-EXP(-KAPPA*RBAR)                                             THR00200
      B=BB(LAMBDA,273.16+TMPC)                                           THR00210
      LI=BB(LAMBDA,273.16+TMPA)                                          THR00220
      WRITE(IOOUT,1000) B,LI                                             THR00230
      LP=(1.0-TRANS)*((1.0-W0)*(1.0+G*W0)*B+W0*(1.0-G)*LI)               THR00240
 1000 FORMAT(1H0,43X,40H  **THERMAL CALCULATION OF PATH RADIANCE/,45X,   THR00250
     122H     BB(LAMBDA,TMPC) = ,1PE10.4,11H W/M2-SR-MU/,45X,           THR00260
     122H     BB(LAMBDA,TMPA) = ,1PE10.4,11H W/M2-SR-MU)                THR00270
      RETURN                                                             THR00280
      END                                                                THR00290
```

```
      SUBROUTINE SSCLD(ALPHA,BETA,GAMMA,DELTA,EPS,S,KAPPA,                SSC00010
     1W0,RGRND,P,LP)                                                      SSC00020
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSSC00030
C*********************************************************************SSC00040
C                                                                       SSC00050
C   SUBROUTINE COMPUTES PATH RADIANCE LP FOR THE CASE OF SINGLE         SSC00060
C   SCATTERING IN A FINITE CLOUD.  INPUTS ARE GEOMETRICAL PARAMETERS    SSC00070
C   ALPHA, BETA, GAMMA, DELTA, AND EPS, S = LENGTH OF OPTICAL PATH      SSC00080
C   THROUGH CLOUD, AND:                                                 SSC00090
C      KAPPA = VOLUME EXTINCTION COEFFICIENT                            SSC00100
C      W0 = SINGLE SCATTERING ALBEDO                                    SSC00110
C      RGRND = SURFACE BACKGROUND RADIANCE                              SSC00120
C      P = SINGLE SCATTERING PHASE FUNCTION.                            SSC00130
C                                                                       SSC00140
C*********************************************************************SSC00150
      REAL KAPPA,LP                                                     SSC00160
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     SSC00170
      ASIN(ARG)=ATAN2(ARG,SQRT(1.-ARG**2))
      WRITE(IOOUT,1500)                                                 SSC00180
      DISCRM=DELTA*DELTA-4.0*GAMMA*EPS                                  SSC00190
      IF(DISCRM.GT.0.0) GO TO 10                                        SSC00200
      WRITE(IOOUT,1000) DISCRM                                          SSC00210
      LP=0.0                                                            SSC00220
      GO TO 20                                                          SSC00230
   10 TGSDEL=2.0*GAMMA*S+DELTA                                          SSC00240
      HS= ASIN(TGSDEL/SQRT(DISCRM))/SQRT(-GAMMA)                        SSC00250
      H0= ASIN(DELTA/SQRT(DISCRM))/SQRT(-GAMMA)                         SSC00260
      TRANS=EXP(-KAPPA*S)                                               SSC00270
      BETAB=W0*KAPPA                                                    SSC00280
      LP=(1.0-BETA*KAPPA)*S+(1.0-ALPHA)*KAPPA*S*S/2.0                   SSC00290
      LP=LP-KAPPA*DISCRM*(HS-H0)/(8.0*GAMMA)                            SSC00300
      LP=LP-KAPPA*(TGSDEL*SQRT(GAMMA*S*S+DELTA*S+EPS)-                  SSC00310
     1DELTA*SQRT(EPS))/(4.0*GAMMA)                                      SSC00320
      LP=BETAB*LP*P*RGRND*TRANS/(4.0*PI)                                SSC00330
   20 RETURN                                                            SSC00340
 1000 FORMAT(9H0DISCRM =,E10.4,25H IN SSCLD.  LP SET TO 0.0)            SSC00350
 1500 FORMAT(1H0,43X,33H  **RESULTS FOR SINGLE SCATTERING)              SSC00360
      END                                                               SSC00370
```

```
      SUBROUTINE PFNN(LAMBDA,CHI,INDEXP,PFN,ETA)                        PFN00010
      REAL LAMBDA,LAMDA1,LAM1,LAM2,KAPPA                                PFN00020
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUPFN00030
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     PFN00040
      DIMENSION PHANG(65),PF1(65),PF2(65)                               PFN00050
      DATA LAMDA1,IP1,CHI1/-1.0,-2,-1.0/                                PFN00060
C*****************************************************************************PFN00070
C                                                                       PFN00080
C  SUBROUTINE COMPUTES THE VALUE OF THE INDEXP-TH PHASE FUNCTION        PFN00090
C  AT SCATTERING ANGLE CHI AND WAVELENGTH LAMBDA USING BILINEAR         PFN00100
C  INTERPOLATION.                                                       PFN00110
C                                                                       PFN00120
C*****************************************************************************PFN00130
      ACOS(ARG)=ATAN2(SQRT(1.-ARG**2),ARG)
      IERR=0                                                            PFN00140
      MAXID=12
C     PFNDAT ONLY CONTAINS ONE PFN FOR THE VISIBLE (.55UM) AND
C     THE NEAR IR (1.06UM); THEREFORE DO NOT INTERPOLATE.
      IWAVE=0
      IF (LAMBDA.LT.2.0) IWAVE=1
C     PRELIMINARIES FOR FINDING ETA
      ETA1=0.0
      ETA2=0.0
      IDCK=0
      IF (INDEXP.LT.0) IDCK=1
      IF (INDEXP.LT.0) INDEXP=-INDEXP
      IF(INDEXP.EQ.IP1.AND.CHI.EQ.CHI1.AND.LAMBDA.EQ.LAMDA1)            PFN00150
     1GO TO 200                                                         PFN00160
      IF(INDEXP.EQ.IP1.AND.LAMBDA.EQ.LAMDA1) GO TO 70                   PFN00170
      REWIND IPHFUN                                                     PFN00180
      NRD=0                                                             PFN00190
      DO 20 I=1,66,11                                                   PFN00200
      READ(IPHFUN,1000) (PHANG(I+J-1),J=1,11)                           PFN00210
      DO 10 J=1,11                                                      PFN00220
      NRD=NRD+1                                                         PFN00230
      IF(PHANG(NRD).GE.999.9) GO TO 30                                  PFN00240
   10 CONTINUE                                                          PFN00250
   20 CONTINUE                                                          PFN00260
   30 NA=NRD-1                                                          PFN00270
      DO 35 I=1,NA                                                      PFN00280
35    PHANG(I)=COS(PHANG(I)*PIRAD)                                      PFN00290
40    CONTINUE                                                          PFN00300
C     END-OF-FILE CHECK                                                 PFN00310
      IF (IERR.EQ.2) GO TO 195                                          PFN00320
      READ(IPHFUN,1100) IANG1,ID,LAM1,W0,KAPPA,BETA                     PFN00330
      IF (LAM1.GE.12.00.AND.ID.EQ.MAXID) IERR=2                         PFN00340
      READ(IPHFUN,1200) (PF1(I),I=1,NA)                                 PFN00350
      SUM=0.                                                            PFN00360
C     START RENORMALIZATION OF PHASE FUNCTION - ALSO SEE BELOW          PFN00370
      DO 45 J=2,NA                                                      PFN00380
45    SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF1(J)+PF1(J-1))/4.               PFN00390
      DO 46 J=1,NA                                                      PFN00400
46    PF1(J)=PF1(J)/SUM                                                 PFN00410
      ETA1=ETAINT(PF1,PHANG,NA)
      IF(ID.NE.INDEXP) GO TO 40                                         PFN00420
      IF (IWAVE.EQ.1.AND.LAMBDA.GT.LAM1)  GO TO 40
      IF (IWAVE.EQ.1) GO TO 75
      IF(LAMBDA.LT.LAM1) GO TO 190                                      PFN00430
50    CONTINUE                                                          PFN00440
      IF (IERR.EQ.2) GO TO 195                                          PFN00450
      READ(IPHFUN,1100) IANG2,ID,LAM2,W0,KAPPA,BETA                     PFN00460
      IF (LAM2.GE.12..AND.ID.EQ.MAXID) IERR=2                           PFN00470
      READ(IPHFUN,1200) (PF2(I),I=1,NA)                                 PFN00480
      SUM=0.                                                            PFN00490
      DO 55 J=2,NA                                                      PFN00500
55    SUM=SUM+(-PHANG(J)+PHANG(J-1))*(PF2(J)+PF2(J-1))/4.               PFN00510
      DO 56 J=1,NA                                                      PFN00520
56    PF2(J)=PF2(J)/SUM                                                 PFN00530
      ETA2=ETAINT(PF2,PHANG,NA)
C     THE PHASE FUNCTION(S) ARE NOW NORMALIZED TO: INTEGRAL OF          PFN00540
```

```
C     PHASE FUNCTION OVER ALL SOLID ANGLE DIVIDED BY 4 PI = 1.                  PFN00550
      IF(ID.NE.INDEXP) GO TO 190                                                PFN00560
      IF(LAMBDA.LE.LAM2) GO TO 70                                               PFN00570
      LAM1=LAM2                                                                 PFN00580
      IANG1=IANG2                                                               PFN00590
      DO 60 I=1,NA                                                              PFN00600
      PF1(I)=PF2(I)                                                             PFN00610
   60 CONTINUE                                                                  PFN00620
      GO TO 50                                                                  PFN00630
70    CONTINUE
      IF (IWAVE.EQ.1) GO TO 75
      DLAM=(LAMBDA-LAM1)/(LAM2-LAM1)                                            PFN00640
75    CONTINUE
      IF(CHI.LT.-1.E-3.OR.CHI.GT.180.001) GO TO 190                             PFN00650
      DO 80 J=2,NA                                                              PFN00670
      IF(CHI.LE.(ACOS(PHANG(J))/PIRAD)) GO TO 90                                PFN00680
   80 CONTINUE                                                                  PFN00690
      J=NA                                                                      PFN00700
   90 J1=J-1                                                                    PFN00710
      DCHI=(COS(CHI*PIRAD)-PHANG(J1))/(PHANG(J)-PHANG(J1))                      PFN00720
      IF (IWAVE.NE.1) GO TO 95
      PFN=PF1(J1)+DCHI*(PF1(J)-PF1(J1))
      GO TO 96
95    PFN=PF1(J1)+DLAM*(PF2(J1)-PF1(J1))+DCHI*(PF1(J)-PF1(J1))                  PFN00730
     1+DLAM*DCHI*(PF2(J)+PF1(J1)-PF2(J1)-PF1(J))                                PFN00740
96    LAMDA1=LAMBDA                                                             PFN00760
      CHI1=CHI                                                                  PFN00770
      IF (IP1.NE.-2) WRITE(IOOUT,1500) CHI,PFN
      IP1=INDEXP
      ETA=ETA1+DLAM*(ETA2-ETA1)
      GO TO 200                                                                 PFN00790
  190 WRITE(IOOUT,1300) ID,INDEXP,LAM1,LAMBDA,CHI                               PFN00800
      STOP                                                                      PFN00810
195   WRITE (IOOUT,1600) IPHFUN                                                 PFN00820
      STOP                                                                      PFN00830
200   CONTINUE
      IF (IDCK.EQ.1) INDEXP=-INDEXP
      RETURN                                                                    PFN00840
 1000 FORMAT(11(F6.2,1X))                                                       PFN00850
 1100 FORMAT(2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))                              PFN00860
 1200 FORMAT(6(E12.6,1X))                                                       PFN00870
 1300 FORMAT(33H0ERROR IN READING PHASE FUNCTION./                              PFN00880
     127H ID,INDEXP,LAM1,LAMBDA,CHI=,2I3,3E13.7)                                PFN00890
 1500 FORMAT(1H0,23HSCATTERING ANGLE CHI = ,F8.2,7X,24H PHASE FN P(LAMBDPFN00900
     +A,CHI)=,E10.4)                                                            PFN00910
1600  FORMAT(1X,32HATTEMPT TO READ PAST EOF ON UNIT,I3,18H IN SUBROUTINEPFN00920
     1 PFN//)                                                                   PFN00930
      END                                                                       PFN00940
```

```
      SUBROUTINE MSCLD(TAU,TAU0,TRANS,TATM,E0,W0,ETA,RHO,LP)              MSC00010
      REAL LP,LBAR,K                                                     MSC00020
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                      MSC00030
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUMSC00040
      SINH(ARG)=.5*(EXP(ARG)-EXP(-ARG))
      COSH(ARG)=.5*(EXP(ARG)+EXP(-ARG))
C*********************************************************************MSC00050
C                                                                      MSC00060
C   SUBROUTINE COMPUTES LP = PATH RADIANCE DUE TO MULTIPLE SCATTERING  MSC00070
C   IN A FINITE CLOUD.  INPUTS ARE:                                    MSC00080
C                                                                      MSC00090
C      TAU = CLOUD OPTICAL DEPTH                                       MSC00100
C      TAU0 = CLOUD OPTICAL THICKNESS                                  MSC00110
C      TRANS = TRANSMITTANCE ALONG LINE OF SIGHT                       MSC00120
C      TATM = TRANSMITTANCE OF ATMMOSPHERE ABOVE CLOUD                 MSC00130
C      E0 = EXTRATERRESTRIAL IRRADIANCE                                MSC00140
C      W0 = SINGLE SCATTERING ALBEDO                                   MSC00150
C      ETA = FORWARD SCATTERING PARAMETER                              MSC00160
C      RHO = BACKGROUND SURFACE REFLECTANCE.                           MSC00170
C                                                                      MSC00180
C*********************************************************************MSC00190
      IF(W0.LT.1.0) GO TO 10                                           MSC00200
      W0=.999                                                          MSC00210
      WRITE(IOOUT,2000)                                                MSC00220
   10 C1=1.0-ETA*W0                                                    MSC00230
      C2=(1.0-ETA)*W0                                                  MSC00240
      K=SQRT((1.0-W0)*(1.0+W0-2.0*ETA*W0))                             MSC00250
      T0K=K*TAU0                                                       MSC00260
      T0TK=K*(TAU0-TAU)                                                MSC00270
      TK2=2.0*K*TAU                                                    MSC00280
      GAMMA=E0*TATM/(C1*SINH(T0K)+K*COSH(T0K))                         MSC00290
      DELTA=RHO*K/((C1-RHO*C2)*SINH(2.0*T0K)+K*COSH(2.0*T0K))          MSC00300
      WRITE(IOOUT,1000)                                                MSC00310
      EPLUS=C2*SINH(T0TK)+DELTA*(C1*SINH(TK2)+K*COSH(TK2))             MSC00320
      EMINUS=C1*SINH(T0TK)+K*COSH(T0TK)+DELTA*C2*SINH(TK2)             MSC00330
      LBAR=GAMMA*(EPLUS+EMINUS)/TWOPI                                  MSC00340
      LP=W0*LBAR*(1.0-TRANS)                                           MSC00350
      WRITE(IOOUT,100)TAU,TAU0,TATM                                    MSC00360
  100 FORMAT(1H0,45X,18H  OPTICAL DEPTH = ,1PE10.4/,45X,               MSC00370
     131H  OPTICAL THICKNESS OF CLOUD = ,1PE10.4/,45X,                 MSC00380
     244H  TRANSMITTANCE OF ATMOSPHERE ABOVE CLOUD = ,1PE10.4)         MSC00390
      RETURN                                                           MSC00400
 1000 FORMAT(36H0  **RESULTS FOR MULTIPLE SCATTERING )                 MSC00410
 2000 FORMAT(47H0  **OMEGA 0 WAS 1.0, NOW SET TO 0.999 IN MSCLD)       MSC00420
      END                                                              MSC00430
```

```
      FUNCTION ETAINT(PFN,PHANG,NA)
C     THIS FUNCTION WILL DETERMINE ETA, THE FORWARD SCATTERING
C     PARAMETER: ETA=.5*INTEGRAL PFN OVER THETA, WHERE THETA GOES
C     FROM ZERO TO PI/2.
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK
      DIMENSION PHANG(65),PFN(65)
      NAM1=NA-1
      ETA=0.
      DO 1 I=1,NAM1
      IF(PHANG(I+1).GT.0.) ETA=ETA+(PHANG(I)-PHANG(I+1))*
     1 (PFN(I+1)+PFN(I))/4.
1     CONTINUE
      ETAINT=ETA
      RETURN
      END
```

```
      SUBROUTINE OVRCST(LAMBDA,TRANS,IERR)                                  OVR00010
      REAL LC,LG,LB0,KAPPA,LAMBDA,MU,LP                                     OVR00020
      DIMENSION IALPH(7)                                                    OVR00030
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUOVR00040
      COMMON /GEOMET/PTS(15),IGEOSW                                         OVR00050
      DATA IZERO/0/                                                         OVR00060
      DATA IALPH/2HOP   ,2HSP   ,2HCL   ,2HBK   ,2HGR   ,2HTE   ,2HGO   /  OVR00070
      BB(X,T)=1.191062E8/(X**5*(EXP(1.4387864E4/(X*T))-1.0))                OVR00080
C*********************************************************************** OVR00090
C                                                                         OVR00100
C  SUBROUTINE COMPUTES BEAM TRANSMITTANCE, PATH RADIANCE, AND             OVR00110
C  CONTRAST TRANSMITTANCE  ALONG AN OPTICAL PATH UNDER AN OVERCAST        OVR00120
C  SKY.  ORDER-INDEPENDENT INPUT CARDS ARE AS FOLLOWS:                    OVR00130
C  (INDIVIDUAL RECORD FORMAT IS (A4,1X,5(E10.4,1X))                       OVR00140
C                                                                         OVR00150
C   MNEMONIC       VARIABLES READ          DESCRIPTION                    OVR00160
C   --------       --------------          -----------                    OVR00170
C                                                                         OVR00180
C    OPOS             XO,YO,ZO             OBSERVER POSITION              OVR00190
C    SPOS             XT,YT,ZT             SOURCE POSITION                OVR00200
C    CLDS       ZC,LC,KAPPA,ETA ,WO        CLOUD PARAMETERS               OVR00210
C    BKGR             LB0                  BACKGROUND RADIANCE            OVR00220
C    GRND             LG                   GROUND RADIANCE                OVR00230
C    TEMP             TEMP                 TEMPERATURE ALONG PATH         OVR00240
C    GO                                    END OF READ SENTINEL           OVR00250
C                                                                         OVR00260
C     THE VARIABLES ZC AND LC REFER TO THE OVERCAST SKY LAYER!            
C     THE VARIABLES KAPPA, ETA, WO, TEMP, REFER TO THE ATMOSPHERE         
C     BETWEEN THE OVERCAST SKY AND GROUND I.E. THE INTERVENING            
C     ATMOSPHERIC PROPERTIES (GAS OR AEROSOL)                             
C  ** NOTE : THE GO CARD MUST BE THE LAST RECORD READ.                    OVR00270
C                                                                         OVR00280
C  THE FOLLOWING ENUMERATES THE VARIABLES LISTED ON THE ABOVE CARDS :     OVR00290
C                                                                         OVR00300
C       (XO,YO,ZO) = OBSERVER COORDINATES                                 OVR00310
C       (XT,YT,ZT) = SOURCE COORDINATES                                   OVR00320
C       ZC = HEIGHT OF CLOUD LAYER                                        OVR00330
C       LC = CLOUD RADIANCE                                               OVR00340
C       LG = GROUND RADIANCE                                              OVR00350
C       LB0 = BACKGROUND RADIANCE                                         OVR00360
C       KAPPA = VOLUME EXTINCTION COEFFICIENT (KM-1)                      OVR00370
C       ETA  = FORWARD SCATTERING PARAMETER                               OVR00380
C       W0 = SINGLE SCATTERING ALBEDO                                     OVR00390
C       TEMP = TEMPERATURE ALONG PATH (DEG. C)                            OVR00400
C                                                                         OVR00410
C  LENGTH UNITS ARE KILOMETERS; RADIANCE UNITS ARE W/M2-SR-MU.            OVR00420
C  IF TEMP >= -99., THERMAL RADIATION IS CALCULATED; IF TEMP < -99.       OVR00430
C  SINGLY SCATTERED RADIATION IS CALCULATED.                              OVR00440
C                                                                         OVR00450
C  SUBROUTINE RETURNS:                                                    OVR00460
C     TRANS = BEAM TRANSMITTANCE                                          OVR00470
C  TO CALLING PROGRAM.                                                    OVR00480
C                                                                         OVR00490
C*********************************************************************** OVR00500
C                                                                         OVR00510
C  DATA INITIALIZATION                                                    OVR00520
C                                                                         OVR00530
      IF(IZERO.NE.0) GO TO 477                                            OVR00540
      XO=0.0                                                              OVR00550
      YO=0.0                                                              OVR00560
      ZO=0.0                                                              OVR00570
      XT=0.0                                                              OVR00580
      YT=0.0                                                              OVR00590
      ZT=0.0                                                              OVR00600
      ZC=0.0                                                              OVR00610
      LC=0.0                                                              OVR00620
      KAPPA=0.0                                                           OVR00630
      ETA =0.0                                                            OVR00640
      W0=0.0                                                              OVR00650
      LB0=0.0                                                             OVR00660
```

```
      LG=0.0                                                                    OVR00670
      TEMP=0.0                                                                  OVR00680
      IZERO=1                                                                   OVR00690
  477 CONTINUE                                                                  OVR00700
      DO 360 K=1,7                                                              OVR00710
      READ(IOIN,334)IA,IA2,R1,R2,R3,R4,R5                                          OVR0
  334 FORMAT(2A2,1X,5(E10.4,1X))                                                 OVR0073
      DO 333 I=1,8                                                              OVR00740
      IF(IA.NE.IALPH(I)) GO TO 333                                              OVR00750
      IND=I                                                                     OVR00760
      IF(IND.EQ.7) GO TO 361                                                    OVR00770
  333 CONTINUE                                                                  OVR00780
      IF(IND.EQ.8) GO TO 355                                                    OVR00790
      IF(K.EQ.7.AND.IND.NE.7) GO TO 358                                         OVR00800
      GO TO (341,342,343,344,345,346),IND                                       OVR00810
  341 XO=R1                                                                     OVR00820
      YO=R2                                                                     OVR00830
      ZO=R3                                                                     OVR00840
      GO TO 360                                                                 OVR00850
  342 XT=R1                                                                     OVR00860
      YT=R2                                                                     OVR00870
      ZT=R3                                                                     OVR00880
      GO TO 360                                                                 OVR00890
  343 ZC=R1                                                                     OVR00900
      LC=R2                                                                     OVR00910
      KAPPA=R3                                                                  OVR00920
      ETA =R4                                                                   OVR00930
      W0=R5                                                                     OVR00940
      GO TO 360                                                                 OVR00950
  344 LB0=R1                                                                    OVR00960
      GO TO 360                                                                 OVR00970
  345 LG=R1                                                                     OVR00980
      GO TO 360                                                                 OVR00990
  346 TEMP=R1                                                                   OVR01000
      GO TO 360                                                                 OVR01010
  355 WRITE(IOOUT,357)                                                          OVR01020
  357 FORMAT(1H0,25X,44H***OVRCST ERROR*** INPUT CARD DETECTED WHICH,           OVR01030
     + 36H DOES NOT MATCH CORRECT INPUT FORMAT,/)                               OVR01040
      GO TO 360                                                                 OVR01050
  358 WRITE(IOOUT,359)                                                          OVR01060
  359 FORMAT(1H0,34X,45H***OVRCST ERROR*** TOO MANY INPUT CARDS OR GO,          OVR01070
     + 16H SENTINEL ABSENT,/)                                                   OVR01080
      IERR=1                                                                    OVR01090
      GO TO 200                                                                 OVR01100
  360 CONTINUE                                                                  OVR01110
  361 CONTINUE                                                                  OVR01120
      IF(IGEOSW.NE.1) GO TO 222                                                 OVR01130
      XO=PTS(4)                                                                 OVR01140
      YO=PTS(5)                                                                 OVR01150
      ZO=PTS(6)                                                                 OVR01160
      XT=PTS(1)                                                                 OVR01170
      YT=PTS(2)                                                                 OVR01180
      ZT=PTS(3)                                                                 OVR01190
  222 CONTINUE                                                                  OVR01200
C                                                                               OVR01210
C  ECHO INPUT                                                                   OVR01220
C                                                                               OVR01230
      WRITE(IOOUT,1000) XO,XT,YO,YT,ZO,ZT,ZC,LC,LAMBDA,LG,                      OVR01240
     1TEMP,LB0,KAPPA,W0,ETA                                                     OVR01250
      IF(TEMP.LT.-99.0) GO TO 4                                                 OVR01260
      BBTEMP=BB(LAMBDA,273.16+TEMP)                                             OVR01270
      WRITE(IOOUT,1800) BBTEMP                                                  OVR01280
      BTE=(W0-1.0)*BBTEMP                                                       OVR01290
      GO TO 8                                                                   OVR01300
    4 WRITE(IOOUT,1700)                                                         OVR01310
    8 F=2.0*(1.0-ETA )                                                          OVR01320
      ZLEN=ZT-ZO                                                                OVR01330
      S=SQRT((XT-XO)**2+(YT-YO)**2+ZLEN**2)                                     OVR01340
      MU=ABS(ZLEN)/S                                                            OVR01350
      TO=KAPPA*ZO                                                               OVR01360
```

```
      TT=KAPPA*ZT                                                          OVR01370
      SO=KAPPA*(ZC-ZO)                                                     OVR01380
      ST=KAPPA*(ZC-ZT)                                                     OVR01390
      TRANS=EXP(-KAPPA*S)                                                  OVR01400
      CONST=W0*ETA /2.0                                                    OVR01410
      IF(ZLEN) 10,40,70                                                    OVR01420
C                                                                          OVR01430
CC    HEIGHT OF OBSERVER > HEIGHT OF SOURCE                                OVR01440
C                                                                          OVR01450
   10 G21=G2(TO,TT,MU)                                                     OVR01460
      G22=G2(SO,ST,-MU)                                                    OVR01470
      T1=EXP(-TO/MU)                                                       OVR01480
      T2=EXP(SO/MU)                                                        OVR01490
      F1=W0*F*KAPPA*S*T1/2.0                                               OVR01500
      XINT1=CONST*T1*G21                                                   OVR01510
      XINT2=CONST*T2*G22                                                   OVR01520
      IF(TEMP.GE.-99.0) GO TO 20                                           OVR01530
      LP=LG*(F1+XINT1)+LC*XINT2                                            OVR01540
      GO TO 100                                                            OVR01550
   20 LP=(LG+BTE)*(F1+XINT1)+(LC+BTE)*XINT2+W0*BTE*(TRANS-1.0)             OVR01560
      GO TO 100                                                            OVR01570
C                                                                          OVR01580
CC    HEIGHT OF OBSERVER = HEIGHT OF SOURCE                                OVR01590
C                                                                          OVR01600
   40 E21=1.0                                                              OVR01610
      E22=1.0                                                              OVR01620
      IF(TO.NE.0.0) E21=EXP(-TO)-TO*E1(TO)                                 OVR01630
      IF(SO.NE.0.0) E22=EXP(-SO)-SO*E1(SO)                                 OVR01640
      IF(TEMP.GE.-99.0) GO TO 50                                           OVR01650
      LP=CONST*(1.0-TRANS)*(LG*E21+LC*E22)                                 OVR01660
      GO TO 100                                                            OVR01670
   50 LP=CONST*(E21*(LG+BTE)+E22*(LC+BTE))                                 OVR01680
      LP=(LP-W0*BTE)*(1.0-TRANS)                                           OVR01690
      GO TO 100                                                            OVR01700
C                                                                          OVR01710
CC    HEIGHT OF OBSERVER < HEIGHT OF SOURCE                                OVR01720
C                                                                          OVR01730
   70 G21=G2(TO,TT,-MU)                                                    OVR01740
      G22=G2(SO,ST,MU)                                                     OVR01750
      T1=EXP(-SO/MU)                                                       OVR01760
      T2=EXP(TO/MU)                                                        OVR01770
      F1=W0*F*KAPPA*S*T1/2.0                                               OVR01780
      XINT1=CONST*T2*G21                                                   OVR01790
      XINT2=CONST*T1*G22                                                   OVR01800
      IF(TEMP.GE.-99.0) GO TO 80                                           OVR01810
      LP=LC*(F1+XINT2)+LG*XINT1                                            OVR01820
      GO TO 100                                                            OVR01830
   80 LP=(LC+BTE)*(F1+XINT2)+(LG+BTE)*XINT1+W0*BTE*(TRANS-1.0)             OVR01840
C                                                                          OVR01850
CC    WRITE RESULTS                                                        OVR01860
C                                                                          OVR01870
  100 TC=1.0/(1.0+LP/(LB0*TRANS))                                          OVR01880
      WRITE(IOOUT,2000) S,TRANS,LP,TC                                      OVR01890
  200 RETURN                                                               OVR01900
 1000 FORMAT(1H0/1H0,55X,34H-- RADIATION UNDER OVERCAST SKY --/            OVR01910
     1 1H0,43X,8HXO      =,5X,F6.3,5H (KM),8X,5HXT   =,5X,F6.3,5H (KM)/   OVR01920
     2 44X,8HYO      =,5X,F6.3,13X,5HYT   =,5X,F6.3/                       OVR01930
     3 44X,8HZO      =,5X,F6.3,13X,5HZT   =,5X,F6.3/                       OVR01940
     4 44X,8HLC      =,5X,F6.3,13X,5HLC   =,1X,1PE10.4,13H (W/M2-SR-MU)/   OVR01950
     5 44X,8HLAMBDA  =,5X,0PF6.3,5H (MU),8X,5HLG   =,1X,1PE10.4/           OVR01960
     6 44X,8HTEMP    =,3X,0PF6.1,10H    (DEG.C),5X,5HLB0 =,1X,1PE10.4/     OVR01970
     7 44X,8HKAPPA   =,1X,E10.4,7H (KM-1),6X,5HW0   =,5X,0PF6.3/           OVR01980
     8 44X,8HETA     =,6X,0PF5.3)                                          OVR01990
 2000 FORMAT(1H0,39X,4HPATH,7X,13HTRANSMITTANCE,2X,13HPATH RADIANCE,       OVR02000
     1 4X,8HCONTRAST/                                                      OVR02010
     2 38X,11HLENGTH (KM),19X,12H(W/M2-SR-MU),15H  TRANSMITTANCE/          OVR02020
     3 36X,4(15H+-------------+)/                                          OVR02030
     4 1H0,39X,F6.3,9X,F7.5,7X,1PE9.3,7X,0PF7.5)                           OVR02040
 1700 FORMAT(1H0,50X,31H**RESULTS FOR SINGLE SCATTERING)                   OVR02050
 1800 FORMAT(1H0,46X,38H**THERMAL CALCULATION OF PATH RADIANCE/            OVR02060
```

```
1 1H0,50X,8HBBTEMP= ,1PE10.4,11H W/M2-SR-MU)                    OVR02070
  END                                                           OVR02080
```

```
      FUNCTION G2(TAU1,TAU2,MU)                                          FUG00010
C*********************************************************************FUG00020
C                                                                        FUG00030
C   SUBROUTINE COMPUTES IN CLOSED FORM INTEGRALS OF THE FUNCTION         FUG00040
C                                                                        FUG00050
C      EXP(TAU/MU)*E2(TAU)                                               FUG00060
C                                                                        FUG00070
C   WHERE E2 IS THE SECOND EXPONENTIAL INTEGRAL.  FOR DETAILS SEE        FUG00080
C   KOURGANOFF, 'BASIC METHODS IN TRANSFER PROBLEMS', APPENDIX I         FUG00090
C   (PAGES 256-257 OF FIRST EDITION, 1952, OXFORD UNIVERSITY PRESS.)     FUG00100
C                                                                        FUG00110
C*********************************************************************FUG00120
      REAL MU                                                            FUG00130
      DATA GAMMA/.5772156649/                                            FUG00140
      IF(MU.LT..9999) GO TO 50                                           FUG00150
C                                                                        FUG00160
C   MU EQ 1.0                                                            FUG00170
C                                                                        FUG00180
      IF(TAU1.NE.0.0) GO TO 10                                           FUG00190
      G2=(TAU2-1.0)*EXP(TAU2)*E1(TAU2)-GAMMA-ALOG(TAU2)                  FUG00200
      GO TO 100                                                          FUG00210
   10 IF(TAU2.NE.0.0) GO TO 20                                           FUG00220
      G2=GAMMA+ALOG(TAU1)+EXP(TAU1)*(1.0-TAU1)*E1(TAU1)                  FUG00230
      GO TO 100                                                          FUG00240
   20 G2=EXP(TAU1)*(1.0-TAU1)*E1(TAU1)-EXP(TAU2)*(1.0-TAU2)*E1(TAU2)     FUG00250
      G2=G2+(ALOG(TAU1)-ALOG(TAU2))                                      FUG00260
      GO TO 100                                                          FUG00270
C                                                                        FUG00280
C   MU NE 1.0                                                            FUG00290
C                                                                        FUG00300
   50 RM=1.0-1.0/MU                                                      FUG00310
      IF(TAU1.NE.0.0) GO TO 60                                           FUG00320
      G2=EXP(TAU2/MU)*(TAU2-MU)*E1(TAU2)                                 FUG00330
      G2=G2+MU*E1(TAU2*RM)                                               FUG00340
      G2=G2+1.0+MU*ALOG(ABS(RM))-EXP(-RM*TAU2)                           FUG00350
      GO TO 100                                                          FUG00360
   60 IF(TAU2.NE.0.0) GO TO 70                                           FUG00370
      G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1)                                 FUG00380
      G2=G2-MU*E1(TAU1*RM)                                               FUG00390
      G2=G2+EXP(-RM*TAU1)-MU*ALOG(ABS(RM))-1.0                           FUG00400
      GO TO 100                                                          FUG00410
   70 G2=EXP(TAU1/MU)*(MU-TAU1)*E1(TAU1)                                 FUG00420
      G2=G2-EXP(TAU2/MU)*(MU-TAU2)*E1(TAU2)                              FUG00430
      G2=G2+(EXP(-TAU1*RM)-EXP(-TAU2*RM))                                FUG00440
      G2=G2+MU*(E1(TAU2*RM)-E1(TAU1*RM))                                 FUG00450
  100 RETURN                                                             FUG00460
      END                                                                FUG00470
```

```
      FUNCTION E1(X)                                                    FUE00010
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUFUE00020
C*********************************************************************FUE00030
C                                                                       FUE00040
C  FUNCTION COMPUTES THE VALUE OF THE FIRST EXPONENTIAL INTEGRAL        FUE00050
C  -EI(-X) WHERE -174 < X < 170.  FOR X OUTSIDE THESE BOUNDS, AN        FUE00060
C  OVERFLOW OR UNDERFLOW MIGHT OCCUR, SO PROGRAM EXECUTION IS           FUE00070
C  HALTED.  POLYNOMIAL AND RATIONAL FUNCTION APPROXIMATIONS ARE         FUE00080
C  ADAPTED FROM THE IBM SCIENTIFIC SUBROUTINE PACKAGE, SUBROUTINE       FUE00090
C  EXPI.                                                                FUE00100
C                                                                       FUE00110
C*********************************************************************FUE00120
      DATA GAMMA/.5772156649/                                           FUE00130
      DATA C1,C2,C3,C4,C5,C6,C7,C8/674.567029,57.411833,6.05529232,     FUE00140
     11699.06552,841.654932,49.3133893,8.01957683,.99979204/            FUE00150
      DATA D1,D2,D3,D4,D5,D6,D7,D8,D9/248.6697,224.4234,32.43665,       FUE00160
     13.061037,.05176245,180.7837,22.63818,38.93944,3.9951617           FUE00170
      DATA F1,F2,F3,F4,F5,F6,F7,F8,F9/9.999999E-1,2.500001E-1,          FUE00180
     15.555682E-2,1.041576E-2,1.664156E-3,2.335379E-4,2.928433E-5,      FUE00190
     21.766345E-6,7.122452E-7/                                          FUE00200
      DATA G1,G2,G3,G4,G5,G6,G7,G8/.2677737343,8.6347608925,            FUE00210
     1118.0590169?3,8.5733287401,3.9584969228,21.0996530827,            FUE00220
     1225.6329561486,9.5733223454/                                      FUE00230
      IF(X.GT.-174.0.AND.X.LE.170.0) GO TO 10                           FUE00240
      WRITE(IOOUT,1000)                                                 FUE00250
      STOP                                                              FUE00260
   10 IF(X.GT.-9.0) GO TO 20                                            FUE00270
      E1=1.0-(C1+C2*X-C3*X*X-X*X*X)/(C4+C5*X+C6*X*X-C7*X*X*X-C8*X*X*X*X)FUE00280
      E1=E1*EXP(-X)/X                                                   FUE00290
      GO TO 100                                                         FUE00300
   20 IF(X.GT.-3.0) GO TO 30                                            FUE00310
      E1=D1+D2*X+D3*X*X+D4*X*X*X+D5*X*X*X*X                             FUE00320
      E1=(1.0-E1/(D6+D7*X+D8*X*X+D9*X*X*X+X*X*X*X))*EXP(-X)/X           FUE00330
      GO TO 100                                                         FUE00340
   30 IF(X.GT.1.0) GO TO 40                                             FUE00350
      E1=F1-X*(F2-X*(F3-X*(F4-X*(F5-X*(F6-X*(F7-X*(F8-X*F9))))))        FUE00360
      E1=X*E1-GAMMA-ALOG(ABS(X))                                        FUE00370
      GO TO 100                                                         FUE00380
   40 E1=(G1+X*(G2+X*(G3+X*(G4+X))))/(G5+X*(G6+X*(G7+X*(G8+X))))        FUE00390
      E1=E1*EXP(-X)/X                                                   FUE00400
  100 RETURN                                                            FUE00410
 1000 FORMAT(4H0X =,3X,E10.4,39H OUT-OF-RANGE FOR E1. EXECUTION HALTED.)FUE00420
      END                                                               FUE00430
```

```
      SUBROUTINE GRNADE(WAVE1,ICLMAT,TRANS,IERR)                        GRN00010
C/*****************************************************************/GRN00020
C/*                      SUBROUTINE GRNADE                         *GRN00030
C/*                   MAIN GRNADE MODULE                           *GRN00040
C/*                     EOSAEL80                                  */GRN00050
C/*****************************************************************/GRN00060
      DIMENSION XA(3),XTRAN(7)                                          GRN00070
      COMMON /MOS/ XDTA(1000),CDTA(1000),CL(1000)                       GRN00080
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK                      GRN00090
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUGRN00100
      COMMON/CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDMAT,CLDHYT,FOGPRB,       GRN00110
     *              WNDVEL,WNDDIR,IPASCT                                GRN00120
      COMMON/MECH0/XM,YM,ZM,XO,YO,ZO,XT,YT,ZT,ISTO,IETO,IDTO,XN,FW,     GRN00130
     *TBURN,ITYPE,EFF,YF,RHA,UW,WD,ICAT,AIRT,TGRAD,BRATE,HEAD,RNG,      GRN00140
     *DLEN,WPOWR,EXTC(8),XMIS(8),XNORTH                                 GRN00150
      COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP                  GRN00160
      COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH                             GRN00170
      COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM            GRN00180
C*****************************************************************GRN00190
C** THE FOLLOWING VARIABLES SUPPLIED BY THE USER.                     GRN00200
C**  FIELD DATA                                                       GRN00210
C**       XNORTH     FIELD COORDINATES FROM NORTH (DEGREES)           GRN00220
C**       HEAD       GRENADE TANK HEADING CLOCKWISE FROM NORTH (DEGREES) GRN00230
C**       RNG        DISTANCE OF GRENADES FROM TANK (METERS)          GRN00240
C**       DLEN       GRENADE SPACING (PERPENDICILAR TO HEADING)       GRN00250
C**       XO,YO,ZO   COORDINATES OF OBSERVER (METERS)                 GRN00260
C**       XM,YM,ZM   COORDINATES OF TANK (METERS)                     GRN00270
C**       XT,YT,ZT   TARGET COORDINATES (METERS)                      GRN00280
C** METEOROLOGICAL DATA                                              GRN00290
C**       WINDP      WIND PROFILE EXPONENT                            GRN00300
C**       HM         HEIGHT OF INVERSION LAYER                        GRN00310
C**       WD         WIND DIRECTION FROM NORTH (DEGREES)              GRN00320
C**       WS         WIND SPEED (METERS PER SECOND)                   GRN00330
C**       RH         RELATIVE HUMIDITY                                GRN00340
C**       ICAT       PASQUILL CATEGORY                                GRN00350
C**       YF         SMOKE YIELD FACTOR                               GRN00360
C** MUNITION DATA                                                    GRN00370
C**       EFF        CLOUD-MAKING EFFIENCY OF MUNITION                GRN00380
C**       QMUN       TOTAL MASS OF SMOKE AGENT (GRAMS)                GRN00390
C** DETECTOR DATA                                                    GRN00400
C**       WAVE1      WAVELENGTH OF INTEREST (MICRONS)                 GRN00410
C** DIFFUSION PARAMETERS                                             GRN00420
C**       SIGZ       REFERENCE SIGMA (METERS)                         GRN00430
C**       XREF       REFERENCE DISTANCE (METERS)                      GRN00440
C**       ZDIFF      VERTICAL DIFFUSION CONSTANT                      GRN00450
C**       YDIFF      CROSSWIND DIFFUSION CONSTANT                     GRN00460
C**       HK         TERRAIN SCAVENGING COEFFICIENT                   GRN00470
C**       VS         PARTICLE SETTLING VELOCITY (CM/SEC)              GRN00480
C**       RC         TERRAIN REFLECTION COEFFICIENT                   GRN00490
C*****************************************************************GRN00500
C**    DEFINITIONS OF OTHER VARIABLES***                              GRN00510
C**       IXMAX      NUMBER OF POINTS ALONG LINE-OF-SIGHT FOR CL COMPUTAIOGRN00520
C**       NBPT       NUMBER OF GRENADE LINES (NBPT=1)                 GRN00530
C**       NTARG      NUMBER OF TARGETS (NTARG=1)                      GRN00540
C**       TBURST     MUNITION DETONATION TIME                         GRN00550
C*****************************************************************   GRN00560
C**    ERROR CODES AND OPTION CODES:                                  GRN00570
C**       IWRIT=1    DEPRESSES RAW DATA PRINTOUT                      GRN00580
C**       IFLAG=4    INVALAD DATA CARD (BUT IGNORED)                  GRN00590
C**       IFLAG=3    OVER 11 DATA CARDS ENTERED BEFORE GO             GRN00600
C**                  REMAINDER IGNORED                                GRN00610
C**       IFLAG=2    NORMAL READ TERMINATION                          GRN00620
C**       IFLAG=1    WAVELENGTH OF INTEREST NOT IN DEFINED BANDS      GRN00630
C**                  TRANS SET TO 1.0                                 GRN00640
C*****SET DEFAULTS*******                                            GRN00650
      TBURST=0.0                                                        GRN00660
      CALL GOGET(WAVE1,KWAVE)                                           GRN00670
C*****READ DATA AND WRITE HEADING*****                               GRN00680
      WRITE(IOOUT,8000)                                                 GRN00690
 8000 FORMAT(1H0,20X,40(2H**),/,21X,1H*,34X,14HPROGRAM GRNADE,30X,1H*,/,GRN00700
```

```
     *21X,1H*,37X,8HEOSAEL80,33X,1H*,/,21X,40(2H**))                    GRN00710
      IWRIT=1                                                           GRN00720
      IFLAG=0                                                           GRN00730
    6 CALL DATRD(IWRIT,IFLAG)                                           GRN00740
      IERR=IFLAG                                                        GRN00750
      IF(IFLAG.EQ.4)GO TO 9999                                          GRN00760
C******CALCULATE INTEGRATION INCREMENT                                  GRN00770
      XX=(XT-XO)**2+(YT-YO)**2+(ZT-ZO)**2                               GRN00780
      DLOS=SQRT(XX)                                                     GRN00790
      IXMAX=IFIX(DLOS)                                                  GRN00800
      IF(IXMAX.GT.1000)IXMAX=1000                                       GRN00810
C******DEFAULT TO CLIMATE DATA OPTION                                   GRN00820
      IF(ICLMAT.NE.1)GO TO 12                                           GRN00830
      RHA=RH                                                            GRN00840
      UW=WNDVEL                                                         GRN00850
      WD=WNDDIR                                                         GRN00860
      ICAT=IPASCT                                                       GRN00870
   12 CONTINUE                                                          GRN00880
      IF(DLEN.EQ.0.0)DLEN=10.0                                          GRN00890
      IF(BRATE.EQ.0.0)BRATE=(1.0/14.3)                                  GRN00900
      QLENTH=XN*DLEN                                                    GRN00910
      BREXP=BRATE                                                       GRN00920
      QMUN=XN*FW                                                        GRN00930
      EMUN=QMUN*YF*EFF/100.0                                            GRN00940
      WS=UW                                                             GRN00950
      IF(XMIS(1).LE.0.0)GO TO 4                                         GRN00960
      XREFZ=100.0                                                       GRN00970
      SIGZR=XMIS(1)                                                     GRN00980
      ZDIFF=XMIS(2)                                                     GRN00990
      YDIFF=XMIS(3)                                                     GRN01000
      HM=XMIS(4)                                                        GRN01010
      HK=XMIS(5)                                                        GRN01020
      RC=XMIS(6)                                                        GRN01030
      VS=XMIS(7)                                                        GRN01040
      GO TO 5                                                           GRN01050
    4 CALL PARMS(ICAT)                                                  GRN01060
    5 CONTINUE                                                          GRN01070
C******REDEFINE WIND PROFILE EXPONENT IF READ IN POSITIVE               GRN01080
      IF(WPOWR.GE.0.0)WINDP=WPOWR                                       GRN01090
      IF(WS.LE.0.0)  WS=0.1                                             GRN01100
      U2=WS                                                             GRN01110
C-------------------------------------------------------------------    GRN01120
C                        LOCATE GENERATING LINE                         GRN01130
C-------------------------------------------------------------------    GRN01140
      THETA=(HEAD-WD)*PIRAD                                             GRN01150
C*** LOCATE CENTER OF GENERATING LINE.                                  GRN01160
      YHEAD=(XNORTH+90.0-HEAD)*PIRAD                                    GRN01170
      XBURST=RNG*SIN(YHEAD)+XM                                          GRN01180
      YBURST=RNG*COS(YHEAD)+YM                                          GRN01190
      ZBURST=ZM                                                         GRN01200
C-------------------------------------------------------------------    GRN01210
C                          TRANSFORM TO OBSERVER COORDINATES            GRN01220
C-------------------------------------------------------------------    GRN01230
C*** TRANSFORM TO ORIGIN UNDER OBSERVER AND X-AXIS UNDER TARGET.        GRN01240
      ANGLR=ATAN2(YT-YO,XT-XO)                                          GRN01250
      XI(1)=(XBURST-XO)*COS(ANGLR)+(YBURST-YO)*SIN(ANGLR)               GRN01260
      YI(1)=(YBURST-YO)*COS(ANGLR)-(XBURST-XO)*SIN(ANGLR)               GRN01270
      ZI(1) = ZBURST                                                    GRN01280
      TTI(1) = TBURST                                                   GRN01290
C*** GET THE WIND DIRECTION ANGLE WITH THE NEW X-AXIS.                  GRN01300
      ANGLD = ANGLR*180./PI                                             GRN01310
      WX=-(WD+XNORTH+ANGLD)                                             GRN01320
      WX=AMOD(WX, 360.0)                                                GRN01330
      WDA = WX*PI/180.                                                  GRN01340
C-------------------------------------------------------------------    GRN01350
C                        WRITE INPUT DATA AND HEADINGS                  GRN01360
C-------------------------------------------------------------------    GRN01370
      WRITE(IOOUT,8001)                                                 GRN01380
      WRITE(IOOUT,1000)                                                 GRN01390
      WRITE(IOOUT,1001)WS,WD,ICAT,RHA,XNORTH                            GRN01400
```

```
      WRITE(IOOUT,998)                                                  GRN01410
      WRITE(IOOUT,995)                                                  GRN01420
      WRITE(IOOUT,997)XO,EXTC(1),YO,EXTC(2),ZO,EXTC(3),                 GRN01430
     *XT,EXTC(4),YT,EXTC(5),ZT,EXTC(6),EXTC(7)                          GRN01440
      WRITE(IOOUT,1002)                                                 GRN01450
      WRITE(IOOUT,1003)XM,SIGZR,YM,XREFZ,ZM,HM                          GRN01460
      WRITE(IOOUT,1004)HEAD,HK,RNG,RC,XN,VS                             GRN01470
      WRITE(IOOUT,1005)QMUN,WINDP,QLENTH,ZDIFF,BREXP,YDIFF              GRN01480
      WRITE(IOOUT,1006)EFF,YF                                           GRN01490
 8001 FORMAT(//,21X,36H*****INPUT*****ALL LENGTHS IN METERS,/,          GRN01500
     *21X,38(2H--))                                                     GRN01510
 1000 FORMAT(21X,15HMETEOROLOGICAL:)                                    GRN01520
 1001 FORMAT(24X,10HWIND SPEED,10X,F6.1,1X,3HM/S,/,                     GRN01530
     *       24X,14HWIND DIRECTION,6X,F6.1,1X,3HDEG,/,                  GRN01540
     *       24X,17HPASQUILL CATEGORY,3X,I3,/,                          GRN01550
     *       24X,17HRELATIVE HUMIDITY,3X,F6.1,1X,1H%,/,                 GRN01560
     *       24X,21HNOTE: X AXIS HEADING:,1X,F6.1,1X,3HDEG,1X,          GRN01570
     *       28HCLOCKWISE FROM NORTH (DCWFN))                           GRN01580
 1002 FORMAT(20X,19HTANK/MUNITION DATA:,16X,21HDIFFUSION PARAMETERS:)   GRN01590
 1003 FORMAT(24X,7HX(TANK),12X,F6.1,10X,10HSIGZ(XREF),21X,F6.1,/,       GRN01600
     *       24X,7HY(TANK),12X,F6.1,10X,4HXREF,27X,F6.1,/,              GRN01610
     *       24X,7HZ(TANK),12X,F6.1,10X,17HMIXING HEIGHT(HM),14X,F6.1)  GRN01620
 1004 FORMAT(24X,14HHEADING(DCWFN),5X,F6.1,10X,20HSCAVENGING COEFF(HK), GRN01630
     *11X,F6.3,/,24X,5HRANGE,14X,F6.1,10X,20HREFLECTION COEFF(RC),11X,  GRN01640
     *F6.3,/,24X,10HNO GRNADES,9X,F6.1,10X,21HSETTLING VELOCITY(VS),    GRN01650
     *5X,F6.3,1X,4HCM/S)                                                GRN01660
 1005 FORMAT(24X,14HSMOKE MASS(GM),5X,F6.1,10X,29HVERTICAL WIND EXPONENTGRN01670
     *(WPOWR),2X,F6.3,/,24X,11HLINE LENGTH,8X,F6.1,10X,                 GRN01680
     *29HVERTICAL DIFF CONSTANT(ZDIFF),2X,F6.3,/,                       GRN01690
     *24X,13HBURN CONSTANT,2X,F6.3,1X,3H1/S,10X,                        GRN01700
     *30HCROSSWIND DIFF CONSTANT(YDIFF),1X,F6.3)                        GRN01710
 1006 FORMAT(24X,10HEFFICIENCY,9X,F6.1,10X,12HYIELD FACTOR,17X,F6.1)    GRN01720
  998 FORMAT(21X,16HOBSERVER/TARGET:,8X,24HEXTINCTION COEFFICIENTS:)    GRN01730
  995 FORMAT(49X,7HMICRONS,3X,7HM**2/GM)                                GRN01740
  997 FORMAT(24X,6HX(OBS),3X,F6.1,10X,7H0.4-0.7,3X,F6.3,/,              GRN01750
     *       24X,6HY(OBS),3X,F6.1,10X,7H0.7-1.2,3X,F6.3,/,              GRN01760
     *       24X,6HZ(OBS),3X,F6.1,10X,7H 1.06  ,3X,F6.3,/,              GRN01770
     *       24X,6HX(TAR),3X,F6.1,10X,7H3.0-5.0,3X,F6.3,/,              GRN01780
     *       24X,6HY(TAR),3X,F6.1,10X,7H8.0-12.,3X,F6.3,/,              GRN01790
     *       24X,6HZ(TAR),3X,F6.1,10X,7H 10.6  ,3X,F6.3,/,              GRN01800
     *       49X,4H94.0,1X,3HGHZ,2X,F6.3)                               GRN01810
      WRITE(IOOUT,996)                                                  GRN01820
  996 FORMAT(1H1,21X,16H*****OUTPUT*****,/,21X,38(2H--))                GRN01830
      WRITE(IOOUT,3000)                                                 GRN01840
 3000 FORMAT(24X,4HTIME,6X,2HCL,23X,12HTRANSMISSION,/,24X,5H(SEC),2X,   GRN01850
     *9H(GM/M**2),2X,7H0.4-0.7,1X,7H0.7-1.2,3X,4H1.06,2X,7H3.0-5.0,1X,  GRN01860
     *7H8.0-12.,2X,4H10.6,4X,5H94GHZ)                                   GRN01870
C-----------------------------------------------------------------------GRN01880
C                        BEGIN CL CALCULATIONS                          GRN01890
C-----------------------------------------------------------------------GRN01900
      DO 400 IT=ISTO,IETO,IDTO                                          GRN01910
      ITT=IT                                                            GRN01920
C     * SET UP LOOP ON SPACIAL DISTRIBUTION                            GRN01930
      XC = 0.0                                                          GRN01940
      YC = 0.0                                                          GRN01950
      ZC = ZO                                                           GRN01960
      DELX=SQRT( (XT-XO)**2+(YT-YO)**2 )/IXMAX                          GRN01970
      DELZ = (ZT-ZO)/IXMAX                                              GRN01980
C** FOR EACH TIME GET THE CONCENTRATION AT IXMAX POINTS ALONG LINE-OF-SIGRN01990
            DO 300 IX=1,IXMAX                                           GRN02000
            XC = XC + DELX                                              GRN02010
            ZC = ZC + DELZ                                              GRN02020
            XA(1) = XC                                                  GRN02030
            XA(2) = YC                                                  GRN02040
            XA(3) = ZC                                                  GRN02050
            T = FLOAT(IT)                                               GRN02060
            UBXB=U2                                                     GRN02070
            CALL CONCN(XA,T,C)                                          GRN02080
            XDTA(IX)=XC                                                 GRN02090
            CDTA(IX)=C                                                  GRN02100
```

```
  300 CONTINUE                                                          GRN02110
C*** INTEGRATE ALONG LINE-OF-SIGHT TO GET CL.                           GRN02120
      CALL SUMA (XDTA,CDTA, CL ,IXMAX)                                  GRN02130
C-----------------------------------------------------------------------GRN02140
C                              CALCULATE TRANSMITTANCE                  GRN02150
C-----------------------------------------------------------------------GRN02160
      DO 500 J=1,7                                                      GRN02170
      XTRAN(J)=EXP(-1.0*EXTC(J)*CL(IXMAX))                              GRN02180
  500 CONTINUE                                                          GRN02190
      WRITE(IOOUT,2000)T,CL(IXMAX),(XTRAN(J),J=1,7)                     GRN02200
 2000 FORMAT(23X,F5.1,2X,F8.3,2X,7(1X,F7.3))                            GRN02210
  400 CONTINUE                                                          GRN02220
C*****                                                                  GRN02230
C*****SET TRANSMISSION FOR RETURN TO EOSAEL                             GRN02240
C*****                                                                  GRN02250
      IF(KWAVE.GT.0)GO TO 3                                             GRN02260
      TRANS=1.0                                                         GRN02270
      IFLAG=1                                                           GRN02280
    3 TRANS=XTRAN(KWAVE)                                                GRN02290
      GO TO 6                                                           GRN02300
 9999 WRITE(IOOUT,9000)                                                 GRN02310
 9000 FORMAT(21X,28H*****PROGRAM GRNADE END*****)                       GRN02320
      RETURN                                                            GRN02330
      END                                                               GRN02340
```

```
      SUBROUTINE CONCN(XA,T,C2)                                          COC00010
C/*****************************************************************/   COC00020
C/*                          SUBROUTINE CONCN                     */   COC00030
C/*                            GRNAD MODULE                       */   COC00040
C/*                             EOSAEL80                          */   COC00050
C/*****************************************************************/   COC00060
C/* PURPOSE:                                                      */   COC00070
C/*      CALCULATES CONCENTRATION AT A SPECIFIED POSITON AND TIME. */   COC00080
C/* USAGE:                                                        */   COC00090
C/*      THE CONCENTRATION IS USED WITH THE EXTINCTION COEFFICIENT */   COC00100
C/*      TO COMPUTE TRANSMITTANCE.                                */   COC00110
C/* DESCRIPTION OF PARAMETERS:                                    */   COC00120
C/*      XA          - POSITION IN METERS, INPUT.                 */   COC00130
C/*      T           - TIME IN SECONDS, INPUT.                    */   COC00140
C/*      C2          - CONCENTRATION, OUTPUT.                     */   COC00150
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:                */   COC00160
C/*                LOCAT                    UMEAN                 */   COC00170
C/* COMMON BLOCK STATEMENTS REQUIRED:                             */   COC00180
C/*      MECH1      MECH2      MECH3                              */   COC00190
C/* REMARKS:                                                      */   COC00200
C/*      CONCN COMPUTES FIVE TERMS AND MULTIPLIES THEM TO         */   COC00210
C/*      GET CONCENTRATION.                                       */   COC00220
C/*****************************************************************/   COC00230
      DIMENSION XA(3)                                                    COC00240
      COMMON/CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDECK                       COC00250
      COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP                   COC00260
      COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH                              COC00270
      COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM             COC00280
C--------------------------------------------------------------------   COC00290
C        CONCENTRATION=TERM1*TERM2*TERM3*TERM4*TERM5                    COC00300
C--------------------------------------------------------------------   COC00310
      C2 = 0.0                                                           COC00320
      X = XA(1)                                                          COC00330
      Y = XA(2)                                                          COC00340
      Z = XA(3)                                                          COC00350
C*** GET CROSSWIND AND DOWNWIND COMPONENTS OF GENERATING LINE.           COC00360
      QC=ABS(QLENTH*COS(THETA))                                          COC00370
      QD=ABS(QLENTH*SIN(THETA))                                          COC00380
      NBPT=1                                                             COC00390
      DO 380 J = 1,NBPT                                                  COC00400
      TJ = T- TTI(J)                                                     COC00410
C--------------------------------------------------------------------   COC00420
C              TERM1 IS CLOUD MASS AS A FUNCTION OF TIME.              COC00430
C--------------------------------------------------------------------   COC00440
      IF(BREXP*TJ.GE.200.0)STOP01                                        COC00450
      TRM1=EMUN*(1.0-EXP(-BREXP*TJ))                                     COC00460
C--------------------------------------------------------------------   COC00470
C                   TERM2 IS TERRAIN SCAVENGING TERM.                  COC00480
C--------------------------------------------------------------------   COC00490
C** CHANGE TO SMOKE COORDINATES.                                         COC00500
      CALL LOCAT(J,X,Y,Z,XB,YB,ZB)                                       COC00510
C*** UPWIND END OF GENERATING LINE WILL BE ORIGIN OF SMOKE SYSTEM.       COC00520
      XB=XB+0.5*QD                                                       COC00530
      CALL UMEAN(J,TJ)                                                   COC00540
      IF(HK*XB/UBXB.GE.200.0)STOP01                                      COC00550
      TRM2 = EXP(-HK*XB/UBXB)                                            COC00560
C--------------------------------------------------------------------   COC00570
C                   TERM3 IS DOWNWIND PROBABILITY DENSITY.             COC00580
C--------------------------------------------------------------------   COC00590
      UT=UBXB*TJ                                                         COC00600
      IF(XB.LE.0.) GO TO 999                                             COC00610
      IF(XB.GT.0..AND.XB.LE.UT)  TERM3=1./(UT+0.5*QD)                    COC00620
      IF(XB.GT.UT.AND.XB.LE.UT+QD) TERM3=(UT+QD-XB)/(UT*QD+0.5*QD*QD)    COC00630
      IF(XB.GT.UT+QD)  TERM3=0.                                          COC00640
      TRM3=TERM3                                                         COC00650
C--------------------------------------------------------------------   COC00660
C                   TERM4 IS CROSSWIND PROBABILITY DENSITY.            COC00670
C--------------------------------------------------------------------   COC00680
      YWIDTH=YDIFF*XB+QC                                                 COC00690
      TRM4= 1./YWIDTH                                                    COC00700
```

```
      IF(YB .GT. 0.5*YWIDTH )  TRM4=0.                                          COC00710
      IF( YB .LT. -0.5*YWIDTH)  TRM4=0.                                         COC00720
C-----------------------------------------------------------------------------COC00730
C                    TERM5 IS VERTICAL PROBABILITY DENSITY.                   COC00740
C-----------------------------------------------------------------------------COC00750
      SIGZT=SIGZR*(XB/XREFZ)**ZDIFF                                             COC00760
      SZT2=SIGZT*SIGZT                                                          COC00770
      HMVT = ZI(J) - (VS/100.0)*(XB/UBXB)                                       COC00780
      TV1 = EXP(-(HMVT-ZB)**2/(2.0*SZT2))                                       COC00790
      TV2 = RC*EXP(-(HMVT+ZB)**2/(2.0*SZT2))                                    COC00800
      TERR = 1.0E-06                                                            COC00810
      JC = 0                                                                    COC00820
      T1 = 0.0                                                                  COC00830
      T2 = 0.0                                                                  COC00840
      T3 = 0.0                                                                  COC00850
      T4= 0.0                                                                   COC00860
      TS = 0.0                                                                  COC00870
  370 JC = JC+1                                                                 COC00880
      R1 = RC**(JC-1)                                                           COC00890
      R2 = RC**JC                                                               COC00900
      R3 = RC**(JC+1)                                                           COC00910
      IF( (2.*JC*HM+HMVT+ZB)**2 .GT. 600.*2.*SZT2)  GO TO 371                   COC00920
      T1 = R1*EXP(-(2.0*JC*HM-HMVT-ZB)**2/(2.0*SZT2))                           COC00930
      T2 = R2*EXP(-(2.0*JC*HM-HMVT+ZB)**2/(2.0*SZT2))                           COC00940
      T3 = R2*EXP(-(2.0*JC*HM+HMVT-ZB)**2/(2.0*SZT2))                           COC00950
      T4 = R3*EXP(-(2.0*JC*HM+HMVT+ZB)**2/(2.0*SZT2))                           COC00960
      T1234 = T1 + T2 + T3 + T4                                                 COC00970
      TS = TS + T1234                                                           COC00980
      IF(T1234 - TERR) 371,371,370                                              COC00990
  371 CONTINUE                                                                  COC01000
      TV3 = TS                                                                  COC01010
      TRM5 = TV1+TV2+TV3                                                        COC01020
      TRM5=TRM5/( SIGZT*SQRT(2.*PI) )                                           COC01030
      C1 = TRM1*TRM2*TRM3*TRM4*TRM5                                             COC01040
      C2 = C2 + C1                                                              COC01050
  380 CONTINUE                                                                  COC01060
  999 RETURN                                                                    COC01070
      END                                                                       COC01080
```

```
      SUBROUTINE GOGET(WAVE1,KWAVE)                                      GOG00010
C/*****************************************************************/GOG00020
C/*                    SUBROUTINE GOGET                            */GOG00030
C/*                      GRNAD MODULE                              */GOG00040
C/*                       EOSAEL80                                 */GOG00050
C/*****************************************************************/GOG00060
C*****SUBROUTINE FINDS SPECTRAL BAND FOR GIVEN SINGLE WAVELENGTH    GOG00070
      KWAVE=0                                                        GOG00080
      IF(WAVE1.GE.0.40.AND.WAVE1.LT.0.70)KWAVE=1                     GOG00090
      IF(WAVE1.GE.0.70.AND.WAVE1.LT.1.20)KWAVE=2                     GOG00100
      IF(WAVE1.GE.3.00.AND.WAVE1.LT.5.00)KWAVE=4                     GOG00110
      IF(WAVE1.GE.8.00.AND.WAVE1.LT.12.0)KWAVE=5                     GOG00120
      IF(WAVE1.EQ.1.06)                  KWAVE=3                     GOG00130
      IF(WAVE1.EQ.10.6)                  KWAVE=6                     GOG00140
      IF(WAVE1.EQ.94.0)                  KWAVE=7                     GOG00150
      IF(WAVE1.GT.3188.0.AND.WAVE1.LT.3195.0)KWAVE=7                 GOG00160
      RETURN                                                         GOG00170
      END                                                            GOG00180
```

```
      SUBROUTINE PARMS(ICAT)                                                PAR00010
C/*****************************************************************************/PAR00020
C/*                      SUBROUTINE PARMS                                     */PAR00030
C/*                       GRNAD MODULE                                        */PAR00040
C/*                         EOSAEL80                                          */PAR00050
C/*****************************************************************************/PAR00060
C*****SETS DIFFUSION PARAMETER DEFAULTS AS FUNCTION OF PASQUILL CATEGORY      PAR00070
      COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM                  PAR00080
      YDIFF=0.355                                                           PAR00090
      HK=0.002                                                              PAR00100
      VS=0.021                                                              PAR00110
      RC=0.70                                                               PAR00120
      XREFZ=100.0                                                           PAR00130
      IF(ICAT.GT.3)GO TO 1                                                  PAR00140
      WINDP=0.10                                                            PAR00150
      ZDIFF=2.08                                                            PAR00160
      HM=1000.0                                                             PAR00170
      SIGZR=14.0                                                            PAR00180
      GO TO 3                                                               PAR00190
    1 IF(ICAT.GT.4)GO TO 2                                                  PAR00200
      WINDP=0.20                                                            PAR00210
      ZDIFF=1.40                                                            PAR00220
      HM=300.0                                                              PAR00230
      SIGZR=7.2                                                             PAR00240
      GO TO 3                                                               PAR00250
    2 WINDP=0.40                                                            PAR00260
      ZDIFF=1.04                                                            PAR00270
      HM=50.0                                                               PAR00280
      SIGZR=5.0                                                             PAR00290
    3 RETURN                                                                PAR00300
      END                                                                   PAR00310
```

```
      SUBROUTINE EXTIN(EX)                                                  EXTN0010
C/*****************************************************************/EXTN0020
C/*                      SUBROUTINE EXTIN                         */EXTN0030
C/*                       GRNAD MODULE                            */EXTN0040
C/*                         EOSAEL80                              */EXTN0050
C/*****************************************************************/EXTN0060
C*****PROGRAM TO SET EXTINCTION COEFFICIENTS FOR WP/RP SMOKE****      EXTN0070
      DIMENSION EX(7),CX(7)                                         EXTN0080
      DATA CX/4.304,2.166,1.541,0.350,0.338,0.364,0.001/            EXTN0090
      DO 1 I=1,7                                                    EXTN0100
      EX(I)=CX(I)                                                   EXTN0110
    1 CONTINUE                                                      EXTN0120
      RETURN                                                        EXTN0130
      END                                                           EXTN0140
```

```
      SUBROUTINE UMEAN(J,TJ)                                                    UME00010
C/*****************************************************************************/UME00020
C/*                        SUBROUTINE UMEAN                                   */UME00030
C/*                          GRNAD MODULE                                     */UME00040
C/*                           EOSAEL80                                        */UME00050
C/*****************************************************************************/UME00060
C*****CALCULATES MEAN WIND SPEED OVER EXTENT OF CLOUD                          UME00070
      COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP                          UME00080
      COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH                                     UME00090
      COMMON/MECH3/ZDIFF,YDIFF,SIGZR,XREFZ,WINDP,HK,VS,RC,HM                    UME00100
      IC = 0                                                                    UME00110
      QD = ABS(QLENTH*SIN(THETA))                                               UME00120
      P = WINDP + 1.0                                                           UME00130
      C1=P*2.0**WINDP                                                           UME00140
      Z2=ZI(J)                                                                  UME00150
      IF(Z2-2.0)10,10,20                                                        UME00160
   10 CONTINUE                                                                  UME00170
      Z2=2.0                                                                    UME00180
      Z1=0.01                                                                   UME00190
      GO TO 200                                                                 UME00200
   20 Z1=ZI(J)-1.5*SIGZR                                                        UME00210
      IF(Z1)30,30,40                                                            UME00220
   30 Z1=0.01                                                                   UME00230
   40 CONTINUE                                                                  UME00240
      UH = (U2/((Z2-Z1)*C1))*(Z2**P-Z1**P)                                      UME00250
      US=UH                                                                     UME00260
   50 CONTINUE                                                                  UME00270
      XC = UH*TJ                                                                UME00280
      Z2 = ZI(J) - (VS/100.0)*(XC/UH)                                           UME00290
      IF(Z2-2.0)151,151,155                                                     UME00300
  151 CONTINUE                                                                  UME00310
      Z2=2.0                                                                    UME00320
      Z1=0.01                                                                   UME00330
      GO TO 160                                                                 UME00340
  155 CONTINUE                                                                  UME00350
      SIGZT = SIGZR*((XC+QD)/XREFZ)**ZDIFF                                      UME00360
      Z1 = ZI(J) - 1.5*SIGZT                                                    UME00370
      IF(Z1)159,159,160                                                         UME00380
  159 CONTINUE                                                                  UME00390
      Z1=0.01                                                                   UME00400
  160 UBXJ = (U2/((Z2-Z1)*C1))*(Z2**P-Z1**P)                                    UME00410
      GO TO 1000                                                                UME00420
  200 CONTINUE                                                                  UME00430
      US=U2                                                                     UME00440
      UH=US                                                                     UME00450
      UBXJ=US                                                                   UME00460
 1000 CONTINUE                                                                  UME00470
      UBXJ = SQRT((UBXJ**2 + UH**2)/2.0)                                        UME00480
      IC = IC + 1                                                               UME00490
      IF(IC .EQ.1) UH = UBXJ                                                    UME00500
      IF(IC.EQ.1) GO TO 50                                                      UME00510
      UBXB=UBXJ                                                                 UME00520
      RETURN                                                                    UME00530
      END                                                                       UME00540
```

```
      SUBROUTINE SUMA(X,Y,Z,NDIM)                                          SUMA0010
C/*****************************************************************/      SUMA0020
C/*                          SUBROUTINE SUMA                      */      SUMA0030
C/*                            GRNAD MODULE                       */      SUMA0040
C/*                             EOSAEL80                          */      SUMA0050
C/*****************************************************************/      SUMA0060
C/*                                                               */      SUMA0070
C/* PURPOSE:                                                      */      SUMA0080
C/*      GENERAL PURPOSE INTEGRATION SUBROUTINE                   */      SUMA0090
C/* USAGE:                                                        */      SUMA0100
C/*      CALLED FROM GRNAD TO GET INTEGRAL OF CONCENTRATION ALONG */      SUMA0110
C/*      LINE-OF-SIGHT.                                           */      SUMA0120
C/* DESCRIPTION OF PARAMETERS:                                    */      SUMA0130
C/*      X          INDEPENDENT VARIABLE                         */      SUMA0140
C/*      Y          DEPENDENT VARIABLE                           */      SUMA0150
C/*      Z          INTEGRAL OF Y OVER X                         */      SUMA0160
C/*      INDIM      NUMBER OF POINTS                             */      SUMA0170
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:                */      SUMA0180
C/*      NONE                                                     */      SUMA0190
C/* COMMON BLOCK STATEMENTS REQUIRED:                             */      SUMA0200
C/*      NONE                                                     */      SUMA0210
C/* REMARKS:                                                      */      SUMA0220
C/*      AS USED BY GRENADE, X IS POSITION ONLINE-OF-SIGHT, Y IS  */      SUMA0230
C/*      LOCAL CONCENTRATION, Z IS TOTAL CONCENTRATION.           */      SUMA0240
C/* METHOD:                                                       */      SUMA0250
C/*      SUMA INCREMENTS INTEGRAL BY AVERAGE OF LAST TWO Y VALUES */      SUMA0260
C/*      TIMES DELTA X.                                           */      SUMA0270
C/*****************************************************************/      SUMA0280
      DIMENSION X(1000),Y(1000),Z(1000)                                    SUMA0290
      SUM2          = 0.0                                                  SUMA0300
      IF              (NDIM - 1) 4,3,1                                     SUMA0310
C                     *** INTEGRATION LOOP ***                             SUMA0320
 1    DO 2 I        = 2,NDIM                                               SUMA0330
        SUM1        =   SUM2                                               SUMA0340
        SUM2        =   SUM2 + 0.500*(X(I)-X(I-1))*(Y(I)+Y(I-1))           SUMA0350
 2      Z(I-1)      =   SUM1                                               SUMA0360
 3    Z(NDIM)       = SUM2                                                 SUMA0370
 4    RETURN                                                               SUMA0380
      END                                                                  SUMA0390
```

```
      SUBROUTINE LOCAT(I,X,Y,Z,XB,YB,ZB)                                LOC00010
C/*****************************************************************/  LOC00020
C/*                          SUBROUTINE LOCAT                     */  LOC00030
C/*                            GRNAD MODULE                       */  LOC00040
C/*                             EOSAEL80                          */  LOC00050
C/*****************************************************************/  LOC00060
C/* PURPOSE:                                                      */  LOC00070
C/*      LOCAT TRANSLATES POSITION INTO BURST AND WIND COORDINATES. */  LOC00080
C/* USAGE:                                                        */  LOC00090
C/*      CALLED BY CONCN                                          */  LOC00100
C/* DESCRIPTION OF PARAMETERS:                                    */  LOC00110
C/*      T          TIME (SECONDS), INPUT                        */  LOC00120
C/*      X,Y,Z      ORIGINAL POSITION (METERS), INPUT            */  LOC00130
C/*      XI,YI,ZI   ORIGINAL BURST POSITION (METERS), INPUT      */  LOC00140
C/*      WDA        ANGLE BETWEEN WIND VECTOR AND LINE OF SIGHT  */  LOC00150
C/*                 (RADIANS), INPUT                             */  LOC00160
C/*      XB,YB,ZB   TRANSLATED POSITION (METERS), OUTPUT         */  LOC00170
C/* SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED:                */  LOC00180
C/*      NONE                                                     */  LOC00190
C/* COMMON BLOCK STATEMENTS REQUIRED:                             */  LOC00200
C/*      MECH1      MECH2                                         */  LOC00210
C/* REMARKS:                                                      */  LOC00220
C/*      ORIGINAL COORDINATES HAVE ORIGIN AT OBSERVER AND X-AXIS  */  LOC00230
C/*      THROUGH TARGET.  NEW COORDINATES HAVE ORIGIN AT BURST    */  LOC00240
C/*      AND X-AXIS IN DIRECTION OF WIND VECTOR.                  */  LOC00250
C/* METHOD:                                                       */  LOC00260
C/*      STANDARD ROTATION AND TRANSLATION OF AXES                */  LOC00270
C/*****************************************************************/  LOC00280
      COMMON/MECH1/XI(1),YI(1),ZI(1),TTI(1),EMUN,BREXP                LOC00290
      COMMON/MECH2/U2,WDA,THETA,UBXB,QLENTH                           LOC00300
      XB = (X -XI(I))*COS(WDA) + (Y-YI(I))*SIN(WDA)                   LOC00310
      YB =-(X-XI(I))*SIN(WDA) + (Y - YI(I))*COS(WDA)                  LOC00320
      ZB=Z                                                            LOC00330
      IF(ZB.LT.0.0.AND.ABS(ZB).GT.ZI(I)) ZB=0.0                       LOC00340
      RETURN                                                          LOC00350
      END                                                             LOC00360
```

```
      SUBROUTINE DATRD(IWRIT,IFLAG)                                          DAT00010
C/*****************************************************************************/DAT00020
C/*                      SUBROUTINE DATRD                                     */DAT00030
C/*                        GRNAD MODULE                                       */DAT00040
C/*                         EOSAEL80                                        /*/DAT00050
C/*****************************************************************************/DAT00060
C*****THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS          DAT00070
C     THE SMOKE(EOSAEL) PROGRAM                                              DAT00080
C     INPUTS                                                                 DAT00090
C       EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,              DAT00100
C       FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.                 DAT00110
C       PER FIELD BEGINNING IN COL 11.  THE CARDS ARE NOT ORDER              DAT00120
C       DEPENDENT.                                                           DAT00130
C       NAME                IGNORED                                          DAT00140
C       BURN                IGNORED                                          DAT00150
C       MUNC    XM,YM,ZM    COORDINATES OF GRENADE FIRING TANK               DAT00160
C               HEAD        HEADING OF GRENADE FIRING TANK CLOCKWISE         DAT00170
C                           FROM NORTH                                       DAT00180
C               RNG         GRENADE FIRING RANGE OF TANK                     DAT00190
C               DLEN        SPACING OF GRENADES ALONG LINE PERPENDICULAR     DAT00200
C                           TO HEADING                                       DAT00210
C       OBSC    XO,YO,ZO    COORDINATES OF THE OBSERVER   (M,M,M)           DAT00220
C       TARC    XT,YT,ZT    COORDINATES OF THE TARGET    (M,M,M)             DAT00230
C       BART                                                                 DAT00240
C               STO         STARTING TIME  (ELAPSED TIME SINCE BLAST)        DAT00250
C               ETO         ENDING TIME FOR CALCULATION                      DAT00260
C               DTO         TIME INCREMENT FOR CALCULATION                   DAT00270
C               XNORTH      X AXIS HEADING CLOCKWISE FROM NORTH              DAT00280
C       MUNT                                                                 DAT00290
C               XN          NUMBER OF MUNITIONS FIRED AT THE SAME            DAT00300
C                           LOCATION AND AT THE SAME TIME                    DAT00310
C               FW          FILL WEIGHT )LBS)                                DAT00320
C               TBURN       BURN TIME OF SMOKE TYPE (IGNORED)                DAT00330
C               ITYPE       TYPE OF SMOKE (DEFAULTS TO 1)                    DAT00340
C                           1.=WP, 2.=PWP, 3.=HC, 4.=FOG OIL                 DAT00350
C               EFF         EFFICIENCY OF BURN (PERCENT).  IF 0.0,           DAT00360
C                           DEFAULTS TO 62.0%.                               DAT00370
C               YF          YIELD FACTOR IF 0.0,DEFAULTS TO ANALYTICAL       DAT00380
C                           MODEL                                            DAT00390
C               BRATE       EXPONENTIAL BURN RATE PARAMETER                  DAT00400
C       METR                                                                 DAT00410
C               RHA         RELATIVE HUMIDITY    (PERCENT)                   DAT00420
C               UW          WIND VELOCITY    (M/S)                           DAT00430
C               WD          WIND DIRECTION  (DEGREES)                        DAT00440
C               ICAT        PASQUILL CATEGORY                                DAT00450
C                             1-A, 2-B, 3-C, 4-D, 5-E, 6-F                   DAT00460
C               AIRT        SURFACE AIR TEMPERATURE (IGNORED)                DAT00470
C               TGRAD       VERT TEMP GRADIENT (IGNORED)                     DAT00480
C               WPOWR       WIND PROFILE EXPONENT (DIMENSIONLESS)            DAT00490
C       EXTC                DESIRED CHANGES IN EXTINCTION COEFF.             DAT00500
C                           (OPTIONAL). IF NOT USED OR READ AS 0.            DAT00510
C                           DEFAULTS TO ALPHA ARRAY VALUE IN STRNS.          DAT00520
C                            BANDS ARE:                                      DAT00530
C                               0.4-0.7 MICRONS                              DAT00540
C                               0.7-1.2 MICRONS                              DAT00550
C                                 1.06  MICRONS                              DAT00560
C                               3.0-5.0 MICRONS                              DAT00570
C                               8.0-12. MICRONS                              DAT00580
C                                 10.6  MICRONS                              DAT00590
C                                 94.0  GHZ.                                 DAT00600
C       MISC                DIFFUSION PARAMETER OPTION CARD FOR GRNAD        DAT00610
C               SIGZR       DOWNWIND REFERENCE AT 100 M REFERENCE DIST.       DAT00620
C               ZDIFF        VERTICAL DIFFUSION COEFFICIENT                  DAT00630
C               YDIFF       CROSSWIND DIFFUSION COEFFICIENT                  DAT00640
C               HM          HEIGHT OF MIXING LAYER  (METERS)                 DAT00650
C               HK          TERRAIN SCAVENGING COEFFICIENT                   DAT00660
C               RC          TERRAIN REFLECTION COEFFICIENT                   DAT00670
C               VS          SETTLING VELOCITY   (CM SEC)                     DAT00680
C       GO              SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT      DAT00690
C       DONE            END OF JOB.                                          DAT00700
```

```
C*****************************************************************      DAT00710
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUDAT00720
      COMMON /GEOMET/PTS(15),IGEOSW                                      DAT00730
      COMMON/CLYMAT/TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,FOGPRB,        DAT00740
     *              WNDVEL,WNDDIR,IPASCT                                 DAT00750
      COMMON/MECH0/XM,YM,ZM,XO,YO,ZO,XT,YT,ZT,ISTO,IETO,IDTO,XN,FW,      DAT00760
     *TBURN,ITYPE,EFF,YF,RHA,UW,WD,ICAT,AIRT,TGRAD,BRATE,HEAD,RNG,       DAT00770
     *DLEN,WPOWR,EXTC(8),XMIS(8),XNORTH                                  DAT00780
      DIMENSION DUMY(8),IR(26),IR1(10),R1(10),EX(7),INAME(35)            DAT00790
      DATA IR/2HME,2HTR,2HMU,2HNT,2HBA,2HRT,2HMU,2HNC,2HOB,2HSC,2HTA,    DAT00800
     *2HRC,2HEX,2HTC,2HBU,2HRN,2HMI,2HSC,2HGO,2H  ,2HDO,2HNE,2HNA,2HME,  DAT00810
     *2HOU,2HTP/                                                         DAT00820
      IF (IFLAG.GT.0) GO TO 8                                            DAT00830
      DO 2 J=1,8                                                         DAT00840
      XMIS(J)=0.0                                                        DAT00850
    2 EXTC(J)=0.                                                         DAT00860
C***                                                                     DAT00870
C***  BEGINNING OF READ LOOP                                             DAT00880
C***                                                                     DAT00890
8     CONTINUE                                                           DAT00900
      IF(IWRIT.EQ.0)GO TO 6                                              DAT00910
      WRITE(IOOUT,200)                                                   DAT00920
  200 FORMAT(1H0,21X,20H*****CARD INPUT*****,/,21X,40(2H--))             DAT00930
  201 FORMAT(2A2,6X,35A2)                                                DAT00940
  202 FORMAT(1H0,21X,2A2,6X,35A2)                                        DAT00950
    6 DO 70 I=1,13                                                       DAT00960
      IF(I.EQ.13) GO TO 310                                              DAT00970
      IF(IFLAG.GT.0) GO TO 4                                             DAT00980
      IFLAG=1                                                            DAT00990
      READ(IOIN,201)IR1(1),IR1(2),(INAME(J),J=1,35)                      DAT01000
      IF(IWRIT.EQ.0) GO TO 4                                             DAT01010
      WRITE(IOOUT,202)IR1(1),IR1(2),(INAME(J),J=1,35)                    DAT01020
    4 READ(IOIN,20)IR1(1),IR1(2),(R1(J),J=2,8)                           DAT01030
      IF(IWRIT.EQ.0) GO TO 5                                             DAT01040
      WRITE(IOOUT,30)IR1(1),IR1(2),(R1(J),J=2,8)                         DAT01050
    5 IF(IR1(1).EQ.IR(21).AND.IR1(2).EQ.IR(22)) GO TO 998                DAT01060
   20 FORMAT(2A2,6X,7F10.3)                                              DAT01070
   30 FORMAT(1H0,21X,2A2,6X,7F10.3)                                      DAT01080
C*****                                                                   DAT01090
C***  RELATING INPUT DATA TO VARIABLE NAMES.                             DAT01100
C*****                                                                   DAT01110
           IF(IR1(1).EQ.IR(1).AND.IR1(2).EQ.IR(2))    GO TO 90           DAT01120
           IF(IR1(1).EQ.IR(3).AND.IR1(2).EQ.IR(4))    GO TO 100          DAT01130
           IF(IR1(1).EQ.IR(5).AND.IR1(2).EQ.IR(6))    GO TO 110          DAT01140
           IF(IR1(1).EQ.IR(7).AND.IR1(2).EQ.IR(8))    GO TO 120          DAT01150
           IF(IR1(1).EQ.IR(9).AND.IR1(2).EQ.IR(10))   GO TO 130          DAT01160
           IF(IR1(1).EQ.IR(11).AND.IR1(2).EQ.IR(12))  GO TO 140          DAT01170
           IF(IR1(1).EQ.IR(13).AND.IR1(2).EQ.IR(14))  GO TO 150          DAT01180
           IF(IR1(1).EQ.IR(15).AND.IR1(2).EQ.IR(16))  GO TO 155          DAT01190
           IF(IR1(1).EQ.IR(17).AND.IR1(2).EQ.IR(18))  GO TO 165          DAT01200
           IF(IR1(1).EQ.IR(19).AND.IR1(2).EQ.IR(20))  GO TO 175          DAT01210
           IF(IR1(1).EQ.IR(21).AND.IR1(2).EQ.IR(22))  GO TO 998          DAT01220
           IF(IR1(1).EQ.IR(23).AND.IR1(2).EQ.IR(24))  GO TO 70           DAT01230
           IF(IR1(1).EQ.IR(25).AND.IR1(2).EQ.IR(26))  GO TO 70           DAT01240
C*****                                                                   DAT01250
C     ERROR CAUTION FOR INVALID DATA CARD                                DAT01260
C*****                                                                   DAT01270
      IFLAG=2                                                            DAT01280
      WRITE(IOOUT,80)                                                    DAT01290
   80 FORMAT(21X,35H*****CAUTION***** INVALID DATA CARD)                 DAT01300
      GO TO 70                                                           DAT01310
   90       RHA    = R1(2)                                               DAT01320
            UW     = R1(3)                                               DAT01330
            WD     = R1(4)                                               DAT01340
            ICAT   =IFIX(R1(5))                                          DAT01350
            AIRT   = R1(6)                                               DAT01360
            TGRAD  = R1(7)                                               DAT01370
            WPOWR=R1(8)                                                  DAT01380
      GO TO 70                                                           DAT01390
  100       XN     = R1(2)                                               DAT01400
```

```
            FW     =453.6*R1(3)                                                  DAT01410
            TBURN  = R1(4)                                                       DAT01420
            ITYPE  = IFIX(R1(5))                                                 DAT01430
            EFF    = R1(6)                                                       DAT01440
            YF=R1(7)                                                             DAT01450
            BRATE=R1(8)                                                          DAT01460
      GO TO 70                                                                   DAT01470
  110       ISTO   = IFIX(R1(2))                                                 DAT01480
            IETO   = IFIX(R1(3))                                                 DAT01490
            IDTO   = IFIX(R1(4))                                                 DAT01500
            XNORTH= R1(5)                                                        DAT01510
            IF(ISTO.LE.0)ISTO=1                                                  DAT01520
      GO TO 70                                                                   DAT01530
  120       XM     = R1(2)                                                       DAT01540
            YM     = R1(3)                                                       DAT01550
            ZM     = R1(4)                                                       DAT01560
            HEAD   = R1(5)                                                       DAT01570
            RNG    = R1(6)                                                       DAT01580
            DLEN   = R1(7)                                                       DAT01590
      GO TO 70                                                                   DAT01600
  130       XO     = R1(2)                                                       DAT01610
            YO     = R1(3)                                                       DAT01620
            ZO     = R1(4)                                                       DAT01630
      GO TO 70                                                                   DAT01640
  140       XT     = R1(2)                                                       DAT01650
            YT     = R1(3)                                                       DAT01660
            ZT     = R1(4)                                                       DAT01670
C*****                                                                           DAT01680
C*****BURN CARD DATA DUMMYED BY PROGRAM GRNAD*****                               DAT01690
C*****                                                                           DAT01700
  155       DO 156 J=1,7                                                         DAT01710
  156       DUMY(J)=R1(J+1)                                                      DAT01720
      GO TO 70                                                                   DAT01730
  150       DO 152 J=1,7                                                         DAT01740
  152       EXTC(J)=R1(J+1)                                                      DAT01750
      GO TO 70                                                                   DAT01760
  165       DO 166 J=1,7                                                         DAT01770
  166       XMIS(J)=R1(J+1)                                                      DAT01780
   70 CONTINUE                                                                   DAT01790
  175 GO TO 311                                                                  DAT01800
C*****                                                                           DAT01810
C*****CAUTION FOR TOO MANY CARDS                                                 DAT01820
C*****                                                                           DAT01830
  310 WRITE(IOOUT,320)                                                           DAT01840
      IFLAG=3                                                                    DAT01850
C*****                                                                           DAT01860
C*****DEFAULT NON USER DEFINED INPUT*****                                        DAT01870
C*****                                                                           DAT01880
  311 IF(ITYPE.EQ.1)GO TO 3                                                      DAT01890
      ITYPE=1                                                                    DAT01900
      WRITE(IOOUT,171)                                                           DAT01910
  171 FORMAT(1H ,21X,17H*****CAUTION*****,/,1H ,21X,54HWRONG SMOKE TYPE          DAT01920
     *FOR PROGRAM GRNAD--DEFAULTED TO WP/RP)                                     DAT01930
    3 IF(EFF.EQ.0.0)EFF=62.0                                                     DAT01940
      IF(YF.EQ.0.0)YF=3.14+0.032*RHA                                             DAT01950
      IF(EXTC(1).GT.0.0)GO TO 1                                                  DAT01960
      CALL EXTIN(EX)                                                             DAT01970
      DO 7 I=1,7                                                                 DAT01980
    7 EXTC(I)=EX(I)                                                              DAT01990
    1 CONTINUE                                                                   DAT02000
  320 FORMAT(21X,17H*****CAUTION*****,/,                                         DAT02010
     *21X,56HMORE THAN 10 DATA CARDS ENTERED--REMAINING CARDS IGNORED)           DAT02020
      GO TO 9999                                                                 DAT02030
  998 IFLAG=4                                                                    DAT02040
 9999 IF(IGEOSW.NE.1) GO TO 555                                                  DAT02050
      DISKTM=1000.
C***  CONVERT KM TO M.
      XT=PTS(1)*DISKTM                                                           DAT02060
      YT=PTS(2)*DISKTM                                                           DAT02070
      ZT=PTS(3)*DISKTM                                                           DAT02080
```

```
      XO=PTS(4)*DISKTM                                                  DAT02090
      YO=PTS(5)*DISKTM                                                  DAT02100
      ZO=PTS(6)*DISKTM                                                  DAT02110
  555 RETURN                                                            DAT02120
      END                                                               DAT02130
```

```
      SUBROUTINE LT4M(H1,H2,ANGLE,ITYPE,IXY,TRAN,RADA,RADG,IEMISS,LEN,   L4M00010
     +MODEL,VIS,V11,V22,T1,ICLMAT,IERR,NR,IHAZE,MULDV)                   L4M00020
      LOGICAL ISPOT,LOREAD,N16                                           L4M00030
      COMMON /CONST/PI,PI2,CA,TWOPI,TORRMB,CDEGK                         L4M00040
      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL      L4M00050
      COMMON /MO2/ WO(34),RO,TBOUND,JP,IM,ML,IP,JSTOR                    L4M00060
      COMMON /MO5/ C1(501),C2(258),C3(86),C4(33),C5(6),C5DUM(9),C8(82),  L4M00070
     1       C11(4),C12(15),C14(21),C15(6)                               L4M00080
      COMMON /MO7/TR(67),FW(67),FO(67)                                   L4M00090
      COMMON /MO8/SUM4,SUM5,SUM8,SUM11,SUM6                              L4M00100
      COMMON /MO9/RADMAX,RADMIN,VRMAX,VRMIN                              L4M00110
      COMMON /MO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),      L4M00120
     1 01(9),02(9),PPMSO2,PPMNH3,PPMNO2                                  L4M00130
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUL4M00140
      COMMON /LOWEX/WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX,     L4M00150
     1 CLDHGT,NCLD                                                       L4M00160
      COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1                     L4M00170
      COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL                        L4M00180
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),      L4M00190
     1 BE(16),SINGWV,PF(65),LMAX                                         L4M00200
      COMMON /SPOTLO/ISPOT,LOREAD,N16                                    L4M00210
      DIMENSION TRAN(16)                                                 L4M00220
      DIMENSION RADA(16),RADG(16)                                        L4M00230
C     PLANCK RADIANCE FUNCTION ;;;;;;;;                                  L4M00240
      FF(T,V)=1.190956E-16*(V**5)/(EXP(1.43879*V/T)-1.)                  L4M00250
C       WATT CM-2 ST-1 MICRON-1                                          L4M00260
C*********************************************************************L4M00280
C                                                                        L4M00290
C     PROGRAM MODIFIED LOWTRAN CALCULATES THE TRANSMITTANCE              L4M00300
C     OF THE ATMOSPHERE FROM 830 TO 1250, 2010 TO 3330, AND              L4M00310
C     5010 TO 39990 CM-1 (0.25 TO 2.0,3.0 TO 5.0,AND 8.0 TO 12.0         L4M00320
C     MICRONS) AT 20 CM-1 SPECTRAL INTERVALS ON A LINEAR WAVENUMBER      L4M00330
C     SCALE.                                                             L4M00340
C     REFRACTION AND EARTH CURVATURE EFFECTS ARE EXCLUDED. ATMOSPHERE    L4M00350
C     IS LAYERED IN ONE KM INTERVALS BETWEEN 0 AND 25 KM, 5 KM INTER-    L4M00360
C     VALS FROM 25 TO 50 KM, A 20 KM LAYER FROM 50 TO 70 KM, A 30 KM     L4M00370
C     LAYER FROM 70 TO 100 KM, AND ONE FROM 100 KM TO INFINITY.          L4M00380
C*********************************************************************L4M00390
C     PROGRAM ACTIVATED BY SUBMISSION OF CARD SEQUENCE AS FOLLOWS        L4M00400
C                                                                        L4M00410
C  CARD 1  MODEL,IHAZE,ITYPE,LEN,JP,NPLT,IM,ML,IEMISS,                  L4M00420
C          RO,TBOUND,BETAEX     FORMAT ( 9I3,3F10.3)                    L4M00430
C          MODEL    =0,METEOROLOGICAL DATA SPECIFIED                    L4M00440
C                   =1,TROPICAL MODEL ATMOSPHERE                        L4M00450
C                   =2,MIDLATITUDE SUMMER                               L4M00460
C                   =3,MIDLATITUDE WINTER                               L4M00470
C                   =4,SUBARCTIC SUMMER                                 L4M00480
C                   =5,SUBARCTIC WINTER                                 L4M00490
C                   =6,1962 US STANDARD                                 L4M00500
C                   =7,NEW MODEL ATMOSPHERE                             L4M00510
C                   =8,ISRAELI STANDARD ATMOSPHERE (YEAR, DAYTIME)      L4M00520
C                   =9,ISRAELI STANDARD ATMOSPHERE (YEAR, NIGHTTIME)    L4M00530
C  ** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK **L4M00540
C     FOR SLANT PATHS IHAZE = 1,2, OR 3 ARE THE ONLY ALLOWED VALUES.     L4M00550
C            IHAZE    =0,NO AEROSOL ATTENUATION                         L4M00560
C                     =1, MARITIME POLAR                                 L4M00570
C                     =2, MARITIME ARCTIC                                L4M00580
C                     =3, CONTINENTAL POLAR                              L4M00590
C                     =4, RAIN                                           L4M00600
C                     =5, SNOW                                           L4M00610
C                     =7,  USER SUPPLIED EXTINCTION COEFFICIENT          L4M00620
C                       (READ ON ATM CARD - SEE CARD 3 BELOW)            L4M00630
C          ITYPE    =1,HORIZONTAL (CONSTANT PRESSURE) PATH              L4M00640
C                   =2,VERTICAL OR SLANT PATH BETWEEN 2 ALTITUDES       L4M00650
C                   =3,VERTICAL OR SLANT PATH TO SPACE                  L4M00660
C          LEN      =0,NORMAL OPERATION                                 L4M00670
C                   =1,DOWNWARD LONG PATH                               L4M00680
C          JP       =0,NORMAL OPERATION                                 L4M00690
C                   =1,SUPPRESS PRINT OF HORIZ AND VERTICAL PROFILES    L4M00700
C        NPLT       =0, NORMAL OPERATION                                L4M00710
```

```
C                  =1, IN TRANSMISSION MODE WRITE, WAVELENGTH (UM),      L4M00720
C                      H2O, CO2+, OZONE, N2 C, H2O C, MOL SCAT,          L4M00730
C                      NITRIC, SO2, HNO3, NO2.  IN EMISSION MODE         L4M00740
C                      WRITE WAVELENGTH (UM) AND RADIANCE PER MICRON,    L4M00750
C          RESULTS WILL BE WRITTEN ON NPLOTU (SEE COMMON BLOCK IOUNIT)   L4M00760
C          IM       =1,RADIOSONDE DATA TO BE READ INITIALLY              L4M00770
C                   =0,NORMAL OPERATION OR WHEN SUBSEQUENT CALCULATIONS  L4M00780
C                  ARE TO BE RUN WITH MODEL = 7                          L4M00790
C          ML       =NUMBER OF LEVELS TO BE READ IN FOR MODEL = 7        L4M00800
C  ***IM AND ML ONLY USED WHEN MODEL = 7  AND THEN ONLY ON               L4M00810
C     FIRST CALCULATIONS WHEN DATA READ IN                               L4M00820
C          IEMISS   DETERMINES MODE OF EXECUTION OF PROGRAM              L4M00830
C                   =0,TRANSMITTANCE MODE                                L4M00840
C                   =1,RADIANCE MODE                                     L4M00850
C          RO       RADIUS OF THE EARTH (KM) AT LOCATION OF CALCULATION  L4M00860
C   ***DEFAULT WILL BE MIDLATITUDE VALUE OF 6371.23 KM WHEN              L4M00870
C                    MODEL = 0 OR = 7   OTHERWISE DEFAULT IS EARTH RADIUSL4M00880
C                    FOR STANDARD MODEL ATMOSPHERE SPECIFIED BY MODEL    L4M00890
C          TBOUND   TEMPERATURE OF EARTH (DEGREES K) AT LOCATION OF CALCUL4M00900
C   ***USED ONLY IN RADIANCE MODE FOR SLANT PATHS WHICH INTERSECT EARTH  L4M00910
C   ***DEFAULT IS TEMPERATURE OF FIRST LAYER   BOUNDARY TEMPERATURE      L4M00920
C        BETAEX   USER SUPPLIED EXTINCTION COEFFICIENT, INPUT ONLY       L4M00930
C                 WHEN IHAZE=7                                          L4M00940
C  CARD 2  H1,H2,ANGLE,RANGE,BETA,VIS,CLDHGT  FORMAT (7F10.3)           L4M00950
C          H1       INITIAL ALTITUDE (KM)                              L4M00960
C          H2       FINAL ALTITUDE (KM)                                L4M00970
C          ANGLE    INITIAL ZENITH ANGLE (DEG)                         L4M00980
C          RANGE    PATH LENGTH (KM)                                   L4M00990
C          BETA     EARTH CENTER ANGLE SUBTENDED BY H1 AND H2 (DEG)    L4M01000
C          VIS      SEA LEVEL VISUAL RANGE (KM)                        L4M01010
C          CLDHGT   HEIGHT OF BOTTOM OF CLOUD LAYER (KM), WHEN IHAZE NE 0L4M01020
C   ***VIS NOT REQUIRED ON THIS CARD IF ICLMAT (EOMAIN) =1 OR            L4M01030
C   ***THIS IS FIRST LOOP THROUGH LT4                                    L4M01040
C   ***SEE MANUAL FOR MORE DETAIL                                        L4M01050
C  CARD 2A  V1,V2,MULDV     FORMAT (2F10.3,I2)                         L4M01060
C          V1       INITIAL FREQUENCY (CM**-1)                         L4M01070
C          V2       FINAL FREQUENCY (CM**-1)                           L4M01080
C          MULDV    MULTIPLIER FOR FREQUENCY INCREMENT, WHERE THE       L4M01090
C                   INCREMENT IS A MULTIPLE OF 20 (CM**-1).            L4M01100
C  OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY NR=1 IN EOMAIN)          L4M01110
C     CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2).     L4M01120
C     CARDS 2 - NUMBER OF VALUES: FORMAT (2(E10.4,1X))                  L4M01130
C       ONE VALUE OF WAVELENGTH (UM) AND RESPONSE FUNCTON PER CARD      L4M01140
C  CARD 3  IXY      FORMAT (I3)                                        L4M01150
C          IXY      =0,EXIT LOWTRAN MODULE                             L4M01160
C                   =1,SELECT NEW WAVE FREQUENCY RANGE (CARD 2A)       L4M01170
C                   =2,SELECT NEW DATA SEQUENCE (CARDS 1,2,2A,3)       L4M01180
C                   =3,SELECT NEW CARD 2 AND CARD 3                    L4M01190
C                   =4,SELECT NEW CARD 1 AND CARD 3                    L4M01200
C                                                                      L4M01210
C  ***FOR NON-STANDARD CONDITIONS SEE MANUAL                           L4M01220
C*********************************************************************L4M01230
      V1=V11                                                           L4M01240
      V2=V22                                                           L4M01250
      KMAX=16                                                          L4M01260
      ISPOT1=0                                                         L4M01270
      RESPFN=0.                                                        L4M01280
      SUMRPF=0.                                                        L4M01290
      SUMINT=0.                                                        L4M01300
      IF (ISPOT) NPLT=0                                                L4M01310
  200 CONTINUE                                                         L4M01320
      IF (.NOT.LOREAD) GO TO 400                                       L4M01330
      LOREAD=.FALSE.                                                   L4M01340
      READ (LOUNIT,3300) IATM,NL                                       L4M01350
      NL4=4*NL                                                         L4M01360
      DO 299 I=1,NL4                                                   L4M01370
  299 READ (LOUNIT,3500) DUMMY                                         L4M01380
      READ (LOUNIT,3510) PPMSO2,PPMNH3,PPMNO2                          L4M01390
      READ (LOUNIT,3700)(TR(I),FW(I),F0(I),I=1,67)                     L4M01400
      READ (LOUNIT,3800)(C1(I),I=1,501)                                L4M01410
```

```
      READ (LOUNIT,3800)(C2(I),I=1,258)                                  L4M01420
      READ (LOUNIT,3800)(C3(I),I=1,86)                                   L4M01430
      READ (LOUNIT,3900)(C4(I),I=1,33)                                   L4M01440
      READ(LOUNIT,3810)(C5(I),I=1,6)                                     L4M01450
      READ (LOUNIT,3900)(C8(I),I=1,82)                                   L4M01460
      READ (LOUNIT,4000)(C11(I),I=1,4)                                   L4M01470
      READ (LOUNIT,4010)(C12(I),I=1,15)                                  L4M01480
      READ (LOUNIT,4020)(C14(I),I=1,21)                                  L4M01490
      READ (LOUNIT,4020)(C15(I),I=1,6)                                   L4M01500
      READ (LOUNIT,4010)(FS(I),S1(I),S2(I),I=1,9)                        L4M01510
      READ (LOUNIT,4010)(FNH3(I),FH1(I),FH2(I),I=1,9)                    L4M01520
      READ (LOUNIT,4010)(FNO2(I),O1(I),O2(I),I=1,9)                      L4M01530
      REWIND LOUNIT                                                      L4M01540
      IF (ISPOT) GO TO 710                                               L4M01550
400   CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN(1))          L4M01560
      IF (IERR.EQ.1) RETURN                                              L4M01570
      JP=0                                                               L4M01580
      JSTOR=0                                                            L4M01590
      IP=0                                                               L4M01600
      IF (.NOT.ISPOT) GO TO 700                                          L4M01610
      BETA=0.                                                            L4M01620
      RANGE=0.                                                           L4M01630
      RO=0.                                                              L4M01640
      IF (ITYPE.EQ.1) RANGE=ANGLE                                        L4M01650
      IF (IXY.EQ.0) GO TO 700                                            L4M01660
401   GO TO (500,700,600,680),IXY                                        L4M01670
500   AVW=1.E+04/V1                                                      L4M01680
      ALAM=1.E+04/V2                                                     L4M01690
      SUMA=0.                                                            L4M01700
      GO TO 1100                                                         L4M01710
  600 IF (MODEL.EQ.0) GO TO 800                                          L4M01720
      ISPOT1=1                                                           L4M01730
      GO TO 1000                                                         L4M01740
  700 CONTINUE                                                           L4M01750
      RE=6371.23                                                         L4M01760
680   IF (.NOT.ISPOT) READ (IOIN,3300) MODEL,IHAZE,ITYPE,                L4M01770
     1     LEN,JP,NPLT,IM,ML,IEMISS,RO,TBOUND                           L4M01780
C*****IEMISS=0=TRANSMISSION MODE / IEMISS=1=EMISSION MODE                L4M01790
      IF ((IEMISS.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,4100)             L4M01800
      IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,4200)             L4M01810
      IF (ISPOT) GO TO 800                                               L4M01820
      IF(MODEL.EQ.0.OR.MODEL.EQ.7) GO TO 210                             L4M01830
  710 READ(LOUNIT,3300)IATM,NL                                           L4M01840
      MSKIP=(MODEL-1)/2                                                  L4M01850
      IF(MSKIP.EQ.0) GO TO 220                                           L4M01860
      IF (MSKIP.EQ.4) MSKIP=3                                            L4M01870
      DO 230 J=1,MSKIP                                                   L4M01880
      DO 230 I=1,NL                                                      L4M01890
  230 READ(LOUNIT,3500) DUMMY                                            L4M01900
  220 CONTINUE                                                           L4M01910
C   ISRAELI STD ATM READS                                                L4M01920
      IF (MODEL.EQ.8) GO TO 270                                          L4M01930
      IF (MODEL.EQ.9) GO TO 250                                          L4M01940
      IF (2*(MODEL/2).EQ.MODEL) GO TO 250                                L4M01950
  270 DO 240 I=1,NL                                                      L4M01960
  240 READ(LOUNIT,3500)Z(I),P(I),T(I),WA(I),WH(I),WO(I)                  L4M01970
      GO TO 210                                                          L4M01980
  250 DO 260 I=1,NL                                                      L4M01990
  260 READ(LOUNIT,3550)Z(I),P(I),T(I),WA(I),WH(I),WO(I)                  L4M02000
  210 REWIND LOUNIT                                                      L4M02010
      IF (ISPOT) RETURN                                                  L4M02020
  800 M=MODEL                                                            L4M02030
  900 IF (RO.GT.0) RE=RO                                                 L4M02040
      LENTOR=LEN                                                         L4M02050
1000  CALL ABSORB(IXY,IERR,W,V1,V2,DV,SUMA,MULDV,ANGLE,LEN,ITYPE,H1,H2,  L4M02060
     1     MODEL,ISPOT1,RANGE,BETA,VIS,ICLMAT,IV1,IV2,IDV)               L4M02070
      IF (IERR.EQ.1) TRAN(1)=1.                                          L4M02080
      IF (IERR.EQ.1) RETURN                                              L4M02090
 1100 CONTINUE                                                           L4M02100
      IF (.NOT.ISPOT) WRITE (IOOUT,4300)                                 L4M02110
```

```
      IF (.NOT.ISPOT) WRITE(IOOUT,4400) (W(I),I=1,6),W(8),W(10)          L4M02120
      IF (.NOT.ISPOT) WRITE(IOOUT,4401)                                  L4M02130
      IF (.NOT.ISPOT) WRITE (IOOUT,4500) W(11),W(12),W(13),W(14),        L4M02140
     * W(15),W(16)                                                       L4M02150
1200  CONTINUE                                                           L4M02160
C     NCLD WILL BE THE INDEX OF THE LAYER ABOVE CLDHGT                   L4M02170
      DO 5 ICLD=2,6                                                      L4M02180
      NCLD=ICLD                                                          L4M02190
      IF (CLDHGT.LT.Z(ICLD)) GO TO 6                                     L4M02200
5     CONTINUE                                                           L4M02210
6     CONTINUE                                                           L4M02220
      IF (CLDHGT.GT.Z(6)) WRITE (IOOUT,7)                                L4M02230
7     FORMAT (1H ,25H**** WARNING FROM LOWTRAN,/,1X,14HCLOUD BASE IS ,   L4M02240
     1  23HLIMITED TO 4 KM MAXIMUM/)                                     L4M02250
      I=1                                                                L4M02260
      L=1                                                                L4M02270
      IV=IV1                                                             L4M02280
      ICOUNT=0                                                           L4M02290
      IF (N16) KWAVE=0                                                   L4M02300
      IF (IEMISS.EQ.0) GO TO 1300                                        L4M02310
      RADSUM=0.0                                                         L4M02320
      FACTOR=0.5                                                         L4M02330
      CALL LTPATH(WLAY,WPATH,TBBY,ANGLE,LEN,ITYPE,H1,H2,MODEL)           L4M02340
      IF (.NOT.ISPOT) WRITE (IOOUT,4600)                                 L4M02350
      IF (.NOT.ISPOT) WRITE (IOOUT,4700)                                 L4M02360
C**** BEGINNING OF TRANSMITTANCE CALCULATIONS                            L4M02370
1300  CONTINUE                                                           L4M02380
      IF (N16) KWAVE=KWAVE+1                                             L4M02390
      SUMV=0.                                                            L4M02400
      TLOLD=1.                                                           L4M02410
      TSOLD=1.                                                           L4M02420
      TX7=1.                                                             L4M02430
      TX10=1.                                                            L4M02440
      IKLO=1                                                             L4M02450
      TOLD=1                                                             L4M02460
      IF (IEMISS.EQ.0) IKMAX=IKLO                                        L4M02470
C     ONLY ONE LOOP FOR TRANSMISSION: LOOP OVER LAYERS FOR EMISSION      L4M02480
      DO 2300 IK=IKLO,IKMAX                                              L4M02490
      IF (IEMISS.EQ.0) GO TO 1500                                        L4M02500
C     TRANSFER CUMULATIVE ABSORBER AMOUNTS FOR TH IK TH LEVEL AND        L4M02510
C     THE K TH ABSORBER - EMISSION ONLY.                                 L4M02520
      DO 1400 K=1,KMAX                                                   L4M02530
      W(K)=WPATH(IK,K)                                                   L4M02540
 1400 CONTINUE                                                           L4M02550
 1500 IJ=IK                                                              L4M02560
      IF (ICOUNT.EQ.0) GO TO 1600                                        L4M02570
      IF (ICOUNT.EQ.50) GO TO 1600                                       L4M02580
      GO TO 1700                                                         L4M02590
 1600 ICOUNT=0                                                           L4M02600
      IF ((IEMISS.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,4800)             L4M02610
 1700 DO 1800 K=1,KMAX                                                   L4M02620
      TX(K)=1.0                                                          L4M02630
 1800 CONTINUE                                                           L4M02640
      ICOUNT=ICOUNT+1                                                    L4M02650
      V=FLOAT(IV)                                                        L4M02660
      ALAM=1.E+04/V                                                      L4M02670
      I=(IV-830)/20+1                                                    L4M02680
      SUM4=0.                                                            L4M02690
      SUM5=0.                                                            L4M02700
      SUM6=0.                                                            L4M02710
      SUM8=0.                                                            L4M02720
      SUM11=0.                                                           L4M02730
      CALL FREQSL(I,IV,W,TX)                                             L4M02740
      TX(9)=SUM4+SUM5+SUM8+SUM11+SUM6                                    L4M02750
      IF (TX(9).EQ.0.0) GO TO 2000                                       L4M02760
      IF (TX(9).LE.0.1) GO TO 1900                                       L4M02770
      IF (TX(9).GT.20.) GO TO 2100                                       L4M02780
      TX(9)=EXP(-TX(9))                                                  L4M02790
      GO TO 2200                                                         L4M02800
 1900 TX(9)=1.0-TX(9)+0.5*TX(9)*TX(9)                                    L4M02810
```

```
      GO TO 2200                                                        L4M02820
 2000 TX(9)=1.0                                                         L4M02830
      GO TO 2200                                                        L4M02840
 2100 TX(9)=0.                                                          L4M02850
 2200 TX(9)=TX(1)*TX(2)*TX(3)*TX(9)*TX(12)*TX(13)*TX(14)                L4M02860
C     AEROSOL COMPUTATIONS UNTIL LABEL 1                                L4M02870
      IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 1                         L4M02880
C                                                                       L4M02890
C    IF SPOT IS CALLING LT4M WITH ITYPE = 3, DO NOT INCLUDE AEROSOLS    L4M02900
C                                                                       L4M02910
      IF (ISPOT.AND.ITYPE.EQ.3) GO TO 1                                 L4M02920
      EXT55=3.914/VIS                                                   L4M02930
C     UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE            L4M02940
C     PASS HORIZONTAL DIST IF ITYPE=1, SLANT DISTANCE IF ITYPE GT 1.    L4M02950
      IF (ITYPE.EQ.1) RNG=RANGE                                         L4M02960
      IF (ITYPE.EQ.2.AND.(H2.GT.H1).AND.(RANGE.GT..5/COS(ANGLE*CA)))    L4M02970
     1  RNG=.5/COS(ANGLE*CA)                                            L4M02980
      IF (ITYPE.EQ.2.AND.(H2.LT.H1).AND.                                L4M02990
     1  (RANGE.GT..5/COS((180.-ANGLE)*CA)))                             L4M03000
     2       RNG=.5/COS((180.-ANGLE)*CA)                                L4M03010
      IF (ITYPE.EQ.3.AND.(RANGE.GT..5/COS(ANGLE*CA)))                   L4M03020
     1       RNG=.5/COS(ANGLE*CA)                                       L4M03030
      IF (ITYPE.EQ.3.AND.RANGE.LT..0001)                                L4M03040
     1       RNG=.5/COS(ANGLE*CA)                                       L4M03050
      ISLANT=ITYPE-1                                                    L4M03060
C     CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL        L4M03070
      CALL XSCALE(ALAM,88.,EXT55,TX7,IERR,ISLANT,IHAZE,RNG,ANGLE)       L4M03080
      IF (IERR.EQ.1) RETURN                                             L4M03090
C     USER OPTIONS                                                      L4M03100
      IF (IHAZE.EQ.7) TX7=EXP(-BETAEX*RANGE)                            L4M03110
      IF (ISPOT.AND.IHAZE.EQ.8) TX7=EXP(-BE(KWAVE)*RANGE)               L4M03120
 1    CONTINUE                                                          L4M03130
      TX(9)=TX(9)*TX7                                                   L4M03140
      IF (IV.GE.13000) TX(3)=TX(8)                                      L4M03150
      TNEW=TX(9)                                                        L4M03160
      IF (IEMISS.EQ.0) GO TO 2500                                       L4M03170
C     COMPUTER LIMITS                                                   L4M03180
      BBIK=0.0                                                          L4M03190
      IF(ABS(1.43879*V/TBBY(IK)).LT.85.) BBIK=FF(TBBY(IK),V)            L4M03200
C     AEROSOL COMPUTATIONS UNTIL LABEL 2                                L4M03210
      IF (IHAZE.EQ.0.OR.IK.NE.(NCLD-1)) GO TO 2                         L4M03220
C     FIND AEROSOL ABSORPTION IN DIFFERENT WAVELENGTH BANDS FROM EXTN   L4M03230
      IF (ALAM.LT.2.) TX10=1.                                           L4M03240
      IF (ALAM.GE.3..AND.ALAM.LE.5.) TX10=TX7**.2                       L4M03250
      IF (ALAM.GE.8..AND.ALAM.LE.12.05) TX10=TX7**.45                   L4M03260
 2    CONTINUE                                                          L4M03270
      TLNEW=TX(9)*TX10/(TX(6)*TX7)                                      L4M03280
      TSNEW=TX7*TX(6)/TX10                                              L4M03290
      DTAU=ABS(TLOLD-TLNEW)                                             L4M03300
      IF (DTAU.LT.1.0E-5.AND.TLNEW.LT.1.0E-5) GO TO 2400                L4M03310
      SUMV=SUMV+(TOLD-TNEW)*BBIK                                        L4M03320
      TLOLD=TLNEW                                                       L4M03330
      TSOLD=TSNEW                                                       L4M03340
      TOLD=TNEW                                                         L4M03350
 2300 CONTINUE                                                          L4M03360
 2400 CONTINUE                                                          L4M03370
      TAUG=0.                                                           L4M03380
      IF (HMIN.LE.0.0.AND.IL.EQ.1) TAUG=TX(9)                           L4M03390
      T1=T(1)                                                           L4M03400
      IF (TBOUND.GT.0.0) T1=TBOUND                                      L4M03410
C     COMPUTER LIMITS                                                   L4M03420
      BBG=0.0                                                           L4M03430
      IF(ABS(1.43879*V/T1).LT.85.) BBG=FF(T1,V)*TAUG                    L4M03440
      IF (N16) RADG(KWAVE)=BBG*1.E+04                                   L4M03450
      IF (N16) RADA(KWAVE)=SUMV*1.E+04                                  L4M03460
      IF (HMIN.LE.0) SUMV=SUMV+BBG                                      L4M03470
      SUMVV=SUMV                                                        L4M03480
      IF (IV.GT.IV1) FACTOR=1.0                                         L4M03490
      IF (IV.GE.IV2) FACTOR=0.5                                         L4M03500
      SUMV=(1.0E+04/V**2)*SUMV                                          L4M03510
```

```
      RADSUM=RADSUM+DV*FACTOR*SUMV                                          L4M03520
      IF (.NOT.ISPOT) WRITE (IOOUT,4900) V,                                 L4M03530
     1    ALAM,SUMV,SUMVV,RADSUM,TX(9),TX7,TX10                             L4M03540
      IF (NPLT.EQ.1) WRITE (NPLOTU,98) ALAM,SUMVV                           L4M03550
98    FORMAT (F7.4,1X,E13.5)                                                L4M03560
      IF (SUMV.GE.RADMAX) VRMAX=V                                           L4M03570
      IF (SUMV.GE.RADMAX) RADMAX=SUMV                                       L4M03580
      IF (SUMV.LE.RADMIN) VRMIN=V                                           L4M03590
      IF (SUMV.LE.RADMIN) RADMIN=SUMV                                       L4M03600
 2500 AB=1.-TX(9)                                                           L4M03610
      IF (IV.EQ.IV1.OR.IV.GE.IV2) AB=0.5*AB                                 L4M03620
      SUMA=SUMA+AB*DV                                                       L4M03630
      IF (IEMISS.EQ.1) GO TO 2600                                           L4M03640
      IF (.NOT.ISPOT) WRITE (IOOUT,5000) IV,                                L4M03650
     1   ALAM,(TX(K),K=1,6),TX(11),TX(12),TX(13),TX(14),                    L4M03660
     2   SUMA,TX(9),TX7                                                     L4M03670
      IF (NPLT.EQ.1) WRITE(NPLOTU,99) ALAM,(TX(K),K=1,6),(TX(J),J=11,14)    L4M03680
99    FORMAT (F7.4,10(1X,F6.4))                                             L4M03690
 2600 CONTINUE                                                              L4M03700
      RESPFN=RESFN(NR,ALAM)                                                 L4M03710
      SUMRPF=SUMRPF+RESPFN                                                  L4M03720
      IF (IV.GT.IV1) SUMINT=SUMINT+.5*(OLDTX9*OLDRFN+TX(9)*RESPFN)*         L4M03730
     +ABS(1./FLOAT(IV)-1./FLOAT(IV-IDV))*1.E+04                             L4M03740
      IF (N16) TRAN(KWAVE)=TX(9)                                            L4M03750
      OLDTX9=TX(9)                                                          L4M03760
      OLDRFN=RESPFN                                                         L4M03770
      IV=IV+IDV                                                             L4M03780
      IF (IV.GE.IV2) GO TO 2700                                             L4M03790
      GO TO 1300                                                            L4M03800
 2700 CONTINUE                                                              L4M03810
      IF (NR.NE.1) SUMRPF=1.                                                L4M03820
      SUMINT=SUMINT/(SUMRPF*1.E+04*ABS(1./FLOAT(IV1)-1./FLOAT(IV2)))        L4M03830
      IF (.NOT.ISPOT) TRAN(1)=SUMINT                                        L4M03840
      IF (.NOT.ISPOT.AND.NR.EQ.1) WRITE (IOOUT,3250) SUMINT                 L4M03850
      IF (.NOT.ISPOT.AND.NR.NE.1) WRITE (IOOUT,3275) SUMINT                 L4M03860
      RESPFN=0.                                                             L4M03870
      SUMRPF=0.                                                             L4M03880
      SUMINT=0.                                                             L4M03890
      IF ((IEMISS.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,5100)                L4M03900
     1     VRMIN,RADMIN,VRMAX,RADMAX                                        L4M03910
      JSTOR=0                                                               L4M03920
      AB=1.0-SUMA/FLOAT(IV-IV1)                                             L4M03930
      IF (ISPOT) RETURN                                                     L4M03940
      WRITE (IOOUT,5200) IV1,IV,SUMA,AB                                     L4M03950
      IF ((IEMISS.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,5300) RADSUM         L4M03960
      IF (.NOT.ISPOT) READ (IOIN,3300) IXY                                  L4M03970
      IF (IXY.EQ.0) GO TO 3100                                              L4M03980
      GO TO (2800,700,2900,680,3100),IXY                                    L4M03990
 2800 CONTINUE                                                              L4M04000
      READ (IOIN,5400) V1,V2,MULDV                                          L4M04010
      CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN(1))             L4M04020
      IF (IERR.EQ.1) RETURN                                                 L4M04030
      AVW=10000./V1                                                         L4M04040
      ALAM=10000./V2                                                        L4M04050
      WRITE (IOOUT,5500) V1,V2,DV,ALAM,AVW                                  L4M04060
      SUMA=0.0                                                              L4M04070
      GO TO 1100                                                            L4M04080
 2900 IF (MODEL.EQ.0) GO TO 800                                             L4M04090
      GO TO 401                                                             L4M04100
 3000 CONTINUE                                                              L4M04110
      READ (IOIN,3300) MODEL,IHAZE,ITYPE,LEN,JP,NPLT,IM,                    L4M04120
     1    ML,IEMISS,RO,TBOUND,BETAEX                                        L4M04130
      IF (IEMISS.EQ.1) WRITE (IOOUT,4100)                                   L4M04140
      IF (IEMISS.EQ.0) WRITE (IOOUT,4200)                                   L4M04150
      LENTOR=LEN                                                            L4M04160
      GO TO 800                                                             L4M04170
3100  RETURN                                                                L4M04180
C                                                                           L4M04190
3250  FORMAT (/,1X,48HWAVELENGTH AND SENSOR INTEGRATED TRANSMISSION =       L4M04200
     +,E10.4)                                                               L4M04210
```

```
3275  FORMAT (/,1X,37HWAVELENGTH INTEGRATED TRANSMISSION = ,E10.4)
 3300 FORMAT ( 9I3,3F10.3)
 3500 FORMAT (F6.1,2(E9.3,F5.1,E9.3,2E7.1))
 3510 FORMAT(3F8.3)
 3550 FORMAT(F6.1,37X,E9.3,F5.1,E9.3,2E7.1)
 3700 FORMAT (4(F6.3,2F7.4))
 3800 FORMAT (15F5.2)
 3810 FORMAT(15F5.3)
 3900 FORMAT (8E9.2)
 4000 FORMAT (12F6.3)
 4010 FORMAT(10F8.4)
 4020 FORMAT(10F8.3)
 4100 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,//,1X,
     1 45HPROGRAM WILL BE EXECUTED IN THE EMISSION MODE)
 4200 FORMAT (1H1,40X,36HLT4M ATMOSPHERIC TRANSMISSION MODULE,//,1X,
     1 49HPROGRAM WILL BE EXECUTED IN THE TRANSMISSION MODE)
 4300 FORMAT (/10X,38H EQUIVALENT SEA LEVEL ABSORBER AMOUNTS
     1//21X,56HWATER VAPOUR     CO2 ETC.        OZONE    NITROGEN (CONT),
     2         42H  H2O (CONT)        MOL  SCAT    OZONE(U-V)/24X,
     3         7HGM CM-2,10X,2HKM,10X,6HATM CM,10X,2HKM,9X,7HGM CM-2,
     4 10X,2HKM,10X,6HATM CM)
 4400 FORMAT(/10X,10H W(1-6,8)=,7(E14.3)/,1X,10X,7H W(10)=,58X,E14.3/)
 4401 FORMAT(//23X,11HNITRIC ACID,8X,3HSO2,11X,3HNH3,11X,3HNO2/)
 4500 FORMAT(/10X,10H W(11-16)=,6(E14.3)/)
 4600 FORMAT (1H1,30X,28HRADIANCE(WATTS/CM2-STER-XXX))
 4700 FORMAT (1H ,10X,37HFR(CM-1)  WVL(MICRON)     PER CM-1      ,
     1         10HPER MICRON,26H     INTEGRAL        TRANS,1X,4(1H-),
     2       11H AERO TRAN ,4(1H-),/,1X,84X,17H EXTN        ABS)
 4800 FORMAT  (1H1,/,1X,2X,15HFREQ WAVELENGTH,2X,3HH2O,3X,4HCO2+,4X,
     1         30HOZONE   N2 C    H2O C    MOL S  ,1X,
     2         22HNITRIC    SO2     HNO3,4X,16HNO2    INTEGRATED,
     3         2X,13HTOTAL AEROSOL/1X,1X,13H CM-1 MICRONS,10(3X,5HTRANS),
     4         2X,24HABSORPTION  TRANS  TRANS)
 4900 FORMAT (1H ,10X,F8.1,F13.6,3E13.5,F13.6,1X,F7.5,3X,F7.5)
 5000 FORMAT (1H ,I6,11F8.4,F11.4,F8.4,1X,F7.5)
 5100 FORMAT (1H0,8H RADMIN ,F12.3,E12.5,/,8H RADMAX ,F12.3,
     1         E12.5)
 5200 FORMAT (1H0,26H INTEGRATED ASORPTION FROM,I5,4H TO ,I5,
     1         7H CM-1 =,F10.2,25H, AVERAGE TRANSMITTANCE =,F6.4)
 5300 FORMAT (1H ,22H INTEGRATED RADIANCE =,E12.5,13H WATT CM -2  ,2HSR)
 5400 FORMAT (2F10.3,I2)
 5500 FORMAT (/10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO ,
     1         4HV2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1   (
     2         ,F6.2,3H - ,F5.2,10H MICRONS ))
      END
```

```
      SUBROUTINE ABSORB(IXY,IERR,W,V1,V2,DV,SUMA,MULDV,ANGLE,LEN,ITYPE,  ABS00010
     1        H1,H2,MODEL,ISPOT1,RANGE,BETA,VIS,ICLMAT,IV1,IV2,IDV)      ABS00020
      COMMON /GEOMET/PTS(15),IGEOSW                                      ABS00030
      COMMON /CLYMAT/TEMP,PRESS,RH1,AH1,DP1,VIS1,CLDAMT,                 ABS00040
     1       CLDHYT,FOGPRB,WNDVEL,WNDDIR,IPASCT                          ABS00050
      COMMON /CONST/PI,PI2,CA,TWOPI,TORRMB,CDEGK                         ABS00060
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUABS00070
      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL      ABS00080
      COMMON /MO2/ WO(34),RO,TBOUND,JP,IM,ML,IP,JSTOR                    ABS00090
      COMMON /MO9/RADMAX,RADMIN,VRMAX,VRMIN                              ABS00100
      COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1                     ABS00110
      COMMON /LOWEX/WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX,     ABS00120
     1 CLDHGT,NCLD                                                       ABS00130
      COMMON /SPOTLO/ISPOT,LOREAD,N16                                    ABS00140
      COMMON /MO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),      ABS00150
     1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2                                  ABS00160
      LOGICAL ISPOT,N16,LOREAD                                           ABS00170
      DIMENSION VH(16),W(16),E(16)                                       ABS00180
C     EH(7,I) REPLACES HSTOR                                             ABS00190
C     EH(9,I) REPLACES HMIX                                              ABS00200
      DATA (EH(9,I),I=1,34) /9*0.,0.1,0.33,0.8,1.2,1.4,1.6,1.8,1.9,      ABS00210
     1 2.0,2.1,2.3,3.0,3.7,4.2,5.2,6.0,3.8,2.6,0.22,6*0.0/              ABS00220
      F(A)=EXP(18.9766-14.9595*A-2.43882*A*A)*A                          ABS00230
      TMPVIS=VIS                                                         ABS00240
      IF (ISPOT1.EQ.1) GO TO 200                                         ABS00250
      IF (MODEL.EQ.0) GO TO 400                                          ABS00260
      IF (IXY.EQ.3) GO TO 100                                            ABS00270
      IF (M.EQ.7.AND.IM.NE.0) GO TO 400                                  ABS00280
      IF (IXY.GT.3) GO TO 1500                                           ABS00290
C     WHEN IXY=0 VIS IS READ IN MAIN                                     ABS00300
  100 IF (.NOT.ISPOT) READ (IOIN,6200) H1,H2,ANGLE,RANGE,               ABS00310
     1     BETA,VIS,CLDHGT                                             ABS00320
      IF(IGEOSW.NE.1) GO TO 111                                          ABS00330
      H1=PTS(3)                                                          ABS00340
      H2=PTS(6)                                                          ABS00350
      RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+                  ABS00360
     +(PTS(3)-PTS(6))**2)                                                ABS00370
  111 CONTINUE                                                           ABS00380
      IF (IXY.EQ.0) VIS=TMPVIS                                           ABS00390
  200 X1=RE+H1                                                           ABS00400
      X2=RE+H2                                                           ABS00410
      IF (ITYPE.EQ.3) GO TO 1000                                         ABS00420
      IF (ITYPE.EQ.1) GO TO 1500                                         ABS00430
      IF (RANGE.EQ.0.) GO TO 1200                                        ABS00440
      IF (.NOT.ISPOT) WRITE (IOOUT,6300) H1,H2,ANGLE,RANGE,              ABS00450
     1     BETA,VIS                                                      ABS00460
      IF (H2.EQ.0.AND.ANGLE.NE.0) GO TO 300                              ABS00470
      ANGLE= ACOS(0.5*((H2-H1)*(1.+X2/X1)/RANGE-RANGE/X1))/CA            ABS00480
      GO TO 1400                                                         ABS00490
  300 X2=SQRT((X1/RANGE+RANGE/X1+2.0*COS(ANGLE*CA))*X1*                  ABS00500
     1       RANGE)                                                      ABS00510
      H2=X2-RE                                                           ABS00520
      GO TO 1400                                                         ABS00530
  400 CONTINUE                                                           ABS00540
      IF (ML.LE.0) ML=1                                                  ABS00550
      DO 900 K=1,ML                                                      ABS00560
C     CLIMATE OPTION - SEE COMMON /CLYMAT/                               ABS00570
      IF (M.EQ.0.AND..NOT.ISPOT) READ (IOIN,6400) H1,P(1),               ABS00580
     1     TMP,DP,RH,WH(K),WO(K),VIS,RANGE                               ABS00590
      IF(IGEOSW.NE.1) GO TO 444                                          ABS00600
      H1=PTS(3)                                                          ABS00610
      H2=PTS(6)                                                          ABS00620
      RANGE=SQRT((PTS(1)-PTS(4))**2+(PTS(2)-PTS(5))**2+                  ABS00630
     +(PTS(3)-PTS(6))**2)                                                ABS00640
  444 CONTINUE                                                           ABS00650
      IF (ICLMAT.NE.1) GO TO 500                                         ABS00660
      TMP=TEMP                                                           ABS00670
      P(1)=PRESS                                                         ABS00680
      DP=DP1                                                             ABS00690
      RH=RH1                                                             ABS00700
```

```
      WH(K)=AH1                                                          ABS00710
      VIS=VIS1                                                           ABS00720
  500 IF (IXY.EQ.0.AND.ICLMAT.NE.1) VIS=TMPVIS                           ABS00730
      IF ((M.GT.0).AND.(.NOT.ISPOT)) READ (IOIN,6400) Z(K),              ABS00740
     1    P(K),TMP,DP,RH,WH(K),WO(K)                                     ABS00750
      IF ((M.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,6500) H1,              ABS00760
     1    P(1),TMP,DP,RH,WH(K),WO(K),VIS,RANGE                           ABS00770
      IF (M.EQ.0) Z(K)=H1                                                ABS00780
      J=IFIX(Z(K)+1.0E-6)+1.                                             ABS00790
      IF (Z(K).GE.25.) J=(Z(K)-25.)/5.0+26.                              ABS00800
      IF (Z(K).GE.50.0) J=(Z(K)-50.)/20.+31.                             ABS00810
      IF (Z(K).GE.70.) J=(Z(K)-70.)/30.+32.                              ABS00820
      IF (J.GT.33) J=33                                                  ABS00830
      FAC=Z(K)-FLOAT(J-1)                                                ABS00840
      IF (J.LT.26) GO TO 600                                             ABS00850
      FAC=(Z(K)-5.0*FLOAT(J-26)-25.)/5.                                  ABS00860
      IF (J.GE.31) FAC=(Z(K)-50.)/20.                                    ABS00870
      IF (J.GE.32) FAC=(Z(K)-70.)/30.                                    ABS00880
      IF (FAC.GT.1.0) FAC=1.0                                            ABS00890
  600 L=J+1                                                              ABS00900
      T(K)=TMP+CDEGK                                                     ABS00910
      TT=CDEGK/T(K)                                                      ABS00920
      IF (RH.LE.0.0) TT=CDEGK/(CDEGK+DP)                                 ABS00930
      IF (WH(K).LE.0.0) WH(K)=F(TT)                                      ABS00940
      IF (RH.GT.0.0) WH(K)=0.01*RH*WH(K)                                 ABS00950
      EH(7,K)=0.0                                                        ABS00960
      IF (EH(9,J).LE.0.) GO TO 700                                       ABS00970
      EH(7,K)=EH(9,J)*(EH(9,L)/EH(9,J))**FAC                             ABS00980
  700 CONTINUE                                                           ABS00990
      IF (MODEL.EQ.0) GO TO 1500                                         ABS01000
      IF ((K.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,6600)                  ABS01010
      IF (.NOT.ISPOT) WRITE (IOOUT,6400) Z(K),P(K),TMP,DP,               ABS01020
     1    RH,WH(K),WO(K)                                                 ABS01030
  900 CONTINUE                                                           ABS01040
      IM=0                                                               ABS01050
      NL=ML                                                              ABS01060
C     NOTE THAT Z(I) MAY NOT CORRESPOND TO THE VALUES GIVEN FOR STANDARDABS01070
C     MODEL ATMOSPHERES                                                  ABS01080
      IF (IXY.GE.3) GO TO 1500                                           ABS01090
      GO TO 100                                                          ABS01100
 1000 IF (RANGE.GT.0.0) GO TO 1100                                       ABS01110
      GO TO 1500                                                         ABS01120
 1100 ITYPE=2                                                            ABS01130
      BETA=_ACOS(0.5*(RANGE*RANGE/(X1*X2)-X2/X1-X1/X2))/CA               ABS01140
 1200 IF (BETA.EQ.0.) GO TO 1300                                         ABS01150
      BET=CA*BETA                                                        ABS01160
      X2=RE+H2                                                           ABS01170
      ANGLE=ATAN(X2*SIN(BET)/(X2*COS(BET)-X1))/CA                        ABS01180
      IF (ANGLE.LT.0.) ANGLE=ANGLE+PI                                    ABS01190
      RANGE=X2*SIN(BET)/SIN(ANGLE*CA)                                    ABS01200
      BET=BETA                                                           ABS01210
      GO TO 1500                                                         ABS01220
 1300 RANGE=(X2/X1)**2-(SIN(ANGLE*CA))**2                                ABS01230
      IF (RANGE.GE.0.0) RANGE=X1*(SQRT(RANGE)-ABS(COS(ANGLE*CA)))        ABS01240
 1400 IF (ANGLE.NE.0..OR.ANGLE.NE.180.) BET=ASIN(RANGE*SIN(ANGLE*CA)/X2)ABS01250
      IF (ANGLE.LT.0.) ANGLE=ANGLE+PI                                    ABS01260
      IF (RANGE.LT.0.0) RANGE=-RANGE                                     ABS01270
      BET=BET/CA                                                         ABS01280
      IF (.NOT.ISPOT) WRITE (IOOUT,6300) H1,H2,ANGLE,RANGE,              ABS01290
     1    BET,VIS                                                        ABS01300
 1500 CONTINUE                                                           ABS01310
      DO 1600 I=1,NL                                                     ABS01320
      DO 1600 J=1,KMAX                                                   ABS01330
 1600 WLAY(I,J)=0.                                                       ABS01340
      SUMA=0.                                                            ABS01350
C     WHEN IXY=0 V1,V2,MULDV ARE READ IN EOSAEL.MAIN                     ABS01360
      IF ((IXY.EQ.1.OR.IXY.EQ.2).AND.(.NOT.ISPOT))                       ABS01370
     1    READ (IOIN,6250) V1,V2,MULDV                                   ABS01380
      IF ((IXY.EQ.1.OR.IXY.EQ.2).AND.(.NOT.ISPOT))                       ABS01390
     + CALL CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TMPVIS)          ABS01400
```

```
      IF (IERR.EQ.1) RETURN                                              ABS01410
      IF ((ITYPE.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,6700)              ABS01420
     1     H1,RANGE                                                      ABS01430
      IF ((ITYPE.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,6800)              ABS01440
     1     H1,H2,ANGLE                                                   ABS01450
      IF ((ITYPE.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,6900)              ABS01460
     1     H1,ANGLE                                                      ABS01470
      IF (MODEL.EQ.0) M=7                                                ABS01480
      IF ((M.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,7200) M                ABS01490
      IF ((M.EQ.2).AND.(.NOT.ISPOT)) WRITE (IOOUT,7300) M                ABS01500
      IF ((M.EQ.3).AND.(.NOT.ISPOT)) WRITE (IOOUT,7400) M                ABS01510
      IF ((M.EQ.4).AND.(.NOT.ISPOT)) WRITE (IOOUT,7500) M                ABS01520
      IF ((M.EQ.5).AND.(.NOT.ISPOT)) WRITE (IOOUT,7600) M                ABS01530
      IF ((M.EQ.6).AND.(.NOT.ISPOT)) WRITE (IOOUT,7700) M                ABS01540
      IF ((M.EQ.8).AND.(.NOT.ISPOT)) WRITE (IOOUT,7800) M                ABS01550
      IF ((M.EQ.9).AND.(.NOT.ISPOT)) WRITE (IOOUT,7900) M                ABS01560
      AVW=10000./V1                                                      ABS01570
      ALAM=10000./V2                                                     ABS01580
      RADMIN=1.0E+38                                                     ABS01590
      RADMAX=0.                                                          ABS01600
      VRMIN=0.                                                           ABS01610
      VRMAX=0.                                                           ABS01620
      IF (.NOT.ISPOT) WRITE (IOOUT,8000) V1,V2,DV,ALAM,AVW               ABS01630
      AVW=0.5E-4*(V1+V2)                                                 ABS01640
      AVW=AVW*AVW                                                        ABS01650
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8100)                 ABS01660
      IF (ITYPE.EQ.1) GO TO 2100                                         ABS01670
      DO 1800 K=1,KMAX                                                   ABS01680
      VH(K)=0.0                                                          ABS01690
 1800 CONTINUE                                                           ABS01700
      BETA=0.0                                                           ABS01710
      SR=0.0                                                             ABS01720
      IP=0                                                               ABS01730
C****  NOW DEFINE CONSTANT PRESSURE PATH QUANTITES  EH(1-8)              ABS01740
      Y=CA*ANGLE                                                         ABS01750
      SPHI=SIN(Y)                                                        ABS01760
      R1=(RE+H1)*SPHI                                                    ABS01770
      IF (H1.GT.Z(NL)) GO TO 1900                                        ABS01780
      GO TO 2100                                                         ABS01790
 1900 X=(RE+Z(NL))/(RE+H1)                                               ABS01800
      IF (SPHI.GT.X) GO TO 2000                                          ABS01810
      H1=Z(NL)                                                           ABS01820
      J1=NL                                                              ABS01830
      SPHI=SPHI/X                                                        ABS01840
      ANGLE=180.0- ASIN(SPHI)/CA                                         ABS01850
      R1=(RE+H1)*SPHI                                                    ABS01860
      GO TO 2100                                                         ABS01870
 2000 HMIN=R1-RE                                                         ABS01880
      IF (.NOT.ISPOT) WRITE (IOOUT,8200) HMIN                            ABS01890
      GO TO 6000                                                         ABS01900
 2100 DO 2400 I=1,NL                                                     ABS01910
      PS=P(I)/1013.0                                                     ABS01920
      TS=CDEGK/T(I)                                                      ABS01930
      X=PS*TS                                                            ABS01940
C--- COMPUTE MASS DENSITY (G M-3) FROM IDEAL GAS LAW ---                 ABS01950
C--- 1292.02 = DENSITY OF STANDARD COMPOSITION AIR AT STP ---            ABS01960
      WA(I) = 1292.02*X                                                  ABS01970
      PT=PS*SQRT(TS)                                                     ABS01980
      D=0.1*WH(I)                                                        ABS01990
      EH(1,I)=D*PT**0.9                                                  ABS02000
      EH(2,I)=X*PT**0.75                                                 ABS02010
      EH(4,I)=0.8*PT*X                                                   ABS02020
      PPW=4.56E-5*D*CDEGK/TS                                             ABS02030
      TS1=(296.0/CDEGK)*TS                                               ABS02040
      EH(5,I)=D*PPW*EXP(6.08*(TS1-1.0))+0.002*D*(PS-PPW)                 ABS02050
      EH(10,I)=D*(PPW+0.12*(PS-PPW))*EXP(4.56*(TS1-1.0))                 ABS02060
      EH(6,I)=X                                                          ABS02070
      EH(8,I)=46.6667*WO(I)                                              ABS02080
      EH(3,I)=EH(8,I)*PT**0.4                                            ABS02090
C EH(11,I)=HNO3 ABSORBER AMOUNT (ATM-CM)/KM                              ABS02100
```

```
      EH(11,I)=PS*TS*EH(9,I)*1.0E-04                                    ABS02110
      IF (MODEL.EQ.0.OR.MODEL.EQ.7) EH(11,I)=PS*TS*EH(7,I)*1.0E-04      ABS02120
C*****EH(12,I) = SO2   ABSORBER AMOUNT (ATM-CM)/KM                      ABS02130
      EH(12,I)=0.772E-04*PPMSO2*WA(I)*PS**0.07122*TS**0.06159           ABS02140
      EH(13,I)=0.772E-04*PPMNH3*WA(I)*PS**0.52125*TS**(-0.60438)        ABS02150
      EH(14,I)=0.772E-04*PPMNO2*WA(I)*PS**0.18066*TS**0.20911           ABS02160
C***K=15 FOR ASL 3.3 - 4.3_ MICRON H2O CONTINUUM                        ABS02170
      EH(15,I)=PPW*D                                                    ABS02180
C***K=16 FOR 4.6 - 4.8 MICRON H2O CONTINUUM                             ABS02190
      EH(16,I)=PPW*(PPW+3.0E-03*(PS-PPW))                               ABS02200
      IF (I.EQ.NL) GO TO 2300                                           ABS02210
      IF (MODEL.EQ.0.AND.I.GE.1) GO TO 3600                             ABS02220
      T2=T(I+1)                                                         ABS02230
      W2=WH(I+1)                                                        ABS02240
      PPW=4.56E-6*W2*T2                                                 ABS02250
 2300 IF (H1.GE.Z(I)) J1=I                                              ABS02260
      IF ((JP.EQ.0).AND.(.NOT.ISPOT))                                   ABS02270
     1WRITE (IOOUT,8300)I,Z(I),(EH(K,I),K=1,6),EH(8,I),(EH(K,I),K=10,14)ABS02280
 2400 CONTINUE                                                          ABS02290
      X1=H1                                                             ABS02300
      CALL POINT (H1,N,NP1,TX)                                          ABS02310
      J1=N                                                              ABS02320
      DO 2500 K=1,KMAX                                                  ABS02330
 2500 E(K)=TX(K)                                                        ABS02340
      JEXTRA=0                                                          ABS02350
      JMIN=0                                                            ABS02360
C**ITYPE=1 MEANS HORIZONTAL PATH ***************************************ABS02370
      IF (ITYPE.EQ.1) GO TO 3600                                        ABS02380
      IF (ITYPE.EQ.3) H2=Z(NL)                                          ABS02390
C** ANGLE GREATER THAN 90 DEGREES MEANS DOWNWARD TRAJECTORY ************ABS02400
      IF (ANGLE.GT.90.0) GO TO 3800                                     ABS02410
C** IF THE PATH IS NOT HORIZONTAL OR DOWNWARD THEN IT IS UPWARD TRAJECTOABS02420
 2600 IF (ANGLE.GT.90.0.AND.NP1.GT.0) J1=J1+1                           ABS02430
      J2=NL                                                             ABS02440
      IF (ITYPE.EQ.3) GO TO 2700                                        ABS02450
      CALL POINT (H2,N,NP,TX)                                           ABS02460
      J2=N                                                              ABS02470
      IF (NP.GT.0) J2=J2-1                                              ABS02480
 2700 DO 2800 K=1,KMAX                                                  ABS02490
      IF (K.EQ.9.OR.K.EQ.7) GO TO 2800                                  ABS02500
      EH(K,J1)=E(K)                                                     ABS02510
      IF (ITYPE.EQ.3) GO TO 2800                                        ABS02520
      EH(K,J2+1)=TX(K)                                                  ABS02530
2800  CONTINUE                                                          ABS02540
C****  NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)                      ABS02550
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8400)                ABS02560
      DO 2900 K=1,KMAX                                                  ABS02570
 2900 W(K)=0.                                                           ABS02580
      DO 3500 I=J1,J2                                                   ABS02590
      X1=Z(I)                                                           ABS02600
      IF (I.LT.NL) X2=Z(I+1)                                            ABS02610
      IF (I.EQ.NL) X2=Z(I)                                              ABS02620
      IF (I.EQ.J1) X1=H1                                                ABS02630
      IF (I.EQ.J2) X2=H2                                                ABS02640
      DZ=X2-X1                                                          ABS02650
      IF (I.EQ.NL) DZ=Z(I)-Z(I-1)                                       ABS02660
      DS=DZ                                                             ABS02670
C*****  UPWARD TRAJECTORY                                              ABS02680
      RX=(RE+X1)/(RE+X2)                                                ABS02690
      THETA= ASIN(SPHI)/CA                                              ABS02700
      PHI= ASIN(SPHI*RX)/CA                                             ABS02710
      BET=THETA-PHI                                                     ABS02720
      SALP=RX*SPHI                                                      ABS02730
      IF (SPHI.GT.1.E-10) DS=(RE+X2)*SIN(BET*CA)/SPHI                   ABS02740
      BETA=BETA+BET                                                     ABS02750
      PSI=BETA+PHI-ANGLE                                                ABS02760
      PHI=180.-PHI                                                      ABS02770
      SR=SR+DS                                                          ABS02780
      JEXTRA=0                                                          ABS02790
      DO 3400 K=1,KMAX                                                  ABS02800
```

```
      IF (K.EQ.7.OR.K.EQ.9) GO TO 3100                                  ABS02810
      EV=DS*EH(K,I)                                                     ABS02820
      IF (I.EQ.NL) GO TO 3000                                           ABS02830
      IF (EH(K,I).EQ.0.0.OR.EH(K,I+1).EQ.0.0) GO TO 3100                ABS02840
      IF (EH(K,I).EQ.EH(K,I+1)) GO TO 3200                              ABS02850
      EV=DS*(EH(K,I)-EH(K,I+1))/ALOG(EH(K,I)/EH(K,I+1))                 ABS02860
      GO TO 3200                                                        ABS02870
 3000 IF (EH(K,I).EQ.0.0) GO TO 3100                                    ABS02880
      IF (EH(K,I-1).EQ.0.0) GO TO 3100                                  ABS02890
      IF (EH(K,I).EQ.EH(K,I-1)) GO TO 3200                              ABS02900
      EV=EV/ALOG(EH(K,I-1)/EH(K,I))                                     ABS02910
      GO TO 3200                                                        ABS02920
 3100 EV=0.                                                             ABS02930
 3200 VH(K)=VH(K)+EV                                                    ABS02940
      IF (I.EQ.JSTOR) GO TO 3300                                        ABS02950
      WLAY(I,K)=EV+W(K)                                                 ABS02960
      W(K)=0.                                                           ABS02970
      GO TO 3400                                                        ABS02980
 3300 W(K)=EV                                                           ABS02990
      IF (J1.NE.J2) GO TO 3400                                          ABS03000
      WLAY(J2+1,K)=W(K)                                                 ABS03010
      W(K)=0.                                                           ABS03020
      JEXTRA=1                                                          ABS03030
 3400 CONTINUE                                                          ABS03040
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8500) I,             ABS03050
     1    X1,(VH(L),L=1,6),VH(8),PSI,PHI,BETA,THETA,SR                  ABS03060
      IF (I.GE.NL) GO TO 3500                                           ABS03070
      SPHI=SPHI*RX                                                      ABS03080
      IF (SALP.GE.1.) SPHI=SALP                                         ABS03090
 3500 CONTINUE                                                          ABS03100
      GO TO 5800                                                        ABS03110
C**** HORIZONTAL PATH                                                   ABS03120
 3600 DO 3700 K=1,KMAX                                                  ABS03130
      IF (K.EQ.7.OR.K.EQ.9) GO TO 3700                                  ABS03140
      W(K)=RANGE*EH(K,1)                                                ABS03150
      IF (MODEL.GT.0) W(K)=RANGE*TX(K)                                  ABS03160
      VH(K)=W(K)                                                        ABS03170
 3700 CONTINUE                                                          ABS03180
      GO TO 6100                                                        ABS03190
 3800 CONTINUE                                                          ABS03200
C****   DOWNWARD TRAJECTORY                                             ABS03210
      K2=0                                                              ABS03220
      IF (NP1.EQ.1) J1=J1-1                                             ABS03230
      IF (J1.LE.0) J1=1                                                 ABS03240
      J2=J1+1                                                           ABS03250
      J=J1+1                                                            ABS03260
      IF (H2.GT.Z(J1+1).OR.H1.EQ.H2) GO TO 4000                         ABS03270
      IF (NP1.EQ.1.AND.H2.GE.Z(J1+1)) GO TO 4000                        ABS03280
      CALL POINT (H2,N,NP2,TX)                                          ABS03290
      DO 3900 K=1,KMAX                                                  ABS03300
 3900 W(K)=TX(K)                                                        ABS03310
      IF (H2.LT.H1) H=H2                                                ABS03320
      J2=N                                                              ABS03330
 4000 A0=(RE+H1)*SPHI                                                   ABS03340
      DO 4100 I=1,J1                                                    ABS03350
      HMIN=A0-RE                                                        ABS03360
      JMIN=I                                                            ABS03370
      IF (HMIN.LE.Z(I+1)) GO TO 4200                                    ABS03380
 4100 CONTINUE                                                          ABS03390
 4200 X=HMIN                                                            ABS03400
      IF (HMIN.LE.0) GO TO 4400                                         ABS03410
      CALL POINT (X,N,NP,TX)                                            ABS03420
      JMIN=N                                                            ABS03430
      HMIN=A0-RE                                                        ABS03440
      IF (ABS(X-HMIN).GT.0.0001) GO TO 4200                             ABS03450
      IF (H2.GE.H1) J2=N                                                ABS03460
      IF (H2.GE.H1.OR.H2.LT.HMIN) H=HMIN                                ABS03470
      IF (.NOT.ISPOT) WRITE (IOOUT,8600) HMIN                           ABS03480
      IF (H2.LT.HMIN) J2=N                                              ABS03490
      IF ((H2.LT.HMIN).AND.(.NOT.ISPOT)) WRITE (IOOUT,8700) HMIN        ABS03500
```

```
      GO TO 4500                                                        ABS03510
 4400 IF (.NOT.ISPOT) WRITE (IOOUT,8600) HMIN                           ABS03520
      IF (H2.LT.H1) GO TO 4500                                          ABS03530
      IF ((ITYPE.EQ.3.OR.H2.GE.H1).AND.(.NOT.ISPOT))                    ABS03540
     1    WRITE (IOOUT,8800)                                           ABS03550
      ITYPE=2                                                           ABS03560
      JMIN=0                                                            ABS03570
      J2=1                                                              ABS03580
      H2=0.0                                                            ABS03590
      H=0.0                                                             ABS03600
C****  NOW DEFINE VERTICAL PATH QUANTITIES VH(1-8)                      ABS03610
 4500 IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8400)                ABS03620
      JSTOR=J-1                                                         ABS03630
      DO 5100 I=1,NL                                                    ABS03640
      J=J-1                                                             ABS03650
      IF (I.NE.1) X1=Z(J+1)                                             ABS03660
      X2=Z(J)                                                           ABS03670
      IF (J.EQ.J2.AND.K2.EQ.0) X2=H                                     ABS03680
      IF (J.EQ.JMIN.AND.K2.EQ.1) X2=HMIN                                ABS03690
      HM=(RE+X1)*SPHI-RE                                                ABS03700
      IF (HM.GT.Z(J).AND.HM.GT.X2) X2=HM                                ABS03710
      RX=(RE+X1)/(RE+X2)                                                ABS03720
      DS=X1-X2                                                          ABS03730
      ALP=90.0                                                          ABS03740
      THET= ASIN(SPHI)/CA                                               ABS03750
      SALP=RX*SPHI                                                      ABS03760
      IF (ABS(2-HM).GT.1.0E-5) ALP= ASIN(SALP)/CA                       ABS03770
      BET=ALP-THET                                                      ABS03780
      IF (SPHI.GT.1.0E-10) DS=(RE+X2)*SIN(BET*CA)/SPHI                  ABS03790
      THETA=180.0-THET                                                  ABS03800
      BETA=BETA+BET                                                     ABS03810
      PSI=BETA-ALP-ANGLE+180.0                                          ABS03820
      SR=SR+DS                                                          ABS03830
      DO 5000 K=1,KMAX                                                  ABS03840
      IF (K.EQ.7.OR.K.EQ.9) GO TO 5000                                  ABS03850
      AJ=EH(K,J)                                                        ABS03860
      BJ=EH(K,J+1)                                                      ABS03870
      IF (J.EQ.J1) BJ=E(K)                                              ABS03880
      IF (J.EQ.J2.AND.H2.LT.H1.AND.H2.GT.0.0) AJ=W(K)                   ABS03890
      IF (J.EQ.JMIN.AND.H2.GE.H1) AJ=TX(K)                              ABS03900
      IF (J.EQ.JMIN.AND.ABS(H2-HM).LT.1.0E-5) AJ=TX(K)                  ABS03910
      IF (K2.EQ.0) GO TO 4600                                           ABS03920
      IF (J.EQ.J2) BJ=W(K)                                              ABS03930
      IF (J.EQ.JMIN) AJ=TX(K)                                           ABS03940
 4600 IF (AJ.EQ.0.0.OR.BJ.EQ.0.0) GO TO 4800                            ABS03950
      IF (AJ.EQ.BJ) GO TO 4700                                          ABS03960
      EV=DS*(AJ-BJ)/ALOG(AJ/BJ)                                         ABS03970
      GO TO 4900                                                        ABS03980
 4700 EV=DS*AJ                                                          ABS03990
      GO TO 4900                                                        ABS04000
 4800 EV=0.0                                                            ABS04010
 4900 VH(K)=VH(K)+EV                                                    ABS04020
 5000 WLAY(J,K)=EV                                                      ABS04030
      IF ((JP.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8500) J,             ABS04040
     1    X1,(VH(L),L=1,6),VH(8),PSI,ALP,BETA,THETA,SR                  ABS04050
      IF (J.EQ.J2.AND.H2.GE.H1) GO TO 5600                              ABS04060
      IF (J.EQ.JMIN.AND.K2.EQ.1) GO TO 5400                             ABS04070
      SPHI=SALP                                                         ABS04080
      IF (J.EQ.J2.AND.K2.EQ.0) GO TO 5200                               ABS04090
 5100 CONTINUE                                                          ABS04100
 5200 IF (HMIN.LE.0) GO TO 5800                                         ABS04110
      IF ((LEN.EQ.0).AND.(.NOT.ISPOT)) WRITE (IOOUT,8900)               ABS04120
      IF ((LEN.EQ.1).AND.(.NOT.ISPOT)) WRITE (IOOUT,9000)               ABS04130
      IF (LEN.EQ.0) GO TO 5800                                          ABS04140
      K2=1                                                              ABS04150
      X1=X2                                                             ABS04160
      IF (ABS(X1-HMIN).LE.0.001) GO TO 5800                             ABS04170
      H=HMIN                                                            ABS04180
      J=J2+1                                                            ABS04190
      IF (NP2.EQ.1) J=J-1                                               ABS04200
```

```
      B=BETA                                                                ABS04210
      PH=180.0- ASIN(SPHI)/CA                                               ABS04220
      TS=SR                                                                 ABS04230
      PS=PSI                                                                ABS04240
      DO 5300 K=1,KMAX                                                      ABS04250
 5300 E(K)=VH(K)                                                            ABS04260
      GO TO 4500                                                            ABS04270
 5400 BETA=2.*BETA-B                                                        ABS04280
      PSI=2.*PSI-PS                                                         ABS04290
      SR=2.*SR-TS                                                           ABS04300
C      LONG PATH  TAKEN                                                     ABS04310
      PHI=PH                                                                ABS04320
      DO 5500 K=1,KMAX                                                      ABS04330
 5500 VH(K)=2.*VH(K)-E(K)                                                   ABS04340
      GO TO 5800                                                            ABS04350
 5600 DO 5700 K=1,KMAX                                                      ABS04360
 5700 VH(K)=2.0*VH(K)                                                       ABS04370
      BETA=2.0*BETA                                                         ABS04380
      SR=2.0*SR                                                             ABS04390
      IF (H2.EQ.H1) GO TO 5800                                              ABS04400
      SPHI=SIN(ANGLE*CA)                                                    ABS04410
      GO TO 2600                                                            ABS04420
 5800 CONTINUE                                                              ABS04430
      DO 5900 K=1,KMAX                                                      ABS04440
      W(K)=VH(K)                                                            ABS04450
 5900 CONTINUE                                                              ABS04460
 6000 CONTINUE                                                              ABS04470
 6100 RETURN                                                                ABS04480
C                                                                           ABS04490
 6200 FORMAT (7F10.3)                                                       ABS04500
 6250 FORMAT (2F10.3,I2)                                                    ABS04510
 6300 FORMAT (1H0,9X,4H H1=,F7.3,6HKM,H2=,F7.3,9HKM,ANGLE=,                 ABS04520
     1        F8.4,13HGEOM. RANGE =,F7.2,8HKM,BETA=,F8.5,                   ABS04530
     2        5H,VIS=,F6.1)                                                 ABS04540
 6400 FORMAT (3F10.3,2F5.1,2E10.3,2F10.3)                                   ABS04550
 6500 FORMAT (10X,26HINPUT METEOROLOGICAL DATA:/10X,2HZ=,                   ABS04560
     1        F7.2,7H KM, P=,F7.2,6H MB,T=,F5.1,8H C, DEW ,                 ABS04570
     2        7HPT.TEMP,F5.1,17H C, REL HUMIDITY=,F5.1,                     ABS04580
     3        16H %, H2O DENSITY=,1PE9.2,7H GM M-3/10X,                     ABS04590
     4        15H OZONE DENSITY=,E9.2,20H GM-3, VISUAL RANGE=               ABS04600
     5        ,0PF6.1,10H KM,RANGE=,F10.3,4H KM )                           ABS04610
 6600 FORMAT (24H  MODEL ATMOSPHERE NO. 7,/4X,6HZ (KM),3X,                  ABS04620
     1        6HP (MB),4X,30HT (C) DEW PT %RH  H2O(GM.M-3) ,                ABS04630
     2        19H03(GM.M-3) NO. DEN.)                                       ABS04640
 6700 FORMAT (//10X,28H HORIZONTAL PATH, ALTITUDE =,F7.3,                   ABS04650
     1        11H KM,RANGE =,F7.3,3H KM)                                    ABS04660
 6800 FORMAT (//10X,37H SLANT PATH BETWEEN ALTITUDES H1 AND ,               ABS04670
     1        13HH2 WHERE H1 =,F7.3,8H KM H2 =,F7.3,                        ABS04680
     2        18H KM,ZENITH ANGLE =,F7.3,8H DEGREES)                        ABS04690
 6900 FORMAT (//10X,39H SLANT PATH TO SPACE FROM ALTITUDE H1 =              ABS04700
     1        ,F7.3,19H KM, ZENITH ANGLE =,F7.3,                            ABS04710
     2        8H DEGREES)                                                   ABS04720
 7200 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,11H = TROPICAL)                 ABS04730
 7300 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,                                ABS04740
     1        21H = MIDLATITUDE SUMMER)                                     ABS04750
 7400 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,                                ABS04760
     1        21H = MIDLATITUDE WINTER)                                     ABS04770
 7500 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,14H = SUB-ARCTIC ,              ABS04780
     1        7HSUMMER )                                                    ABS04790
 7600 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,14H = SUB-ARCTIC ,              ABS04800
     1        7HWINTER )                                                    ABS04810
 7700 FORMAT (/20X,18H MODEL ATMOSPHERE ,I1,11H = 1962 US ,                 ABS04820
     1        10HSTANDARD   )                                               ABS04830
 7800 FORMAT(/20X,18H MODEL ATMOSPHERE ,I1,20H = ISRAELI STANDARD ,         ABS04840
     1       16H(YEAR, DAYTIME) )                                           ABS04850
 7900 FORMAT(/20X,18H MODEL ATMOSPHERE ,I1,20H = ISRAELI STANDARD ,         ABS04860
     1       18H(YEAR, NIGHTTIME) )                                         ABS04870
 8000 FORMAT (/10X,21H FREQUENCY RANGE V1= ,F7.1,9H CM-1 TO ,               ABS04880
     1        4HV2= ,F7.1,14H CM-1 FOR DV =,F6.1,9H CM-1   ('              ABS04890
     2        ,F6.2,3H - ,F5.2,10H MICRONS ))                               ABS04900
```

```
 8100 FORMAT (1H1,///10X,20H HORIZONTAL PROFILES/)                        ABS04910
 8200 FORMAT (38H TRAJECTORY MISSES EARTHS ATMOSPHERE. ,                  ABS04920
     1         31HCLOSEST DISTANCE OF APPROACH IS,F10.2,1X,/              ABS04930
     2         ,1X,18HEND OF CALCULATION)                                 ABS04940
 8300 FORMAT (1X,I4,F6.1,12(E9.3))                                        ABS04950
 8400 FORMAT (1H1,///10X,21H  VERTICAL PROFILES  ,53X,3HPSI,              ABS04960
     1         6X,3HPHI,6X,4HBETA,3X,13HTHETA   RANGE)                    ABS04970
 8500 FORMAT (I3,F6.1,7E10.3,4F9.4,F6.1)                                  ABS04980
 8600 FORMAT (8H HMIN = ,F10.3)                                           ABS04990
 8700 FORMAT (40H H2 WAS SET LESS THAN HMIN AND HAS BEEN ,                ABS05000
     1         34HRESET EQUAL TO     HMIN I.E. H2 = ,F10.3)             ABS05010
 8800 FORMAT (41H PATH INTERSECTS EARTH - PATH CHANGED TO ,               ABS05020
     1         23HTYPE 2 WITH H2 = 0.0 KM)                              ABS05030
 8900 FORMAT (36H CHOICE OF TWO PATHS FOR THIS CASE -,                    ABS05040
     1         42HSHORTEST PATH TAKEN.  FOR LONGER PATH SET ,           ABS05050
     2         6HLEN=1,)                                                ABS05060
 9000 FORMAT (44H CHOICE OF TWO PATHS FOR THIS CASE -LONGEST ,            ABS05070
     1         41HPATH TAKEN.   FOR SHORT PATH SET LEN = 0 )            ABS05080
      END                                                                 ABS05090
```

```
      SUBROUTINE CKER (V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,TRAN)       CKR00010
      LOGICAL ISPOT                                                     CKR00020
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCKR00030
      IERR=0                                                            CKR00040
      IF (MULDV.LE.0) MULDV=1                                           CKR00050
C     CHECK FOR PROPER WINDOW REGION                                    CKR00060
      IF(V2.LE.830.0) GO TO 1090                                        CKR00070
      IF (V1.GE.1250.0.AND.V2.LE.2010.0) GO TO 1090                     CKR00080
      IF (V1.GE.3330.0.AND.V2.LT.5010.0) GO TO 1090                     CKR00090
      IF (V1.GE.39990.0) GO TO 1090                                     CKR00100
      GO TO 1100                                                        CKR00110
 1090 WRITE(IOOUT,3190)                                                 CKR00120
      TRAN=1.                                                           CKR00130
      IERR=1                                                            CKR00140
      RETURN                                                            CKR00150
C     CHECK FOR PROPER INTEGER VALUES                                   CKR00160
1100  IV1=20*IFIX((V1-10.0)/20.0)+10                                    CKR00170
      IV2=20*IFIX((V2-10.0)/20.0+0.99)+10                               CKR00180
      IF (IV1.LT.830) IV1=830                                           CKR00190
      IF (IV1.LT.1250.AND.IV2.GT.1250) IV2=1250                         CKR00200
      IF (IV1.LT.2010.AND.IV2.GT.2010) IV1=2010                         CKR00210
      IF (IV1.LT.3330.AND.IV2.GT.3330) IV2=3330                         CKR00220
      IF (IV1.LT.5010.AND.IV2.GT.5010) IV1=5010                         CKR00230
      IF (IV1.LT.39990.AND.IV2.GT.39990) IV2=39990                      CKR00240
      V1=FLOAT(IV1)                                                     CKR00250
      V2=FLOAT(IV2)                                                     CKR00260
      DV=20.*FLOAT(MULDV)                                               CKR00270
      IWVCK=0                                                           CKR00280
      IDV=IFIX(DV)                                                      CKR00290
C     WHEN CALLED FROM SPOT CHECK FOR MORE THAN 15 DIVISIONS            CKR00300
C     (16 WAVENUMBER VALUES) WHICH IS ARRAY SIZE.                       CKR00310
C     CHECK HERE FOR ROUNDING PROBLEMS ON V1, V2 CAUSING                CKR00320
C     TOO SMALL AN INCREMENT                                            CKR00330
      IF (ISPOT.AND.(FLOAT(IV2-IV1)/15..GT.(FLOAT(IDV)+.001))) IWVCK=1  CKR00340
      IF (IWVCK.NE.1) GO TO 91                                          CKR00350
      CKDV=FLOAT(IV2-IV1)/(15.*20.)                                     CKR00360
      MULDV=IFIX(CKDV)                                                  CKR00370
      IF (FLOAT(MULDV)/CKDV.LT..99) MULDV=MULDV+1                       CKR00380
      DVHOLD=20.*FLOAT(MULDV)                                           CKR00390
      WRITE (IOOUT,93) DV,DVHOLD                                        CKR00400
      DV=DVHOLD                                                         CKR00410
      IF (DV.LT.20.) DV=20.                                             CKR00420
      IDV=IFIX(DV)                                                      CKR00430
93    FORMAT (1X,'DIVISION LIMITS CHANGED FROM ',F10.3,                 CKR00440
     +' TO ',F10.3)                                                     CKR00450
91    CONTINUE                                                          CKR00460
      RETURN                                                            CKR00470
 3190 FORMAT (6X,'******FREQUENCY IS OUTSIDE OF THE WINDOW******'/      CKR00480
     1        6X,'******TOTAL TRANSMITTANCE IS 1.0000******')           CKR00490
      END                                                               CKR00500
```

```
      SUBROUTINE FREQSL(I,IV,W,TX)                                      FRE00010
      COMMON /MO5/ C1(501),C2(258),C3(86),C4(33),C5(6),C5DUM(9),C8(82),  FRE00020
     1        C11(4),C12(15),C14(21),C15(6)                              FRE00030
      COMMON /MO7/ TR(67),FW(67),FO(67)                                 FRE00040
      COMMON /MO8/SUM4,SUM5,SUM8,SUM11,SUM6                             FRE00050
      DIMENSION W(16),TX(16)                                            FRE00060
      IF (I.EQ.1) GO TO 10                                              FRE00070
      IF (I.GE.2.AND.I.LE.3) GO TO 11                                   FRE00080
      IF (I.GE.4.AND.I.LE.5) GO TO 12                                   FRE00090
      IF (I.GE.6.AND.I.LE.12) GO TO 13                                  FRE00100
      IF (I.GE.13.AND.I.LE.21) GO TO 15                                 FRE00110
      IF (I.EQ.22) GO TO 16                                             FRE00120
      IF ((I.GE.23.AND.I.LE.59).OR.(I.GE.127.AND.I.LE.209)) RETURN      FRE00130
      IF ((I.GE.60.AND.I.LE.63)) GO TO 14                               FRE00140
      IF (I.GE.64.AND.I.LE.76) GO TO 18                                 FRE00150
      IF ((I.GE.77.AND.I.LE.81).OR.(I.GE.87.AND.I.LE.96)) GO TO 18      FRE00160
      IF (I.GE.82.AND.I.LE.86) GO TO 30                                 FRE00170
      IF ((I.GE.97.AND.I.LE.101).OR.(I.GE.105.AND.I.LE.109)) GO TO 14   FRE00180
      IF (I.GE.102.AND.I.LE.104) GO TO 9                                FRE00190
      IF (I.GE.110.AND.I.LE.112) GO TO 21                               FRE00200
      IF (I.GE.113.AND.I.LE.123) GO TO 22                               FRE00210
      IF ((I.GE.124.AND.I.LE.126).OR.(I.GE.210.AND.I.LE.363)) GO TO 23  FRE00220
      IF ((I.GE.364.AND.I.LE.419).OR.(I.GE.454.AND.I.LE.599)) GO TO 24  FRE00230
      IF ((I.GE.420.AND.I.LE.453).OR.(I.GE.600.AND.I.LE.606).OR.(I.GE   FRE00240
     1.1160.AND.I.LE.1334)) GO TO 35                                    FRE00250
      IF (I.GE.607.AND.I.LE.609) GO TO 25                               FRE00260
      IF (I.GE.610.AND.I.LE.621) GO TO 26                               FRE00270
      IF ((I.GE.622.AND.I.LE.629).OR.(I.GE.686.AND.I.LE.1159).OR.(I.GE  FRE00280
     1.1335)) GO TO 27                                                  FRE00290
      IF (I.GE.630.AND.I.LE.685) GO TO 28                               FRE00300
4     CALL H2OVAP(I,W,C1,TX)                                            FRE00310
      GO TO 40                                                          FRE00320
6     CALL OZONE(I,W,C3,TX)                                             FRE00330
7     CALL UNIMIX(I,W,C2,TX)                                            FRE00340
      GO TO 4                                                           FRE00350
8     CALL NO2(I,W,C15,TX)                                              FRE00360
      GO TO 6                                                           FRE00370
9     CALL H2O410(I,IV,W,C5,TX,SUM5)                                    FRE00380
      GO TO 8                                                           FRE00390
10    CALL NH3(I,W,C14,TX)                                              FRE00400
      GO TO 9                                                           FRE00410
11    CALL NITRIC(I,W,C11,SUM11,TX)                                     FRE00420
      GO TO 10                                                          FRE00430
12    CALL NITRIC(I,W,C11,SUM11,TX)                                     FRE00440
13    CALL NH3(I,W,C14,TX)                                              FRE00450
14    CALL H2O410(I,IV,W,C5,TX,SUM5)                                    FRE00460
      GO TO 6                                                           FRE00470
15    CALL SO2(I,W,C12,TX)                                              FRE00480
      GO TO 13                                                          FRE00490
16    CALL SO2(I,W,C12,TX)                                              FRE00500
      GO TO 14                                                          FRE00510
18    CALL H2O410(I,IV,W,C5,TX,SUM5)                                    FRE00520
19    CALL NITRO(I,W,C4,TX,SUM4)                                        FRE00530
      GO TO 6                                                           FRE00540
20    CALL SO2(I,W,C12,TX)                                              FRE00550
      GO TO 19                                                          FRE00560
21    CALL MOLSCT(IV,W,TX,SUM6)                                         FRE00570
      GO TO 14                                                          FRE00580
22    CALL MOLSCT(IV,W,TX,SUM6)                                         FRE00590
      GO TO 6                                                           FRE00600
23    CALL MOLSCT(IV,W,TX,SUM6)                                         FRE00610
      GO TO 7                                                           FRE00620
24    CALL MOLSCT(IV,W,TX,SUM6)                                         FRE00630
      GO TO 4                                                           FRE00640
25    CALL UNIMIX(I,W,C2,TX)                                            FRE00650
35    CALL MOLSCT(IV,W,TX,SUM6)                                         FRE00660
      GO TO 40                                                          FRE00670
26    CALL UVOZNE(I,W,C8,TX,SUM8)                                       FRE00680
      GO TO 25                                                          FRE00690
27    CALL UVOZNE(I,W,C8,TX,SUM8)                                       FRE00700
```

```
      CALL MOLSCT(IV,W,TX,SUM6)                                        FRE00710
      GO TO 40                                                         FRE00720
28    CALL H2OVAP(I,W,C1,TX)                                           FRE00730
      GO TO 27                                                         FRE00740
30    CALL H2O410(I,IV,W,C5,TX,SUM5)                                   FRE00750
      GO TO 20                                                         FRE00760
40    RETURN                                                           FRE00770
      END                                                              FRE00780
```

```
      SUBROUTINE H2OVAP(I,W,C1,TX)                                      H2V00010
      DIMENSION C1(501),TX(1),W(1)                                      H2V00020
C*****************TRANSMITTANCE FOR WATER VAPOR************************H2V00030
      IF (W(1).LT.1.0E-20) GO TO 500                                    H2V00040
      IF (I.LE.22) I1=I                                                 H2V00050
      IF (I.GE.60.AND.I.LE.126) I1=I-37                                 H2V00060
      IF(I.GE.210.AND.I.LE.419) I1=I-120                                H2V00070
      IF(I.GE.454.AND.I.LE.599) I1=I-154                                H2V00080
      IF (I.GE.630) I1=I-184                                            H2V00090
      WS1=ALOG10(W(1))+C1(I1)                                           H2V00100
      TX(1)=EXP(-10**(-1.14619+0.55013*WS1))                            H2V00110
  500 RETURN                                                            H2V00120
      END                                                               H2V00130
```

```
      SUBROUTINE H20410(I,IV,W,C5,TX,SUM5)                                  H2F00010
C***WATER VAPOR CONTINUUM, 3-5 AND 8-12 MICRON REGIONS                      H2F00020
      DIMENSION C5(6),TX(5),W(16)                                           H2F00030
      V=FLOAT(IV)                                                           H2F00040
      IF(I.GT.109) GO TO 100                                                H2F00050
      IF(I.GT.22) GO TO 200                                                 H2F00060
C***1.LE.I.LE.22:CALCULATE OPTICAL DEPTH DUE TO 8-12 MICRON CONTINUUM       H2F00070
      TX(5)=(4.18+5578.0*EXP(-7.87E-03*V))*W(5)                             H2F00080
      GO TO 300                                                             H2F00090
  200 CONTINUE                                                              H2F00100
      IF(I.LT.63) GO TO 100                                                 H2F00110
      IF(I.GT.68) GO TO 400                                                 H2F00120
C***63.LE.I.LE.68:CALCULATE OPTICAL DEPTH DUE TO 4.6-4.8 MICRON             H2F00130
C***CONTINUUM,MODEL FROM BEN-SHALOM ET AL.1980,SPIE,VOL.253,261.            H2F00140
      II=I-62                                                               H2F00150
      TX(5)=C5(II)*577.6*W(16)                                              H2F00160
      GO TO 300                                                             H2F00170
  400 CONTINUE                                                              H2F00180
      IF(I.LT.77) GO TO 100                                                 H2F00190
C***77.LE.I.LE.109:CALCULATE OPTICAL DEPTH DUE TO 3.3-4.3 MICRON            H2F00200
C***CONTINUUM, MODEL FROM WATKINS ET AL.1979,APPL.OPT.,VOL.18,1149.         H2F00210
      V=V*1.0E-03                                                           H2F00220
      V2=V*V                                                                H2F00230
      V3=V2*V                                                               H2F00240
      CBURCH=46.4745-48.0898*V+16.3988*V2-1.83217*V3                        H2F00250
      CASL=-370.082+508.137*V-225.822*V2+32.7744*V3                         H2F00260
      TX(5)=CBURCH*W(10)+CASL*W(15)                                         H2F00270
      GO TO 300                                                             H2F00280
  100 TX(5)=0.0                                                             H2F00290
  300 SUM5=TX(5)                                                            H2F00300
      IF(TX(5).LT.1.0E-05) GO TO 500                                        H2F00310
      IF(TX(5).GT.20.0) GO TO 600                                           H2F00320
      TX(5)=EXP(-TX(5))                                                     H2F00330
      RETURN                                                                H2F00340
  500 TX(5)=1.0                                                             H2F00350
      RETURN                                                                H2F00360
  600 TX(5)=0.0                                                             H2F00370
      RETURN                                                                H2F00380
      END                                                                   H2F00390
```

```
      SUBROUTINE LTPATH(WLAY,WPATH,TBBY,ANGLE,LEN,ITYPE,H1,H2,MODEL)     LTP00010
      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL      LTP00020
      DIMENSION WLAY(34,16),TBBY(68),WPATH(68,16)                        LTP00030
      COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1                     LTP00040
      COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL                        LTP00050
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTULTP00060
      COMMON /SPOTLO/ISPOT,LOREAD,N16                                    LTP00070
      LOGICAL ISPOT,N16,LOREAD                                           LTP00080
  555 IL=0                                                               LTP00090
      IF (ITYPE.EQ.1) GO TO 1000                                         LTP00100
      IF (ITYPE.EQ.2.AND.H1.EQ.H2) J2=J1                                 LTP00110
      IF (H2.GT.H1.AND.ANGLE.GT.90..AND.NP1.EQ.1) J1=J1-1                LTP00120
      IF (JEXTRA.EQ.1) J2=J2+1                                           LTP00130
      IF ((ITYPE.EQ.2).AND.(H1.GT.H2).AND.(LENTOR.EQ.1)) J2=             LTP00140
     1    J2-1                                                          LTP00150
      IF (ITYPE.EQ.3) J2=NL                                              LTP00160
      IF (.NOT.ISPOT) WRITE (IOOUT,1200)                                 LTP00170
      DO 100 IK=1,68                                                     LTP00180
      TBBY(IK)=0.                                                        LTP00190
      DO 100 K=1,KMAX                                                    LTP00200
      WPATH(IK,K)=0.                                                     LTP00210
  100 CONTINUE                                                           LTP00220
      LEN=0                                                              LTP00230
      NLL=NL-1                                                           LTP00240
      IL=J1+1                                                            LTP00250
      IJ=IL+NLL                                                          LTP00260
      DO 200 K=1,KMAX                                                    LTP00270
      E(K)=0.                                                            LTP00280
  200 CONTINUE                                                           LTP00290
      IF (ANGLE.GT.90.0) GO TO 300                                       LTP00300
      LEN=1.                                                             LTP00310
      IL=J1-1                                                            LTP00320
      HMIN=1.0E-6                                                        LTP00330
      IJ=NLL                                                             LTP00340
  300 CONTINUE                                                           LTP00350
      DO 800 IK=1,68                                                     LTP00360
      IF (LEN.EQ.0) IL=IL-1                                              LTP00370
      IF (LEN.EQ.1) IL=IL+1                                              LTP00380
      IJ=IJ-1                                                            LTP00390
      IF (IL.EQ.0) GO TO 800                                             LTP00400
      DO 400 K=1,KMAX                                                    LTP00410
      W(K)=E(K)+WLAY(IL,K)                                               LTP00420
      WPATH(IK,K)=W(K)                                                   LTP00430
  400 CONTINUE                                                           LTP00440
      IF (IL.LE.0.OR.IL.GE.NL) GO TO 500                                 LTP00450
      TBAR=(T(IL)+T(IL+1))*0.5                                           LTP00460
C     JEXTRA IS 1 ONLY WHEN PROGRAM NEVER LEAVES ONE LAYER               LTP00470
      IF (JEXTRA.EQ.1) TBAR=(T(J1)+T(J1+1))*0.5                          LTP00480
  500 CONTINUE                                                           LTP00490
      TBBY(IK)=TBAR                                                      LTP00500
      DO 600 K=1,KMAX                                                    LTP00510
      E(K)=W(K)                                                          LTP00520
  600 CONTINUE                                                           LTP00530
      IF (ANGLE.LE.90.0.AND.IL.EQ.NLL) GO TO 900                         LTP00540
      IF (ITYPE.EQ.3.AND.ANGLE.LE.90.0) GO TO 700                        LTP00550
      IF (ITYPE.EQ.3.AND.LEN.EQ.1.AND.IL.EQ.J2) GO TO 900                LTP00560
      IF (ITYPE.EQ.2.AND.LENTOR.EQ.0.AND.IL.EQ.J2) GO TO 900             LTP00570
      IF (IL.EQ.JMIN.AND.HMIN.GT.0) LEN=1                                LTP00580
      IF (IL.EQ.1.AND.HMIN.LE.0.0) GO TO 900                             LTP00590
      IF (LEN.EQ.0) GO TO 700                                            LTP00600
      IF (IL.EQ.JMIN.AND.IJ.EQ.IL+NLL) IL=IL-1                           LTP00610
      IF (ITYPE.EQ.2.AND.IL.EQ.J2) GO TO 900                             LTP00620
  700 CONTINUE                                                           LTP00630
      IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=              LTP00640
     1    1,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK)                LTP00650
  800 CONTINUE                                                           LTP00660
      IKMAX=68                                                           LTP00670
      LEN=LENTOR                                                         LTP00680
      RETURN                                                             LTP00690
  900 CONTINUE                                                           LTP00700
```

```
      IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=                LTP00710
     1    1,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK)                 LTP00720
      IKMAX=IK                                                            LTP00730
      LEN=LENTOR                                                          LTP00740
      RETURN                                                              LTP00750
 1000 DO 1100 K=1,KMAX                                                    LTP00760
      WPATH(1,K)=W(K)                                                     LTP00770
 1100 CONTINUE                                                            LTP00780
      IF (MODEL.EQ.0) J1=1                                                LTP00790
      J2=J1                                                               LTP00800
      TBBY(1)=T(J1)                                                       LTP00810
      IKMAX=1                                                             LTP00820
      IK=1                                                                LTP00830
      IF (.NOT.ISPOT) WRITE (IOOUT,1200)                                  LTP00840
      IF (.NOT.ISPOT) WRITE (IOOUT,1300) IK,(WPATH(IK,K),K=               LTP00850
     1    1,6),WPATH(IK,8),(WPATH(IK,K),K=10,14),TBBY(IK)                 LTP00860
      HMIN=1.0E-6                                                         LTP00870
      RETURN                                                              LTP00880
C                                                                         LTP00890
 1200 FORMAT (//,20X,37H CUMULATIVE ABSORBER AMOUNTS FOR THE ,            LTP00900
     1         16HATMOSPHERIC PATH,// 8X,3HH2O,5X,4HCO2+,6X,              LTP00910
     2         2HO3,7X,2HN2,6X,5HH2O C,4X,5HMOL S,4X,5HO3 UV,             LTP00920
     3         4X,5HH2O C,5X,4HHNO3,6X,3HSO2,6X,3HNH3,6X,3HNO2,           LTP00930
     4         5X,4HTAVE)                                                LTP00940
 1300 FORMAT (I5,12E9.3,F10.3)                                            LTP00950
      END                                                                 LTP00960
```

```
      SUBROUTINE MOLSCT(IV,W,TX,SUM6)                                   MOL00010
C*********TRANSMITTANCE FOR MOLECULAR SCATTERING **********************MOL00020
C    C6 EXPRESSION MODIFIED AS PER SHETTLE ET AL 1980,APPL.OPT.,VOL.19,  MOL00030
C    2873.                                                              MOL00040
      DIMENSION TX(6),W(6)                                              MOL00050
      V=FLOAT(IV)                                                       MOL00060
      C6=(V**4)/(9.67578E+18-1.11836E+09*V**2)                          MOL00070
      TX(6)=C6*W(6)                                                     MOL00080
      SUM6=TX(6)                                                        MOL00090
      IF (TX(6).EQ.0.0) GO TO 200                                       MOL00100
      IF (TX(6).LE.0.1) GO TO 100                                       MOL00110
      IF (TX(6).GT.20.) GO TO 300                                       MOL00120
      TX(6)=EXP(-TX(6))                                                 MOL00130
      GO TO 400                                                         MOL00140
  100 TX(6)=1.0-TX(6)+0.5*TX(6)*TX(6)                                   MOL00150
      GO TO 400                                                         MOL00160
  200 TX(6)=1.0                                                         MOL00170
      GO TO 400                                                         MOL00180
  300 TX(6)=0.0                                                         MOL00190
  400 RETURN                                                            MOL00200
      END                                                               MOL00210
```

```
      SUBROUTINE NH3(I,W,C13,TX)                                        NHA00010
      COMMON /MO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),     NHA00020
     1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2                                 NHA00030
      DIMENSION C13(21),TX(13),W(13)                                    NHA00040
C*******************************************************************   NHA00050
C                                                                       NHA00060
C     THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY NH3 ( PPM READ IN NHA00070
C     THE MAIN PROGRAM).                                                NHA00080
C                                                                       NHA00090
C*******************************************************************   NHA00100
      II=I                                                              NHA00110
      IF (W(13).LT.1.0E-20) GO TO 3                                     NHA00120
      WS13=ALOG10(W(13))+C13(II)                                        NHA00130
      DO 1 J=1,9                                                        NHA00140
      IF (WS13-FNH3(J)) 2,2,1                                           NHA00150
1     CONTINUE                                                          NHA00160
2     TX(13)=EXP(-10**(FH1(J)+FH2(J)*WS13))                             NHA00170
3     RETURN                                                            NHA00180
      END                                                               NHA00190
```

```
      SUBROUTINE NITRIC(I,W,C11,SUM11,TX)                                NITC0010
      DIMENSION C11(4),TX(11),W(11)                                      NITC0020
C******** TRANSMITTANCE FOR NITRIC ACID*********************************NITC0030
      HABS=0.                                                            NITC0040
      IF (I.LT.2.OR.I.GT.46) GO TO 100                                   NITC0050
      IF (I.GT.5.AND.I.LT.23) GO TO 100                                  NITC0060
      I1=I-1                                                             NITC0070
      HABS=C11(I1)                                                       NITC0080
  100 CONTINUE                                                           NITC0090
      TX(11)=HABS*W(11)                                                  NITC0100
      SUM11=TX(11)                                                       NITC0110
      IF (TX(11).EQ.0.0) GO TO 300                                       NITC0120
      IF (TX(11).LE.0.1) GO TO 200                                       NITC0130
      IF (TX(11).GT.20.) GO TO 400                                       NITC0140
      TX(11)=EXP(-TX(11))                                                NITC0150
      GO TO 500                                                          NITC0160
  200 TX(11)=1.0-TX(11)+0.5*TX(11)*TX(11)                                NITC0170
      GO TO 500                                                          NITC0180
  300 TX(11)=1.0                                                         NITC0190
      GO TO 500                                                          NITC0200
  400 TX(11)=0.0                                                         NITC0210
  500 RETURN                                                             NITC0220
      END                                                                NITC0230
```

```
      SUBROUTINE NITRO(I,W,C4,TX,SUM4)                                  NITR0010
      DIMENSION C4(33),TX(4),W(4)                                       NITR0020
C*********TRANSMITTANCE FOR NITROGEN CONTINUUM*************************NITR0030
      IF (I.LT.64) GO TO 200                                            NITR0040
      I1=I-63                                                           NITR0050
C         TEMP FIX FOLLOWS                                              NITR0060
C     IF (I1.GT.10) GO TO 300                                           NITR0070
      TX(4)=C4(I1)*W(4)                                                 NITR0080
      SUM4=TX(4)                                                        NITR0090
      IF (TX(4).EQ.0.0) GO TO 200                                       NITR0100
      IF (TX(4).LE.0.1) GO TO 100                                       NITR0110
      IF (TX(4).GT.20.) GO TO 300                                       NITR0120
      TX(4)=EXP(-TX(4))                                                 NITR0130
      GO TO 400                                                         NITR0140
  100 TX(4)=1.0-TX(4)+0.5*TX(4)*TX(4)                                   NITR0150
      GO TO 400                                                         NITR0160
  200 TX(4)=1.0                                                         NITR0170
      GO TO 400                                                         NITR0180
  300 TX(4)=0.0                                                         NITR0190
  400 RETURN                                                            NITR0200
      END                                                               NITR0210
```

```
      SUBROUTINE NO2(I,W,C14,TX)                                        NO2X0010
      COMMON /MO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),     NO2X0020
     1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2                                 NO2X0030
      DIMENSION C14(6),TX(14),W(14)                                     NO2X0040
C*****************************************************************    NO2X0050
C                                                                       NO2X0060
C     THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY NO2 ( PPM READ IN NO2X0070
C     THE MAIN PROGRAM).                                                NO2X0080
C                                                                       NO2X0090
C*****************************************************************    NO2X0100
      IF (I.GE.102.AND.I.LE.104) I1=I-98                                NO2X0110
      IF (I.LE.3) I1=I                                                  NO2X0120
      IF (W(14).LT.1.0E-20) GO TO 3                                     NO2X0130
      WS14=ALOG10(W(14))+C14(I1)                                        NO2X0140
      DO 1 J=1,9                                                        NO2X0150
      IF (WS14-FNO2(J)) 2,2,1                                           NO2X0160
1     CONTINUE                                                          NO2X0170
2     TX(14)=EXP(-10**(O1(J)+O2(J)*WS14))                               NO2X0180
3     RETURN                                                            NO2X0190
      END                                                               NO2X0200
```

```
      SUBROUTINE OZONE(I,W,C3,TX)                                       OZN00010
      DIMENSION C3(86),TX(3),W(3)                                       OZN00020
C*********************TRANSMITTANCE FOR OZONE*****************************OZN00030
      IF (W(3).LT.1.0E-20) GO TO 500                                    OZN00040
      IF (I.LE.22) I1=I                                                 OZN00050
      IF (I.GE.60) I1=I-37                                              OZN00060
      WS3=ALOG10(W(3))+C3(I1)                                           OZN00070
      TX(3)=1/(1+EXP(-3.08019+2.11127*WS3))                             OZN00080
  500 RETURN                                                            OZN00090
      END                                                               OZN00100
```

```
      SUBROUTINE POINT (X,N,NP,TX)                                          POI00010
      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL          POI00020
      COMMON /EM1/HMIN,KMAX,IJ,J1,J2,JMIN,JEXTRA,NP1                        POI00030
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUPOI00040
      COMMON /SPOTLO/ISPOT,LOREAD,N16                                       POI00050
      DIMENSION TX(16)                                                      POI00060
      LOGICAL ISPOT,N16,LOREAD                                              POI00070
C*************************************************************************POI00080
C     SUBROUTINE POINT INTERPOLATES EXPONENTIALLY TO                        POI00090
C     DETERMINE THE EQUIVALENT ABSORBER AMOUNTS AT THAT ALTITUDE.           POI00100
C                                                                           POI00110
C      X IS THE HEIGHT IN QUESTION                                          POI00120
C      N IS THE LEVEL INTEGER CORRESPONDING TO X OR THE LEVEL BELOW X       POI00130
C      NP =1 IF X COINCIDES WITH MODEL ATMOSPHERE LEVEL ,IF NOT NP = 0      POI00140
C      TX(1-8) ARE ABSORBER  AMOUNTS PER KM AT HEIGHT X                     POI00150
C*************************************************************************POI00160
555   N=NL                                                                  POI00170
      NP=0                                                                  POI00180
      IF (X.LT.0.0) X=0.                                                    POI00190
      IF (X.GT.Z(NL)) GO TO 400                                             POI00200
      DO 100 I=1,NL                                                         POI00210
      N=I                                                                   POI00220
      IF (X-Z(I)) 200,400,100                                               POI00230
  100 CONTINUE                                                              POI00240
  200 J2=N                                                                  POI00250
      N=N-1                                                                 POI00260
      FAC=(X-Z(N))/(Z(J2)-Z(N))                                             POI00270
      DO 300 K=1,KMAX                                                       POI00280
      IF (K.EQ.9.OR.K.EQ.7) GO TO 300                                       POI00290
      TX(K)=0.0                                                             POI00300
      IF (EH(K,N).EQ.0.0) GO TO 300                                         POI00310
      IF (EH(K,N).GT.1000.0) GO TO 300                                      POI00320
      TX(K)=EH(K,N)*(EH(K,J2)/EH(K,N))**FAC                                 POI00330
  300 CONTINUE                                                              POI00340
      GO TO 700                                                             POI00350
  400 NP=1                                                                  POI00360
      DO 500 K=1,KMAX                                                       POI00370
  500 TX(K)=EH(K,N)                                                         POI00380
700   RETURN                                                                POI00390
      END                                                                   POI00400
```

```
      FUNCTION RESFN (NR,WAVE)                                          RES00010
C     THIS FUNCTION WILL READ IN UP TO 20 VALUES OF A RESPONSE FUNCTION RES00020
C     IF THE RESF CARD IS READ IN EQMAIN.  ONLY ONE RESPONSE FUNCTION   RES00030
C     PER RUN IS ALLOWED.  THIS FUNCTION WILL ALSO DO A LINEAR INTERPOLARES00040
C     OVER WAVELENGTH.  IF RESF CARD IS NOT READ A VALUE OF 1. IS RETURNEDRES00050
C     TO THE CALLING PROGRAM WHEN THIS FUNCTION IS REFERENCED.          RES00060
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTURES00070
      DIMENSION WAVELN(20),RESPFN(20)                                   RES00080
      DATA WAVELN,RESPFN,ICOUNT,NBR /20*0.,20*0.,0,1/                   RES00090
      IF (NR.NE.1) GO TO 6                                              RES00100
      ICOUNT=ICOUNT+1                                                   RES00110
      IF (ICOUNT.GT.1) GO TO 2                                          RES00120
      READ (IOIN,100) NBR                                               RES00130
      IF (NBR.GT.20) WRITE (IOOUT,102)                                  RES00140
      IF (NBR.GT.20) STOP                                               RES00150
      WRITE (IOOUT,103)                                                 RES00160
      DO 3 I=1,NBR                                                      RES00170
      READ (IOIN,101) WAVELN(I),RESPFN(I)                               RES00180
3     WRITE (IOOUT,104) WAVELN(I),RESPFN(I)                             RES00190
2     IF (WAVE.LT.(WAVELN(1)-.0001).OR.WAVE.GT.(WAVELN(NBR)+.0001))     RES00200
     + GO TO 6                                                          RES00210
      DO 4 I=1,NBR                                                      RES00220
      K=I                                                               RES00230
4     IF (WAVE.GE.WAVELN(I)) GO TO 5                                    RES00240
5     IF (WAVE/WAVELN(K).GE..99.AND.WAVE/WAVELN(K).LE.1.01) GO TO 7     RES00250
      IF (K.EQ.NBR) GO TO 8                                             RES00260
      RESFN=(WAVE-WAVELN(K))*(RESPFN(K+1)-RESPFN(K))/                   RES00270
     1 (RESPFN(K+1)-RESPFN(K))+RESPFN(K)                                RES00280
      RETURN                                                            RES00290
7     RESFN=RESPFN(K)                                                   RES00300
      RETURN                                                            RES00310
8     RESFN=RESPFN(NBR)                                                 RES00320
      RETURN                                                            RES00330
6     RESFN=1.                                                          RES00340
      RETURN                                                            RES00350
 100   FORMAT (I2)                                                      RES00360
 101   FORMAT (2(E10.4,1X))                                             RES00370
102   FORMAT (1H ,51HTHE NUMBER OF VALUES FOR THE RESPONSE FUNCTION IS GRES00380
     +,58HREATER THAN THE DIMENSIONS LIMITS OF WAVELN( ) AND RESPFN( ),  RES00390
     +/,1X,19HPROGRAM TERMINATED.)                                      RES00400
103   FORMAT (1H ,20X,23HINPUT RESPONSE FUNCTION,/,1X,15X,10HWAVELENGTH,RES00410
     +5X,10HR FUNCTION)                                                 RES00420
104   FORMAT (1H ,15X,2(E10.4,1X))                                      RES00430
      END                                                               RES00440
```

```
      SUBROUTINE SO2(I,W,C12,TX)                                        SO2X0010
      COMMON /MO3/ FS(9),S1(9),S2(9),FNH3(9),FH1(9),FH2(9),FNO2(9),     SO2X0020
     1 O1(9),O2(9),PPMSO2,PPMNH3,PPMNO2                                 SO2X0030
      DIMENSION C12(15),TX(12),W(12)                                    SO2X0040
C*******************************************************************SO2X0050
C                                                                       SO2X0060
C     THIS SUBROUTINE CALCULATES THE TRANSMITTANCE BY SO2 ( PPM READ IN SO2X0070
C     THE MAIN PROGRAM).                                                SO2X0080
C                                                                       SO2X0090
C*******************************************************************SO2X0100
      IF (W(12).LT.1.0E-20) GO TO 5                                     SO2X0110
      IF (I.GE.13.AND.I.LE.22) I1=I-12                                  SO2X0120
      IF (I.GE.82) I1=I-71                                              SO2X0130
      WS12=ALOG10(W(12))+C12(I1)                                        SO2X0140
      DO 1 J=1,9                                                        SO2X0150
      IF(WS12-FS(J)) 2,2,1                                              SO2X0160
1     CONTINUE                                                          SO2X0170
2     TX(12)=EXP(-10**(S1(J)+S2(J)*WS12))                               SO2X0180
5     RETURN                                                            SO2X0190
      END                                                               SO2X0200
```

```
      SUBROUTINE UNIMIX(I,W,C2,TX)                                      UNI00010
      DIMENSION C2(258),TX(2),W(2)                                      UNI00020
C*********************TRANSMITTANCE FOR UNIFORMLY MIXED GASES**********UNI00030
      IF (W(2).LT.1.0E-20) GO TO 500                                    UNI00040
      IF (I.LE.22) I1=I                                                 UNI00050
      IF (I.GE.60.AND.I.LE.126) I1=I-37                                 UNI00060
      IF(I.GE.210.AND.I.LE.363) I1=I-120                                UNI00070
      IF (I.GE.607) I1=I-363                                            UNI00080
      WS2=ALOG10(W(2))+C2(I1)                                           UNI00090
      TX(2)=EXP(-10**(-1.14619+0.55013*WS2))                            UNI00100
  500 RETURN                                                            UNI00110
      END                                                               UNI00120
```

```
      SUBROUTINE UVOZNE(I,W,C8,TX,SUM8)                                 UVZ00010
      DIMENSION C8(82),TX(8),W(8)                                       UVZ00020
C********* TRANSMITTANCE FOR UV OZONE ***********************************UVZ00030
      AI=I                                                              UVZ00040
      IF(I.LE.1159) GO TO 90                                            UVZ00050
      IF (I.GE.1335) GO TO 100                                          UVZ00060
   90 XX=10.0                                                           UVZ00070
      XI=(AI-610.0)/XX+1.0                                              UVZ00080
      L1=1                                                              UVZ00090
      L2=53                                                             UVZ00100
      GO TO 200                                                         UVZ00110
  100 XX=25.0                                                           UVZ00120
      XI=(AI-1335.0)/XX+57.0                                            UVZ00130
      L1=57                                                             UVZ00140
      L2=102                                                            UVZ00150
  200 DO 300 N=L1,L2                                                    UVZ00160
      XD=XI-FLOAT(N)                                                    UVZ00170
      IF (XD) 500,400,300                                               UVZ00180
  300 CONTINUE                                                          UVZ00190
  400 TX(8)=W(8)*C8(N)                                                  UVZ00200
      GO TO 600                                                         UVZ00210
  500 TX(8)=C8(N)+XD*(C8(N)-C8(N-1))                                    UVZ00220
      TX(8)=W(8)*TX(8)                                                  UVZ00230
  600 SUM8=TX(8)                                                        UVZ00240
      IF (TX(8).EQ.0.0) GO TO 800                                       UVZ00250
      IF (TX(8).LE.0.1) GO TO 700                                       UVZ00260
      IF (TX(8).GT.20.0) GO TO 900                                      UVZ00270
      TX(8)=EXP(-TX(8))                                                 UVZ00280
      GO TO 1000                                                        UVZ00290
  700 TX(8)=1.0-TX(8)+0.5*TX(8)*TX(8)                                   UVZ00300
      GO TO 1000                                                        UVZ00310
  800 TX(8)=1.0                                                         UVZ00320
      GO TO 1000                                                        UVZ00330
  900 TX(8)=0.0                                                         UVZ00340
 1000 RETURN                                                            UVZ00350
      END                                                               UVZ00360
```

```
      SUBROUTINE SPOT(WAVN1,WAVN2,VIS,NR,IERR,MULDV)                    SPOT0010
C*****************************************************************     SPOT0020
C     INPUT:  EXCLUDING THE OPTIONAL RESPONSE FUNCTION CARDS,           SPOT0030
C     THERE IS A MAXIMUM OF 7 CARDS TO EXECUTE THIS MODULE.             SPOT0040
C     THE CARDS MAY BE INSERTED IN ANY ORDER WITH THE EXCEPTION OF      SPOT0050
C     THE LAST CARD WHICH SIGNIFIES THAT EXECUTION IS TO BEGIN.         SPOT0060
C     THE CARDS ARE INPUT WITH FORMAT (A4,6X,7E10.4)                    SPOT0070
C     EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1 - 4          SPOT0080
C     FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL PER           SPOT0090
C     FIELD BEGINNING IN COL 11.                                        SPOT0100
C           THE CARDS ARE NOT ORDER DEPENDENT.                          SPOT0110
C     CARD 1                                                            SPOT0120
C  ENVR       ISORC, ITARG, IHAZE, MODEL, NLAM                         SPOT0130
C             ISORC = 0 SUNLIGHT ONLY                                   SPOT0140
C                     1 MOONLIGHT ONLY                                  SPOT0150
C                     2 EMISSION ONLY                                   SPOT0160
C                     3 SUNLIGHT AND EMISSION                           SPOT0170
C                     4 MOONLIGHT AND EMISSION                          SPOT0180
C             ITARG = 0 BACKGROUND ONLY                                 SPOT0190
C                     1 GROUND REFLECTANCE/EMISSION                     SPOT0200
C                     2 TARGET REFLECTANCE/EMISSION                     SPOT0210
C  ** AEROSOL ATTENUATION LIMITED TO 4 KM BASE HEIGHT AND 500 M THICK **SPOT0220
C     FOR SLANT PATHS IHAZE = 1,2, OR 3 ARE THE ONLY ALLOWED VALUES.    SPOT0230
C           IHAZE   =0,NO AEROSOL ATTENUATION                           SPOT0240
C                   =1, MARITIME POLAR                                  SPOT0250
C                   =2, MARITIME ARCTIC                                 SPOT0260
C                   =3, CONTINENTAL POLAR                               SPOT0270
C                   =4, RAIN                                            SPOT0280
C                   =5, SNOW                                            SPOT0290
C                   =7, USER SUPPLIED EXTINCTION COEFFICIENT            SPOT0300
C                       (READ ON ATM CARD - SEE CARD 3 BELOW)           SPOT0310
C                   =8, EXTINCTION COEFFICIENT WILL BE READ FROM        SPOT0320
C                       PHASE FUNCTION DATA FILE                        SPOT0330
C           MODEL = 1 TROPICAL MODEL ATMOSPHERE                         SPOT0340
C                   2 MIDLATITUDE SUMMER                                SPOT0350
C                   3 MIDLATITUDE WINTER                                SPOT0360
C                   4 SUBARCTIC SUMMER                                  SPOT0370
C                   5 SUBARCTIC WINTER                                  SPOT0380
C                   6 1962 US STANDARD                                  SPOT0390
C                   8 ISRAELI STANDARD (YEAR, DAYTIME)                  SPOT0400
C                   9 ISRAELI STANDARD (YEAR, NIGHTTIME)                SPOT0410
C           NLAM    OPTION FOR AEROSOL PHASE FUNCTION                   SPOT0420
C                   =  0 NO AEROSOL ATTENUATION                         SPOT0430
C                   NE 0 READ PHASE FUNCTION DATA SET - ALSO SEE        SPOT0440
C                        ID BELOW AND EXPLN OF PFN DATA SET BELOW       SPOT0450
C           ID      PHASE FUNCTION IDENTIFIER                           SPOT0460
C                   =0,   USER SUPPLIED                                 SPOT0470
C                   =1,   MARITIME ARCTIC, VIS=0.1 TO 2.0 KM            SPOT0480
C                   =2,   MARITIME POLAR, VIS=0.2 KM                    SPOT0490
C                   =3,   MARITIME POLAR, VIS=02. KM                    SPOT0500
C                   =4,   CONTINENTAL POLAR, VIS= 0.2 TO 2.5 KM         SPOT0510
C                   =5,   WHITE PHOSPHORUS                              SPOT0520
C                   =6,   HEXACHLOROETHANE                              SPOT0530
C                   =7,   FOG OIL                                       SPOT0540
C                   =8,   DUST (MODERATE AEROSOL LOADING)               SPOT0550
C                   =9,   DUST (HEAVY AEROSOL LOADING)                  SPOT0560
C                   =10,  MARITIME MODEL B, VIS=5KM, RH=95%             SPOT0570
C                   =11,  MARITIME MODEL B, VIS=10KM,RH=90%             SPOT0580
C                   =12,  MARITIME MODEL B, VIS=50KM,RH=50%             SPOT0590
C     CARD 2   ****  IF ISORC LT 2 OR ITARG LT 1 THIS CARD IS NOT NEEDEDSPOT0600
C  EMIS   EM(1), TM(1), EM(2), TM(2)                                    SPOT0610
C           EM(1)     EMISSIVITY OF GROUND                            SPOT0620
C           TM(1)     TEMPERATURE OF GROUND (KELVIN)                  SPOT0630
C           EM(2)     EMISSIVITY OF TARGET                            SPOT0640
C           TM(2)     TEMPERATURE OF TARGET (KELVIN)                  SPOT0650
C     CARD 3                                                          SPOT0660
C  ATM    ZENTH, CLDHGT, PHASE,BETAEX                                  SPOT0670
C           ZENTH     INCIDENT ANGLE OF SUNLIGHT OR MOONLIGHT (DEGREES)SPOT0680
C           CLDHGT    HEIGHT OF BOTTOM OF CLOUD LAYER (KM)            SPOT0690
C                     ONLY NEEDED WHEN IHAZE NE 0  (DEFAULT IS 0.)    SPOT0700
```

```
C             PHASE     PHASE ANGLE FOR MOONLIGHT (DEGREES)                  SPOT0710
C             BETAEX    USER SUPPLIED EXTINCTION COEFFICIENT                 SPOT0720
C                       VALID ONLY WHEN IHAZE=7                              SPOT0730
C     CARD 4                                                                 SPOT0740
C  TARG   RTARG,COSX,COSY,COSZ                                               SPOT0750
C             RTARG     SLANT RANGE FROM RECEIVER TO TARGET (KM)             SPOT0760
C             COSX      X-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)       SPOT0770
C             COSY      Y-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)       SPOT0780
C             COSZ      Z-DIRECTIONAL ANGLE OF TARGET NORMAL (DEGREES)       SPOT0790
C     CARD 5    ****  IF ITARG LT 1 THIS CARD IS NOT NEEDED                  SPOT0800
C  REFL   A0(1),A1(1),IALB(1),A0(2),A1(2),IALB(2)                            SPOT0810
C             A0(1)     ALBEDO COEFFICIENT FOR GROUND                        SPOT0820
C             A1(1)     ALBEDO COEFFICIENT FOR GROUND                        SPOT0830
C             IALB(1)   TYPE OF REFLECTION DISTRIBUTION FOR GROUND           SPOT0840
C             A0(2)     ALBEDO COEFFICIENT FOR TARGET                        SPOT0850
C             A1(2)     ALBEDO COEFFICIENT FOR TARGET                        SPOT0860
C             IALB(2)   TYPE OF REFLECTION DISTRIBUTION FOR TARGET           SPOT0870
C             IALB   = 0 LAMBERTIAN REFLECTION SURFACE                       SPOT0880
C                      1 ISOTROPIC                                           SPOT0890
C     CARD 6                                                                 SPOT0900
C  SENS   ALT, THETA, PHI, SANG2                                             SPOT0910
C             ALT       ALTITUDE OF RECEIVER (KM)                            SPOT0920
C             THETA     POLAR DIRECTION OF LOOK ANGLE (DEGREES)              SPOT0930
C             PHI       AZMITH DIRECTION OF LOOK ANGLE (DEGREES)             SPOT0940
C                                                                            SPOT0950
C*** METEOROLOGICAL AZIMUTH CONVENTION ASSUMED: N = 0 DEG,                   SPOT0960
C*** E = 90 DEG, S = 180 DEG, W = 270 DEG **************************        SPOT0970
C                                                                            SPOT0980
C             SANG2     HALF ANGLE DEFINING RECEIVER FIELD-OF-VIEW           SPOT0990
C                         (DEGREES)                                          SPOT1000
C     CARD 7                                                                 SPOT1010
C     GO          SIGNIFIES TO BEGIN EXECUTION, NO MORE INPUT FOR            SPOT1020
C                 THIS CALL.  NOTE THAT IF A DATA CARD IS NOT READ           SPOT1030
C                 THEN ANY VALUES ESTABLISHED FROM PREVIOUS CALLS            SPOT1040
C                 TO THE MODULE ARE STILL IN EFFECT.                         SPOT1050
C                                                                            SPOT1060
C  OPTIONAL CARDS FOR RESPONSE FUNTION (SET BY NR=1 IN EOMAIN)               SPOT1070
C  THESE CARDS MUST FOLLOW THE GO CARD AND CAN ONLY BE INSERTED ONCE         SPOT1080
C     CARD 1: NUMBER OF VALUES FOR RESPONSE FUNCTION - FORMAT (I2).          SPOT1090
C     CARDS 2 - NUMBER OF VALUES; FORMAT (2(E10.4,1X))                       SPOT1100
C               ONE VALUE OF WAVE (UM) AND RESPONSE FUNCTON PER CARD         SPOT1110
C               N.B. ONLY ONE RESPONSE FUNCTION PER EOSAEL RUN.              SPOT1120
C                                                                            SPOT1130
C    AUXILLARY READ FROM UNIT IPHFUN (ASL DATA SET PROVIDED WITH EOSAEL)     SPOT1140
C             ANG       ANGLES AT WHICH PHASE MATRIX IS DEFINED,             SPOT1150
C                       NANG VALUES  (DEFAULT IS 65), FORMAT(11(F6.2,1X))    SPOT1160
C             NANG,ID,WAVE,OMEGA0,BETAEX,BETABS                              SPOT1170
C                       NUMBER OF ANGLES AT WHICH THE PHASE FUNCTION HAS     SPOT1180
C                       VALUES, PFN IDENTIFIER, WAVELENGTH(UM), SINGLE       SPOT1190
C                       SCATTERING ALBEDO, EXTINCTION COEFFICIENTS (TOTAL    SPOT1200
C                       AND SCATTERING).                                     SPOT1210
C                       FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X)).       SPOT1220
C             PF        PHASE FUNCTION AT SPECIFIED ANGLES,                  SPOT1230
C                       FORMAT (6(E12.6,1X))                                 SPOT1240
C*************************************************************************  SPOT1250
      LOGICAL L1,L2,L3,L4,L5,L6,L7,ISPOT,N16,LOREAD,HORIZ                     SPOT1260
      DIMENSION DUMMY(16)                                                    SPOT1270
      EQUIVALENCE (ITARG,IT)                                                 SPOT1280
      COMMON /ANSW2/TTR(16),TBR(16),CNTRST(16)                               SPOT1290
      COMMON /ALBED/A0(2),A1(2),IALB(2)                                      SPOT1300
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB                      SPOT1310
      COMMON /CGEOM/COSGM,COSBT,COSIN                                        SPOT1320
      COMMON /COMI1/ISORC,ITARG,IWN,JHL                                      SPOT1330
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                          SPOT1340
      COMMON /CTARG/RTARG,COSX,COSY,COSZ                                     SPOT1350
      COMMON /EMISS/EM(2),TM(2)                                              SPOT1360
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUSPOT1370
      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,         SPOT1380
     +            RRS(16,34),SCOE(16,34),                                    SPOT1390
     +            TRANS(16,5),RADA(16,2),WAVE(16),SS(16),                    SPOT1400
```

```
     1       DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),                        SPOT1410
     2       PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)                     SPOT1420
      COMMON /MO2/ WO(34),RO,TBOUND,JP,IM,ML,IP,JSTOR                          SPOT1430
      COMMON /EM2/W(16),E(16),IL,IKMAX,LENTOR,NLL                              SPOT1440
      COMMON /SPOTLO/ ISPOT,LOREAD,N16                                         SPOT1450
      COMMON /LOWEX/ WPATH(68,16),WLAY(34,16),TBBY(68),TX(16),BETAEX,          SPOT1460
     1 CLDHGT,NCLD                                                             SPOT1470
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),            SPOT1480
     1 BE(16),SINGWV,PF(65),LMAX                                               SPOT1490
      COMMON /LOGIC/L1,L2,L3,L4,L5,L6,L7                                       SPOT1500
      DATA L1,L2,L3,L4,L5,L6,L7/7*.FALSE./                                     SPOT1510
      DATA ITR1,ITR2,ITR3,ITR4/2,3,5,1/                                        SPOT1520
      N16=.TRUE.                                                               SPOT1530
      INBR=16                                                                  SPOT1540
      DUM=1.                                                                   SPOT1550
      ICLMAT=0                                                                 SPOT1560
      LOREAD=.TRUE.                                                            SPOT1570
      ISPOT=.TRUE.                                                             SPOT1580
C     INITIALIZE AND READ INPUT PARAMETERS                                     SPOT1590
      CALL ZERO                                                                SPOT1600
      CALL INDAT(IEMISS,IHAZE,IM,LEN,ML,MODEL,                                 SPOT1610
     1     TBOUND,0,CLDHGT,BETAEX)                                             SPOT1620
C     CHECK FOR ERROR IN INPUT DATA                                            SPOT1630
      IF (IHAZE.EQ.9) IERR=1                                                   SPOT1640
      IF (IERR.EQ.1) RETURN                                                    SPOT1650
      IF (ISORC.NE.2) ISWTCH=1                                                 SPOT1660
C     FIRST CALL IS TO READ LOWTRAN DATA FILE ONLY                             SPOT1670
      CALL LT4M(ALT,DUM,ZENTH,3,0,TRANS(1,1),DUMMY,DUMMY,                      SPOT1680
     1                  IEMISS,LEN,MODEL,VIS,   V1,V2,TGRD,                  SPOT1690
     2     ICLMAT,IERR,NR,IHAZE,MULDV)                                         SPOT1700
      V1=WAVN1                                                                 SPOT1710
      V2=WAVN2                                                                 SPOT1720
      CALL CKER(V1,V2,DV,IV1,IV2,IDV,IERR,MULDV,ISPOT,DUM)                     SPOT1730
      WAVE(1)=10000./V1                                                        SPOT1740
      DO 300 IW=2,INBR                                                         SPOT1750
      V2=V1+20.*FLOAT(MULDV)*FLOAT(IW-1)                                       SPOT1760
      IF (V2.GE.WAVN2) GO TO 400                                               SPOT1770
      WAVE(IW)=10000./V2                                                       SPOT1780
  300 CONTINUE                                                                 SPOT1790
      L1=.TRUE.                                                                SPOT1800
C   MAXIMUM NO. OF WAVELENGTHS                                                 SPOT1810
  400 IWN=IW-1                                                                 SPOT1820
      IF (L1) IWN=INBR                                                         SPOT1830
      NWVL=IWN                                                                 SPOT1840
C                                                                              SPOT1850
C     ARRAY WVL IS USED ONLY IN SUBROUTINE PFUNC. THE WAVELENGTHS              SPOT1860
C     IN THIS ARRAY INCREASE WITH INCREASING ARRAY INDEX. THE                  SPOT1870
C     VALID RANGES FOR VALUES IN THE WVL ARRAY ARE : 0.2-2.0, 3.0-5.0,         SPOT1880
C     AND 8.0-12.0 MICROMETERS.                                                SPOT1890
C                                                                              SPOT1900
      DO 355 JX=1,NWVL                                                         SPOT1910
      INDM=NWVL-JX+1                                                           SPOT1920
  355 WVL(JX)=WAVE(INDM)                                                       SPOT1930
      DO 500 I=1,NL                                                            SPOT1940
      IF (ALT.LE.Z(I)) GO TO 600                                               SPOT1950
  500 CONTINUE                                                                 SPOT1960
      WRITE (IOOUT,3700) ALT,I,Z(I)                                            SPOT1970
      IERR=1                                                                   SPOT1980
      RETURN                                                                   SPOT1990
  600 CONTINUE                                                                 SPOT2000
      IF (I.EQ.1) WRITE (IOOUT,3800) ALT                                       SPOT2010
      IF (I.EQ.1) IERR=1                                                       SPOT2020
      IF (IERR.EQ.1) RETURN                                                    SPOT2030
      JHL=I-1                                                                  SPOT2040
      NLL=NL-1                                                                 SPOT2050
      SANG=TWOPI*(1.0-COS(SANG2*PIRAD))                                        SPOT2060
      CZNTH=COS(ZENTH*PIRAD)                                                   SPOT2070
      SZNTH=SIN(ZENTH*PIRAD)                                                   SPOT2080
      CTHTA=COS(THETA*PIRAD)                                                   SPOT2090
      STHTA=SIN(THETA*PIRAD)                                                   SPOT2100
```

```
      CPHI=COS(PHI*PIRAD)                                                    SPOT2110
      SPHI=SIN(PHI*PIRAD)                                                    SPOT2120
      IF (ISORC.EQ.2) GO TO 900                                              SPOT2130
C     CALCULATE SOURCE TERM ... FOR SUNLIGHT (ISORC=0,3)                     SPOT2140
C                               OR MOONLIGHT (ISORC=1,4)                     SPOT2150
      DO 700 IW=1,IWN                                                        SPOT2160
      SS(IW)=0.0                                                             SPOT2170
      IF (ISORC.EQ.0.OR.ISORC.EQ.3) SS(IW)=SOLARS(WAVE(IW))                  SPOT2180
      IF (ISORC.EQ.1.OR.ISORC.EQ.4) SS(IW)=SMOON(WAVE(IW),PHASE)             SPOT2190
  700 CONTINUE                                                               SPOT2200
C     CALCULATE DIRECT INTENSITY ... FOR SUNLIGHT (ISORC=0,3)                SPOT2210
C                                    OR MOONLIGHT (ISORC=1,4)                SPOT2220
      COSIN=STHTA*CPHI*SZNTH+CTHTA*CZNTH                                     SPOT2230
      ANGIN=ACOS(COSIN)/PIRAD                                                SPOT2240
C     READ PHASE FUNCTION FILE                                               SPOT2250
      CALL INDAT(IEMISS,IHAZE,IM,LEN,ML,MODEL,                               SPOT2260
     1     TBOUND,1,CLDHGT,BETAEX)                                           SPOT2270
      IF (ANGIN.GT.SANG2) L2=.TRUE.                                          SPOT2280
      IF (ZENTH.GT.80.0) L3=.TRUE.                                           SPOT2290
      CALL LT4M(ALT,DUM,ZENTH,3,0,TRANS(1,1),DUMMY,DUMMY,                    SPOT2300
     1                IEMISS,LEN,MODEL,VIS,   V1,V2,TGRD,                  SPOT2310
     2     ICLMAT,IERR,NR,IHAZE,MULDV)                                       SPOT2320
      IF (L2.OR.L3) GO TO 900                                                SPOT2330
      DO 800 IW=1,IWN                                                        SPOT2340
      DIR(IW)=SS(IW)*TRANS(IW,1)                                             SPOT2350
  800 CONTINUE                                                               SPOT2360
  900 IF (ITARG.EQ.0.AND.THETA.GT.90.0) L7=.TRUE.                            SPOT2370
      IF (L7) GO TO 3500                                                     SPOT2380
      IF (ITARG.EQ.0) GO TO 1200                                             SPOT2390
      IF (ITARG.EQ.1) GO TO 1000                                             SPOT2400
C     TARGET...                                                              SPOT2410
      ZTARG=RTARG*CTHTA+ALT                                                  SPOT2420
      COSTX=STHTA*CPHI                                                       SPOT2430
      COSTY=STHTA*SPHI                                                       SPOT2440
      COSTZ=CTHTA                                                            SPOT2450
      COSBT=COSX*SZNTH+COSZ*CZNTH                                            SPOT2460
      COSGM=-(COSTX*COSX+COSTY*COSY+COSTZ*COSZ)                              SPOT2470
      IF (THETA.LE.90.0) GO TO 1100                                          SPOT2480
      COSBTG=CZNTH                                                           SPOT2490
      COSGMG=COS((180.0-THETA)*PIRAD)                                        SPOT2500
      GO TO 1100                                                             SPOT2510
C     GROUND...                                                              SPOT2520
 1000 ZTARG=0.0                                                              SPOT2530
      COSBT=CZNTH                                                            SPOT2540
      COSGM=COS((180.0-THETA)*PIRAD)                                         SPOT2550
      COSBTG=COSBT                                                           SPOT2560
      COSGMG=COSGM                                                           SPOT2570
C     CALCULATE ATMOSPHERIC TRANSMISSION/RADIANCE FOR VARIOUS PATHS          SPOT2580
 1100 IF (COSGM.LE.0.0) L4=.TRUE.                                            SPOT2590
      IF (COSBT.LE.0.0) L5=.TRUE.                                            SPOT2600
      IF (THETA.LE.90.0.AND.ITARG.EQ.1) L6=.TRUE.                            SPOT2610
      IF (L4.OR.L6) GO TO 3500                                               SPOT2620
 1200 CONTINUE                                                               SPOT2630
      IF (THETA.EQ.90.0)                                                     SPOT2640
     1     CALL LT4M(ALT,DUM,1000.0,1,2,TRANS(1,4),RADA(1,1),                SPOT2650
     2    DUMMY,            IEMISS,LEN,MODEL,VIS,   V1,V2,                  SPOT2660
     3    TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)                                  SPOT2670
      IF (THETA.EQ.90.0.AND.ITARG.EQ.2)                                      SPOT2680
     1     CALL LT4M(ALT,DUM,RTARG,1,2,TRANS(1,2),RADA(1,2),                 SPOT2690
     2    DUMMY,            IEMISS,LEN,MODEL,VIS,   V1,V2,                  SPOT2700
     3    TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)                                  SPOT2710
      IF (THETA.EQ.90.0) GO TO 1300                                          SPOT2720
      IF (THETA.LT.90.0)                                                     SPOT2730
     1     CALL LT4M(ALT,DUM,THETA,3,2,TRANS(1,4),RADA(1,2),DUMMY,           SPOT2740
     2               IEMISS,LEN,MODEL,VIS,   V1,V2,TGRD,                     SPOT2750
     3    ICLMAT,IERR,NR,IHAZE,MULDV)                                       SPOT2760
      IF (ITARG.EQ.0) GO TO 2000                                             SPOT2770
      IF (THETA.LT.90.0)                                                     SPOT2780
     1     CALL LT4M(ALT,ZTARG,THETA,2,2,TRANS(1,2),RADA(1,2)                SPOT2790
     2    ,DUMMY,           IEMISS,LEN,MODEL,VIS,   V1,V2,                  SPOT2800
```

```
     3    TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)                                  SPOT2810
      IF (THETA.LT.90.0) GO TO 1300                                         SPOT2820
      CALL LT4M(ALT,ZTARG,THETA,2,2,TRANS(1,2),RADA(1,2),                   SPOT2830
     1     RADG(1),                IEMISS,LEN,MODEL,VIS,    V1,V2,         SPOT2840
     2     TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)                                 SPOT2850
      IF (ITARG.NE.2) ITR4=2                                                SPOT2860
      IF (ITARG.NE.2) GO TO 1300                                            SPOT2870
      IF (ZTARG.LE.0.0) ITR4=2                                              SPOT2880
      IF (ZTARG.LE.0.0) GO TO 1300                                          SPOT2890
      ITR1=4                                                                SPOT2900
      CALL LT4M(ALT,0.0,THETA,2,2,TRANS(1,4),RADA(1,1),                     SPOT2910
     1     RADG(1),                IEMISS,LEN,MODEL,VIS,    V1,V2,         SPOT2920
     2     TGRD,ICLMAT,IERR,NR,IHAZE,MULDV)                                 SPOT2930
 1300 IF (ISORC.LT.2) GO TO 1600                                            SPOT2940
C     CALCULATE UNCOLLIDED EMISSION ... FROM GROUND (ITARG=1)               SPOT2950
C                                        OR TARGET (ITARG=2)                SPOT2960
      DO 1500 IW=1,IWN                                                      SPOT2970
      WAVEM=WAVE(IW)/1.0E+4                                                 SPOT2980
      IF (ITARG.EQ.2) GO TO 1400                                            SPOT2990
      BKG(IW)=BLACK(WAVEM,TGRD)*EM(1)                                       SPOT3000
      RADG(IW)=RADG(IW)*EM(1)*COSGMG                                        SPOT3010
      BK(IW)=BKG(IW)                                                        SPOT3020
      UTEM(IW)=RADG(IW)                                                     SPOT3030
      GO TO 1500                                                            SPOT3040
 1400 BK(IW)=BLACK(WAVEM,TM(2))*EM(2)                                       SPOT3050
      UTEM(IW)=BK(IW)*COSGM*TRANS(IW,2)                                     SPOT3060
      IF (THETA.LE.90.0) GO TO 1500                                         SPOT3070
      BKG(IW)=BLACK(WAVEM,TGRD)*EM(1)                                       SPOT3080
      RADG(IW)=RADG(IW)*EM(1)*COSGMG                                        SPOT3090
 1500 CONTINUE                                                              SPOT3100
 1600 IF (ISORC.EQ.2) GO TO 2900                                            SPOT3110
C     CALCULATE UNCOLLIDED REFLECTANCE ... FROM GROUND (ITARG=1)            SPOT3120
C                                          OR TARGET (ITARG=2)            SPOT3130
      IEMISS=0                                                              SPOT3140
      IF (L5) GO TO 2000                                                    SPOT3150
      HORIZ=ABS(ZTARG-ALT).LT.0.001                                         SPOT3160
      IF(HORIZ) ITR2=1                                                      SPOT3170
      IF(HORIZ) GO TO 1700                                                  SPOT3180
      CALL LT4M(ZTARG,DUM,ZENTH,3,2,TRANS(1,3),DUMMY,DUMMY,                 SPOT3190
     1                IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,                SPOT3200
     2     ICLMAT,IERR,NR,IHAZE,MULDV)                                      SPOT3210
 1700 IF (ZTARG.LE.0.0) ITR3=ITR2                                           SPOT3220
      IF (ZTARG.LE.0.0) GO TO 1800                                          SPOT3230
      IF (THETA.LE.90.0) GO TO 1800                                         SPOT3240
      CALL LT4M(0.0,DUM,ZENTH,3,2,TRANS(1,5),DUMMY,DUMMY,                   SPOT3250
     1                IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,                SPOT3260
     2     ICLMAT,IERR,NR,IHAZE,MULDV)                                      SPOT3270
 1800 DO 1900 IW=1,IWN                                                      SPOT3280
      ALB=ALBEDO(IT)                                                        SPOT3290
      UTRF(IW)=SS(IW)*COSBT*ALB*TRANS(IW,2)*TRANS(IW,ITR2)                  SPOT3300
      IF (THETA.LE.90.0) GO TO 1900                                         SPOT3310
      UERF(IW)=SS(IW)*COSBTG*ALBEDO(1)*TRANS(IW,ITR1)*                      SPOT3320
     1     TRANS(IW,ITR3)                                                   SPOT3330
 1900 CONTINUE                                                              SPOT3340
C     CALCULATE SINGLE-SCATTERED PATH RADIANCE ...                          SPOT3350
C               FROM SUNLIGHT (ISORC=0,3)                                   SPOT3360
C                 OR MOONLIGHT (ISORC=1,4)                                  SPOT3370
2000  CALL COEFS(P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX)                      SPOT3380
      IF (IERR.EQ.1) RETURN                                                 SPOT3390
      IF (ITARG.EQ.0) GO TO 2700                                            SPOT3400
      CALL PATHRD(CTHTA,ALT,RTARG,1,IHAZE,NR,                               SPOT3410
     1     IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)              SPOT3420
      DO 2100 IW=1,IWN                                                      SPOT3430
 2100 PATHR2(IW)=SS(IW)*PATHR(IW)                                           SPOT3440
      IF (ABS(CTHTA).LE.1.0E-3) GO TO 2600                                  SPOT3450
      IF (CTHTA.LT.0.0) GO TO 2300                                          SPOT3460
 2200 ZZ=Z(NLL)                                                             SPOT3470
      IQ=2                                                                  SPOT3480
      RT=(Z(NLL)-ZTARG)/CTHTA                                               SPOT3490
      GO TO 2800                                                            SPOT3500
```

```
 2300 IF (ZTARG.GT.0.0) GO TO 2500                                      SPOT3510
      DO 2400 IW=1,IWN                                                  SPOT3520
 2400 PATHR(IW)=0.0                                                     SPOT3530
      GO TO 2900                                                        SPOT3540
 2500 Z2=0.0                                                            SPOT3550
      IO=1                                                              SPOT3560
      RT=-ZTARG/CTHTA                                                   SPOT3570
      GO TO 2800                                                        SPOT3580
 2600 Z2=ZTARG                                                          SPOT3590
      RT=1000.0                                                         SPOT3600
      IO=3                                                              SPOT3610
      GO TO 2800                                                        SPOT3620
 2700 IEMISS=0                                                          SPOT3630
      ZTARG=ALT                                                         SPOT3640
      IF (ABS(CTHTA).LE.1.0E-03) GO TO 2600                             SPOT3650
      GO TO 2200                                                        SPOT3660
 2800 CALL PATHRD(CTHTA,ZTARG,RT,IO,IHAZE,NR,                           SPOT3670
     1      IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)         SPOT3680
C     CALCULATE BACKGROUND AND TOTAL INTENSITY, PLUS                    SPOT3690
C                CONTRAST RATIO                                         SPOT3700
 2900 DO 3100 IW=1,IWN                                                  SPOT3710
      PATHR(IW)=PATHR2(IW)+SS(IW)*PATHR(IW)                             SPOT3720
      TTR(IW)=PATHR2(IW)+UTRF(IW)+RADA(IW,2)+UTEM(IW)                   SPOT3730
      TBR(IW)=PATHR(IW)+UERF(IW)+RADA(IW,ITR4)+RADG(IW)                 SPOT3740
      DIF=TTR(IW)-TBR(IW)                                               SPOT3750
      IF (TBR(IW).GT.0.0) GO TO 3000                                    SPOT3760
      IF (TTR(IW).EQ.TBR(IW)) CNTRST(IW)=0.0                            SPOT3770
      IF (TTR(IW).NE.TBR(IW)) CNTRST(IW)=1.0E30                         SPOT3780
      GO TO 3100                                                        SPOT3790
 3000 CNTRST(IW)=DIF/TBR(IW)                                            SPOT3800
 3100 CONTINUE                                                          SPOT3810
C     CALCULATE TOTAL RADIANCES INTEGRATED OVER DETECTOR RESPONSE       SPOT3820
      DV2=DV*0.5                                                        SPOT3830
      SUMRPF=0.                                                         SPOT3840
      DO 3200 IW=1,IWN                                                  SPOT3850
      NW=10000./WAVE(IW)                                                SPOT3860
      W2=10000./(FLOAT(NW)-DV2)                                         SPOT3870
      W1=10000./(FLOAT(NW)+DV2)                                         SPOT3880
      IF (IW.EQ.1) W2=WAVE(1)                                           SPOT3890
      IF (IW.EQ.IWN) W1=WAVE(IWN)                                       SPOT3900
      RESPFN=RESFN(NR,WAVE(IW))                                         SPOT3910
      SUMRPF=SUMRPF+RESPFN                                              SPOT3920
      DW=(W2-W1)*RESPFN                                                 SPOT3930
      TOT(1)=TOT(1)+DW*UTEM(IW)                                         SPOT3940
      TOT(2)=TOT(2)+DW*UTRF(IW)                                         SPOT3950
      TOT(3)=TOT(3)+DW*RADA(IW,2)                                       SPOT3960
      TOT(4)=TOT(4)+DW*PATHR2(IW)                                       SPOT3970
      TOT(5)=TOT(5)+DW*TTR(IW)                                          SPOT3980
      TOT(6)=TOT(6)+DW*RADG(IW)                                         SPOT3990
      TOT(7)=TOT(7)+DW*UERF(IW)                                         SPOT4000
      TOT(8)=TOT(8)+DW*RADA(IW,ITR4)                                    SPOT4010
      TOT(9)=TOT(9)+DW*PATHR(IW)                                        SPOT4020
      TOT(10)=TOT(10)+DW*TBR(IW)                                        SPOT4030
 3200 TOT(11)=TOT(11)+DW*DIR(IW)                                        SPOT4040
      IF (NR.NE.1) SUMRPF=1.                                            SPOT4050
      DO 3250 I=1,11                                                    SPOT4060
 3250 TOT(I)=TOT(I)/SUMRPF                                              SPOT4070
      IF (TOT(10).GT.0.0) GO TO 3300                                    SPOT4080
      IF (TOT(5).EQ.TOT(10)) TOT(12)=0.0                                SPOT4090
      IF (TOT(5).NE.TOT(10)) TOT(12)=1.0E30                             SPOT4100
      GO TO 3400                                                        SPOT4110
 3300 TOT(12)=(TOT(5)-TOT(10))/TOT(10)                                  SPOT4120
 3400 CONTINUE                                                          SPOT4130
 3500 CALL OUTPUT(MODEL,IHAZE,CLDHGT)                                   SPOT4140
      RETURN                                                            SPOT4150
C                                                                       SPOT4160
 3700 FORMAT (1H ,11H ALTITUDE (,F10.3,17H) GREATER THAN Z(,            SPOT4170
     1        I2,2H)=,F10.3,27H  CONTROL RETURNED TO MAIN               SPOT4180
     2        10HFROM SPOT.)                                           SPOT4190
 3800 FORMAT (1H ,11H ALTITUDE (,F10.3,16H) LESS THAN ZERO,             SPOT4200
```

```
1 37H  CONTROL RETURNED TO MAIN FROM SPOT.)                    SPOT4210
 END                                                           SPOT4220
```

```
      FUNCTION ALBEDO(I)                                                    ALB00010
      COMMON /ALBED/A0(2),A1(2),IALB(2)                                     ALB00020
      COMMON /CGEOM/COSGM,COSBT,COSIN                                       ALB00030
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                         ALB00040
C     CALCULATE ALBEDO FOR GROUND (ITARG=1) OR TARGET (ITARG=2)             ALB00050
C     IALB(I) = 0   LAMBERTIAN REFLECTION SURFACE                           ALB00060
C               1   ISOTROPIC                                               ALB00070
      A=A0(I)+A1(I)*COSBT                                                   ALB00080
      IF (IALB(I).EQ.0) ALBEDO=A*COSGM/PI                                   ALB00090
      IF (IALB(I).EQ.1) ALBEDO=A/TWOPI                                      ALB00100
      RETURN                                                                ALB00110
      END                                                                   ALB00120
```

```
      FUNCTION BLACK(W,T)                                               BLA00010
C     BLACK(W,T) = PLANCK FUNCTION (UNITS: WATT PER SQUARE METER PER    BLA00020
C     MICROMETER PER STERADIAN), GIVEN WAVELENGTH W IN CM AND TEMP-     BLA00030
C     ERATURE T IN K                                                    BLA00040
C                                                                       BLA00050
C     EXP OVERFLOW PROTECTION                                           BLA00060
C                                                                       BLA00070
      ARG=1.43879/(W*T)                                                 BLA00080
      IF(ARG.LT.88.) GO TO 1                                            BLA00090
      BLACK=0.0                                                         BLA00100
      RETURN                                                            BLA00110
    1 BLACK=1.19106E-12/(W**5*(EXP(ARG)-1.0))                           BLA00120
      RETURN                                                            BLA00130
      END                                                               BLA00140
```

```
      SUBROUTINE COEFS(P,T,VIS,IHAZE,ZTARG,NCLD,IERR,BETAEX)              COE00010
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB                   COE00020
      COMMON /CTARG/RTARG,COSX,COSY,COSZ                                  COE00030
      COMMON /MO1/DUMMIE(715),MHOLD,NL,                                   COE00040
     +            RRS(16,34),SCOE(16,34),                                COE00050
     +            TRANS(16,5),RADA(16,2),WAVE(16),SS(16),                COE00060
     1      DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),                    COE00070
     2      PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)                 COE00080
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),       COE00090
     1 BE(16),SINGWV,PF(65)                                               COE00100
      COMMON /COMI1/ISORC,ITARG,IWN,JHL                                   COE00110
      COMMON /CONST/ PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                      COE00120
      DIMENSION P(34),T(34)                                               COE00130
C     CALCULATE THE WAVELENGTH-DEPENDENT CONSTANT PRESSURE COEFFICIENTS    COE00140
C     FOR MOLECULAR SCATTERING.                                           COE00150
C   LOOP OVER LAYERS                                                      COE00160
      DO 600 I=1,NL                                                       COE00170
      PS=P(I)/1013.0                                                      COE00180
      TS=CDEGK/T(I)                                                       COE00190
      RSCAT=PS*TS                                                         COE00200
C   LOOP OVER WAVELENGTHS                                                 COE00210
      DO 600 IW=1,IWN                                                     COE00220
      RAYS=0.0                                                            COE00230
      NW=10000./WAVE(IW)                                                  COE00240
C     RAYLEIGH SCATTERING = 0.FOR WAVELENGTH GT 3.33 UM                   COE00250
      IF (NW.LT.3000) GO TO 200                                           COE00260
      WN=FLOAT(NW)                                                        COE00270
      RAYS=RSCAT*(WN**4)/(9.67578E+18-1.11836E+09*WN**2)                  COE00280
200   CONTINUE                                                            COE00290
      AEXT=1.                                                             COE00300
      AABS=1.                                                             COE00310
      IF (IHAZE.EQ.0.OR.I.NE.(NCLD-1)) GO TO 1                            COE00320
      EXT55=3.912/VIS                                                     COE00330
C     UPPER LIMIT OF 500 METERS VERTICAL DISTANCE FOR XSCALE              COE00340
      ZTALT=ZTARG/ALT                                                     COE00350
      IF (ABS(ZTALT-1.).LT..01) RNG=RTARG                                 COE00360
      IF ((ZTARG.GT.ALT).AND.(RTARG.GT..5/COS(THETA*PIRAD)))              COE00370
     1  RNG=.5/COS(THETA*PIRAD)                                           COE00380
      IF (ZTARG.LT.RTARG.AND.(RTARG.GT..5/COS((180.-THETA)*PIRAD)))       COE00390
     1      RNG=.5/COS((180.-THETA)*PIRAD)                                COE00400
      IF (ITARG.EQ.0.AND.(RTARG.GT..5/COS(THETA*PIRAD)))                  COE00410
     1      RNG=.5/COS(THETA*PIRAD)                                       COE00420
      ISLANT=1                                                            COE00430
      IF (ABS(ZTALT-1.).LT..01) ISLANT=0                                  COE00440
C     CALL XSCALE FOR TOTAL PATH LENGTH TRANSMISSION FOR AEROSOL          COE00450
      CALL XSCALE(WAVE(IW),88.,EXT55,XSTRN,IERR,ISLANT,IHAZE,RNG,THETA)   COE00460
      IF (IERR.EQ.1) RETURN                                               COE00470
      AEXT=-ALOG(XSTRN)/RNG                                               COE00480
C     USER SUPPLIED COEFF(IHAZE=7), OR READ FROM PFN DATA FILE(IHAZE=8)   COE00490
      IF (IHAZE.EQ.7) AEXT=BETAEX                                         COE00500
      IF (IHAZE.EQ.8) AEXT=BE(IW)                                         COE00510
      IF (WAVE(IW).LT.2.) AABS=1.                                         COE00520
      IF (WAVE(IW).GE.3..AND.WAVE(IW).LE.5.) AABS=AEXT*.2                 COE00530
      IF (WAVE(IW).GE.8..AND.WAVE(IW).LE.12.) AABS=AEXT*.45               COE00540
1     CONTINUE                                                            COE00550
      SCOE(IW,I)=AEXT-AABS+RAYS                                           COE00560
C     CHECK FOR NO AEROSOL PRESENT                                        COE00570
      IF(SCOE(IW,I).LT.1.E-20)RRS(IW,I)=1.0                               COE00580
C     AEROSOL AND RAYLEIGH PRESENT                                        COE00590
      IF(SCOE(IW,I).GE.1.E-20)RRS(IW,I)=RAYS/SCOE(IW,I)                   COE00600
C     CHECK FOR NO RAYLEIGH SCATTERING                                    COE00610
      IF (RAYS.LT.1.E-20) RRS(IW,I)=0.0                                   COE00620
  600 CONTINUE                                                            COE00630
      RETURN                                                              COE00640
      END                                                                 COE00650
```

```
      SUBROUTINE DIAG                                                   DIAG0010
C        THIS SUBROUTINE PRODUCES DIAGNOSTIC COMMENTARY FOR THE         DIAG0020
C        SPOT PROGRAM.                                                  DIAG0030
C                                                                       DIAG0040
C        CALLING SEQUENCE:  CALL DIAG                                   DIAG0050
C                                                                       DIAG0060
C        EXTERNAL VARIABLES REQUIRED:                                   DIAG0070
C                 THETA (COMMON BLOCK BKDAT)                            DIAG0080
C                 HLIGHT,HTYPE (COMMON BLOCK HOLRTH)                    DIAG0090
C                 L1,L2,L3,L4,L5,L6 (COMMON BLOCK LOGIC)                DIAG0100
C                                                                       DIAG0110
      LOGICAL L1,L2,L3,L4,L5,L6,L7                                      DIAG0120
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB                 DIAG0130
      COMMON /HOLRTH/                HITARG(8,3),HISORC(6,5),           DIAG0140
     1       HMODEL(5,6),HLIGHT(3),HMNLT(3),HSNLT(3),                   DIAG0150
     2       HTRGT(2),HTYPE(2),HGRND(2)                                 DIAG0160
      COMMON /LOGIC/L1,L2,L3,L4,L5,L6,L7                                DIAG0170
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUDIAG0180
      DATA I/1/                                                         DIAG0190
C        WRITE HEADING.                                                 DIAG0200
      WRITE (IOOUT,900)                                                 DIAG0210
C        IF ERRORS, GO TO 5; OTHERWISE PRINT CLEAN RUN MESSAGE.         DIAG0220
      IF (L1.OR.L2.OR.L3.OR.L4.OR.L5.OR.L6.OR.L7) GO TO 100             DIAG0230
      WRITE (IOOUT,1000)                                                DIAG0240
      GO TO 700                                                         DIAG0250
C        HERE IF THERE WERE ERRORS                                      DIAG0260
  100 WRITE (IOOUT,1100)                                                DIAG0270
      IF (.NOT.L1) GO TO 200                                            DIAG0280
      WRITE (IOOUT,1200) I                                              DIAG0290
      I=I+1                                                             DIAG0300
  200 IF (.NOT.L2) GO TO 300                                            DIAG0310
      WRITE (IOOUT,1300) I,HLIGHT                                       DIAG0320
      I=I+1                                                             DIAG0330
  300 IF (.NOT.L3) GO TO 400                                            DIAG0340
      WRITE (IOOUT,1400) I,HLIGHT                                       DIAG0350
      I=I+1                                                             DIAG0360
  400 IF (.NOT.L4) GO TO 500                                            DIAG0370
      WRITE (IOOUT,1500) I,HTYPE                                        DIAG0380
      I=I+1                                                             DIAG0390
  500 IF (.NOT.L5) GO TO 600                                            DIAG0400
      WRITE (IOOUT,1600) I,HLIGHT                                       DIAG0410
      I=I+1                                                             DIAG0420
  600 IF (.NOT.L6) GO TO 700                                            DIAG0430
      WRITE (IOOUT,1700) I,THETA                                        DIAG0440
      I=I+1                                                             DIAG0450
  700 IF (.NOT.L7) GO TO 800                                            DIAG0460
      WRITE (IOOUT,1800) I,THETA                                        DIAG0470
C        WRITE FOOTING.                                                 DIAG0480
  800 WRITE (IOOUT,1900)                                                DIAG0490
C                                                                       DIAG0500
  900 FORMAT (1H0,21X,90(1H*),3(/,21X,1H*,88X,1H*))                     DIAG0510
 1000 FORMAT (21X,1H*,28X,29HNO SPOT DIAGNOSTICS FOR THIS               DIAG0520
     1        3HRUN,28X,1H*,/,21X,1H*,28X,8H-- ---- ,11(1H-)           DIAG0530
     2        ,13H --- ---- ---,28X,1H*)                                DIAG0540
 1100 FORMAT (21X,1H*,28X,25HSPOT DIAGNOSTIC MESSAGES                   DIAG0550
     1        7HFOLLOW:,28X,1H*,/,21X,1H*,28X,5H---- ,10(1H-           DIAG0560
     2        ),1X,8(1H-),1X,6(1H-),29X,1H*,2(/,21X,1H*,88X,           DIAG0570
     3        1H*))                                                     DIAG0580
 1200 FORMAT (21X,1H*,9X,I1,30H.  NUMBER OF WAVELENGTHS (IWN)           DIAG0590
     1        9H EXCEEDS ,31HALLOWABLE DIMENSIONS; IWN RESET            DIAG0600
     2        ,8X,1H*,/,21X,1H*,13X,6HTO 16.,69X,1H*,/,21X,            DIAG0610
     3        1H*,88X,1H*)                                              DIAG0620
 1300 FORMAT (21X,1H*,9X,I1,12H.  NO DIRECT,3A4,9HINCIDENT               DIAG0630
     1        7HWITHIN ,25HRECEIVER'S FIELD OF VIEW.,13X,1H*            DIAG0640
     2        ,/,21X,1H*,88X,1H*)                                       DIAG0650
 1400 FORMAT (21X,1H*,9X,I1,25H.  ANGLE OF INCIDENCE FOR,               DIAG0660
     1        3A4,8HGREATER ,21HTHAN 80.0 DEGREES; NO,12X,              DIAG0670
     2        1H*,/,21X,1H*,13X,26HCALCULATIONS WILL BE MADE            DIAG0680
     3        14HFOR ITARG = 0.,35X,1H*,/,21X,1H*,88X,1H*)              DIAG0690
 1500 FORMAT (21X,1H*,9X,I1,1H.,1X,2A4,14HDOES NOT FACE                 DIAG0700
```

```
     1          9HRECEIVER.,45X,1H*,/,21X,1H*,88X,1H*)                    DIAG0710
1600 FORMAT (21X,1H*,9X,I1,1H.,3A4,22HILLUMINATES BACK SIDE               DIAG0720
     1       3HOF ,7HTARGET.,33X,1H*,/,21X,1H*,88X,1H*)                   DIAG0730
1700 FORMAT (21X,1H*,9X,I1,30H.  THETA LESS THAN 90 DEGREES               DIAG0740
     1       10HAND ITARG 13H = 1; THETA =,F6.4,19X,1H*,/                 DIAG0750
     2       ,21X,1H*,88X,1H*)                                            DIAG0760
1800 FORMAT (21X,1H*,9X,I1,25H.  THETA GREATER THAN 90                    DIAG0770
     1       13HDEGREES AND  19HITARG = 0; THETA = ,F7.4,                 DIAG0780
     2       13X,1H*,/,21X,1H*,88X,1H*)                                   DIAG0790
1900 FORMAT (2(21X,1H*,88X,1H*,/),21X,90(1H*))                            DIAG0800
     RETURN                                                               DIAG0810
     END                                                                  DIAG0820
```

```
      SUBROUTINE INDAT(IEMISS,IHAZE,IM,LEN,ML,                          IND00010
     1           MODEL,TBOUND,ISW,CLDHGT,BETAEX)                        IND00020
      COMMON /ALBED/A0(2),A1(2),IALB(2)                                 IND00030
      COMMON /MO1/DUMMIE(715),MHOLD,NLHOLD,DUMMYS(1088),                IND00040
     +           TRANS(16,5),RADA(16,2),WAVE(16),SS(16),                IND00050
     1     DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),                   IND00060
     2     PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)                IND00070
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB                 IND00080
      COMMON /COMI1/ISORC,ITARG,IWN,JHL                                 IND00090
      COMMON /CONST/PI,PI2,PIRAD,TWOPI,TORRMB,CDEGK                     IND00100
      COMMON /CTARG/RTARG,COSX,COSY,COSZ                                IND00110
      COMMON /EMISS/EM(2),TM(2)                                         IND00120
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUIND00130
      COMMON /GEOMET/PTS(15),IGEOSW                                     IND00140
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),     IND00150
     1 BE(16),SINGWV,PF(65),LMAX                                        IND00160
      DIMENSION DAT(7),TYPE(7)                                          IND00170
      DATA TYPE /4HENVR,4HEMIS,4HATM ,4HTARG,4HREFL,4HSENS,4HGO  /      IND00180
      DATA IZERO /0/                                                    IND00190
C                                                                       IND00200
CCC   SUBROUTINE INDAT IS CALLED UPON TO A) READ INPUT CONTROL          IND00210
CCC   PARAMETERS, WITH CARD ORDER INDEPENDENT INPUT (SEE SPOT           IND00220
CCC   FOR MORE DETAIL) AND, B) TO READ VALUES OF THE PHASE              IND00230
CCC   FUNCTION AT SPECIFIED ID AND WAVELENGTH.                          IND00240
CCC   NOTE.  IF ISORC LT 2 OR ITARG LT 1 EM(1),TM(1),EM(2),TM(2)        IND00250
CCC          ARE NOT NEEDED.                                           IND00260
CCC          IF ITARG LT 1 A0( ),A1( ),IALB( ) ARE NOT NEEDED.         IND00270
CCC                                                                    IND00280
C     ISW=1 ON SECOND CALL TO INDAT                                     IND00290
      IF (ISW.EQ.1) GO TO 400                                           IND00300
      IF (IZERO.GT.0) GO TO 9                                           IND00310
      ISORC=0                                                           IND00320
      ITARG=0                                                           IND00330
      MODEL=0                                                           IND00340
      IHAZE=0                                                           IND00350
      NLAM=0                                                            IND00360
      EM(1)=0.                                                          IND00370
      TM(1)=0.                                                          IND00380
      EM(2)=0.                                                          IND00390
      TM(2)=0.                                                          IND00400
      ZENTH=0.                                                          IND00410
      CLDHGT=0.                                                         IND00420
      PHASE=0.                                                          IND00430
      RTARG=0.                                                          IND00440
      COSX=0.                                                           IND00450
      COSY=0.                                                           IND00460
      COSZ=0.                                                           IND00470
      A0(1)=0.                                                          IND00480
      A1(1)=0.                                                          IND00490
      IALB(1)=0                                                         IND00500
      A0(2)=0.                                                          IND00510
      A1(2)=0.                                                          IND00520
      IALB(2)=0                                                         IND00530
      ALT=0.                                                            IND00540
      THETA=0.                                                          IND00550
      PHI=0.                                                            IND00560
      SANG2=0.                                                          IND00570
      IZERO=1                                                           IND00580
9     CONTINUE                                                          IND00590
      WRITE(IOOUT,600)                                                  IND00600
  600 FORMAT(1H0,'SPOT CONTROL CARDS READ FOR THIS RUN:'/)              IND00610
      DO 10 I=1,7                                                       IND00620
      READ (IOIN,11) (DAT(J),J=1,7)                                     IND00630
11    FORMAT (A4,6X,7E10.4)                                             IND00640
      WRITE(IOOUT,610)(DAT(J),J=1,7)                                    IND00650
  610 FORMAT(1H ,A4,6X,7E10.4)                                          IND00660
      IF (DAT(1).EQ.TYPE(1)) GO TO 1                                    IND00670
      IF (DAT(1).EQ.TYPE(2)) GO TO 2                                    IND00680
      IF (DAT(1).EQ.TYPE(3)) GO TO 3                                    IND00690
      IF (DAT(1).EQ.TYPE(4)) GO TO 4                                    IND00700
```

```
      IF (DAT(1).EQ.TYPE(5)) GO TO 5                                    IND00710
      IF (DAT(1).EQ.TYPE(6)) GO TO 6                                    IND00720
      IF (DAT(1).EQ.TYPE(7)) GO TO 7                                    IND00730
C     ERROR RETURN                                                      IND00740
      WRITE (IOOUT,101)                                                 IND00750
101   FORMAT(1H ,48HINCORRECT INPUT CARD FOR SPOT, CONTROL RETURNED ,   IND00760
     1 18HTO MAIN FROM INDAT)                                           IND00770
      IHAZE=9                                                           IND00780
      RETURN                                                            IND00790
C     OPERATING ENVIRONMENT                                             IND00800
1     ISORC=IFIX(DAT(2))                                                IND00810
      ITARG=IFIX(DAT(3))                                                IND00820
      IHAZE=IFIX(DAT(4))                                                IND00830
      MODEL=IFIX(DAT(5))                                                IND00840
      NLAM=IFIX(DAT(6))                                                 IND00850
      ID=IFIX(DAT(7))                                                   IND00860
      GO TO 8                                                           IND00870
C     EMISSIVITY AND TEMPERATURE OF GROUND AND TARGET, RESPECTIVELY     IND00880
2     EM(1)=DAT(2)                                                      IND00890
      TM(1)=DAT(3)                                                      IND00900
      EM(2)=DAT(4)                                                      IND00910
      TM(2)=DAT(5)                                                      IND00920
      TBOUND=TM(1)                                                      IND00930
      GO TO 8                                                           IND00940
C     INCIDENT ZENITH ANGLE OF RADIATION, CLOUD BOTTOM HEIGHT,          IND00950
C     PHASE ANGLE OF MOON, OPTIONAL EXTN COEF (VALID WHEN IHAZE=8)      IND00960
3     ZENTH=DAT(2)                                                      IND00970
      CLDHGT=DAT(3)                                                     IND00980
      PHASE=DAT(4)                                                      IND00990
      BETAEX=DAT(5)                                                     IND01000
      GO TO 8                                                           IND01010
C     TARGET PROPERTIES                                                 IND01020
4     RTARG=DAT(2)                                                      IND01030
      COSX=DAT(3)                                                       IND01040
      COSY=DAT(4)                                                       IND01050
      COSZ=DAT(5)                                                       IND01060
      GO TO 8                                                           IND01070
C     ALBEDO COEFFICIENTS AND TYPE OF REFLECTION SURFACE FOR            IND01080
C     GROUND AND TARGET, RESPECTIVELY                                   IND01090
5     A0(1)=DAT(2)                                                      IND01100
      A1(1)=DAT(3)                                                      IND01110
      IALB(1)=IFIX(DAT(4))                                              IND01120
      A0(2)=DAT(5)                                                      IND01130
      A1(2)=DAT(6)                                                      IND01140
      IALB(2)=IFIX(DAT(7))                                              IND01150
      GO TO 8                                                           IND01160
C     SENSOR CHARACTERISTICS                                            IND01170
6     ALT=DAT(2)                                                        IND01180
      THETA=DAT(3)                                                      IND01190
      PHI=DAT(4)                                                        IND01200
C                                                                       IND01210
C   EXPECTING INPUT AZIMUTH IN METEOROLOGICAL CONVENTION                IND01220
C   (I.E., N = 0 DEG, E = 90 DEG, S = 180 DEG, W = 270 DEG),            IND01230
C   SO CONVERT TO MATHEMATICAL CONVENTION FOR PURPOSES OF               IND01240
C   SPOT (ASSUMING Y-AXIS IS POSITIVE NORTHWARD, X-AXIS POSI-           IND01250
C   TIVE EASTWARD).                                                     IND01260
C                                                                       IND01270
      PHI=90.-PHI                                                       IND01280
      SANG2=DAT(5)                                                      IND01290
8     CONTINUE                                                          IND01300
10    CONTINUE                                                          IND01310
7     CONTINUE                                                          IND01320
      IEMISS=0                                                          IND01330
      IM=0                                                              IND01340
      LEN=0                                                             IND01350
      ML=0                                                              IND01360
      IF (ISORC.GT.1) IEMISS=1                                          IND01370
      COSX=COS(COSX*PIRAD)                                              IND01380
      COSY=COS(COSY*PIRAD)                                              IND01390
      COSZ=COS(COSZ*PIRAD)                                              IND01400
```

```
C       GEOMETRICAL OPTION                                              IND01410
        IF(IGEOSW.NE.1)GO TO 311                                        IND01420
        RTARG=SQRT((PTS(4)-PTS(1))**2+(PTS(5)-PTS(2))**2+               IND01430
       +(PTS(6)-PTS(3))**2)                                             IND01440
        THETA=ACOS((PTS(3)-PTS(6))/RTARG)                               IND01450
        RTDCON=1.0/PIRAD                                                IND01460
        THETA=THETA*RTDCON                                              IND01470
        ALT=PTS(6)                                                      IND01480
        DELX=PTS(1)-PTS(4)                                              IND01490
        DELY=PTS(2)-PTS(5)                                              IND01500
        HDIS=SQRT(DELX**2+DELY**2)                                      IND01510
        PHI=ACOS(DELX/HDIS)                                             IND01520
        PHI=PHI*RTDCON                                                  IND01530
        IF(DELY.LT.0.0)PHI=360.0-PHI                                    IND01540
  311   CONTINUE                                                        IND01550
        IF (ITARG.EQ.1) RTARG=ALT/ABS(COS(THETA*PIRAD))                 IND01560
        IF (ITARG.EQ.0) RTARG=1000.0                                    IND01570
        RETURN                                                          IND01580
  400   CONTINUE                                                        IND01590
        REWIND IPHFUN                                                   IND01600
        DO 500 I=1,IWN                                                  IND01610
500     PF(I)=0.                                                        IND01620
        IF (NLAM.NE.0) CALL PFUNC(ID)                                   IND01630
        RETURN                                                          IND01640
        END                                                             IND01650
```

```
      SUBROUTINE OUTPUT(MODEL,IHAZE,CLDHGT)                                  OUTP0010
      LOGICAL LMNLT                                                          OUTP0020
      COMMON /MO1/ DUMMIE(715),MHOLD,NLHOLD,DUMMYS(1088),                    OUTP0030
     +             TRANS(16,5),RADA(16,2),WAVE(16),SS(16),                   OUTP0040
     1       DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),                      OUTP0050
     2       PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)                   OUTP0060
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB                      OUTP0070
      COMMON /COMI1/ISORC,ITARG,IWN,JHL                                      OUTP0080
      COMMON /ANSW2/TTR(16),TBR(16),CNTRST(16)                               OUTP0090
      COMMON /HOLRTH/              HITARG(8,3),HISORC(6,5),                  OUTP0100
     1       HMODEL(5,6),HLIGHT(3),HMNLT(3),HSNLT(3),                        OUTP0110
     2       HTRGT(2),HTYPE(2),HGRND(2)                                      OUTP0120
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTU     OUTP0130
      DATA HGRND/4H GRO,4HUND /                                              OUTP0140
      DATA HISORC/4HSUNL,4HIGHT,4H ONL,4HY   ,4H    ,4H    ,                 OUTP0150
     1      4HMOON,4HLIGH,4HT ON,4HLY  ,4H    ,4H    ,4HEMIS,                OUTP0160
     2      4HSION,4H ONL,4HY   ,4H    ,4H    ,4HSUNL,4HIGHT,                 OUTP0170
     3      4H AND,4H EMI,4HSSIO,4HN   ,4HMOON,4HLIGH,4HT AN,                OUTP0180
     4      4HD EM,4HISSI,4HON  /                                            OUTP0190
      DATA HITARG/4HNO R,4HEFLE,4HCTAN,4HCE  ,4H    ,4H    ,                 OUTP0200
     1      4H    ,4H    ,4HGROU,4HND R,4HEFLE,4HCTAN,4HCE /,                OUTP0210
     2      4H EMI,4HSSIO,4HN   ,4HTARG,4HET R,4HEFLE,4HCTAN,                OUTP0220
     3      4HCE /,4H EMI,4HSSIO,4HN   /                                     OUTP0230
      DATA HMODEL/4HTROP,4HICAL,4H    ,4H    ,4H    ,4HMIDA,                OUTP0240
     1      4HLTIT,4HUDE ,4HSUMM,4HER  ,4HMIDA,4HLTIT,4HUDE ,                OUTP0250
     2      4HWINT,4HER  ,4HSUBA,4HRCTI,4HC SU,4HMMER,4H    ,                OUTP0260
     3      4HSUBA,4HRCTI,4HC WI,4HNTER,4H    ,4H1962,4H U.S,                OUTP0270
     4      4H. ST,4HANDA,4HRD  /                                            OUTP0280
      DATA HMNLT/4H  MO,4HONLI,4HGHT /                                       OUTP0290
      DATA HSNLT/4H  SU,4HNLIG,4HHT  /                                       OUTP0300
      DATA HTRGT/4H TAR,4HGET /                                              OUTP0310
      DATA HTYPE/4H    ,4H    /                                              OUTP0320
      DATA LMNLT/.FALSE./                                                    OUTP0330
      IF (ISORC.NE.0.AND.ISORC.NE.3) GO TO 200                               OUTP0340
      DO 100 I=1,3                                                           OUTP0350
  100 HLIGHT(I)=HSNLT(I)                                                     OUTP0360
      GO TO 400                                                              OUTP0370
  200 DO 300 I=1,3                                                           OUTP0380
  300 HLIGHT(I)=HMNLT(I)                                                     OUTP0390
      LMNLT=.TRUE.                                                           OUTP0400
  400 IF (ITARG.EQ.0) GO TO 600                                              OUTP0410
      IF (ITARG.EQ.2) GO TO 500                                              OUTP0420
      HTYPE(1)=HGRND(1)                                                      OUTP0430
      HTYPE(2)=HGRND(2)                                                      OUTP0440
      GO TO 600                                                              OUTP0450
  500 HTYPE(1)=HTRGT(1)                                                      OUTP0460
      HTYPE(2)=HTRGT(2)                                                      OUTP0470
  600 CALL DIAG                                                              OUTP0480
      IF (MODEL.GT.7) GO TO 700                                              OUTP0490
      WRITE (IOOUT,1400) ISORC,(HISORC(I,ISORC+1),I=1,6),                    OUTP0500
     1       ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,                          OUTP0510
     2       (HMODEL(I,MODEL),I=1,5),IHAZE                                   OUTP0520
      GO TO 750                                                              OUTP0530
  700 IF(MODEL.EQ.8)WRITE (IOOUT,1450) ISORC,(HISORC(I,ISORC+1),I=1,6),      OUTP0540
     1       ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE                     OUTP0550
      IF(MODEL.EQ.9)WRITE (IOOUT,1500) ISORC,(HISORC(I,ISORC+1),I=1,6),      OUTP0560
     1       ITARG,(HITARG(I,ITARG+1),I=1,8),MODEL,IHAZE                     OUTP0570
  750 IF (IHAZE.GT.0) WRITE (IOOUT,1600) CLDHGT                              OUTP0580
  800 WRITE (IOOUT,1700)                                                     OUTP0590
      IF (LMNLT) WRITE (IOOUT,1800) PHASE                                    OUTP0600
      WRITE (IOOUT,1900) HLIGHT                                              OUTP0610
      DO 900 I=1,IWN                                                         OUTP0620
      NW=10000./WAVE(I)                                                      OUTP0630
  900 WRITE (IOOUT,2000) WAVE(I),NW,SS(I),BK(I),BKG(I)                       OUTP0640
      WRITE (IOOUT,2100)                                                     OUTP0650
      DO 1000 I=1,IWN                                                        OUTP0660
      NW=10000./WAVE(I)                                                      OUTP0670
 1000 WRITE (IOOUT,2200) WAVE(I),NW,UTEM(I),UTRF(I),                         OUTP0680
     1       RADA(I,2),PATHR2(I),TTR(I)                                      OUTP0690
      WRITE (IOOUT,2300)                                                     OUTP0700
```

```
      DO 1100 I=1,IWN                                                   OUTP0710
      NW=10000./WAVE(I)                                                 OUTP0720
 1100 WRITE (IOOUT,2200) WAVE(I),NW,RADG(I),UERF(I),                    OUTP0730
     1       RADA(I,1),PATHR(I),TBR(I)                                  OUTP0740
      WRITE (IOOUT,2400) HLIGHT                                         OUTP0750
      IF (LMNLT) WRITE (IOOUT,1800) PHASE                               OUTP0760
      WRITE (IOOUT,2500) HLIGHT,HLIGHT                                  OUTP0770
      DO 1200 I=1,IWN                                                   OUTP0780
      NW=10000./WAVE(I)                                                 OUTP0790
 1200 WRITE (IOOUT,2600) WAVE(I),NW,SS(I),DIR(I)                        OUTP0800
      WRITE (IOOUT,2700)                                                OUTP0810
      DO 1300 I=1,IWN                                                   OUTP0820
      NW=10000./WAVE(I)                                                 OUTP0830
 1300 WRITE (IOOUT,2800) WAVE(I),NW,TTR(I),TBR(I),CNTRST(I)             OUTP0840
      WRITE (IOOUT,2900) (TOT(I),I=1,10),TOT(12),HLIGHT,                OUTP0850
     1       TOT(11)                                                    OUTP0860
      RETURN                                                            OUTP0870
C                                                                       OUTP0880
 1400 FORMAT (43X,37HDEFINITION  OF  CONTROL  PARAMETERS                OUTP0890
     1        8HFOLLOWS:,/,43X,10(1H-),6H  --  ,7(1H-),2X,             OUTP0900
     2        10(1H-),2X,7(1H-),///,43X,9HPARAMETER,3X,                 OUTP0910
     3        5HVALUE,3X,11HDESCRIPTION,/,43X,9(1H-),3X,5(              OUTP0920
     4        1H-),3X,11(1H-),//,45X,5HISORC,7X,I1,5X,6A4,//            OUTP0930
     5        ,45X,5HITARG,7X,I1,5X,8A4,//,45X,5HMODEL,7X,              OUTP0940
     6        I1,5X,5A4,//,45X,5HIHAZE,7X,I1,/)                         OUTP0950
 1450 FORMAT (43X,37HDEFINITION  OF  CONTROL  PARAMETERS                OUTP0960
     1        8HFOLLOWS:,/,43X,10(1H-),6H  --  ,7(1H-),2X,             OUTP0970
     2        10(1H-),2X,7(1H-),///,43X,9HPARAMETER,3X,                 OUTP0980
     3        5HVALUE,3X,11HDESCRIPTION,/,43X,9(1H-),3X,5(              OUTP0990
     4        1H-),3X,11(1H-),//,45X,5HISORC,7X,I1,5X,6A4,//            OUTP1000
     5        ,45X,5HITARG,7X,I1,5X,8A4,//,45X,5HMODEL,7X,              OUTP1010
     6 I1,5X,32HISRAELI STANDARD (YEAR, DAYTIME)//45X,5HIHAZE,7X,I1/)   OUTP1020
 1500 FORMAT (43X,37HDEFINITION  OF  CONTROL  PARAMETERS                OUTP1030
     1        8HFOLLOWS:,/,43X,10(1H-),6H  --  ,7(1H-),2X,             OUTP1040
     2        10(1H-),2X,7(1H-),///,43X,9HPARAMETER,3X,                 OUTP1050
     3        5HVALUE,3X,11HDESCRIPTION,/,43X,9(1H-),3X,5(              OUTP1060
     4        1H-),3X,11(1H-),//,45X,5HISORC,7X,I1,5X,6A4,//            OUTP1070
     5        ,45X,5HITARG,7X,I1,5X,8A4,//,45X,5HMODEL,7X,              OUTP1080
     6 I1,5X,34HISRAELI STANDARD (YEAR, NIGHTTIME)//45X,5HIHAZE,7X,I1/) OUTP1090
 1600 FORMAT (1H ,45X,22HCLOUD BOTTOM HEIGHT = ,F5.3,3H KM)             OUTP1100
 1700 FORMAT (1H1,56X,19HSOURCE  INTENSITIES,/,56X,7H------             OUTP1110
     1        12H ----------,//)                                        OUTP1120
 1800 FORMAT (44X,27HPHASE ANGLE FOR MOONLIGHT: ,F6.2,                  OUTP1130
     1        10H (DEGREES),//)                                         OUTP1140
 1900 FORMAT (15X,10HWAVELENGTH,3X,10HWAVENUMBER,3X,3A4,                OUTP1150
     1        6HSOURCE,11X,13HTARGET SOURCE,15X,7HGROUND                OUTP1160
     2        6HSOURCE,/,15X,9H(MICRONS),6X,6H(CM-1),11X,               OUTP1170
     3        8HSTRENGTH,17X,8HSTRENGTH,20X,8HSTRENGTH,/                OUTP1180
     4        ,41X,20H(WATTS M-2 MICRON-1),2(3X,9H(WATTS M-             OUTP1190
     5        16H2 MICRON-1 SR-1)),/,15X,10(1H-),3X,10(1H-),            OUTP1200
     6        3X,20(1H-),3X,25(1H-),3X,25(1H-),//)                      OUTP1210
 2000 FORMAT (15X,1PE10.4,3X,I7,11X,1PE10.4,15X,1PE10.4,18X,            OUTP1220
     1        1PE10.4)                                                  OUTP1230
 2100 FORMAT (1H1,46X,33HCOMPONENTS  FOR  RADIANCE  FROM                OUTP1240
     1        6HTARGET,/,46X,29H---------  ---  --------  --           OUTP1250
     2        10H--  ------,//,53X,23H(WATTS M-2 MICRON-1 SR-           OUTP1260
     3        2H1),///,22X,10HWAVELENGTH,3X,10HWAVENUMBER,              OUTP1270
     4        5X,6HTARGET,7X,6HTARGET,7X,7HPARTIAL,6X,                  OUTP1280
     5        7HPARTIAL,6X,5HTOTAL,/,22X,9H(MICRONS),6X,                OUTP1290
     6        6H(CM-1),6X,8HEMISSION,4X,11HREFLECTANCE,2X,              OUTP1300
     7        11HATMOSPHERIC,5X,4HPATH,8X,6HTARGET,/,75X,               OUTP1310
     8        8HEMISSION,5X,8HRADIANCE,5X,8HRADIANCE,/                  OUTP1320
     9        ,22X,10(1H-),3X,10(1H-),4X,8(1H-),4X,11(1H-),             OUTP1330
     X        2X,11(1H-),3X,8(1H-),5X,8(1H-),//)                        OUTP1340
 2200 FORMAT (22X,1PE10.4,3X,I7,3X,1P5E13.4)                            OUTP1350
 2300 FORMAT (1H1,47X,29HCOMPONENTS  FOR  BACKGROUND                    OUTP1360
     1        8HRADIANCE,/,47X,27H----------  ---  ----------          OUTP1370
     2        10H  --------,//,53X,23H(WATTS M-2 MICRON-1 SR-           OUTP1380
     3        2H1),///,22X,10HWAVELENGTH,3X,10HWAVENUMBER,              OUTP1390
     4        5X,6HGROUND,7X,6HGROUND,8X,5HTOTAL,8X,5HTOTAL,            OUTP1400
```

```
     5          7X,5HTOTAL,/,22X,9H(MICRONS),6X,6H(CM-1),6X,                    OUTP1410
     6          8HEMISSION,4X,11HREFLECTANCE,2X,11HATMOSPHERIC                  OUTP1420
     7          ,5X,4HPATH,6X,10HBACKGROUND,/,75X,8HEMISSION,                   OUTP1430
     8          2(5X,8HRADIANCE),/,22X,10(1H-),3X,10(1H-),4X,                   OUTP1440
     9          8(1H-),4X,11(1H-),2X,11(1H-),3X,8(1H-),4X,10(                   OUTP1450
     X          1H-),//)                                                        OUTP1460
2400 FORMAT (1H1,58X,6HDIRECT,3A4,/,57X,6(1H-),1X,12(1H-),/                     OUTP1470
     1          /,56X,20H(WATTS M-2 MICRON-1),//)                               OUTP1480
2500 FORMAT (41X,10HWAVELENGTH,3X,10HWAVENUMBER,2X,3A4,1X,                      OUTP1490
     1          3A4,/,41X,9H(MICRONS),6X,6H(CM-1),7X,6HSOURCE,                  OUTP1500
     2          8X,4HFLUX,/,68X,8HSTRENGTH,/,41X,10(1H-),3X,                    OUTP1510
     3          10(1H-),2X,12(1H-),1X,12(1H-),//)                               OUTP1520
2600 FORMAT (41X,1PE10.4,3X,I7,6X,1PE10.4,3X,1PE10.4)                           OUTP1530
2700 FORMAT (1H1,58X,15HTOTAL   RADIANCE,/,58X,11H-----   ----                 OUTP1540
     1          4H----,//,53X,25H(WATTS M-2 MICRON-1 SR-1),///                  OUTP1550
     2          ,35X,10HWAVELENGTH,3X,10HWAVENUMBER,5X,                         OUTP1560
     3          6HTARGET,5X,10HBACKGROUND,4X,8HCONTRAST,/                       OUTP1570
     4          ,35X,9H(MICRONS),6X,6H(CM-1),33X,5HRATIO,/                      OUTP1580
     5          ,35X,10(1H-),3X,10(1H-),5X,6(1H-),5X,10(1H-),                   OUTP1590
     6          4X,8(1H-),//)                                                   OUTP1600
2800 FORMAT (35X,1PE10.4,3X,I7,3X,1P3E13.4)                                     OUTP1610
2900 FORMAT (1H1,46X,30HDETECTOR-RESPONSE  WAVELENGTH-                          OUTP1620
     1          10HINTEGRATED,/,46X,17(1H-),2X,21(1H-),//                       OUTP1630
     2          ,58X,16H(WATTS M-2 SR-1),//,46X,7HTARGET                        OUTP1640
     3          8HEMISSION,16X,1PE10.4,/,46X,7HTARGET                           OUTP1650
     4          11HREFLECTANCE,13X,1PE10.4,/,46X,8HPARTIAL                      OUTP1660
     5          20HATMOSPHERIC EMISSION,3X,1PE10.4,/,46X,                       OUTP1670
     6          21HPARTIAL PATH RADIANCE,10X,1PE10.4,/,46X,                     OUTP1680
     7          21HTOTAL TARGET RADIANCE,10X,1PE10.4,/,46X,                     OUTP1690
     8          15HGROUND EMISSION,16X,1PE10.4,/,46X,7HGROUND                   OUTP1700
     9          11HREFLECTANCE,13X,1PE10.4,/,46X,6HTOTAL                        OUTP1710
     X          20HATMOSPHERIC EMISSION,5X,1PE10.4,/,46X,                       OUTP1720
     1          19HTOTAL PATH RADIANCE,12X,1PE10.4,/,46X,                       OUTP1730
     2          25HTOTAL BACKGROUND RADIANCE,6X,1PE10.4,//                      OUTP1740
     3          ,46X,41(1H*),//,46X,8HCONTRAST,22X,1PE11.4,//                   OUTP1750
     4          ,46X,41(1H*),//,46X,6HDIRECT,3A4,13X,1PE10.4,/                  OUTP1760
     5          ,46X,11H(WATTS M-2))                                            OUTP1770
      END                                                                       OUTP1780
```

```
      SUBROUTINE PATHRD(CTHED,HP,RT,IO,IHAZE,NR,                        PAD00010
     1           IEMISS,LEN,MODEL,VIS,V1,V2,TGRD,DUMMY,ICLMAT,MULDV)    PAD00020
      LOGICAL HORIZ                                                     PAD00030
      DIMENSION TR1(16),TR2(16),DUMMY(16),ANS(16)                       PAD00040
      COMMON /BASPOT/ ANG(65),SUM(65),WVL(16),NWVL,ALBB(16),BS(16),     PAD00050
     1 BE(16),SINGWV,SAER(65),LDUM                                      PAD00060
      COMMON /BKDAT/ALT,THETA,PHI,SANG2,ZENTH,PHASE,ALB                 PAD00070
      COMMON /CGEOM/COSGM,COSBT,CSA                                     PAD00080
      COMMON /COMI1/ISORC,ITARG,IWN,JHL                                 PAD00090
      COMMON /MO1/EH(16,34),P(34),T(34),WH(34),Z(34),WA(34),RE,M,NL,    PAD00100
     +            RRS(16,34),SCOE(16,34),                               PAD00110
     +            TRANS(16,5),RADA(16,2),WAVE(16),SS(16),               PAD00120
     1       DIR(16),RADG(16),UTEM(16),UTRF(16),BK(16),                 PAD00130
     2       PATHR(16),UERF(16),PATHR2(16),TOT(12),BKG(16)              PAD00140
C **** CONST1 = (1.-.0295)*.5*3./(8.*PI)/(1.+.5*.0295)                  PAD00150
C **** CONST2 = .0295/(1.+.5*.0295)*3./(8.*PI)                          PAD00160
      DATA CONST1,CONST2/.0570805145,.00347011897/                      PAD00170
      NLL=NL-1                                                          PAD00180
      SRAYL=CONST1*(1.+CSA*CSA)+CONST2                                  PAD00190
C **** INITIALIZE VARIABLES ****                                        PAD00200
      DO 800 IW=1,IWN                                                   PAD00210
  800 PATHR(IW)=0.0                                                     PAD00220
      DO 900 J=2,NLL                                                    PAD00230
      IF (HP.LT.Z(J)) GO TO 1000                                        PAD00240
  900 CONTINUE                                                          PAD00250
      J=NLL                                                             PAD00260
 1000 JU=J                                                              PAD00270
      JL=JU-1                                                           PAD00280
      HORIZ=.FALSE.                                                     PAD00290
      DS1=0.0                                                           PAD00300
      H2=HP                                                             PAD00310
      RAT=(H2-Z(JL))/(Z(JU)-Z(JL))                                      PAD00320
      GO TO (1100,1200,1300),IO                                         PAD00330
 1100 IX=RT*1.99999+1                                                   PAD00340
      DX=RT/FLOAT(IX)                                                   PAD00350
      DH=DX*CTHED                                                       PAD00360
      HORIZ=ABS(CTHED).LT.1.E-3                                         PAD00370
      DIST=0.5*DX                                                       PAD00380
      GO TO 1400                                                        PAD00390
 1200 IX=NLL-JL                                                         PAD00400
      DH=Z(JU)-HP                                                       PAD00410
      DX=DH/CTHED                                                       PAD00420
      GO TO 1400                                                        PAD00430
C     HORIZONTAL PATH FOLLOWS                                           PAD00440
 1300 IX=100                                                            PAD00450
      DIST=DIST-0.5*DX+0.25                                             PAD00460
      DX=0.5                                                            PAD00470
      HORIZ=.TRUE.                                                      PAD00480
      GO TO 1500                                                        PAD00490
 1400 H2=HP+DH*0.5                                                      PAD00500
 1500 DO 2600 K=1,IX                                                    PAD00510
      IF (HORIZ) GO TO 2000                                             PAD00520
      CALL LT4M(HP,H2,THETA,2,2,TR1,DUMMY,DUMMY,                        PAD00530
     1     IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,ICLMAT,IERR,             PAD00540
     2  NR,IHAZE,MULDV)                                                PAD00550
      CALL LT4M(H2,DUM,ZENTH,3,2,TR2,DUMMY,DUMMY,                       PAD00560
     1     IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,ICLMAT,IERR,             PAD00570
     2  NR,IHAZE,MULDV)                                                PAD00580
      IF (IO.EQ.2) GO TO 1800                                           PAD00590
      DO 1600 J=2,NLL                                                   PAD00600
      IF (H2.LT.Z(J)) GO TO 1700                                        PAD00610
 1600 CONTINUE                                                          PAD00620
      J=NLL                                                             PAD00630
 1700 JU=J                                                              PAD00640
      JL=JU-1                                                           PAD00650
      RAT=(H2-Z(JL))/(Z(JU)-Z(JL))                                      PAD00660
 1800 DO 1900 IW=1,IWN                                                  PAD00670
      SC=SCOE(IW,JL)+RAT*(SCOE(IW,JU)-SCOE(IW,JL))                      PAD00680
      RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL))                         PAD00690
      SSCAT=RS*SRAYL+(1.0-RS)*SAER(IW)                                  PAD00700
```

```
1900  PATHR(IW)=PATHR(IW)+TR1(IW)*TR2(IW)*SC*SSCAT*DX                        PAD00710
      GO TO (2400,2500,2300),IO                                              PAD00720
 2000 CONTINUE                                                               PAD00730
      CALL LT4M(HP,DUM,DIST,1,2,TR1,DUMMY,DUMMY,                            PAD00740
     1     IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,ICLMAT,IERR,                  PAD00750
     2  NR,IHAZE,MULDV)                                                     PAD00760
      IF (K.EQ.1) CALL LT4M(HP,DUM,ZENTH,3,2,TR2,DUMMY,DUMMY,                PAD00770
     1                IEMISS,LEN,MODEL,VIS,    V1,V2,TGRD,                  PAD00780
     2     ICLMAT,IERR,                                                      PAD00790
     2  NR,IHAZE,MULDV)                                                     PAD00800
      DS=0.0                                                                 PAD00810
      DO 2200 IW=1,IWN                                                       PAD00820
      IF (K.GT.1) GO TO 2100                                                 PAD00830
      SC=SCOE(IW,JL)+RAT*(SCOE(IW,JU)-SCOE(IW,JL))                           PAD00840
      RS=RRS(IW,JL)+RAT*(RRS(IW,JU)-RRS(IW,JL))                              PAD00850
      ANS(IW)=TR2(IW)*SC*(RS*SRAYL+(1.0-RS)*SAER(IW))                        PAD00860
 2100 DPATH=ANS(IW)*TR1(IW)*DX                                               PAD00870
      DS=DS+DPATH                                                            PAD00880
 2200 PATHR(IW)=PATHR(IW)+DPATH                                              PAD00890
      DS1=DS1+DS                                                             PAD00900
      DS=DS*0.5/DX                                                           PAD00910
      IF (IO.EQ.3.AND.K.GT.1.AND.DS/DS1.LT.0.001) RETURN                     PAD00920
      IK=(K/20)*20                                                           PAD00930
      IF (IO.EQ.3.AND.IK.EQ.K) DX=DX*2.0                                     PAD00940
 2300 DIST=DIST+DX                                                           PAD00950
      GO TO 2600                                                             PAD00960
 2400 H2=H2+DH                                                               PAD00970
      GO TO 2600                                                             PAD00980
 2500 JU=JU+1                                                                PAD00990
      JL=JL+1                                                                PAD01000
      IF (K.EQ.1) RAT=0.5                                                    PAD01010
      H2=(Z(JU)+Z(JL))*0.5                                                   PAD01020
      DX=(Z(JU)-Z(JL))/CTHED                                                 PAD01030
 2600 CONTINUE                                                               PAD01040
      RETURN                                                                 PAD01050
      END                                                                    PAD01060
```

```
      SUBROUTINE ZERO                                                   ZER00010
      COMMON /MO1/ DUMMIE(715),M,NL,DUMMYS(1088),RDATA(300)             ZER00020
      DO 100 I=1,300                                                    ZER00030
      RDATA(I)=0.0                                                      ZER00040
  100 CONTINUE                                                          ZER00050
      RETURN                                                            ZER00060
      END                                                               ZER00070
```

```
      SUBROUTINE CLIMAT(LOCAT,MONTH,NHOUR,IWIND,NPRT,TEMP,PRESS,RH,AH,DPCLI00010
     1,VIS,WNDVEL,WINDIR,IPASCT)                                        CLI00020
C                                                                       CLI00030
CC*********************************************************************CLI00040
CC                                                                      CLI00050
CC                CLIMATOLOGY MODULE - CLIMAT                           CLI00060
CC                                                                      CLI00070
CC     PURPOSE - TO PROVIDE THE CENTRAL EUROPEAN AND MID-EASTERN        CLI00080
CC               CLIMATOLOGY DATA REQUIRED BY OTHER MODULES OF EOSAEL.  CLI00090
CC                                                                      CLI00100
CC     PARAMETER DESCRIPTION                                            CLI00110
CC          LOCAT - CLIMATOLOGY REGION INDICATOR. LOCAT IS AN INTEGER   CLI00120
CC                  (1-4) FOR CENTRAL EUROPE AND                        CLI00130
CC                  (5-10) FOR MID-EAST.                                CLI00140
CC                                                                      CLI00150
CC                  REGION  1 - EUROPEAN LOWLANDS,                      CLI00160
CC                  REGION  2 - EUROPEAN RHINE VALLEY,                  CLI00170
CC                  REGION  3 - EUROPEAN HIGHLANDS,                     CLI00180
CC                  REGION  4 - EUROPEAN ALPINE,                        CLI00190
CC                  REGION  5 - MIDEAST DESERTS,                        CLI00200
CC                  REGION  6 - MIDEAST COASTAL,                        CLI00210
CC                  REGION  7 - MIDEAST PERSIAN GULF,                   CLI00220
CC                  REGION  8 - MIDEAST RED SEA,                        CLI00230
CC                  REGION  9 - MIDEAST EASTERN MOUNTAINS, AND          CLI00240
CC                  REGION 10 - MIDEAST INDUS VALLEY.                   CLI00250
CC                                                                      CLI00260
CC          MONTH - AN INTEGER (1-12) INDICATING THE MONTH OF THE YEAR. CLI00270
CC                    MONTH IS USED TO SELECT THE SEASON WHICH IS       CLI00280
CC                    APPLICABLE TO THE REGION LOCAT.                   CLI00290
CC          NHOUR - AN INTEGER (0-23) INDICATING THE TIME OF DAY LOCAL  CLI00300
CC                    STANDARD TIME (LST). NHOUR IS USED TO SELECT ONE OFCLI00310
CC                    FOUR TIME PERIODS OF THE DAY 20-02, 03-09, 10-14, CLI00320
CC                    AND 15-19.                                        CLI00330
CC          IWIND - *** NOT USED ***                                    CLI00340
CC          NPRT - A PRINT SELECTOR.                                    CLI00350
CC                   NPRT LE ZERO - DO NOT PRINT CLIMATOLOGICAL DATA.   CLI00360
CC                   NPRT GT ZERO - PRINT ALL AVAILABLE MEANS, STANDARD CLI00370
CC                    DEVIATIONS, AND PERCENT OCCURRENCES.              CLI00380
CC          TEMP - MEAN TEMPERATURE (C).                                CLI00390
CC          PRESS - MEAN SEA LEVEL PRESSURE (MB).                       CLI00400
CC          RH - MEAN RELATIVE HUMIDITY (PERCENT).                      CLI00410
CC          AH - MEAN ABSOLUTE HUMIDITY (GM/CU M).                      CLI00420
CC          DP - MEAN DEW-POINT TEMPERATURE (C).                        CLI00430
CC          VIS - MEAN HORIZONTAL VISIBILITY (KM).                      CLI00440
CC          WNDVEL - MEAN WIND SPEED (MPS).                             CLI00450
CC          WINDIR - MOST PROBABLE WIND DIRECTION (DEGREES). WINDIR IS  CLI00460
CC                    GIVEN IN 30 DEGREE INCREMENTS (015,045,075,...    CLI00470
CC                   ,345).                                             CLI00480
CC          IPASCT - INDICATOR (1-6) FOR THE MOST PROBABLE PASQUILL     CLI00490
CC                    STABILITY CATEGORY (A-F).                         CLI00500
CC          CLDHT - MEAN CLOUD HEIGHT (KM).                             CLI00510
CC          CLDCVR - MEAN TOTAL CLOUD COVER (PERCENT).                  CLI00520
CC          WNDDIR - WIND DIRECTION (DEGREES).                          CLI00530
CC                                                                      CLI00540
CC     SUBROUTINES AND FUNCTIONS - NONE                                 CLI00550
CC                                                                      CLI00560
CC     CARD INPUT - NONE                                                CLI00570
CC                                                                      CLI00580
CC     TAPE INPUT - YES. BE SURE TO ASSIGN THE CLIMATOLOGY DATA TAPE TO CLI00590
CC                   UNIT NCLIMT.                                       CLI00600
CC                                                                      CLI00610
CC*********************************************************************CLI00620
C                                                                       CLI00630
      COMMON /IOUNIT/IOIN,IOOUT,IPHFUN,LOUNIT,NDIRTU,NCLIMT,KSTOR,NPLOTUCLI00640
      DIMENSION REGION(70),SEASON(8),HOUR(8),DATA(18),DIR(13)           CLI00650
C                                                                       CLI00660
      DATA REGION/4HEURO,4HPEAN,4H LOW,4HLAND,4HS   ,                   CLI00670
     1 2*4H    ,4HEURO,4HPEAN,4H RHI,4HNE V,4HALLE,4HY   ,4H    ,4HEURO,CLI00680
     2 4HPEAN,4H HIG,4HHLAN,4HDS  ,2*4H    ,4HEURO,4HPEAN,4H ALP,4HINE  ,CLI00690
     3 3*4H    ,4HMIDE,4HAST ,4HDESE,4HRTS ,3*4H    ,4HMIDE,4HAST ,4HCOACLI00700
```

```
     4S,4HTAL  ,3*4H     ,4HMIDE,4HAST ,4HPERS,4HIAN ,4HGULF,2*4H     ,4HMICLI00710
     5DE,4HAST ,4HRED  ,4HSEA  ,3*4H     ,4HMIDE,4HAST ,4HEAST,4HERN  ,4HMOUCLI00720
     6N,4HTAIN,4HS     ,4HMIDE,4HAST  ,4HINDU,4HS VA,4HLLEY,2*4H     /      CLI00730
      DATA SEASON/4HWINT,4HER   ,4HSPRI,4HNG   ,4HSUMM,4HER   ,4HAUTU,4HMN CLI00740
     1 /                                                                    CLI00750
      DATA HOUR/4H20-0,4H2    ,4H03-0,4H9    ,4H10-1,4H4    ,4H15-1,4H9    /CLI00760
      DATA DIR/4H 015,4H 045,4H 075,4H 105,4H 135,4H 165,4H 195,4H 225,     CLI00770
     14H 255,4H 285,4H 315,4H 345,4H VBL/                                   CLI00780
C                                                                           CLI00790
CCC        POSITION THE TAPE NCLIMT FOR READING                             CLI00800
C                                                                           CLI00810
      REWIND NCLIMT                                                         CLI00820
C                                                                           CLI00830
CCC        SKIP OVER ALL DATA FOR REGIONS 1,2,...,LOCAT-1                   CLI00840
C                                                                           CLI00850
      IF(LOCAT.LT.1.OR.LOCAT.GT.10) LOCAT=1                                 CLI00860
      LSKIP=1056*(LOCAT-1)                                                  CLI00870
C                                                                           CLI00880
CCC        SKIP OVER DATA FOR SEASONS 1,2,...,SEASON-1 FOR REGION LOCAT     CLI00890
C                                                                           CLI00900
      NSEASN=1                                                              CLI00910
      IF(MONTH.GE.3.AND.MONTH.LE.5) NSEASN=2                                CLI00920
      IF(MONTH.GE.6.AND.MONTH.LE.8) NSEASN=3                                CLI00930
      IF(MONTH.GE.9.AND.MONTH.LE.10) NSEASN=4                               CLI00940
      IF(LOCAT.GE.5.AND.MONTH.EQ.11) NSEASN=4                               CLI00950
      NSKIP=LSKIP+176*(NSEASN-1)                                            CLI00960
C                                                                           CLI00970
CCC        SKIP OVER DATA FOR TIME PERIODS 0,1,...,PERIOD-1 FOR REGION      CLI00980
CCC             LOCAT DURING SEASON.                                        CLI00990
C                                                                           CLI01000
      NTIME=1                                                               CLI01010
      IF(NHOUR.GE.3.AND.NHOUR.LE.9) NTIME=2                                 CLI01020
      IF(NHOUR.GE.10.AND.NHOUR.LE.14) NTIME=3                               CLI01030
      IF(NHOUR.GE.15.AND.NHOUR.LE.19) NTIME=4                               CLI01040
      NSKIP=NSKIP+44*(NTIME-1)                                              CLI01050
      IF(NSKIP.LE.0) GO TO 2                                                CLI01060
      DO 1 J=1,NSKIP                                                        CLI01070
      READ(NCLIMT,9) A                                                      CLI01080
    1 CONTINUE                                                              CLI01090
C                                                                           CLI01100
CCC        IF NPRT GT 0, PRINT A HEADING FOR THE THERMODYNAMIC DATA         CLI01110
C                                                                           CLI01120
    2 CONTINUE                                                              CLI01130
      IF(NPRT.LE.0) GO TO 3                                                 CLI01140
      WRITE(IOOUT,15)                                                       CLI01150
      ILOC=7*LOCAT-6                                                        CLI01160
      ILOC6=ILOC+6                                                          CLI01170
      WRITE(IOOUT,10) (REGION(J),J=ILOC,ILOC6),SEASON(2*NSEASN-1),          CLI01180
     1SEASON(2*NSEASN),HOUR(2*NTIME-1),HOUR(2*NTIME)                        CLI01190
C                                                                           CLI01200
CCC        READ THE THERMODYNAMIC DATA FOR REGION LOCAT AT NTIME            CLI01210
CCC             DURING NSEASN.                                              CLI01220
C                                                                           CLI01230
    3 DO 4 J=1,22                                                           CLI01240
      READ(NCLIMT,11) NCLASS,(DATA(K),K=1,18)                               CLI01250
C                                                                           CLI01260
CCC        CONVERT FROM METERS TO KILOMETERS                                CLI01270
C                                                                           CLI01280
      DATA(6)=0.001*DATA(6)                                                 CLI01290
      DATA(10)=0.001*DATA(10)                                               CLI01300
C                                                                           CLI01310
CCC        IF NPRT GT 0, PRINT THE THERMODYNAMIC DATA                       CLI01320
C                                                                           CLI01330
      IF(NPRT.LE.0) GO TO 4                                                 CLI01340
      WRITE(IOOUT,12) NCLASS,(DATA(K),K=1,18)                               CLI01350
    4 CONTINUE                                                              CLI01360
C                                                                           CLI01370
CCC        EXTRACT THE VALUES OF TEMP, PRESS, RH, AH, DP, VIS, AND          CLI01380
CCC             WNDVEL.                                                     CLI01390
C                                                                           CLI01400
```

```
      TEMP=DATA(2)                                                          CLI01410
      DP=DATA(3)                                                            CLI01420
      AH=DATA(4)                                                            CLI01430
      RH=DATA(5)                                                            CLI01440
      VIS=DATA(6)                                                           CLI01450
      PRESS=DATA(7)                                                         CLI01460
      WNDVEL=DATA(8)                                                        CLI01470
C                                                                           CLI01480
CC          DETERMINE THE VALUE OF IPASCT                                   CLI01490
C                                                                           CLI01500
      IPASCT=1                                                              CLI01510
      FREQ=DATA(13)                                                         CLI01520
      DO 5 J=2,6                                                            CLI01530
      IF(DATA(J+12).LE.FREQ) GO TO 5                                        CLI01540
      FREQ=DATA(J+12)                                                       CLI01550
      IPASCT=J                                                              CLI01560
    5 CONTINUE                                                              CLI01570
C                                                                           CLI01580
CC          GET TO THE WIND DATA ON NCLIMT                                  CLI01590
C                                                                           CLI01600
      NSKIP=(NSKIP-LSKIP)/44                                                CLI01610
      NSKIP=44*(15-NSKIP)+22*NSKIP                                          CLI01620
      DO 16 J=1,NSKIP                                                       CLI01630
      READ(NCLIMT,9) A                                                      CLI01640
   16 CONTINUE                                                              CLI01650
C                                                                           CLI01660
CC          IF NPRT GT 0, PRINT A HEADING FOR THE WIND DATA                 CLI01670
C                                                                           CLI01680
      IF(NPRT.LE.0) GO TO 6                                                 CLI01690
      WRITE(IOOUT,13) (DIR(J),J=1,13)                                       CLI01700
C                                                                           CLI01710
CC          READ THE WIND DATA FOR REGION LOCAT AT NTIME DURING NSEASN      CLI01720
C                                                                           CLI01730
    6 DO 7 J=1,22                                                           CLI01740
      READ(NCLIMT,14) NCLASS,(DATA(K),K=1,14)                               CLI01750
C                                                                           CLI01760
CC          IF NPRT GT 0, PRINT THE WIND DATA                               CLI01770
C                                                                           CLI01780
      IF(NPRT.LE.0) GO TO 7                                                 CLI01790
      WRITE(IOOUT,17) NCLASS,(DATA(K),K=1,14)                               CLI01800
    7 CONTINUE                                                              CLI01810
C                                                                           CLI01820
CC          DETERMINE THE VALUE OF WINDIR                                   CLI01830
C                                                                           CLI01840
      NDIR=1                                                                CLI01850
      FREQ=DATA(2)                                                          CLI01860
      DO 8 J=2,12                                                           CLI01870
      IF(DATA(J+1).LE.FREQ) GO TO 8                                         CLI01880
      FREQ=DATA(J+1)                                                        CLI01890
      NDIR=J                                                                CLI01900
    8 CONTINUE                                                              CLI01910
      WINDIR=30*NDIR-15                                                     CLI01920
C                                                                           CLI01930
CC          RETURN FROM CLIMAT                                              CLI01940
C                                                                           CLI01950
      RETURN                                                                CLI01960
C                                                                           CLI01970
CC          FORMAT STATEMENTS                                               CLI01980
C                                                                           CLI01990
    9 FORMAT(A1)                                                            CLI02000
   10 FORMAT(25H1 EGSAEL CLIMATOLOGY FOR ,7A4,8H DURING ,2A4,4H AT ,2A4,CLI02010
     17H(LST),  //126H CLASS  FREQCY   MEAN   MEAN   MEAN   MEAN   MEAN  CLI02020
     2 MEAN MEAN/STDEV MEAN  MEAN/STDEV FREQCY FREQCY FREQCY FREQCY FREQCLI02030
     3CY FREQCY/126H  NO.    CLASS   TEMP    DP      AH      RH      VIS  PCLI02040
     4RESS   WNDVEL   CLDHT   CLDCVR      A       B       C       D       E CLI02050
     5     F   /126H          (%)     (C)     (C) (GM/CU.M) (%)    (KM)    (CLI02060
     6MB)   (MPS)    (KM)     (%)      (%)     (%)     (%)     (%)     (%) CLI02070
     7    (%) /)                                                            CLI02080
   11 FORMAT(6X,I3,5F5.1,F7.0,F6.1,2F5.1,F6.0/8F5.1)                        CLI02090
   12 FORMAT(I5,F9.1,4F7.1,F7.3,F7.1,F4.1,1H/,F4.1,F7.3,F6.1,1H/,F5.1,6FCLI02100
```

```
     17,1)                                                                  CLI02110
  13 FORMAT(1H0/7H CLASS ,14(7H FREQCY)/14H   NO.     CLASS,13(7H WNDDIR)CLI02120
     1/10X,3H(%),1X,13(2X,A4,1X)/14X,13(3X,3H(%),1X)/)                      CLI02130
  14 FORMAT(6X,I3,14F5.1)                                                   CLI02140
  15 FORMAT(18H1CLIMATOLOGY MODEL///39H  DEFINITIONS OF METEOROLOGICAL      CLI02150
     1CLASSES//48H CLASS  1 = FOG, HAZE AND MIST WITH VIS LT 1 KM./53H CLI02160
     2LASS  2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM./53H CLASS  3 =CLI02170
     3 FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM./48H CLASS  4 = FOG, HAZCLI02180
     4E AND MIST WITH VIS GE 7 KM./34H CLASS  5 = DUST WITH VIS LT 3 KM.CLI02190
     5/34H CLASS  6 = DUST WITH VIS GE 3 KM./53H CLASS  7 = DRIZZLE, RAICLI02200
     6N AND TSTMS WITH VIS LT 1 KM./58H CLASS  8 = DRIZZLE, RAIN AND TSTCLI02210
     7MS WITH 1 LE VIS LT 3 KM./58H CLASS  9 = DRIZZLE, RAIN AND TSTMS WCLI02220
     8ITH 3 LE VIS LT 7 KM./53H CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH CLI02230
     9VIS GE 7 KM./34H CLASS 11 = SNOW WITH VIS LT 1 KM./39H CLASS 12 = CLI02240
     ASNOW WITH 1 LE VIS LT 3 KM./39H CLASS 13 = SNOW WITH 3 LE VIS LT 7CLI02250
     B KM./34H CLASS 14 = SNOW WITH VIS GE 7 KM./59H CLASS 15 = NO WEATHCLI02260
     CER AND ABSOLUTE HUMIDITY LT 10 GM/CU M./59H CLASS 16 = NO WEATHER CLI02270
     DAND ABSOLUTE HUMIDITY GE 10 GM/CU M./52H CLASS 17 = VIS LT 1 KM ANCLI02280
     ED CEILING HEIGHT LT 300 M./53H CLASS 18 = VIS LT 3 KM AND CEILING CLI02290
     FHEIGHT LT 1000 M./36H CLASS 19 = CEILING HEIGHT LT 300 M./37H CLASCLI02300
     GS 20 = CEILING HEIGHT LT 1000 M./23H CLASS 21 = NO CEILING./36H CLCLI02310
     HASS 22 = ALL CONDITIONS COMBINED.)                                    CLI02320
  17  FORMAT(I5,F9.1,13F7.1)                                                CLI02330
C                                                                           CLI02340
      END                                                                   CLI02350
```

```
      PROGRAM AGAUS
C                              PROGRAM AGAUS                                    AGX00020
C                                                                               AGX00030
C           REVISION DATE 22 JANUARY, 1982                                      AGX00040
C                                                                               AGX00050
C                                                                               AGX00060
C     PURPOSE:                                                                  AGX00070
C             TO CALCULATE EXTINCTION COEFFICIENTS, ETC., AND PRODUCE           AGX00080
C     LEGENDRE EXPANSION COEFFICIENTS, PHASE FUNCTIONS AND (OPTIONALLY)         AGX00090
C     SCATTERING FRACTIONS UNDER A VARIETY OF CONDITIONS AND AEROSOL            AGX00100
C     DISTRIBUTIONS AT ONE OR MORE WAVELENGTHS.  THE PHASE FUNCTION             AGX00110
C     IS NORMALIZED TO 4 PI OMEGA ZERO AND MAY BE RENORMALIZED BY               AGX00120
C     DIVISION BY THE APPROPRIATE CONSTANT(S).                                  AGX00130
C                                                                               AGX00140
C     ************************  INPUT  ****************************             AGX00150
C                                                                               AGX00160
C     CARD 1 - IDENTIFIER - 80 ALPHA CHARACTERS                                 AGX00170
C                                                                               AGX00180
C     CARD 2 - INTEGER CONTROL PARAMETERS: NWAVE, NINDX, IW, IDSTP,             AGX00190
C              NRADI, IT, MQRTE, IANG, IEO, NEOU                                AGX00200
C              FORMAT (10I5)                                                    AGX00210
C   NWAVE: IS THE NO. OF WAVELENGTHS, OR REL. HUMIDITY VALUES TO BE             AGX00220
C     TREATED IN THIS RUN. SEE COMMENTS CIRCA READ OF WAVE, DWAVE, ETC.         AGX00230
C     N.B. NWAVE MUST BE .LE. 10 - TO CHANGE THE NUMBER OF WAVELENGTHS          AGX00240
C     CHANGE THE FIRST INDEX OF ARRAY OUT(I,J) TO AGREE WITH NWAVE.             AGX00250
C   NINDX: IS THE NBR OF AEROSOL COMPONENTS WHICH WILL HAVE DIFFERENT           AGX00260
C     OPTICAL CONSTANTS, MASS DENSITIES OR MASS CONCENTRATIONS.                 AGX00270
C   IW: =0 WILL SET THE REFRACTIVE INDEX OF THE AEROSOL EQUAL                   AGX00280
C     TO THAT OF WATER AT THE INPUT WAVELENGTH AND TEMP.  IF IW .NE. 0          AGX00290
C     AND HANEL'S GROWTH FACTOR IS ZERO (EMUA=0. - CARD 5), THEN THE            AGX00300
C     INPUT REFRACTIVE INDEX (EMA,CAYA) WILL BE USED FOR THE AEROSOL.           AGX00310
C     OTHERWISE THE REFRACTIVE INDEX IS ADJUSTED PER HANEL (SEE BELOW).         AGX00320
C   IDSTP: IDENTIFIES TYPE OF AEROSOL SIZE DISTRIBUTION TO BE USED.             AGX00330
C   NRADI: NO. OF PARTICLE RADII TO BE EXPECTED FOR IDSTP=0 OR 3;               AGX00340
C     THE INPUT VALUE OF NRADI IS IGNORED FOR IDSTP NOT ZERO OR 3.              AGX00350
C     NRADI MUST BE .LE. 1+2**JDIMCK(2) - C.F. BLOCK DATA                       AGX00360
C   IT: IS THE NUMBER OF GAUSS-LEGENDRE ANGLES (ORDER OF EXPANSION)             AGX00370
C     IF ONLY EXTINCION COEFFICIENTS, ETC. ARE DESIRED, I.E. NOT PHASE          AGX00380
C     FUNCTIONS, THEN SET -IT- EQUAL TO ONE.                                    AGX00390
C   MQRTE:=12345 WILL CAUSE PRINTS OF MIE EFFICIENCY FACTORS AT                 AGX00400
C     EVERY VALUE OF PARTICLE RADIUS USED IN THE MIE CALCULATIONS;              AGX00410
C     SET MQRTE = 0 IF SUCH PRINTS ARE NOT DESIRED.                             AGX00420
C  IANG:=0 FOR COMPUTATIONS OF PHASE FN. AT -IT- GAUSS LEGENDRE                 AGX00430
C     QUADRATURE ANGLES; IANG=1 FOR COMPUTATIONS OF PHASE FN AT                 AGX00440
C     -IT- EQUALLY SPACED ANGLES BETWEEN 0 AND 180 DEGREES.                     AGX00450
C     IANG=2 WILL ALLOW -IT- USER SUPPLIED ANGLES TO BE READ -                  AGX00460
C     FORMAT (16F5.1).  THIS REQUIRES AT LEAST ONE CARD OF TYPE 2A.             AGX00470
C     IF IANG.GT.0 NO LEGENDRE COEFFICIENTS WILL BE GENERATED.                  AGX00480
C  IEO=1,2,3,4 WILL CONSTRUCT A PHASE FUNCTION FILE (ON NEOU).                  AGX00490
C     IEO=1  65 PREDETERMINED ANGLES   INDIVIDUAL WAVELENGTHS ONLY             AGX00500
C     IEO=2  65 PREDETERMINED ANGLES   COMPOSITE WAVELENGTH ONLY               AGX00510
C     IEO=3  USER INPUT ANGLES         INDIVIDUAL WAVELENGTHS ONLY             AGX00520
C     IEO=4  USER INPUT ANGLES         COMPOSITE WAVELENGTH ONLY               AGX00530
C     IEO=5  65 PREDETERMINED ANGLES   INDIVIDUAL & COMPOSITE WAVELENGTHA      AGX00540
C                                      THE COMPOSITE WILL BE THE LAST DA        AGX00550
C                                      SET WRITTEN ON UNIT -NEOU- .             AGX00560
C     FOR USER INPUT ANGLES SEE -IT- AND IANG ABOVE.  THE COMPOSITE             AGX00570
C     VALUES ARE SIMPLE AVERAGES OVER THE NUMBER OF WAVELENGTHS.                AGX00580
C     THIS FILE WILL CONTAIN THE FOLLOWING INFORMATION:                         AGX00590
C     1) ANGLES (65 MAX) - FORMAT(11(F6.2,1X))                                  AGX00600
C     2)NBR OF ANGLES, PHASE FUNCTION IDENTIFIER (=0= IMPLIED USER INPUTAGX00610
C     IN EOSAEL), WAVELENGTH (UM), SINGLE SCATTERING ALBEDO, EXTINCTION AGX00620
C     AND SCATTERING COEFFICIENTS IN INVERSE KM -                               AGX00630
C     FORMAT (2(I2,1X),F5.2,1X,F8.6,1X,2(E12.6,1X))                             AGX00640
C     3) PHASE FUNCTION AT ANGLES SPECIFIED ABOVE.  N.B. THE PHASE FUNCTAGX00650
C     AS WRITTEN OUT HERE IS NORMALIZED TO 4 PI OMEGA ZERO;  THE ROUTINEAGX00660
C     IN EOSAEL WILL RENORMALIZE THE PHASE FUNCTION TO ONE.                     AGX00670
C     FORMAT (6(E12.6,1X))                                                      AGX00680
C  NEOU= UNIT NUMBER UPON WHICH EOSAEL PHASE FUNCTION IS TO BE STORED.          AGX00690
C                                                                               AGX00700
```

```
C     CARD 2A - USER SUPPLIED SET OF -IT- ANGLES  FORMAT (16F5.1)          AGX00710
C               16 VALUES PER CARD, MORE THAN 1 CARD MAY BE NEEDED.        AGX00720
C                 THIS CARD IS ONLY NEEDED WHEN IANG=2.                    AGX00730
C                                                                          AGX00740
C     CARD 3 - DISTRIBUTION PARAMETERS; ONLY ONE TYPE PER RUN.             AGX00750
C              FORMAT (6E12.6)  - READ IN AGXP1                            AGX00760
C     *****  ALL DIMENSIONS ARE IN MICRONS  *****                         AGX00770
C      TYPE 0. USER SUPPLIED - NRADI CARDS, ONE VALUE OF RADIUS AND
C              NUMBER DENSITY PER CARD.  IT IS SUGGESTED THAT DELTA BE
C              INPUT NO GREATER THAN .001 IN ORDER TO FORCE THE MAXIMUM
C              NUMBER OF RADII TO BE USED DUE TO THE POSSIBLE IRREGULAR
C              NATURE OF THIS DISTRIBUTION.
C              R(I), FF(I), I = 1, NRADI
C      TYPE 1. LOG-NORMAL                                                  AGX00810
C              RBAR, SIGMA, RLO, RHI                                       AGX00820
C      TYPE 2. DOUBLE EXPONENTIAL                                          AGX00830
C              RLO, RHI, CUE, A, B                                         AGX00840
C      TYPE 3. DEIRMENDJIAN MODEL C                                        AGX00850
C              - NO INPUT -                                                AGX00860
C      TYPE 4. POWER LAW (JUNGE)                                           AGX00870
C              RLO, RHI, CUE, A                                            AGX00880
C      TYPE 5. MODIFIED GAMMA                                              AGX00890
C              RLO, RHI, RC, ALF, GAM                                      AGX00900
C      TYPE 6. MODIFIED GAMMA FOG MODEL                                    AGX00910
C              RLO, RHI, RC, ALF, GAM, ELWC                                AGX00920
C      TYPE 7. POWER LAW                                                   AGX00930
C              VIS                                                         AGX00940
C      TYPE 8. CONTINENTAL BIMODAL                                         AGX00950
C              - NO INPUT -                                                AGX00960
C      TYPE 9. MARITIME BIMODAL                                            AGX00970
C              - NO INPUT -                                                AGX00980
C      TYPE 10.URBAN BIMODAL                                               AGX00990
C              - NO INPUT -                                                AGX01000
C      TYPE 11.USER SUPPLIED BIMODAL                                       AGX01010
C              FOA, RBARA, SGA, FOC, RBARC, SGC                            AGX01020
C      TYPE 12.MARSHALL-PALMER RAIN MODEL                                  AGX01030
C              RAIN                                                        AGX01040
C                                                                          AGX01050
C     CARD 4 - CONTROL PARAMETERS: FORMAT (6E12.6)                         AGX01060
C              WAVE, DWAVE, RELHUM, DENSH, TEMP, DELTA                     AGX01070
C              FOR LOOPING OVER RELATIVE HUMIDITY ADD                      AGX01080
C              NWAVE-1 CARDS CONTAINING RELHUM,TEMP - FORMAT(2E12.6)       AGX01090
C              SEE DWAVE BELOW.                                            AGX01100
C   WAVE: IS WAVELENGTH IN MICROMETERS.                                    AGX01110
C   DWAVE: IS THE WAVELENGTH INCREMENT IN MICROMETERS. IF DWAVE IS         AGX01120
C     LESS THAN 1.E-4, A SPECIAL CASE APPLIES USED FOR LOOPING OVER        AGX01130
C     NWAVE VALUES OF RELHUM: THE FIRST TIME THIS CARD IS READ IT          AGX01140
C     MUST CONTAIN WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA; THE SECOND AND      AGX01150
C     SUBSEQUENT TIMES IT MUST ONLY HAVE RELHUM,TEMP ON IT.  THIS          AGX01160
C     ALSO REQUIRES REPETITION OF CARD 5.                                  AGX01170
C   RELHUM: IS RELATIVE HUMIDITY IN PERCENT.                               AGX01180
C   DENSH: IS PARTICLE NUMBER PER CUBIC CENTIMETER.                        AGX01190
C     USER-SUPPLIED VALUE OF DENSH WILL BE IGNORED FOR IDSTP=3 OR GT 6     AGX01200
C     BECAUSE THOSE DISTRIBUTIONS CARRY PRE-DETERMINED DENSITY VALUES.     AGX01210
C     ALSO, IF DENSH IS LESS THAN 1E-4, THE PARTICLE NUMBER DENSITY        AGX01220
C     WILL BE CALCULATED FROM MASS DENSITY AND MASS CONCENTRATION.         AGX01230
C   TEMP: IS THE TEMPERATURE OF THE ATMOSPHERE IN DEGREES C.               AGX01240
C   DELTA:  IS THE CONVERGENCE CRITERION WITHIN A PARTICULAR SIZE          AGX01250
C     RANGE INTERVAL; HALVING IS TERMINATED WHEN THE QUANTITY DEL          AGX01260
C     IS LESS THAN DELTA. **N.B.** THE AMOUNT OF CPU TIME USED BY          AGX01270
C     THIS PROGRAM IS CLOSELY CONNECTED WITH DELTA.  THE SMALLER DELTA     AGX01280
C     IS THE LARGER YOUR RUN TIME WILL BE.  IT IS SUGGESTED THAT           AGX01290
C     DELTA BE SET EQUAL TO .001 FOR MOST RUNS.                            AGX01300
C                                                                          AGX01310
C     CARD 5 -  OPTICAL AND PHYSICAL DATA: FORMAT (4F10.5,E15.7)           AGX01320
C               EMA, CAYA, EMUA, RHOA, CONC                                AGX01330
C     REPEAT NWAVE*NINDX TIMES: IF IDSTP=6 THIS CARD IS NOT NEEDED.        AGX01340
C   EMA: IS THE REAL PART OF THE INDEX OF REFRACTION OF DRY AEROSOL.       AGX01350
C   CAYA: IS THE IMAGINARY PART OF REFRACTIVE INDEX FOR DRY AEROSOL.       AGX01360
C     ***** CAYA IS ASSUMED TO BE NEGATIVE  ****                           AGX01370
```

```
C     **** DO NOT ENTER CAYA WITH A NEGATIVE SIGN !!!!! *****                  AGX01380
C   EMUA:  IS HANEL'S GROWTH FACTOR (MU-BAR)/ACCRETION COEF.                   AGX01390
C   RHOA:  IS THE MASS DENSITY(SP. GRAV) OF DRY AEROSOL.                       AGX01400
C   CONC:  IS THE MASS CONCENTRATION(GM/CC) OF DRY AEROSOL.                    AGX01410
C                                                                              AGX01420
C     ************************  END INPUT  ************************            AGX01430
C                                                                              AGX01440
C     ******************** MISCELLANEOUS INFO ********************            AGX01450
C                                                                              AGX01460
C     THE INPUT AND OUTPUT UNITS, ALONG WITH A EXTRA, CURRENTLY UNUSED,        AGX01470
C     UNIT (NUNIT) ARE ASSIGNED VALUES IN THE BLOCK DATA SUBROUTINE.           AGX01480
C                                                                              AGX01490
C     REL. HUMIDITY TREATMENT PER G. HANEL/1976 ADV. IN GEOPHYS.               AGX01500
C                                                                              AGX01510
C     FOR DIMENSION SIZES REFER TO THE BLOCK DATA SUBROUTINE.  THERE           AGX01520
C     ALSO IS A ERROR ROUTINE (DIMER) THAT CKS ON YOUR DIMENSIONS.             AGX01530
C                                                                              AGX01540
C     SCATTERING FRACTIONS REQUIRE THAT NUNIT BE ASSIGNED AND A                AGX01550
C     SIMPLE CHANGE BE MADE IN SUBROUTINE AGXP3:  FOR FURTHER INFO             AGX01560
C     REFER TO THAT SUBROUTINE.                                                AGX01570
C                                                                              AGX01580
C     THE FUNCTION ATAN2(SQRT(1.-C(I)**2),C(I)) IS EQUIVALENT                  AGX01590
C     TO ARCOS(C(I)).                                                          AGX01600
C                                                                              AGX01610
C     ****************** ACKNOWLEDGEMENTS ************************            AGX01620
C                                                                              AGX01630
C       THIS PROGRAM HAS BEEN CONSTRUCTED BY THE ATMOSPHERIC SCIENCES          AGX01640
C     LABORATORY AND NEW MEXICO STATE UNIVERSITY, DEPT OF PHYSICS.  THE        AGX01650
C     FOLLOWING PEOPLE HAVE PARTICIPATED IN THIS UNDERTAKING:                  AGX01660
C                                                                              AGX01670
C               DR. A.U. MILLER          NMSU                                  AGX01680
C               DR. R.C. SHIRKEY         ASL                                   AGX01690
C               DR. G.H. GOEDECKE        NMSU                                  AGX01700
C               MR. E.J. BURLBAW         NMSU                                  AGX01710
C                                                                              AGX01720
C     THE POINT OF CONTACT IS R.C. SHIRKEY, ASL.  PHONE (505) 678-5470         AGX01730
C     OR AV 258-5470.                                                          AGX01740
C                                                                              AGX01750
C     ************************************************************             AGX01760
      REAL KEXTT,KSCAT,KBAKT                                                   AGX01770
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                              AGX01780
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)                           AGX01790
     +,NRADI,PI,IDSTP,NKG,NHALV,NI                                             AGX01800
      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)              AGX01810
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                                     AGX01820
      DIMENSION OUT(10,4),NTITLE(40)                                           AGX01830
C     *****  READ AND WRITE IDENTIFIER  *****                                  AGX01840
      READ (IOIN,88) (NTITLE(I),I=1,40)                                        AGX01850
88    FORMAT (40A2)                                                            AGX01860
      WRITE (IOUT,89) (NTITLE(I),I=1,40)                                       AGX01870
89    FORMAT(1H1,40A2//)                                                       AGX01880
C     ***** READ INTEGER CONTROL PARAMETERS FOR THIS RUN  *****                AGX01890
      READ (IOIN,103) NWAVE,NINDX,IW,IDSTP,NRADI,IT,MQRTE,IANG,IEO,NEOU        AGX01900
C     ERROR CHECKS                                                             AGX01910
      IF (IT.LE.0) IT=1                                                        AGX01920
      IF (IT.GT.JDIMCK(1)) CALL DIMER(1)                                       AGX01930
      IF(JDIMCK(1).LT.65) WRITE(IOUT,129)                                      AGX01940
C     CHECK FOR CONFLICTING EOSAEL OPTIONS                                     AGX01950
      IF ((IEO.EQ.2.OR.IEO.EQ.5).AND.NWAVE.EQ.1) IEO=1                         AGX01960
      IF ((IEO.EQ.4.OR.IEO.EQ.5).AND.NWAVE.EQ.1) IEO=3                         AGX01970
      IF (IEO.GT.1.AND.IT.GT.65) GO TO 20                                      AGX01980
C     EOSAEL OPTION                                                            AGX01990
      IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) IT=65                              AGX02000
      JDIMCK(3)=1+2**JDIMCK(2)                                                 AGX02010
      IF (IDSTP.GT.12) GO TO 1                                                 AGX02020
      IF (NWAVE.EQ.0) NWAVE=1                                                  AGX02030
      IF (IDSTP.EQ.12) IW=1                                                    AGX02040
      IF (NINDX.LT.1.OR.IDSTP.EQ.6.OR.IDSTP.EQ.12) NINDX=1                     AGX02050
      WRITE (IOUT,104) NWAVE,NINDX,IW,IDSTP,NRADI,IT,MQRTE,IANG,IEO,NEOU       AGX02060
      IF (IW.EQ.0) WRITE (IOUT,122)                                            AGX02070
```

```
C       INITIALIZE QUANTITIES USED IN SUMMATIONS                                AGX02080
        DO 2 I=1,IT                                                             AGX02090
        OLT(I)=0.                                                               AGX02100
      2 PSUMT(I)=0.E0                                                           AGX02110
        WAVAVG=0.                                                               AGX02120
        ALBDOT=0.                                                               AGX02130
        KEXTT=0.E0                                                              AGX02140
        KSCAT=0.E0                                                              AGX02150
        KBAKT=0.E0                                                              AGX02160
        CATTN=0.E0                                                              AGX02170
        ITT=IT-1                                                                AGX02180
        PI=3.1415926535898E+00                                                  AGX02190
        IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) GO TO 3                           AGX02200
        IF ((IANG.EQ.1).OR.(IANG.EQ.2)) GO TO 3                                 AGX02210
C       WHEN IANG=0 ROUTINE GUSET IS CALLED TO SET-UP THE ABSCISSAE AND         AGX02220
C       WEIGHTS USED FOR CALCULATING THE PHASE-FUNCTION AT -IT- POINTS          AGX02230
C       USED FOR NUMERICAL INTEGRATION VIA GAUSS-LEGENDRE QUADRATURE AND        AGX02240
C       THE PHASE FUNCTION EXPANSION COEFS, OL( ).                              AGX02250
C       THE WEIGHTS ARE PLACED IN THE ARRAY W(), AND THE COSINES OF THE         AGX02260
C       ANGLES ARE PLACED IN THE ARRAY C().                                     AGX02270
        CALL GUSET(IT)                                                          AGX02280
        IF (ITT.LT.3) ITT=3                                                     AGX02290
        GO TO 7                                                                 AGX02300
      3 CALL ANGLE (PI,IANG,IT)                                                 AGX02310
C       SUBROUTINE ANGLE IS CALLED WHEN IANG=1 OR 2 TO SET UP THE               AGX02320
C       ANGLES AT WHICH PHASE FUNCTIONS WILL BE CALCULATED.  ANGLES             AGX02330
C       GO INTO ARRAY W() AND COSINES IN C().                                   AGX02340
      7 CONTINUE                                                                AGX02350
C       WRITE ANGLES FOR EOSAEL DATA FILE                                       AGX02360
        IF (IEO.LE.0) GO TO 21                                                  AGX02370
        DO 22 I=1,IT                                                            AGX02380
     22 C(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI                               AGX02390
        ITP1=IT+1                                                               AGX02400
        IF (ITP1.GT.JDIMCK(1)) ITP1=IT                                          AGX02410
        IF (IT.LT.65.AND.JDIMCK(1).GT.65) C(ITP1)=999.99                        AGX02420
        WRITE (NEOU,125)  (C(I),I=1,ITP1)                                       AGX02430
        DO 23 I=1,IT                                                            AGX02440
     23 C(I)=COS(C(I)*PI/180.)                                                  AGX02450
     21 CONTINUE                                                                AGX02460
C       DETERMINE DETAILS OF AEROSOL SIZE-DISTRIBUTION VIA AGXP1                AGX02470
        CALL AGXP1(DENS,FSUM,VOL,JDIMCK)                                        AGX02480
        IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) ELWC=DENS                                AGX02490
C       DRYVOL IS THE AVERAGE VOLUME OF THE DRY AEROSOL PARTICLES IN            AGX02500
C       CUBIC MICROMETERS.                                                      AGX02510
        DRYVOL=VOL                                                              AGX02520
C       *** READ INPUT PARAMETERS ***                                           AGX02530
        READ (IOIN,105) WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA                      AGX02540
        IF (NWAVE.EQ.1) DWAVE=0.E0                                              AGX02550
        WRITE (IOUT,106) WAVE,DWAVE,RELHUM,DENSH,TEMP,DELTA                     AGX02560
        IF (NINDX.GT.1) WRITE (IOUT,107) NINDX                                  AGX02570
        IF ((DWAVE.LT.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,108) NWAVE           AGX02580
        IF ((DWAVE.GE.1E-04).AND.(NWAVE.GT.1)) WRITE (IOUT,109) NWAVE           AGX02590
        IF (DENSH.LT.1E-04) WRITE (IOUT,110)                                    AGX02600
        ENWAV= FLOAT(NWAVE)                                                     AGX02610
        IF (DWAVE.LT.1.E-4) GO TO 8                                             AGX02620
        WAVE=WAVE-DWAVE                                                         AGX02630
      8 DO 9 NWV=1,NWAVE                                                        AGX02640
        IF (DWAVE.GT.1.E-4) GO TO 10                                            AGX02650
        IF (NWV.EQ.1) GO TO 11                                                  AGX02660
        READ (IOIN,105) RELHUM,TEMP                                             AGX02670
        GO TO 11                                                                AGX02680
     10 WAVE=WAVE+DWAVE                                                         AGX02690
     11 VOL=DRYVOL                                                              AGX02700
C       DETERMINE WHETHER THE USER SUPPLIED PARTICLE NUMBER DENSITY DENSH       AGX02710
C       SHOULD BE OVERIDDEN BECAUSE THE CHOSEN IDSTP CASE HAS FIXED             AGX02720
C       PARAMETERS, AND/OR IF NUMBER DENSITIES ARE TO BE CALCULATED LATER       AGX02730
C       FROM THE AVG PARTICLE VOLUME, MASS DENSITY, AND MASS CONCENTRATION      AGX02740
        LLLL=0                                                                  AGX02750
        IF (IDSTP.EQ.6) GO TO 12                                                AGX02760
        IF (IDSTP.EQ.3.OR.IDSTP.GE.7) LLLL=1                                    AGX02770
```

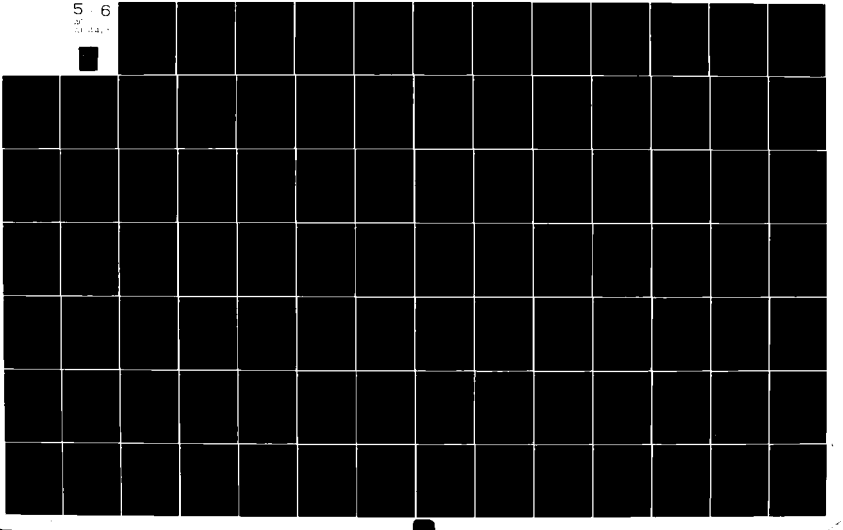
```
      IF (LLLL.EQ.1) GO TO 12                                                  AGX02780
      IF (DENSH.LE.1.E-4) GO TO 12                                             AGX02790
      LLLL=1                                                                   AGX02800
      DENS=DENSH                                                               AGX02810
   12 CONTINUE                                                                 AGX02820
C     RESTRICT RELATIVE HUMIDITY TO MAX OF 99 PERCENT.                         AGX02830
      IF (RELHUM.GE.99.E+00) RELHUM=99.0E+00                                   AGX02840
      WRITE (IOUT,111) RELHUM,WAVE                                             AGX02850
      IF (DENS.EQ.0.0) DENS=1.0E+00                                            AGX02860
      GNU=1.0E+04/WAVE                                                         AGX02870
      IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) DENS=ELWC                                 AGX02880
C     DENS IS USED AS AN ALIAS TO PASS ELWC TO ROUTINE AGXP2.                  AGX02890
      CALL AGXP2(RELHUM,CTSUM,CSSUM,CRSUM,VOL,TMASS,DENS,QATTN,TEMP,           AGX02900
     1DELTA,NINDX,IW,OLSTAR,OM2,LLLL,IT,WAVE,EM,CAY,EMM,MQRTE,PFNZRO)          AGX02910
      LMAX=3*IFIX(2.E+0*PI*EMM*R(NRADI)/WAVE)                                  AGX02920
      IF (LMAX.GT.IT) WRITE (IOUT,112) LMAX,IT                                 AGX02930
   13 CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,WAVE,EM,CAY,EMM,IT,0,        AGX02940
     +IANG)                                                                    AGX02950
C     SUM QUANTITIES OVER INDEX NWV.                                           AGX02960
      DO 14 IK=1,IT                                                            AGX02970
      OLT(IK)=OLT(IK)+OL(IK)                                                   AGX02980
   14 PSUMT(IK)=PSUMT(IK)+PSUM(IK)                                             AGX02990
C     ALBDOT BECOMES THE TOTAL SINGLE SCATTERING ALBEDO                        AGX03000
C     KEXTT BECOMES THE TOTAL EXTINCTION COEF. (PER KILOMETER)                 AGX03010
C     KSCAT BECOMES THE TOTAL SCATTERING COEF. (PER KM)                        AGX03020
C     KBAKT BECOMES THE TOTAL BACK-SCATTERING(RADAR) COEF (PER KM)             AGX03030
C     ARRAY OUT(,) HOLDS SOME QUANTITIES FOR LATER PRINTOUTS                   AGX03040
      ALBDO=CSSUM/CTSUM                                                        AGX03050
      ALBDOT=ALBDOT+ALBDO                                                      AGX03060
      KEXTT=KEXTT+CTSUM                                                        AGX03070
      KSCAT=KSCAT+CSSUM                                                        AGX03080
      KBAKT=KBAKT+CRSUM                                                        AGX03090
      CATTN=CATTN+QATTN                                                        AGX03100
      WAVAVG=WAVAVG+WAVE                                                       AGX03110
      OUT(NWV,1)=WAVE                                                          AGX03120
      OUT(NWV,2)=RELHUM                                                        AGX03130
      OUT(NWV,3)=TMASS*1.E5                                                    AGX03140
      OUT(NWV,4)=CTSUM                                                         AGX03150
      IF ((NWAVE.GT.1).AND.(DWAVE.GE.1.E-04)) WRITE (IOUT,113) NWV             AGX03160
      IF ((NWAVE.GT.1).AND.(DWAVE.LT.1.E-04)) WRITE (IOUT,114) NWV             AGX03170
C     EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH, SINGLE SCATTERING        AGX03180
C     ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR                AGX03190
C     INDIVIDUAL WAVELENGTHS.                                                  AGX03200
      IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,127)                   AGX03210
     +                        IT,WAVE,ALBDO,CTSUM,CSSUM                        AGX03220
C     EOSAEL OPTION: WRITE PHASE FUNCTION FOR INDIVIDUAL WAVELENGTHS.          AGX03230
      IF (IEO.EQ.1.OR.IEO.EQ.3.OR.IEO.EQ.5) WRITE (NEOU,128)                   AGX03240
     +    (PSUM(I),I=1,IT)                                                     AGX03250
    9 CONTINUE                                                                 AGX03260
C     END OF NWAVE LOOP                                                        AGX03270
      IF (NWAVE.LE.1) GO TO 19                                                 AGX03280
C     DIVIDE BY NBR OF VALUES OF NWV TO GET AVERAGED RESULTS                   AGX03290
      DO 16 I=1,IT                                                             AGX03300
      OL(I)=OLT(I)/ENWAV                                                       AGX03310
      PSUM(I)=PSUMT(I)/ENWAV                                                   AGX03320
   16 CONTINUE                                                                 AGX03330
      ALBDOT=ALBDOT/ENWAV                                                      AGX03340
      KEXTT=KEXTT/ENWAV                                                        AGX03350
      KSCAT=KSCAT/ENWAV                                                        AGX03360
      KBAKT=KBAKT/ENWAV                                                        AGX03370
      CATTN=CATTN/ENWAV                                                        AGX03380
      WAVAVG=WAVAVG/ENWAV                                                      AGX03390
      WRITE (IOUT,117) NWAVE                                                   AGX03400
      WRITE (IOUT,118)                                                         AGX03410
      DO 18 J=1,NWAVE                                                          AGX03420
   18 WRITE (IOUT,119) (OUT(J,JJ),JJ=1,4)                                      AGX03430
      WRITE (IOUT,123) KEXTT,KSCAT,KBAKT,CATTN,ALBDOT                          AGX03440
      CALL AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,WAVE,EM,CAY,EMM,IT,1,        AGX03450
     +IANG)                                                                    AGX03460
C     EOSAEL OPTION: WRITE NBR OF ANGLES, WAVELENGTH ,SINGLE SCATTERING        AGX03470
```

```
C      ALBEDO, EXTINCTION COEFFICIENT (TOTAL AND SCATTERING) FOR          AGX03480
C      COMPOSITE VALUES.                                                  AGX03490
       IF (IEO.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,127)             AGX03500
      +                        IT,WAVAVG,ALBDOT,KEXTT,KSCAT               AGX03510
C      EOSAEL OPTION: WRITE COMPOSITE PHASE FUNCTION.                     AGX03520
       IF (IEO.EQ.2.OR.IEO.EQ.4.OR.IEO.EQ.5) WRITE (NEOU,128)             AGX03530
      +   (PSUM(I),I=1,IT)                                               AGX03540
       GO TO 19                                                          AGX03550
     1 WRITE (IOUT,120) IDSTP                                            AGX03560
    19 WRITE (IOUT,126)                                                  AGX03570
       STOP                                                              AGX03580
20     WRITE (IOUT,124)                                                  AGX03590
       STOP                                                              AGX03600
   103 FORMAT (10I5)                                                     AGX03610
   104 FORMAT(1H ,78HINTEGER CONTROL PARAMETERS: NWAVE NINDX IW IDSTP NRAAGX03620
      +DI IT MQRTE  IANG IEO NEOU/,1X,29X,2(I2,4X),I1,3X,I2,             AGX03630
      +3X,I3,1X,I3,1X,I5,1X,I5,2(2X,I2))                                 AGX03640
   105 FORMAT (6E12.6)                                                   AGX03650
   106 FORMAT(1H ,/,17H INPUT PARAMETERS/,1X,6X,9HWAVE =    ,E12.6,8H MICRAGX03660
      +ONS/,1X,6X,9HDWAVE =   ,E12.6,8H MICRONS/,1X,6X,9HRELHUM =  ,E12.6,AGX03670
      +8H PERCENT/,1X,6X,9HDENSH =   ,E12.6,13H PARTICLES/CC/,1X,6X,      AGX03680
      +9HTEMP =    ,E12.6,6H DEG C/,1X,6X,19HDELTA (CONVERGENCE ,         AGX03690
      +13HCRITERION) = ,E12.6)                                           AGX03700
   107 FORMAT (/,1H ,29HLOOPING OPTION IN EFFECT FOR ,I2,                AGX03710
      +19H AEROSOL COMPONENTS)                                           AGX03720
   108 FORMAT (/,1H ,39HRELATIVE HUMIDITY OPTION IN EFFECT FOR ,I2,      AGX03730
      +7H VALUES)                                                        AGX03740
   109 FORMAT (/,1H ,40HWAVELENGTH LOOPING OPTION IN EFFECT FOR ,I2,     AGX03750
      +12H WAVELENGTHS)                                                  AGX03760
   110 FORMAT(1H ,52H***  PARTICLE NUMBER DENSITY WILL BE CALCULATED FROMAGX03770
      +,41H MASS DENSITY AND MASS CONCENTRATION  ***)                    AGX03780
   111 FORMAT (1H1,//,1X,33HRELATIVE HUMIDITY FOR THIS RUN = ,F6.2,      AGX03790
      +25H  PERCENT.  WAVELENGTH = ,F10.3,8H MICRONS,/)                  AGX03800
   112 FORMAT (/49H *** WARNING *** OPTIMAL PF EXPANSION ORDER OF   ,I3, AGX03810
      +         22H   EXCEEDS INPUT IT = ,I3,24H.   PF VALUES SHOULD BE ,AGX03820
      +         15HUSED CAUTIOUSLY/)                                     AGX03830
   113 FORMAT (1H //,1X,40(1H*),3X,31HEND OF WAVELENGTH CYCLE NUMBER ,I3,AGX03840
      +3X,40(1H*))                                                       AGX03850
   114 FORMAT (1H //,1X,40(1H*),3X,38HEND OF RELATIVE HUMIDITY CYCLE NUMBAGX03860
      +ER ,I3,3X,40(1H*))                                                AGX03870
   117  FORMAT(1H1,/,47H SUMMARY OF RESULTS FOR THIS RUN AVERAGED OVER ,  AGX03880
      +I2,30H WAVELENGTH(S) ARE AS FOLLOWS://)                           AGX03890
   118  FORMAT(1H ,4X,48HWAVELENGTH    REL.HUMIDITY    AEROSOL MASS    K(,AGX03900
      +  11HEXTINCTION),/1X,29H  (MICROMETERS)    (PERCENT)              AGX03910
      +  ,6H   (GM,25H/(SQ.CM-KM)      (PER KM)/)                        AGX03920
   119 FORMAT (2F15.6,1P2E16.5)                                          AGX03930
   120 FORMAT (//13H *** IDSTP = ,I5,35H IS ILLEGAL. EXECUTION TERMINATEDAGX03940
      +         2H* //)                                                  AGX03950
   122 FORMAT (/,1X,23H*** WATER ONLY CASE ***/)                         AGX03960
   123  FORMAT (/,20H EXTINCTION COEF. = ,5X,1PE13.7,9H (PER KM),/,       AGX03970
      120H SCATTERING COEF. = ,8X,1PE13.7,9H (PER KM),/,                 AGX03980
      225H BACK-SCATTERING COEF. = ,3X,1PE13.7,9H (PER KM),/,            AGX03990
      321H ATTENUATION COEF. = ,7X,1PE13.7,13H SQ-METERS/MG,/,           AGX04000
      428H SINGLE SCATTERING ALBEDO = ,1PE13.7,/)                        AGX04010
   124  FORMAT(1H ,58H*** MORE THAN 65 ANGLES FOR EOSAEL OPTION - PGM TERMAGX04020
      +INATED)                                                           AGX04030
   125  FORMAT (11(F6.2,1X))                                            AGX04040
   126  FORMAT(1H1)                                                      AGX04050
   127  FORMAT(I2,1X,2H00,1X,F5.2,1X,F8.6,1X,2(E12.6,1X))                AGX04060
   128  FORMAT(6(E12.6,1X))                                              AGX04070
   129  FORMAT(1H ,23H**** AGAUS WARNING ****,/,1X,                      AGX04080
      +  37HTHE ARRAY W IS ASSIGNED 65 VALUES IN ,/,1X,                  AGX04090
      +  43HBLOCK DATA WHICH IS LARGER THAN ARRAY SIZE ,/,1X,            AGX04100
      +  47HYOU MAY BE CLOBBERING INSTRUCTIONS AND/OR DATA./)            AGX04110
       END                                                               AGX04120
```

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 9/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF

UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL

5 6



```
      SUBROUTINE AGXP1 (DENS,FSUM,VOL,JDIMCK)                                AGA00010
      COMMON /PT1/  F(513),R(513),DR(8),RR(9),FF(514)                        AGA00020
     +,NRADI,PI,IDSTP,NKG,NHALV,NI                                           AGA00030
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                                   AGA00040
      DIMENSION JDIMCK(3)                                                    AGA00050
      EXTERNAL GAMNA                                                         AGA00060
      WRITE (IOUT,2)                                                         AGA00070
C     CHOOSE AND SET UP PARTICLE SIZE DISTRIBUTION                           AGA00080
      IF (IDSTP.NE.0) GO TO (12,15,16,17,19,19,17,20,21,22,24,26),IDSTP      AGA00090
C**   TYPE 0: ARBITRARY USER-SUPPLIED DISTRIBUTION. NRADI VALUES OF
C     R(J) AND FF(J) MUST BE GIVEN, ONE PER CARD, AND READ IN ORDER
C     FROM SMALLEST RADIUS, RLO TO THE LARGEST.
C     NRADI MUST BE LESS THAN OR EQUAL TO 1+2**JDIMCK(2)
      WRITE (IOUT,7)                                                         AGA00140
      IF (NRADI.GT.JDIMCK(3)) CALL DIMER(2)                                  AGA00160
      DO 9 J=1,NRADI                                                         AGA00170
    9 READ (IOIN,1) R(J),FF(J)                                               AGA00190
      RLO=R(1)
      DO 10 J=1,NRADI,5                                                      AGA00200
      JK=J+4                                                                 AGA00210
      IF(JK.GT.NRADI)JK=NRADI                                                AGA00220
   10 WRITE (IOUT,11)(R(K),FF(K),K=J,JK)                                     AGA00230
      WRITE (IOUT,11)                                                        AGA00240
      FF(NRADI+1)=FF(NRADI)                                                  AGA00250
      RR(1)=RLO                                                              AGA00260
      RR(2)=R(NRADI)                                                         AGA00270
      MIN=0                                                                  AGA00280
      GO TO 28                                                               AGA00290
C**   TYPE 1: LOG-NORMAL DISTRIBUTION                                        AGA00300
   12 READ (IOIN,1) RBAR,SIGMA,RLO,RHI                                       AGA00310
C     SIGMA IS STANDARD DEVIATION, NOT LN(SIGMA)                             AGA00320
      SIGIN=SIGMA                                                            AGA00330
      SIGMA=ALOG(SIGMA)                                                      AGA00340
      A=ABS(1.E0/(2.5066283E0*SIGMA))                                        AGA00350
      IF ((RHI-RLO).LE.1.E-4) GO TO 13                                       AGA00360
      RR(1)=RLO                                                              AGA00370
      RR(3)=RHI                                                              AGA00380
      GO TO 14                                                               AGA00390
   13 RR(1)=RBAR*EXP(-4.E0*SIGMA)                                            AGA00400
      RR(3)=RBAR*EXP(4.E0*SIGMA)                                             AGA00410
   14 RR(2)=RBAR                                                             AGA00420
      MIN=1                                                                  AGA00430
      WRITE (IOUT,3) RBAR,SIGIN,RLO,RHI                                      AGA00440
      AVOL=4.18879E0*(RBAR**3.E0)*EXP(4.5E0*SIGMA*SIGMA)                     AGA00450
C     HERE AND ELSEWHERE, AVOL IS THE VOLUME OBTAINED VIA                    AGA00460
C     ANALYTICAL INTEGRATION OVER THE LIMITS RLO =0 TO RHI =                 AGA00470
C     INFINITY; THAT CAN ONLY BE DONE FOR A FEW IDSTP CASES.                 AGA00480
      GO TO 28                                                               AGA00490
C** TYPE 2: DOUBLE EXPONENTIAL F(R)=CUE*A*EXP(-A*R)+(1-CUE)*B*EXP(-B*R).   AGA00500
C       RESTRICTIONS: RHI.GT.RLO, B.GT.A.GE.0, 0.LE.CUE.LE.1.0.             AGA00510
   15 READ (IOIN,1) RLO,RHI,CUE,A,B                                          AGA00520
      WRITE (IOUT,4) RLO,RHI,CUE,A,B                                         AGA00530
      RR(1)=RLO                                                              AGA00540
      RR(3)=RHI                                                              AGA00550
      RR(2)=0.5E0*(RLO+RHI)                                                  AGA00560
      MIN=1                                                                  AGA00570
      GO TO 28                                                               AGA00580
C**   TYPE 3: DEIRMENDJIAN MODEL C. F(R) = 1.0, RLO.LE.R.LE.4*DELRD,         AGA00590
C     F(R)=(4*DELRD/R)**4, R.GE.(4*DELRD)                                    AGA00600
C     NRADI IS READ IN EARLIER IN THE MAIN PROGRAM.                          AGA00610
   16 DENS=1.378E+04                                                         AGA00620
      DELRD=0.02E0                                                           AGA00630
      RLO=0.02E0                                                             AGA00640
      RHI=RLO+DELRD* FLOAT(NRADI-1)                                          AGA00650
      RR(1)=RLO                                                              AGA00660
      RR(3)=RHI                                                              AGA00670
      MIN=1                                                                  AGA00680
      RR(2)=RLO+4.E0*DELRD                                                   AGA00690
      GO TO 28                                                               AGA00700
C**   TYPE 4 AND TYPE 7: POWER LAW. F(R) = CUE*R**-A                         AGA00710
```

```
C       RLO,LE,R,LF,RHI; VIS=VISIBILITY IN KILOMETERS.                          AGA00720
   17   IF (IDSTP.EQ.4) GO TO 18                                                AGA00730
C**     TYPE 7 PRESCRIBED PARAMETER.                                            AGA00740
        READ (IOIN,1) VIS                                                       AGA00750
        RLO=0.1E0                                                               AGA00760
        RHI=15.E0                                                               AGA00770
        CUE=30.E0                                                               AGA00780
        A=4.E0                                                                  AGA00790
        DENS=11.E0**(5.E0-ALOG10(VIS))                                          AGA00800
C**     TYPE 4 PRESCRIBED PARAMETERS.                                           AGA00810
   18   IF (IDSTP.EQ.4) READ (IOIN,1) RLO,RHI,CUE,A                             AGA00820
        WRITE (IOUT,5) RLO,RHI,CUE,A,VIS                                        AGA00830
        RR(1)=RLO                                                               AGA00840
        RR(3)=RHI                                                               AGA00850
        RR(2)=(0.5E0*(RLO**(-A)+RHI**(-A)))**(-1.E0/A)                          AGA00860
        MIN=1                                                                   AGA00870
        GO TO 28                                                                AGA00880
C**     TYPE 5: MODIFIED GAMMA/GENERALIZED KHIRGIAN-MAZIN                       AGA00890
C       F(R) = (R**ALF)*EXP(-ALF*((R/RC)**GAM)/GAM)                             AGA00900
C       RLO,LE,R,LE,RHI.                                                        AGA00910
C**     TYPE 6: SPECIAL CASE FOR WATER FOGS OR CLOUDS,                          AGA00920
C       IN WHICH CASE ELWC IS LIQUID WATER CONTENT                              AGA00930
C       IN GRAMS PER CUBIC CENTIMETER:                                          AGA00940
C       ELWC IS IGNORED IF IDSTP = 5.                                           AGA00950
   19   READ (IOIN,1) RLO,RHI,RC,ALF,GAM,ELWC                                   AGA00960
        IF (IDSTP.EQ.6) DENS=ELWC                                               AGA00970
        WRITE (IOUT,6) RLO,RHI,RC,ALF,GAM                                       AGA00980
        RR(1)=RLO                                                               AGA00990
        RR(2)=RC                                                                AGA01000
        RR(3)=RHI                                                               AGA01010
        MIN=1                                                                   AGA01020
        B=ALF/(GAM*RC**GAM)                                                     AGA01030
        AVOL=4.1888*B**(-3./GAM)*GAMMA((ALF+4.)/GAM)/GAMMA((ALF+1.)/GAM)       AGA01040
        GO TO 28                                                                AGA01050
C**     TYPES 8,9,10: BIMODAL LOG-NORMAL DISTRIBUTIONS.                         AGA01060
C       METHOD BELOW VALID FOR RBARC*EXP(-SGA).GT.RBARA*EXP(SGA)                AGA01070
C**     TYPE 8: CONTINENTAL BIMODAL.                                            AGA01080
   20   FOA=4.E03                                                               AGA01090
        FOC=2.1E0                                                               AGA01100
        SGA=0.74E0                                                              AGA01110
        SGC=0.81E0                                                              AGA01120
        RBARA=0.03E0                                                            AGA01130
        RBARC=0.4E0                                                             AGA01140
        GO TO 23                                                                AGA01150
C**     TYPE 9: MARITIME BIMODAL.                                               AGA01160
   21   FOA=4.E02                                                               AGA01170
        FOC=3.8E0                                                               AGA01180
        SGA=0.68E0                                                              AGA01190
        SGC=0.74E0                                                              AGA01200
        RBARA=0.05E0                                                            AGA01210
        RBARC=0.65E0                                                            AGA01220
        GO TO 23                                                                AGA01230
C**     TYPE 10: URBAN BIMODAL.                                                 AGA01240
   22   FOA=2.E04                                                               AGA01250
        FOC=0.6E0                                                               AGA01260
        SGA=0.63E0                                                              AGA01270
        SGC=.77E0                                                               AGA01280
        RBARA=0.04E0                                                            AGA01290
        RBARC=0.63E0                                                            AGA01300
C       CALCULATE RADII FOR TYPES 8,9,10.                                       AGA01310
   23   RR(1)=RBARA*EXP(-4.E0*ABS(SGA))                                         AGA01320
        RR(2)=RBARA                                                             AGA01330
        RR(3)=RBARA*EXP(4.E0*ABS(SGA))                                          AGA01340
        RR(4)=RBARC*EXP(-4.E0*ABS(SGC))                                         AGA01350
        RR(5)=RBARC*EXP(4.E0*ABS(SGC))                                          AGA01360
        MIN=2                                                                   AGA01370
        DO 60 J=1,4                                                             AGA01380
        DO 60 I=1,4                                                             AGA01390
        IF (RR(I+1).GT.RR(I)) GO TO 60                                          AGA01400
        HH=RR(I)                                                                AGA01410
```

```
      RR(I)=RR(I+1)                                                     AGA01420
      RR(I+1)=HH                                                        AGA01430
60    CONTINUE                                                          AGA01440
      GO TO 28                                                          AGA01450
C**   TYPE 11:                                                          AGA01460
C     USER SUPPLIED BIMODAL CASE: FOA AND FOC ARE THE NUMBER DENSITIES  AGA01470
C     FOR THE ACCUMULATION (SMALLER RBAR) AND COARSE MODES,             AGA01480
C     RESPECTIVELY, IN PARTICLES PER CUBIC CENTIMETER.                  AGA01490
C     SGA IS STD.DEVIATION FOR MODE A ** NOT LN(SIGMA) ***              AGA01500
C     SGC IS STD.DEVIATION FOR MODE C ** NOT LN(SIGMA) ***              AGA01510
C     *** NOTE, HOWEVER, THAT SGA AND SGC ARE THE LOGS OF THE           AGA01520
C     STANDARD DEVIATIONS IN THE PRE-CODED CASES TYPE  8-10.            AGA01530
   24 READ (IOIN,1) FOA,RBARA,SGA,FOC,RBARC,SGC                         AGA01540
      WRITE (IOUT,25) FOA,RBARA,SGA,FOC,RBARC,SGC                       AGA01550
      SGA=ABS(ALOG(SGA))                                                AGA01560
      SGC=ABS(ALOG(SGC))                                                AGA01570
      GO TO 23                                                          AGA01580
C**   TYPE 12:  MARSHALL-PALMER RAIN MODEL.                             AGA01590
C     C.F. MASON, PHYSICS OF CLOUDS, CH. ON RADAR METEOROLOGY.          AGA01600
C     INPUT PARAMETER RAIN IS RAIN RATE IN MILLIMETERS/HOUR;            AGA01610
C     ** EMA,CAYA, AND RHOA ARE REQUIRED FOR THIS DISTRIBUTION.         AGA01620
   26 READ (IOIN,1) RAIN                                                AGA01630
      ENZERO=0.08E0                                                     AGA01640
      CAPL=41.E0*RAIN**(-0.21E0)                                        AGA01650
      DENS=ENZERO/CAPL                                                  AGA01660
      AVOL=PI*(CAPL**(-3.E0))*1.E12                                     AGA01670
C     CONVERT UNITS FROM CM-4 TO (CM-3)*(MICROMETERS**(-1));            AGA01680
C     THE FACTOR OF 2 CONVERTS THE M-P FORMULA FROM DIAMETER-DATA TO    AGA01690
C     RADIUS BASED FORM.                                                AGA01700
      ENZERO=2.E-4*ENZERO                                               AGA01710
      CAPL=2.E-4*CAPL                                                   AGA01720
      MIN=0                                                             AGA01730
      RR(1)=1.E-4                                                       AGA01740
      RR(2)=2500.E0                                                     AGA01750
      WRITE (IOUT,27) RAIN,DENS                                         AGA01760
C     THE NEXT BLOCK IS COMMON TO ALL DISTRIBUTIONS.                    AGA01770
C     IT SETS THE NMAX VALUES OF RADIUS, R(KK).                         AGA01780
   28 MAX=JDIMCK(2)                                                     AGA01790
      NHALV=MAX-MIN                                                     AGA01800
      NMAX=1+2**MAX                                                     AGA01810
      NI=2**MIN                                                         AGA01820
      IF (NMAX.GT.JDIMCK(3).OR.NI.GT.JDIMCK(3)) CALL DIMER(3)           AGA01830
      NLAST=NI+1                                                        AGA01840
      NKG=2**NHALV                                                      AGA01850
      ENKG= FLOAT(NKG)                                                  AGA01860
      IF (IDSTP.EQ.0) GO TO 30
      DO 29 I=1,NI                                                      AGA01870
      DR(I)=RR(I+1)-RR(I)                                               AGA01880
      DO 29 K=1,NKG                                                     AGA01890
      KK=(I-1)*NKG+K                                                    AGA01900
   29 R(KK)=RR(I)+( FLOAT(K-1))*DR(I)/ENKG                              AGA01910
      R(NMAX)=RR(NLAST)                                                 AGA01920
C     BRANCH AGAIN CALCULATE THE DIFFERENT F(R) ON THE NMAX POINTS R(K) AGA01930
                GO TO (31,33,35,38,41,41,39,43,43,43,43,46),IDSTP       AGA01940
C**   TYPE 0: ARBITRARY                                                 AGA01950
C     INTERPOLATE TO EQUAL INCREMENTS OVER RADII
30    DELR=(R(NRADI)-RLO)/ENKG                                          AGA01960
      F(1)=FF(1)
      NMAXM1=NMAX-1
      DO 64 KK=1,NMAXM1
      RADUS=RLO+DELR*FLOAT(KK)
      DO 62 J=1,NRADI
      K=J
      IF (R(J).GE.RADUS) GO TO 61
62    CONTINUE
61    CONTINUE
      F(KK+1)=(RADUS-R(K-1))*(FF(K)-FF(K-1))/
     1        (R(K)-R(K-1))+FF(K-1)
64    CONTINUE
      DO 65 I=1,NI                                                      AGA01870
```

```
      DR(I)=RR(I+1)-RR(I)                                                    AGA01880
      DO 65 K=1,NKG                                                          AGA01890
      KK=(I-1)*NKG+K                                                         AGA01900
   65 R(KK)=RR(I)+( FLOAT(K-1))*DR(I)/ENKG                                   AGA01910
      R(NMAX)=RR(NLAST)                                                      AGA01920
      GO TO 48                                                               AGA02010
C**   TYPE 1: LOG NORMAL                                                     AGA02020
   31 DEN=2.E0*SIGMA*SIGMA                                                   AGA02030
      DO 32 KK=1,NMAX                                                        AGA02040
      GNUM=ALOG(R(KK)/RBAR)                                                  AGA02050
   32 F(KK)=EXP(-GNUM*GNUM/DEN)*A/R(KK)                                      AGA02060
      GO TO 48                                                               AGA02070
C**   TYPE 2: DOUBLE EXPONENTIAL                                             AGA02080
   33 DO 34 KK=1,NMAX                                                        AGA02090
      FKK=(1.E0-CUE)*B*EXP(-B*R(KK))                                         AGA02100
   34 F(KK)=FKK+CUE*A*EXP(-A*R(KK))                                          AGA02110
      GO TO 48                                                               AGA02120
C**   TYPE 3: DEIRMENDJIAN  MODEL C.                                         AGA02130
   35 DO 36 KK=1,NMAX                                                        AGA02140
   36 F(KK)=1.E0                                                             AGA02150
      NKG1=NKG+1                                                             AGA02160
      DO 37 KK=NKG1,NMAX                                                     AGA02170
   37 F(KK)=(RR(2)/R(KK))**4.E0                                              AGA02180
      GO TO 48                                                               AGA02190
C**   TYPES 4 AND 7: POWER LAW                                               AGA02200
   38 GO TO 39                                                               AGA02210
   39 DO 40 KK=1,NMAX                                                        AGA02220
   40 F(KK)=CUE*R(KK)**(-A)                                                  AGA02230
      GO TO 48                                                               AGA02240
C**   TYPE 5 AND TYPE 6:  MODIFIED GAMMA                                     AGA02250
   41 DO 42 KK=1,NMAX                                                        AGA02260
   42 F(KK)=(EXP(-B*R(KK)**GAM))*R(KK)**ALF                                  AGA02270
      GO TO 48                                                               AGA02280
C**   TYPES 8,9,10,11:   BIMODAL LOG-NORMAL DISTRIBUTIONS                    AGA02290
   43 DENA=2.E0*SGA*SGA                                                      AGA02300
      DENC=2.E0*SGC*SGC                                                      AGA02310
      FAA=FOA/SGA                                                            AGA02320
      FCC=FOC/SGC                                                            AGA02330
      DO 44 KK=1,NMAX                                                        AGA02340
      GNUMA=ALOG(R(KK)/RBARA)                                                AGA02350
      GNUMC=ALOG(R(KK)/RBARC)                                                AGA02360
      FA=FAA*EXP(-GNUMA*GNUMA/DENA)                                          AGA02370
      FC=FCC*EXP(-GNUMC*GNUMC/DENC)                                          AGA02380
   44 F(KK)=(FA+FC)/R(KK)                                                    AGA02390
      DENS=FOA+FOC                                                           AGA02400
      WRITE (IOUT,45) DENS                                                   AGA02410
      VOLA=4.18879E0*(RBARA**3.E0)*EXP(4.5E0*SGA*SGA)*FOA                    AGA02420
      VOLC=4.18879E0*(RBARC**3.E0)*EXP(4.5E0*SGC*SGC)*FOC                    AGA02430
      AVOL=(VOLA+VOLC)/DENS                                                  AGA02440
      GO TO 48                                                               AGA02450
C**   TYPE 12: MARSHALL-PALMER RAIN MODEL                                    AGA02460
   46 DO 47 KK=1,NMAX                                                        AGA02470
   47 F(KK)=ENZERO*EXP(-CAPL*R(KK))                                          AGA02480
C     CALCULATE NORMALIZED F(KK) AND SOME DRY VOLUMES USING ALL NMAX         AGA02490
C     VALUES OF RADII.                                                       AGA02500
C     (VOL=AVERAGE PARTICLE VOLUME IN A DISTRIBUTION). THE                   AGA02510
C     NORMALIZATION AND FURTHER VOLUMES ARE RECALCULATED LATER               AGA02520
C     BY THE HALVING INTEGRATION METHOD.                                     AGA02530
   48 FSUM=0.E0                                                              AGA02540
      IF(F(1).LT.0.E0)F(1)=0.0E0                                             AGA02550
      DO 49 J=2,NMAX                                                         AGA02560
      IF(F(J).LT.0.E0)F(J)=0.E0                                              AGA02570
   49 FSUM=FSUM+0.5E0*(F(J)+F(J-1))*(R(J)-R(J-1))                            AGA02580
      DO 50 J=1,NMAX                                                         AGA02590
   50 F(J)=F(J)/FSUM                                                         AGA02600
      WRITE (IOUT,8) FSUM                                                    AGA02610
      NRADI =NMAX                                                            AGA02620
      IF (IDSTP.EQ.1.OR.IDSTP.EQ.5.OR.IDSTP.GE.8) WRITE (IOUT,51) AVOL       AGA02630
      VOL=0.E0                                                               AGA02640
      DO 52 J=2,NMAX                                                         AGA02650
```

```
   52 VOL=VOL+2.0944E0*(F(J)*R(J)**3.E0+F(J-1)*R(J-1)**3.E0)*(R(J)-R(J-  AGA02660
     1          1))                                                     AGA02670
      WRITE (IOUT,53) VOL                                              AGA02680
C     THE VOLUME PER PARTICLE CALCULATED HERE IS OBTAINED USING        AGA02690
C     ALL AVAILABLE (NMAX VALUES) VALUES FOR THE PARTICLE RADII.       AGA02700
      WRITE (IOUT,54)                                                  AGA02710
      DO 56 INT=1,NI                                                   AGA02720
      INF=INT+1                                                        AGA02730
   56 WRITE (IOUT,55) INT,RR(INT),RR(INF)                              AGA02740
    1 FORMAT (6E12.6,I3)                                               AGA02750
    2 FORMAT (1H ,//24H AEROSOL PARAMETERS ARE )                       AGA02760
    3 FORMAT (1H ,24X,6HRBAR= ,E12.6,5X,7HSIGMA= ,E12.6,7H RLO = ,E12.6,AGA02770
     +         7H RHI = ,E12.6/)                                       AGA02780
    4 FORMAT (1H ,24X,5HRLO= ,E10.4,1X,5HRHI= ,E10.4,1X,5HCUE= ,E10.4,  AGA02790
     +         1X,3HA= ,E10.4,1X,3HB= ,E10.4/)                         AGA02800
    5 FORMAT (1H ,24X,5HRLO= ,E10.4,1X,5HRHI= ,E10.4,1X,5HCUE= ,E10.4,  AGA02810
     +         1X,3HA= ,E10.4,1X,4HVIS=,E10.4/)                        AGA02820
    6 FORMAT (1H ,24X,5HRLO= ,E10.4,1X,5HRHI= ,E10.4,1X,4HRC= ,E10.4,1X,AGA02830
     +         5HALF= ,E10.4,1X,5HGAM= ,E10.4/)                        AGA02840
    7 FORMAT (/1H ,5(26H   RADIUS    RELATIVE NO. )//)                 AGA02850
    8 FORMAT (/46H NORMALIZATION FACTOR FOR SIZE DISTRIBUTION = ,E14.7) AGA02860
   11 FORMAT (1X,10(1PE12.6,1X))                                       AGA02870
   25 FORMAT (/1X,7HN(A) = ,E12.6,2X,9H RBARA = ,E12.6,2X,12H SIGMA(A) =AGA02880
     +         ,E12.6,/,1X,7HN(C) = ,E12.6,2X,9H RBARC = ,E12.6,2X,    AGA02890
     +         12H SIGMA(C) = ,E12.6/)                                 AGA02900
   27 FORMAT (/1X,42H MARSHALL-PALMER RAIN MODEL : RAIN RATE = ,1PE10.3,AGA02910
     +         21H MM PER HOUR, DENS = ,1PE12.6,8H PART/CC)            AGA02920
   45 FORMAT (/,1H ,50H*** BIMODAL DISTRIBUTION...EQUIVALENT  DENSITY ISAGA02930
     +         ,1PE13.6,18H  PARTICLES PER CC,/)                       AGA02940
   51 FORMAT(/45H AVERAGE ANALYTIC DRY VOLUME PER PARTICLE IS ,3X,     AGA02950
     +         1PE12.6,18H CUBIC MICROMETERS)                          AGA02960
   53 FORMAT (1X,47HAVERAGE NUMERICAL DRY VOLUME IS                ,   AGA02970
     +         1PE12.6,18H CUBIC MICROMETERS/)                         AGA02980
   54 FORMAT (1X,10X,35H SIZE-INTERVALS USED ARE AS FOLLOWS/)          AGA02990
   55 FORMAT (1H ,14H INTERVAL NO. ,I3,5X,7HRMIN = ,F11.5,5X,8H RMAX =  AGA03000
     +         ,F11.5)                                                 AGA03010
      RETURN                                                           AGA03020
      END                                                              AGA03030
```

```
      SUBROUTINE AGXP2(RELHUM,CTSUM,CSSUM,CRSUM,TVOL,TMASS,DENS,CATTN,   AGB00010
     1 TEMP,DELTA,NINDX,IW,OLSTAR,OM2,LLLL,IT,WAVE,EM,CAY,EMM,MQRTE,    AGB00020
     2 PFNZRO)                                                          AGB00030
      REAL KEXT,KEXTT,KEXOLD                                            AGB00040
C     N.B. FFF IS AN ALIS FOR ARRAY FF                                  AGB00050
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FFF(514),                  AGB00060
     +NRADI,PI,IDSTP,NKG,NHALV,NI                                       AGB00070
      COMMON /PT2/ PHH(65),PSUMTT(65),PGG(65),PSUM(65),PSUMT(65),       AGB00080
     1P(65)                                                             AGB00090
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                              AGB00100
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                       AGB00110
C     IN THIS SUBROUTINE THE FOLLOWING CONVENTIONS ARE USED IN          AGB00120
C     PREFIXING VARIABLE NAMES:                                         AGB00130
C     THE LETTER  C  IS USED FOR CROSS-SECTIONS                         AGB00140
C     THE LETTER  Q  IS USED FOR EFFICIENCY FACTORS                     AGB00150
C     THE LETTER  K  IS USED FOR EXTINCTION COEF. PER UNIT PATH (KM)    AGB00160
C     THE LETTER  T  IS A SUFFIX FOR TOTAL VALUES                       AGB00170
C     THE LETTER  O  IS A PREFIX FOR OMEGA SUB 1 AND 2 CALCULATIONS     AGB00180
C     FOR THE IDSTP=6 AND 12 CASES, DENS IS USED TO TRANSFER THE        AGB00190
C     LIQUID WATER CONTENT FROM THE MAIN PROGRAM TO THIS SUBROUTINE;    AGB00200
C     ELWC IS USED AS THE AEROSOL CONCENTRATION FOR THOSE CASES.        AGB00210
      IF (IDSTP.EQ.6.OR.IDSTP.EQ.12) ELWC=DENS                          AGB00220
      PZRSMT=0.                                                         AGB00230
      OLSTAR=0.0E0                                                      AGB00240
      OM2=0.E0                                                          AGB00250
      CTSUMT=0.0E+00                                                    AGB00260
      CSSUMT=0.0E+00                                                    AGB00270
      DENST=0.E0                                                        AGB00280
      CRSUMT=0.0E+00                                                    AGB00290
      EMM=1.E0                                                          AGB00300
      NLINES=0                                                          AGB00310
      BH=1.056E-3                                                       AGB00320
C     FACTORS BH AND CH ARE USED IN SIZE ADJUSTMENTS                    AGB00330
C     FH IS THE SATURATION RATIO                                        AGB00340
      FH=RELHUM/100.E0                                                  AGB00350
      CH=FH/(1.E0-FH)                                                   AGB00360
      CONCT=0.0E0                                                       AGB00370
      KEXTT=0.E0                                                        AGB00380
C     CONVERT VOL PER PARTICLE RECEIVED FROM MAIN PROGRAM VIA VARIABLE  AGB00390
C     TVOL TO DRY VOLUME PER PARTICLE IN CUBIC CENTIMETERS              AGB00400
      DRYVOL=TVOL*1.0E-12                                               AGB00410
      TVOL=0.E0                                                         AGB00420
      TMASS=0.E0                                                        AGB00430
      DO 6 J=1,IT                                                       AGB00440
      PSUMTT(J)=0.0E0                                                   AGB00450
      PHH(J)=0.0E0                                                      AGB00460
6     PGG(J)=0.E0                                                       AGB00470
C     CONVERT TEMP. TO KELVIN FOR SUBROUTINE WATER USAGE                AGB00480
      TEMK=TEMP+273.16E0                                                AGB00490
C     SKIP SUBROUTINE WATER FOR THE IDSTP = 12 CASE, AND READ THE       AGB00500
C     OPTICAL DATA FOR RAIN AS EMUA,CAYA,ETC...NEEDED BECAUSE CASE      AGB00510
C     IDSTP=12 MAY BE AT WAVELENGTHS LONGER THAN FOUND IN ROUTINE       AGB00520
C     WATER.                                                            AGB00530
      IF (IDSTP.EQ.12) GO TO 8                                          AGB00540
C     SUBROUTINE WATER RETURNS INTERPOLATED VALUES FOR EMW, CAYW AND    AGB00550
C     RHOW AT WAVELENGTH = WAVE AND AT TEMPERATURE = TEMK (DEG K).      AGB00560
C     EMW IS REAL PART OF INDEX OF REFR FOR PURE WATER AT TEMP(DEG C).  AGB00570
C     CAYW IS IMAG. PART OF INDEX OF REFR. FOR PURE WATER;              AGB00580
C     CAYW, HERE IS POSITIVE, BUT TREATED AS NEGATIVE IN MIE-ROUTINE.   AGB00590
C     RHOW IS MASS DENSITY(GM/CC) AT TEMPERATURE = TEMP (DEG C).        AGB00600
      CALL WATER(WAVE,EMW,CAYW,TEMK,RHOW)                               AGB00610
      WRITE (IOUT,9) EMW,CAYW,TEMP,RHOW                                 AGB00620
C     BEGIN LOOP OVER AEROSOL COMPONENTS INDEXED BY NK                  AGB00630
8     DO 32 NK=1,NINDX                                                  AGB00640
C     BYPASS READ OF EMA,CAYA,ETC. FOR IDSTP=6 CASE..USE WATER DATA     AGB00650
      IF (IDSTP.NE.6) GO TO 10                                          AGB00660
      EMA=EMW                                                           AGB00670
      CAYA=CAYW                                                         AGB00680
      RHOA=RHOW                                                         AGB00690
      CONC=ELWC                                                         AGB00700
```

```
      EMUA=0.0E0                                                        AGB00710
      GO TO 11                                                          AGB00720
C     ***  READ OPTICAL AND PHYSICAL DATA    ***                      AGB00730
   10 READ (IOIN,2) EMA,CAYA,EMUA,RHOA,CONC                             AGB00740
      IF (IW.EQ.0) EMA=EMW                                              AGB00750
      IF (IW.EQ.0) CAYA=CAYW                                            AGB00760
      IF(IDSTP.NE.12) GO TO 11                                          AGB00770
      EMUA=0.0E0                                                        AGB00780
      RHOW=RHOA                                                         AGB00790
      EMW=EMA                                                           AGB00800
      CAYW=CAYA                                                         AGB00810
   11 IF (RHOA.LE.0.E0) RHOA=1.E0                                       AGB00820
      WRITE (IOUT,3) NK,EMA,CAYA,RHOA,EMUA,CONC                         AGB00830
      IF (EMA.LT.1.E-30) GO TO 44                                       AGB00840
      BHT=BH*(298.E0/TEMK)                                              AGB00850
      IF (EMUA.LE.0.01) CH=0.0                                          AGB00860
      BC=BHT*CH                                                         AGB00870
      A=1.E0+((RHOA/RHOW)*EMUA*CH)                                      AGB00880
      AC=A**(1.E0/3.E0)                                                 AGB00890
C     ADJUST EM,RHO AND CAY PER G. HANEL/ADVANCES IN GEOPHYS/1976       AGB00900
      RHO=RHOW+(RHOA-RHOW)/A                                            AGB00910
      EM=EMW+(EMA-EMW)/A                                                AGB00920
      CAY=CAYW+(CAYA-CAYW)/A                                            AGB00930
      CAY=CAY/EM                                                        AGB00940
C     INITIALIZE QUANTITIES USED TO HOLD RUNNING SUMMATIONS OVER        AGB00950
C     RADII FOR THE CURRENT COMPONENT                                   AGB00960
      CTSUM=0.E0                                                        AGB00970
      CSSUM=0.E0                                                        AGB00980
      CRSUM=0.0E0                                                       AGB00990
      VOL=0.0E0                                                         AGB01000
      QL1SUM=0.0E0                                                      AGB01010
      QL2SUM=0.E0                                                       AGB01020
      PZRSUM=0.                                                         AGB01030
      DO 13 J=1,IT                                                      AGB01040
   13 PSUM(J)=0.0E0                                                     AGB01050
C     PRINT HEADER IF DETAILED MIE RESULTS ARE TO BE PRINTED            AGB01060
      IF (MQRTE.EQ.12345) WRITE (IOUT,5)                                AGB01070
C     BEGIN ACTUAL LOOP OVER RADIUS INTERVALS FOR THE CURRENT NK VALUE  AGB01080
C     THIS LOOP IS THE ONE IN WHICH THE MIE CALCULATIONS ARE CALLED     AGB01090
C     INTERVALS ARE INDEXED BY I.  THERE ARE NI SUCH INTERVALS.         AGB01100
      DO 26 I=1,NI                                                      AGB01110
      NRADI=2                                                           AGB01120
      D=RR(I+1)-RR(I)                                                   AGB01130
C     RIT IS THE ADJUSTED RADIUS FOR THE RELATIVE HUMIDITY TO BE USED   AGB01140
C     IN THIS PARTICULAR RUN OR PASS                                    AGB01150
      RIT=RR(I)*AC -(BC/AC)                                             AGB01160
      IF (RIT.LT.RR(I).OR.RR(I).LT.0.04E0) RIT=RR(I)                    AGB01170
      ALPHA=2.E0*PI*RIT/WAVE                                            AGB01180
C     ROUTINE MIEGX DOES THE ACTUAL MIE CALCULATIONS.                   AGB01190
C     NOTE THAT THE IMAG. PART OF THE REFRACTIVE INDEX (CAY) HAS BEEN   AGB01200
C     NORMALIZED THROUGH DIVISION BY THE REAL PART (EM) BEFORE ITS      AGB01210
C     VALUE IS PASSED TO THE MIE-ROUTINE.                               AGB01220
C     MIEGX RETURNS THE EXTINCTION EFFICIENCY FACTOR AS QT              AGB01230
C     MIEGX RETURNS THE SCATTERING EFFICIENCY FACTOR AS QS              AGB01240
C     MIEGX RETURNS THE BACK-SCATTERING (RADAR) EFFIC. FACTOR AS QR     AGB01250
C     MIEGX RETURNS THE AVERAGE INTENSITY (I1+I2)/2 IN THE ARRAY P()    AGB01260
C     AT ANGLES = ARCCOS(  C(  )  ),WHERE C( ) IS SET-UP BY             AGB01270
C     SUBROUTINE GUSET OR ANGLE                                         AGB01280
C     MIEGX ALSO RETURNS THE 2-ND AND 3-RD LEGENDRE EXPANSION COEF.     AGB01290
C     (OMEGA SUB 1 AND OMEGA SUB 2) AS O1STAR AND O2STAR.               AGB01300
      EMD= (EM)                                                         AGB01310
      CAYD= (CAY)                                                       AGB01320
      ALPHAD= (ALPHA)                                                   AGB01330
      CALL MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,O1STRD,O2STRD,           AGB01340
     +C,IT,PFNZRO)                                                      AGB01350
      EM=(EMD)                                                          AGB01360
      CAY=(CAYD)                                                        AGB01370
      ALPHA=(ALPHAD)                                                    AGB01380
      QT=(QTD)                                                          AGB01390
      QS=(QSD)                                                          AGB01400
```

```
      QR=(QRD)                                                          AGB01410
      O1STAR=(O1STRD)                                                   AGB01420
      O2STAR=(O2STRD)                                                   AGB01430
      KK=1+(I-1)*NKG                                                    AGB01450
      IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I),F(KK),ALPHA,QT,QS,QR AGB01440
      FKK=F(KK)                                                         AGB01460
      FKKA=FKK*PI*RIT**2.E0                                             AGB01470
      VOLHH=4.1888*FKK*RIT**3.E0                                        AGB01480
      OL1HH=O1STAR*FKKA*QT                                              AGB01490
      OL2HH=O2STAR*FKKA*QT                                              AGB01500
      CTHH=QT*FKKA                                                      AGB01510
      CSHH=QS*FKKA                                                      AGB01520
      CRHH=QR*FKKA                                                      AGB01530
      DO 14 J=1,IT                                                      AGB01540
      PHH(J)=P(J)*FKK                                                   AGB01550
14    CONTINUE                                                          AGB01560
      PFNZER=PFNZRO*FKK                                                 AGB01570
      RIT=RR(I+1)*AC-(BC/AC)                                            AGB01580
      IF (RIT.LT.RR(I+1).OR.RR(I+1).LT.0.04E0) RIT=RR(I+1)              AGB01590
      ALPHA=2.E0*PI*RIT/WAVE                                            AGB01600
      EMD= (EM)                                                         AGB01610
      CAYD= (CAY)                                                       AGB01620
      ALPHAD= (ALPHA)                                                   AGB01630
      CALL MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,O1STRD,O2STRD,           AGB01640
     +C,IT,PFNZRO)                                                      AGB01650
      EM=(EMD)                                                          AGB01660
      CAY=(CAYD)                                                        AGB01670
      ALPHA=(ALPHAD)                                                    AGB01680
      QT=(QTD)                                                          AGB01690
      QS=(QSD)                                                          AGB01700
      QR=(QRD)                                                          AGB01710
      O1STAR=(O1STRD)                                                   AGB01720
      O2STAR=(O2STRD)                                                   AGB01730
      KK1=1+NKG*I                                                       AGB01740
      FKK1=F(KK1)                                                       AGB01750
      IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,RR(I+1),FKK1,ALPHA,QT,QS,QRAGB01760
      FKK1A=FKK1*PI*RIT**2.E0                                           AGB01770
      VOLHH=(VOLHH+4.188E0*FKK1*RIT**3.E0)*D*0.5E0                      AGB01780
      OL1HH=(OL1HH+FKK1A*QT*O1STAR)*D*0.5E0                             AGB01790
      OL2HH=(OL2HH+FKK1A*QT*O2STAR)*D*0.5E0                             AGB01800
      CTHH=(CTHH+QT*FKK1A)*D*.5E0                                       AGB01810
      CSHH=(CSHH+QS*FKK1A)*D*.5E0                                       AGB01820
      CRHH=(CRHH+QR*FKK1A)*D*.5E0                                       AGB01830
      DO 15 J=1,IT                                                      AGB01840
      PHH(J)=(PHH(J)+P(J)*FKK1)*D*0.5E0                                 AGB01850
15    CONTINUE                                                          AGB01860
      PFNZER=(PFNZER+PFNZRO*FKK1)*D*0.5                                 AGB01870
      FF=0.5E0*D*(FKK+FKK1)                                             AGB01880
      NT=1                                                              AGB01890
      N=1                                                               AGB01900
   16 NJ=NT                                                             AGB01910
      NT=2*NT                                                           AGB01920
      D=0.5E0*D                                                         AGB01930
      VOLGG=0.0E0                                                       AGB01940
      OL1GG=0.0E0                                                       AGB01950
      OL2GG=0.E0                                                        AGB01960
      CTGG=0.E0                                                         AGB01970
      CSGG=0.E0                                                         AGB01980
      CRGG=0.E0                                                         AGB01990
      FT=0.E0                                                           AGB02000
      DO 17 J=1,IT                                                      AGB02010
   17 PGG(J)=0.E0                                                       AGB02020
      PZRTMP=0.                                                         AGB02030
C     NEXT LOOP HANDLES INTERMEDIATE PARTICLE SIZES..THOSE LYING BETWEEN AGB02040
C     RMIN AND RMAX FOR THE CURRENT INTERVAL WHOSE INDEX IS I.          AGB02050
      DO 19 JG=1,NJ                                                     AGB02060
      KK=1+(I-1)*NKG+(2*JG-1)*(NKG/NT)                                  AGB02070
      RIT=R(KK)*AC -(BC/AC)                                             AGB02080
      IF (RIT.LT.R(KK).OR.R(KK).LT.0.04E0) RIT=R(KK)                    AGB02090
      ALPHA=2.E0*PI*RIT/WAVE                                            AGB02100
```

```
      EMD= (EM)                                                          AGB02110
      CAYD= (CAY)                                                        AGB02120
      ALPHAD= (ALPHA)                                                    AGB02130
      CALL MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,Q1STRD,Q2STRD,            AGB02140
     +C,IT,PFNZRO)                                                       AGB02150
      EM=(EMD)                                                           AGB02160
      CAY=(CAYD)                                                         AGB02170
      ALPHA=(ALPHAD)                                                     AGB02180
      QT=(QTD)                                                           AGB02190
      QS=(QSD)                                                           AGB02200
      QR=(QRD)                                                           AGB02210
      Q1STAR=(Q1STRD)                                                    AGB02220
      Q2STAR=(Q2STRD)                                                    AGB02230
      IF (MQRTE.EQ.12345) WRITE (IOUT,4) RIT,R(KK),F(KK),ALPHA,QT,QS,QR  AGB02240
      NRADI=NRADI+1                                                      AGB02250
      FKK=F(KK)                                                          AGB02260
      FKKA=FKK*PI*RIT**2.E0                                              AGB02270
      VOLGG=4.1888E0*FKK*RIT**3.0E0+VOLGG                                AGB02280
      OL1GG=OL1GG+Q1STAR*FKKA*QT                                         AGB02290
      OL2GG=OL2GG+Q2STAR*FKKA*QT                                         AGB02300
      CTGG=CTGG+QT*FKKA                                                  AGB02310
      CSGG=CSGG+QS*FKKA                                                  AGB02320
      CRGG=CRGG+QR*FKKA                                                  AGB02330
      DO 18 J=1,IT                                                       AGB02340
      PGG(J)=PGG(J)+P(J)*FKK                                             AGB02350
18    CONTINUE                                                           AGB02360
      PZRTMP=PZRTMP+PFNZRO*FKK                                           AGB02370
   19 FT=FT+FKK                                                          AGB02380
C     ADD RESULTS ACCUMULATED DURING PREVIOUS HALVINGS TO THOSE FOUND    AGB02390
C     FOR THE NEW RADII TREATED WITHIN THE LOOP OVER INDEX JG            AGB02400
      VOLHHT=0.5E0*VOLHH+D*VOLGG                                         AGB02410
      OL1HHT=0.5E0*OL1HH+D*OL1GG                                         AGB02420
      OL2HHT=0.5E0*OL2HH+D*OL2GG                                         AGB02430
      CTHHT=0.5E0*CTHH+D*CTGG                                            AGB02440
      CSHHT=0.5E0*CSHH+D*CSGG                                            AGB02450
      CRHHT=0.5E0*CRHH+D*CRGG                                            AGB02460
      DO 20 J=1,IT                                                       AGB02470
   20 PHH(J)=.5E0*PHH(J)+D*PGG(J)                                        AGB02480
      PFNZER=.5E0*PFNZER+D*PZRTMP                                        AGB02490
      FFT=0.5E0*FF+D*FT                                                  AGB02500
      IF (CTHHT.LT.1.E-30) GO TO 22                                      AGB02510
      DEL=ABS(VOLHHT-VOLHH)/ABS(VOLHHT)                                  AGB02520
      IF (DEL.LE.DELTA) GO TO 21                                         AGB02530
      GO TO 22                                                           AGB02540
   21 IF (N.GT.2) GO TO 24                                               AGB02550
C     DO NOT ALLOW DEL LESS THAN DELTA EXIT UNLESS AT LEAST TWO          AGB02560
C     HAVINGS HAVE BEEN DONE                                             AGB02570
   22 IF (N.EQ.NHALV) GO TO 24                                           AGB02580
C     MUST EXIT WHEN NHALV HALVINGS HAVE BEEN DONE EVEN IF THE DELTA     AGB02590
C     CRITERION HAS NOT BEEN SATISFIED..SINCE NO MORE VALUES OF RADII    AGB02600
C     ARE AVAILABLE.                                                     AGB02610
      FF=FFT                                                             AGB02620
      CRHH=CRHHT                                                         AGB02630
      OL1HH=OL1HHT                                                       AGB02640
      OL2HH=OL2HHT                                                       AGB02650
      CSHH=CSHHT                                                         AGB02660
      CTHH=CTHHT                                                         AGB02670
      VOLHH=VOLHHT                                                       AGB02680
      N=N+1                                                              AGB02690
      GO TO 16                                                           AGB02700
24    CONTINUE                                                           AGB02710
      IF(N.EQ.NHALV) WRITE(IOUT,124) I                                   AGB02720
C     SUM QUANTITIES OVER ALL INTERVALS TREATED UP UNTIL NOW             AGB02730
      CTSUM=CTSUM+CTHHT                                                  AGB02740
      CSSUM=CSSUM+CSHHT                                                  AGB02750
      CRSUM=CRSUM+CRHHT                                                  AGB02760
      VOL=VOL+VOLHHT                                                     AGB02770
      OL1SUM=OL1SUM+OL1HHT                                               AGB02780
      OL2SUM=OL2SUM+OL2HHT                                               AGB02790
C     AT THIS POINT, PSUM( ) IS THE RUNNING SUM OF THE AVG. INTENSITY    AGB02800
```

```
C       AS SUMMED OVER SIZES                                                    AGB02810
        DO 25 J=1,IT                                                            AGB02820
   25   PSUM(J)=PSUM(J)+PHH(J)                                                  AGB02830
        PZRSUM=PZRSUM+PFNZER                                                    AGB02840
        WRITE (IOUT,27) NK,I,NRADI,CTHHT                                        AGB02850
        NLINES=NLINES+NRADI                                                     AGB02860
C        END LOOP OVER HALVING INTERVALS INDEXED BY I                           AGB02870
   26   CONTINUE                                                                AGB02880
C       CALCULATE PARTICLE NUMBER DENSITY(NO. PER CC) AS DENSC                  AGB02890
        DENSC=CONC/(RHOA*DRYVOL)                                                AGB02900
C       OVERRIDE CALCULATED VALUE OF DENSC WITH DENS IF LLLL =1                 AGB02910
        IF (LLLL.EQ.1) DENSC=DENS                                               AGB02920
C       RECALCULATE CONC FROM OTHER INPUT DATA IF LLLL=1                        AGB02930
        IF (LLLL.EQ.1) CONC=DENS*RHOA*DRYVOL                                    AGB02940
C       REPLACE DENS BY DENSC FOR LATER USE BY AGXP3                            AGB02950
        DENS=DENSC                                                              AGB02960
C       WEIGHT CTSUM,ETC. BY NUMBER DENSITIES (DENSC) FOR THIS COMPONENT        AGB02970
        CTSUM=CTSUM*DENSC                                                       AGB02980
        CSSUM=CSSUM*DENSC                                                       AGB02990
        CRSUM=CRSUM*DENSC                                                       AGB03000
        VOL=VOL*DENSC                                                           AGB03010
        OL1SUM=OL1SUM*DENSC                                                     AGB03020
        OL2SUM=OL2SUM*DENSC                                                     AGB03030
        DO 29 J=1,IT                                                            AGB03040
   29   PSUM(J)=PSUM(J)*DENSC                                                   AGB03050
        PZRSUM=PZRSUM*DENSC                                                     AGB03060
C       NOW, SUM OVER COMPONENTS INDEXED BY NK                                  AGB03070
C       CONCT IS THE TOTAL DRY-AEROSOL CONCENTRATION IN MG PER CC               AGB03080
        CONCT=CONCT+1.E3*CONC                                                   AGB03090
        DENST=DENST+DENSC                                                       AGB03100
        OLSTAR=OL1SUM+OLSTAR                                                    AGB03110
        OM2=OL2SUM+OM2                                                          AGB03120
        CTSUMT=CTSUMT+CTSUM                                                     AGB03130
C       AT THIS POINT, CTSUMT IS THE TOTAL EXTINCTION CROSS SECTION             AGB03140
C       (IN SQ. MICRONS) AS SUMMED OVER ALL COMPONENTS WHICH                    AGB03150
C       HAVE BEEN DEALT WITH THUS FAR                                           AGB03160
        CSSUMT=CSSUMT+CSSUM                                                     AGB03170
        CRSUMT=CRSUMT+CRSUM                                                     AGB03180
        DO 30 J=1,IT                                                            AGB03190
   30   PSUMTT(J)=PSUM(J)+PSUMTT(J)                                             AGB03200
        PZRSMT=PZRSMT+PZRSUM                                                    AGB03210
        VOL=VOL*1.E-12                                                          AGB03220
C       TVOL IS THE TOTAL VOLUME (IN CM**3) OCCUPIED BY THE AEROSOL             AGB03230
C       PARTICLES.  TVOL IS NOT ACTUALLY USED IN THIS VERSION OF                AGB03240
C       PROGRAM AGAUS.                                                          AGB03250
        TVOL=VOL+TVOL                                                           AGB03260
        EMASS=VOL*RHO                                                           AGB03270
        TMASS=TMASS+EMASS                                                       AGB03280
        KEXOLD=KEXTT                                                            AGB03290
        KEXTT=CTSUMT*1.E-3                                                      AGB03300
        KEXT=KEXTT-KEXOLD                                                       AGB03310
        WRITE (IOUT,31) NK,VOL,EMASS,KEXT                                       AGB03320
C       VPF(VOL) IS THE VOLUME PACKING FRACTION: THAT IS, THE FRACTION          AGB03330
C       OF EACH CC OF SPACE WHICH IS FILLED BY AEROSOL MATERIAL BELONGING       AGB03340
C       TO THE CURRENT COMPONENT NK.                                            AGB03350
C       TMASS IS THE TOTAL MASS OF AEROSOL FOUND IN 1 CC OF SPACE.              AGB03360
C       EMASS IS THE MASS OF AEROSOL MATERIAL PER CC ASSOCIATED WITH            AGB03370
C       THE CURRENT COMPONENT NK.                                               AGB03380
C       KEXT IS THE EXTINCTION COEF.(PER KM) WHICH IS ASSOCIATED WITH           AGB03390
C       THE CURRENT COMPONENT--AS IF IT ALONE WERE PRESENT.                     AGB03400
C       KEXTT IS THE SUM OF THE KEXT'S OVER ALL COMPONENTS.                     AGB03410
C       END LOOP OVER AEROSOL COMPONENTS INDEXED BY NK.                         AGB03420
        IF (NINDX.GT.1) WRITE (IOUT,42) NK                                      AGB03430
        IF (MQRTE.EQ.12345) WRITE (IOUT,43)                                     AGB03440
   32   CONTINUE                                                                AGB03450
        IF (NINDX.GT.1) WRITE (IOUT,33) TMASS,KEXTT                             AGB03460
        WRITE (IOUT,34) NLINES                                                  AGB03470
        DENS=DENST                                                              AGB03480
        NRADI=NLINES                                                            AGB03490
C       NOW, PERFORM THE FINAL RENORMALIZATIONS TO OBTAIN CTSUM, ETC.           AGB03500
```

```
C     VALUES REPRESENTATIVE OF A SINGLE AVERAGE PARTICLE.                    AGB03510
C     CTSUM BECOMES THE EXTINCTION CROSSECTION IN SQ. MICROMETERS PER        AGB03520
C     AVERAGE PARTICLE. THE OTHER QUANTITIES CARRY SIMILAR MEANINGS.         AGB03530
      DO 35 J=1,IT                                                           AGB03540
   35 PSUM(J)=PSUMTT(J)/DENST                                                AGB03550
      OLSTAR=OLSTAR/CTSUMT                                                   AGB03560
      OM2=OM2/CTSUMT                                                         AGB03570
      CTSUM=CTSUMT/DENST                                                     AGB03580
      CSSUM=CSSUMT/DENST                                                     AGB03590
      CRSUM=CRSUMT/DENST                                                     AGB03600
C     ...AND CONVERT ANG INTENS AT ZERO DEGREES INTO PHASE FUNCT.            AGB03610
C     WITH NORMALIZATION                                                     AGB03620
      PFNZRO=PZRSMT/DENST*(WAVE*WAVE/(PI*CTSUM*EMM*EMM))                     AGB03630
      WRITE(IOUT,36) OLSTAR,OM2,PFNZRO                                       AGB03640
C     CALCULATE ATTENUATION COEFS. IN SQ.METERS PER MILLIGRAM                AGB03650
      CATTN=CTSUMT*1.E-12/CONCT                                              AGB03660
      CATTNW=1.E-12*KEXTT/TMASS                                              AGB03670
      WRITE (IOUT,37) CATTN                                                  AGB03680
      IF(RELHUM.GT.1.0)WRITE (IOUT,38) CATTNW                                AGB03690
      GO TO 41                                                               AGB03700
   44 WRITE (IOUT,45) EMA                                                    AGB03710
      STOP                                                                   AGB03720
    2 FORMAT (4F10.6,E15.7)                                                  AGB03730
    3 FORMAT (1H ,6H INDX=,I3,4H M= ,F10.6,6H K = -,F10.6,9HI.  MASS ,       AGB03740
     +        10HDENSITY = ,F8.6,                                           AGB03750
     +        17H GROWTH FACTOR = ,F8.4,9H. CONC = ,1PE12.5,7H  GM/CC/)     AGB03760
    4 FORMAT (1X,F10.5,6(2X,1PE11.5))                                        AGB03770
    5 FORMAT (//54H R(MICRONS)  DRY  RADIUS     N(R)       MIE  SIZE        AGB03780
     +        ,36HQ (EXT)      Q(SCA)      Q(RADAR)//)                      AGB03790
    9 FORMAT (1H ,39HINDEX OF REFRACTION FOR PURE WATER IS: ,F8.6,           AGB03800
     +        3H - ,F8.6,1HI//,1X,25HMASS DENSITY OF WATER AT ,F6.2,        AGB03810
     +          11H DEG C IS: ,F8.2,6H GM/CC,/)                             AGB03820
  124 FORMAT(/52H *** CONVERGENCE LEVEL NOT REACHED FOR INTERVAL NO. ,       AGB03830
     +        I3,4H ***/)                                                   AGB03840
   27 FORMAT(1H ,19H FOR COMPONENT NO. ,I3,15H  INTERVAL NO. ,I3,1H ,I4,     AGB03850
     +        43H RADII WERE USED.  CONTRIBUTION TO CTSUM = ,1PE12.6)       AGB03860
   31 FORMAT (1H ,/,20H  FOR COMPONENT NO. ,I2,12H :    VPF = ,1PE12.5,     AGB03870
     +7H PER CC,24H   MASS CONCENTRATION = ,E12.5,21H  GM/CC.  KEXT =       AGB03880
     +        E12.5,7H PER KM)                                              AGB03890
   33 FORMAT (1H ,/,29H  TOTAL MASS CONCENTRATION = ,1PE12.5,7H GM/CC;,      AGB03900
     +15H  TOTAL KEXT = ,E12.5,7H PER KM)                                    AGB03910
   34 FORMAT (/1X,32H TOTAL NUMBER OF RADII USED WAS ,I5)                    AGB03920
   36 FORMAT (/1X,19H ANALYTIC SOLUTIONS,/,                                  AGB03930
     +16H  OMEGA SUB 1 = ,1PE14.7/,16H  OMEGA SUB 2 = ,1PE14.7/              AGB03940
     +,16H  PFN AT ZERO = ,1PE14.7,/)                                        AGB03950
   37 FORMAT (1H ,21H ATTENUATION COEF. = ,1PE12.5,12H  SQ-METERS/,          AGB03960
     +        33HMILLIGRAM OF DRY AEROSOL MATERIAL)                         AGB03970
   38 FORMAT (1H ,21H ATTENUATION COEF. = ,1PE12.5,12H  SQ-METERS/,          AGB03980
     +        33HMILLIGRAM OF WET AEROSOL MATERIAL/)                        AGB03990
   42 FORMAT (1H /,1X,10X,30(1H*),33H  END OF AEROSOL COMPONENT CYCLE ,      AGB04000
     +        7HNUMBER ,I3,2X,30(1H*)//)                                    AGB04010
   43 FORMAT (1H )                                                           AGB04020
   45 FORMAT (///,1X,11H***** EMA (,F10.6,20H) IS EITHER ZERO OR ,          AGB04030
     +35HNEGATIVE - PROGRAM TERMINATED *****)                                AGB04040
   41 RETURN                                                                 AGB04050
      END                                                                    AGB04060
```

```
      SUBROUTINE AGXP3(CTSUM,CSSUM,CRSUM,GNU,DENS,NINDX,                AGC00010
     +WAVE,EM,CAY,EMM,IT,IEND,IANG)                                     AGC00020
      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)       AGC00030
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                              AGC00040
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                       AGC00050
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)                    AGC00060
     +,NRADI,PI,IDSTP,NKG,NHALV,NI                                      AGC00070
C     IEND=1 WHEN THE COMPOSITE PHASE FUNCTION IS BEING WRITTEN         AGC00080
      IF (IEND.EQ.1) GO TO 6                                            AGC00090
      ALBDO=CSSUM/CTSUM                                                 AGC00100
C     PFACT IS USED TO CONVERT AVG. INTENSITY PSUM() INTO PHASE-        AGC00110
C     FUNCTIONS.  SFACT IS USED TO CONVERT PSUM INTO SCATTERING         AGC00120
C     FRACTIONS, NORMALIZED PER SOM. THE INTEGRAL OF SCAT OVER SOLID    AGC00130
C     ANGLE SHOULD YIELD THE TOTAL SCATTERING CROSS-SECTION IN SQ. M.   AGC00140
      SFACT=WAVE*WAVE*DENS*1.E-6/(4.*PI*PI)                             AGC00150
      PFACT=WAVE*WAVE/(PI*CTSUM*EMM*EMM)                                AGC00160
      DO 15 J=1,IT                                                      AGC00170
      SCAT=PSUM(J)*SFACT                                                AGC00180
      PSUM(J)=PSUM(J)*PFACT                                             AGC00190
C     UNCOMMENT THE FOLLOWING STMT IF U WANT SCAT FRACT,COSINES AND A   AGC00200
C     COUNTER WRITTEN ON NUNIT. NUNIT IS SET IN THE BLOCK DATA ROUTINE. AGC00210
C     WRITE (NUNIT,3) SCAT,C(J),J                                       AGC00220
   15 CONTINUE                                                          AGC00230
      IF (NINDX.GE.2) WRITE (IOUT,12)                                   AGC00240
      WRITE (IOUT,4) IDSTP,WAVE,EM,CAY,CTSUM,CSSUM,ALBDO                AGC00250
C     CONVERT AVG. CROSS-SECTIONS TO COEFFICIENTS (PER KM)              AGC00260
      CTSUM=CTSUM*1.0E-3*DENS                                           AGC00270
      CSSUM=CSSUM*1.0E-3*DENS                                           AGC00280
      CRSUM=CRSUM*1.0E-3*DENS                                           AGC00290
      WRITE (IOUT,13) CTSUM,CSSUM,CRSUM                                 AGC00300
      WRITE (IOUT,14) GNU,DENS                                          AGC00310
      IF (IT.LT.2) GO TO 21                                             AGC00320
C     WRITE PHASE FUNCTION AT SPECIFIED ANGLES                          AGC00330
6     WRITE (IOUT,1)                                                    AGC00340
      WRITE (IOUT,5)                                                    AGC00350
C     FIND ANGLES FROM COSINES                                          AGC00360
      DO 2 I=1,IT                                                       AGC00370
2     FF(I)=180.*ATAN2(SQRT(1.-C(I)**2),C(I))/PI                        AGC00380
      DO 19 J=1,IT,4                                                    AGC00390
      K=J+3                                                             AGC00400
      IF (K.GT.IT) K=IT                                                 AGC00410
   19 WRITE (IOUT,8) (C(I),FF(I),PSUM(I),I=J,K)                         AGC00420
      IF (IANG.NE.0) RETURN                                             AGC00430
C     ROUTINE GAUS GENERATES AND PRINTS THE LEGENDRE                    AGC00440
C     EXPANSION COEFS (OMEGAS) FOR THE PHASE FUNCTION.                  AGC00450
      CALL GAUS(IT)                                                     AGC00460
C     CHECK TO SEE IF SNG. SCAT. ALBEDO (ALBDO) COMPUTED DIRECTLY       AGC00470
C     FROM CROSS-SECTIONS AGREES WITH THAT FOUND FROM THE LEGENDRE      AGC00480
C     EXPANSION OF PHASE-FUNCTION.                                      AGC00490
      IF ((ABS(OL(1)-ALBDO)/ALBDO.GT.5.E-3).AND.(IEND.NE.1))            AGC00500
     1      WRITE (IOUT,20)                                            AGC00510
    1 FORMAT (//1H ,50X,14HPHASE FUNCTION/,1X,42X,31H(NORMALIZED TO 4 PIAGC00520
     + OMEGA ZERO)//)                                                   AGC00530
3     FORMAT (2(E13.7,1X),I3)                                           AGC00540
    4 FORMAT (1H1,/,41H  DISTRIBUTION  WAVELENGTH     REFRACTIVE,9X,    AGC00550
     +20HEXTINCTION X SECTION,8X,20HSCATTERING X SECTION,12X,5HALBDO/   AGC00560
     +1H ,6X, 4HTYPE,6X,9H(MICRONS),8X,5HINDEX,16X,12H(SQ MICRONS),13X, AGC00570
     +        12H(SQ MICRONS)/1H ,I9,4X,F11.4,F10.4,3H(1-,F7.4,2HI),    AGC00580
     +        7X,1PE14.7,11X,1PE14.7,12X,1PE14.7,/)                     AGC00590
    5 FORMAT (1H ,3X,4(5H   MU,2X,' ANGLE ',17H PHASE FUNCTION  )/)     AGC00600
    8 FORMAT (1H ,F9.5,F7.2,E12.5,3(3X,F9.5,F7.2,E12.5))                AGC00610
   12 FORMAT (52H THIS IS A MIXED CASE * SUBSEQUENT REFRACTIVE INDEX ,   AGC00620
     +        34HPRINT-OUTS ARE NOT GENERALLY VALID/)                   AGC00630
   13 FORMAT (1H ,10H K(EXT) = ,1PE13.7,11H; K(SCA) = ,E13.7,           AGC00640
     +        11H; K(RAD) = ,E13.7,11H ALL PER KM/)                     AGC00650
   14 FORMAT (/14H WAVENUMBER = ,1PE12.6,5H CM-1,5X,10HDENSITY = ,E12.6,AGC00660
     +        17H PARTICLES PER CC/)                                    AGC00670
   20 FORMAT (//12H *** VALUES                                          AGC00680
     +        ,55HOF ALBDO AND OL(1) DISAGREE BY MORE THAN 0.5 PERCENT **AGC00690
     +        ,34HLARGER VALUE OF 'IT' IS NEEDED ***/)                  AGC00700
```

```
      21    RETURN                                                    AGC00710
            END                                                       AGC00720
```

```
      SUBROUTINE ANGLE(PI,IANG,IT)                                      ANG00010
C     THIS ROUTINE IS TO BE USED TO REPLACE GUSET FOR THE PURPOSE       ANG00020
C     OF USING AGAUS TO DO PHASE FUNCTION CALCULATIONS AT -IT-          ANG00030
C     ANGLES BETWEEN 0 AND 180 DEGREES, RATHER THAN AT THE G-L          ANG00040
C     QUADRATURE ABSCISSA VALUES.  IT ALSO READS THE INPUT ANGLES       ANG00050
C     IF IANG=2,                                                        ANG00060
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                              ANG00070
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                       ANG00080
      RADS=PI/180.                                                      ANG00090
      DEL=180./FLOAT(IT-1)                                              ANG00100
      IF (IEO.EQ.1.OR.IEO.EQ.2.OR.IEO.EQ.5) GO TO 4                     ANG00110
      IF (IANG.EQ.2) GO TO 2                                            ANG00120
      DO 1 I=1,IT                                                       ANG00130
      W(I)=DEL*FLOAT(I-1)                                               ANG00140
1     C(I)=COS(W(I)*RADS)                                               ANG00150
      RETURN                                                            ANG00160
2     READ(5,100) (W(I),I=1,IT)                                         ANG00170
4     DO 3 I=1,IT                                                       ANG00180
3     C(I)=COS(W(I)*RADS)                                               ANG00190
      IANG=1                                                            ANG00200
      RETURN                                                            ANG00210
100   FORMAT(16F5.1)                                                    ANG00220
      END                                                               ANG00230
```

```
      FUNCTION GAMMA(X)                                                 GMA00010
C     GAMMA FUNCTION: TAKEN FROM HANDBOOK OF MATHEMATICAL FUNCTIONS,    GMA00020
C     ABRAMOWITZ AND STEGUN, NOV 1964, PP 256-257.  RECURRENCE FORMULA  GMA00030
C     6.1.16, POLYNOMIAL APPROXIMATION 6.1.35.                          GMA00040
      DATA A1,A2,A3,A4,A5                                               GMA00050
     + /-.5748646,.9512363,-.6998588,.4245549,-.1010678/                GMA00060
C     COMPUTER AND GAMMA FUNCTION LIMITS                                GMA00070
      IF (X.GT.34..OR.X.LT.0.) GO TO 3                                  GMA00080
      GSUM=1.                                                           GMA00090
      N=IFIX(X+.00001)                                                  GMA00100
C     FIND Z .LE. 1.                                                    GMA00110
      Z=X-FLOAT(N)                                                      GMA00120
C     CK FOR Z BEING INTEGER                                            GMA00130
      IF (Z.LT.1.E-04) N=N-1                                            GMA00140
      IF (Z.LT.1.E-04) Z=1.                                             GMA00150
C     COMPENSATE FOR N-1 IN FORMULA                                     GMA00160
      N=N-1                                                             GMA00170
C     IF Z .LE. 1. SKIP LOOP                                            GMA00180
      IF (N.LE.0) GO TO 2                                               GMA00190
C     RECURRENCE RELATION: G(N+Z)=(N-1+Z)*(N-2+Z)...(1+Z)*G(1+Z)        GMA00200
      DO 1 I=1,N                                                        GMA00210
      VALUE=FLOAT(I)+Z                                                  GMA00220
1     GSUM=GSUM*VALUE                                                   GMA00230
C     POLYNOMIAL APPROXIMATION: Z.LE.1                                  GMA00240
2     GAMMA=1.+A1*Z+A2*Z*Z+A3*Z*Z*Z+A4*Z*Z*Z*Z+A5*Z*Z*Z*Z*Z             GMA00250
      GAMMA=GAMMA*GSUM                                                  GMA00260
      RETURN                                                            GMA00270
3     WRITE (1,100) X                                                   GMA00280
100   FORMAT(1H ,' ***** THE VALUE OF X (',2PE11.4,') IS EITHER ',      GMA00290
     +'OUTSIDE COMPUTER LIMITS',/,' OR NEGATIVE - PGM STOPPED *****')   GMA00300
      STOP                                                              GMA00310
      END                                                               GMA00320
```

```
      SUBROUTINE GUSET(IT)                                                  GUSET010
C      THIS ROUTINE CALCULATES THE ABSCISSAE C()                            GUSET020
C      AND GAUSS-LEGENDRE WEIGHTS W() FOR NUMERICAL INTEGRATION             GUSET030
C      VIA GAUSS-LEGENDRE QUADRATURE OF ORDER N                             GUSET040
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                                  GUSET050
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                           GUSET060
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)                        GUSET070
     +,NRADI,PI,IDSTP,NKG,NHALV,NI                                          GUSET080
      N=IT                                                                  GUSET090
      TOL=1.0E-06                                                           GUSET100
      AA=2.0E+00/PI**2.0E+00                                                GUSET110
      AB=-62.E+00/(3.0E+00*PI**4.0E+00)                                     GUSET120
      AC=15116.0E+00/(15.0E+00*PI*6.0E+00)                                  GUSET130
      AD=-12554474.0E+00/(105.0E+00*PI*8.0E+00)                             GUSET140
      F(1)=1.0E+00                                                          GUSET150
      EN= FLOAT(N)                                                          GUSET160
      NP1=N+1                                                               GUSET170
      U=1.0E+00-(2.0E+00/PI)**2.0E+00                                       GUSET180
      D=1.0E+00/SQRT((EN+0.5E+00)**2.E+00+U/4.0E+00)                        GUSET190
      DO 1 I=1,N                                                            GUSET200
      S= FLOAT(I)                                                           GUSET210
      A=4.0E+00*S-1.0E+00                                                   GUSET220
      AE=AA/A                                                               GUSET230
      AF=AB/A**3.0E+00                                                      GUSET240
      AG=AC/A**5.0E+00                                                      GUSET250
      AH=AD/A**7.0E+00                                                      GUSET260
    1 R(I)=PI*(A+AE+AF+AG+AH)/4.E+00                                        GUSET270
      DO 6 K=1,N                                                            GUSET280
      X=COS(R(K)*D)                                                         GUSET290
    2 F(2)=X                                                                GUSET300
      DO 3 NN=3,NP1                                                         GUSET310
      ENN= FLOAT(NN-1)                                                      GUSET320
      F(NN)=((2.0E+00*ENN-1.E+00)*X*F(NN-1)-(ENN-1.E+00)*F(NN-2))/ENN       GUSET330
    3 IF (ABS(F(NN)).GT.1E+35) F(NN)=SIGN(1.E+35,F(NN))                     GUSET340
      PNP=EN*(F(N)-X*F(NP1))/(1.0E+00-X*X)                                  GUSET350
      XI=X-F(NP1)/PNP                                                       GUSET360
      XD= ABS(XI-X)                                                         GUSET370
      XDD=XD-TOL                                                            GUSET380
      IF (XDD) 5,5,4                                                        GUSET390
    4 X=XI                                                                  GUSET400
      GO TO 2                                                               GUSET410
    5 C(K)=X                                                                GUSET420
    6 W(K)=2.0E+00*(1.0E+00-X*X)/(EN*F(N)*EN*F(N))                          GUSET430
      DO 7 I=1,N                                                            GUSET440
      R(I)=0.00                                                             GUSET450
    7 F(I)=0.00                                                             GUSET460
      RETURN                                                                GUSET470
      END                                                                   GUSET480
```

```
      SUBROUTINE MIEGX(EMD,CAYD,ALPHAD,QTD,QSD,QRD,P,S1STRD,S2STRD,      MIEGX010
     +C,IT,PFNZRO)                                                       MIEGX020
C     THIS ROUTINE IS CURRENTLY SINGLE PRECISION COMPLEX                 MIEGX030
C     CHANGE THE VALUE OF NDIM IF YOU CHANGE THE DIM OF A IN NEXT LINE   MIEGX040
      COMPLEX  A(600),ACAPN,ZNUM,ZDEN,ZPDT,ZRPDT,ZAN,ZANP,Y,RF,RRF,      MIEGX050
     1 RRFX,WM1,FNA,FNB,TC1,FNAP,FNBP,FNAPP,FNBPP,TC2,WFN(2)             MIEGX060
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                               MIEGX070
      DIMENSION P(65),C(65)                                              MIEGX080
      DIMENSION T(4),TA(4),TB(2),TC(2),TD(2),TE(2),TF(2),TG(2)           MIEGX090
      DIMENSION ELTRMX(4,76),PI(3,76),TAU(3,76)                          MIEGX100
      EQUIVALENCE (WFN(1),TA(1)),(FNA,TB(1)),(FNB,TC(1)),(FNAP,TD(1))    MIEGX110
      EQUIVALENCE (FNBP,TE(1)),(FNAPP,TF(1)),(FNBPP,TG(1))               MIEGX120
C      THESE EQUIVALENCES ALLOW USE OF REAL AND IMAG PARTS INDIVIDUALLY  MIEGX130
      TOL = 1.E-06                                                       MIEGX140
      ITT=IT                                                             MIEGX150
      X=(ALPHAD)                                                         MIEGX160
      CAY=(CAYD)                                                         MIEGX170
      EM=(EMD)                                                           MIEGX180
      CAYE=CAY*EM                                                        MIEGX190
      QRT=0.0                                                            MIEGX200
      S=1.0                                                              MIEGX210
      RF=CMPLX(EM,-CAYE)                                                 MIEGX220
      NMX=IFIX(X*(EM+CAYE))+9                                            MIEGX230
      RRF=1.0/RF                                                         MIEGX240
      RX=1.0/X                                                           MIEGX250
      RRFX=RRF*RX                                                        MIEGX260
C     LOOP POINT FOR CALCULATING PFN AT ZERO DEGREES                     MIEGX270
      IAPXCT=0                                                           MIEGX280
   21 CONTINUE                                                           MIEGX290
C     THESE ARE THE PI AND TAU FUNCTIONS                                 MIEGX300
      DO 1 J=1,IT                                                        MIEGX310
      PI(1,J)=0.0                                                        MIEGX320
      PI(2,J)=1.0                                                        MIEGX330
      TAU(1,J)=0.0                                                       MIEGX340
      TAU(2,J)=C(J)                                                      MIEGX350
    1 CONTINUE                                                           MIEGX360
      T(1)=COS(X)                                                        MIEGX370
      T(2)=SIN(X)                                                        MIEGX380
      WM1=CMPLX(T(1),-T(2))                                              MIEGX390
      WFN(1)=CMPLX(T(2),T(1))                                            MIEGX400
      WFN(2)=RX*WFN(1)-WM1                                               MIEGX410
      T(1)=CAYE*X                                                        MIEGX420
      N=1                                                                MIEGX430
C   NDIM MUST EQUAL THE DIMENSION OF A( ).                               MIEGX440
      NDIM=600                                                           MIEGX450
      IF (NMX.LT.NDIM) NDELTA=NMX                                        MIEGX460
      IF (NMX.GT.NDIM) NDELTA=NDIM                                       MIEGX470
      NMX=0                                                              MIEGX480
      IF (N.EQ.1) GO TO 4                                                MIEGX490
    2 EN=FLOAT(N)                                                        MIEGX500
      T(1)=2.0*EN-1.0                                                    MIEGX510
      T(2)=EN-1.0                                                        MIEGX520
      T(3)=2.0*EN+1.0                                                    MIEGX530
      DO 3 J=1,IT                                                        MIEGX540
      PI1J=PI(1,J)                                                       MIEGX550
      PI2J=PI(2,J)                                                       MIEGX560
      CJ= C(J)                                                           MIEGX570
C     SWITCH FOR CALCULATING PFN AT ZERO DEGREES                         MIEGX580
      IF (IAPXCT.EQ.1) CJ=1.0                                            MIEGX590
      S2T=(1.0-CJ*CJ)                                                    MIEGX600
      PI(3,J)=(T(1)*PI2J*CJ-EN*PI1J)/T(2)                                MIEGX610
      TAU(3,J)=CJ*(PI(3,J)-PI1J)-T(1)*S2T*PI2J+TAU(1,J)                  MIEGX620
    3 CONTINUE                                                           MIEGX630
      WM1=WFN(1)                                                         MIEGX640
      WFN(1)=WFN(2)                                                      MIEGX650
      WFN(2)=T(1)*RX*WFN(1)-WM1                                          MIEGX660
    4 CONTINUE                                                           MIEGX670
C     CALCULATE RATIO OF BESSEL FNS OF CONSECUTIVE ORDER                 MIEGX680
      IF (N.LT.(NMX+1)) GO TO 9                                          MIEGX690
      NMX=NMX+NDELTA                                                     MIEGX700
```

```
      NMIN=NMX+1-NDELTA                                                    MIEGX710
      V=FLOAT(NMX)+0.50                                                    MIEGX720
      Y=RF*X                                                               MIEGX730
      ZANP=2.0/Y                                                           MIEGX740
      ZNUM=ZANP*V                                                          MIEGX750
      ZPDT=ZNUM                                                            MIEGX760
      V=V+1.0                                                              MIEGX770
      ZDEN=ZANP*V                                                          MIEGX780
      ZNUM=ZDEN-1.0/ZNUM                                                   MIEGX790
    5 ZRPDT=ZNUM/ZDEN                                                      MIEGX800
      ZPDT=ZRPDT*ZPDT                                                      MIEGX810
      IF (ABS(REAL(ZRPDT)-1.0).LT. TOL ) GO TO 7                           MIEGX820
      IF (V.LT.20000.0) GO TO 6                                            MIEGX830
      WRITE (IOUT,1000) X,EM,CAYE                                          MIEGX840
      STOP                                                                 MIEGX850
    6 V=V+1.0                                                              MIEGX860
      ZAN=ZANP*V                                                           MIEGX870
      ZNUM=ZAN-1.0/ZNUM                                                    MIEGX880
      ZDEN=ZAN-1.0/ZDEN                                                    MIEGX890
      GO TO 5                                                              MIEGX900
    7 CONTINUE                                                             MIEGX910
      J=NMX                                                                MIEGX920
    8 JJ=J-NMX+NDELTA                                                      MIEGX930
      A(JJ)=-(FLOAT(J))/Y+ZPDT                                             MIEGX940
      J=J-1                                                                MIEGX950
      IF (J.LT.NMIN) GO TO 9                                               MIEGX960
      ZPDT=(2.0*FLOAT(J)+1.0)/Y-1.0/ZPDT                                   MIEGX970
      GO TO 8                                                              MIEGX980
    9 CONTINUE                                                             MIEGX990
      J=N-NMX+NDELTA                                                       MIEGX000
      ACAPN=A(J)                                                           MIEGX010
      IF (N.GT.1) GO TO 11                                                 MIEGX020
C     THIS PART FOR N EQUAL 1 ONLY                                         MIEGX030
      TC1=ACAPN*RRF+RX                                                     MIEGX040
      TC2=ACAPN*RF+RX                                                      MIEGX050
C     SEE EQUIVALENCE STMTS FOR EXPANATION OF TA( ), ETC.                  MIEGX060
      FNA=(TC1*TA(3)-TA(1))/(TC1*WFN(2)-WFN(1))                            MIEGX070
      FNB=(TC2*TA(3)-TA(1))/(TC2*WFN(2)-WFN(1))                            MIEGX080
      FNAP=FNA                                                             MIEGX090
      FNBP=FNB                                                             MIEGX100
      T(1)=1.50                                                            MIEGX110
      TB(1)=T(1)*TB(1)                                                     MIEGX120
      TB(2)=T(1)*TB(2)                                                     MIEGX130
      TC(1)=T(1)*TC(1)                                                     MIEGX140
      TC(2)=T(1)*TC(2)                                                     MIEGX150
      DO 10 J=1,IT                                                         MIEGX160
      TAU2J=TAU(2,J)                                                       MIEGX170
      ELTRMX(1,J)=TB(1)+TC(1)*TAU2J                                        MIEGX180
      ELTRMX(2,J)=TB(2)+TC(2)*TAU2J                                        MIEGX190
      ELTRMX(3,J)=TC(1)+TB(1)*TAU2J                                        MIEGX200
      ELTRMX(4,J)=TC(2)+TB(2)*TAU2J                                        MIEGX210
   10 CONTINUE                                                             MIEGX220
      QEXT=2.0*(TB(1)+TC(1))                                               MIEGX230
      QSCAT=(TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2))/0.750        MIEGX240
      Q1STAR=0.0                                                           MIEGX250
      Q2STAR=0.0                                                           MIEGX260
      SUMRR=2.0*(TB(1)-TC(1))                                              MIEGX270
      SUMRI=2.0*(TB(2)-TC(2))                                              MIEGX280
      N=2                                                                  MIEGX290
      GO TO 2                                                              MIEGX300
   11 CONTINUE                                                             MIEGX310
      TC1=ACAPN*RRF+EN*RX                                                  MIEGX320
      TC2=ACAPN*RF+EN*RX                                                   MIEGX330
C     SEE EQUIVALENCE STMTS FOR EXPLANATION OF TA( ), ETC.                 MIEGX340
      FNA=(TC1*TA(3)-TA(1))/(TC1*WFN(2)-WFN(1))                            MIEGX350
      FNB=(TC2*TA(3)-TA(1))/(TC2*WFN(2)-WFN(1))                            MIEGX360
      T(4)=T(1)/(EN*T(2))                                                  MIEGX370
      T(2)=(T(2)*(EN+1.0))/EN                                              MIEGX380
      S=-S                                                                 MIEGX390
      SUMRR=SUMRR+S*T(3)*(TB(1)-TC(1))                                     MIEGX400
```

```
      SUMRI=SUMRI+S*T(3)*(TB(2)-TC(2))                                  MIEGX410
C     SEE LATER COMMENTS ABOUT FOLLOWING STATEMENTS                     MIEGX420
      QRTL1=QRT                                                         MIEGX430
      QRT=SUMRR*SUMRR+SUMRI*SUMRI                                       MIEGX440
C     01STAR CALCULATION                                                MIEGX450
      01STAR=01STAR+(TB(1)*TD(1)+TB(2)*TD(2)+TC(1)*TE(1)+TC(2)*TE(2))   MIEGX460
     1 *T(2)*4.0+4.0*T(4)*(TD(1)*TE(1)+TD(2)*TE(2))                     MIEGX470
      IF (N.LT.3) GO TO 12                                              MIEGX480
C     02STAR CALCULATION                                                MIEGX490
      F1=TF(1)*TF(1)+TF(2)*TF(2)+TG(1)*TG(1)+TG(2)*TG(2)                MIEGX500
      F2=TB(1)*TF(1)+TB(2)*TF(2)+TC(1)*TG(1)+TC(2)*TG(2)                MIEGX510
      F3=TD(1)*TG(1)+TD(2)*TG(2)+TE(1)*TF(1)+TE(2)*TF(2)                MIEGX520
      ENL1=EN-1.0                                                       MIEGX530
      COF1=2.50*((EN-2.0)*ENL1-3.0)*((EN-2.0)*ENL1-3.0)*(2.0*EN-3.0)/   MIEGX540
     1    ((EN-2.0)*ENL1*(2.0*EN-1.0)*(2.0*EN-5.0))                     MIEGX550
      COF2=7.50*(EN-2.0)*(EN+1.0)/(2.0*EN-1.0)                          MIEGX560
      COF3=15.0/ENL1                                                    MIEGX570
      02STAR=02STAR+COF1*F1+COF2*F2+COF3*F3                             MIEGX580
   12 CONTINUE                                                          MIEGX590
      QEXT=QEXT+T(3)*(TB(1)+TC(1))                                      MIEGX600
      T(4)=TB(1)*TB(1)+TB(2)*TB(2)+TC(1)*TC(1)+TC(2)*TC(2)              MIEGX610
      QSCAT=QSCAT+T(3)*T(4)                                             MIEGX620
      T(2)=EN*(EN+1.0)                                                  MIEGX630
      T(1)=T(3)/T(2)                                                    MIEGX640
      DO 13 J=1,IT                                                      MIEGX650
      PI3J=PI(3,J)                                                      MIEGX660
      TAU3J=TAU(3,J)                                                    MIEGX670
      ELTRMX(1,J)=ELTRMX(1,J)+T(1)*(TB(1)*PI3J+TC(1)*TAU3J)             MIEGX680
      ELTRMX(2,J)=ELTRMX(2,J)+T(1)*(TB(2)*PI3J+TC(2)*TAU3J)             MIEGX690
      ELTRMX(3,J)=ELTRMX(3,J)+T(1)*(TC(1)*PI3J+TB(1)*TAU3J)             MIEGX700
      ELTRMX(4,J)=ELTRMX(4,J)+T(1)*(TC(2)*PI3J+TB(2)*TAU3J)             MIEGX710
   13 CONTINUE                                                          MIEGX720
      IF (N.LT.5) GO TO 14                                              MIEGX730
      QRTR=ABS((QRT-QRTL1)/QRT)                                         MIEGX740
C     TEST FOR CONVERGENCE ON QEXT, QSCA, AND QRADAR                    MIEGX750
      IF ((T(4).LT. TOL ).AND.(QRTR.LT. TOL )) GO TO 16                 MIEGX760
   14 N=N+1                                                             MIEGX770
      DO 15 J=1,IT                                                      MIEGX780
      PI(1,J)=PI(2,J)                                                   MIEGX790
      PI(2,J)=PI(3,J)                                                   MIEGX800
      TAU(1,J)=TAU(2,J)                                                 MIEGX810
      TAU(2,J)=TAU(3,J)                                                 MIEGX820
   15 CONTINUE                                                          MIEGX830
      FNAPP=FNAP                                                        MIEGX840
      FNBPP=FNBP                                                        MIEGX850
      FNAP=FNA                                                          MIEGX860
      FNBP=FNB                                                          MIEGX870
      GO TO 2                                                           MIEGX880
   16 CONTINUE                                                          MIEGX890
      DO 18 J=1,IT                                                      MIEGX900
      DO 17 I=1,4                                                       MIEGX910
      T(I)=ELTRMX(I,J)                                                  MIEGX920
   17 CONTINUE                                                          MIEGX930
      ELTRMX(1,J)=T(3)*T(3)+T(4)*T(4)                                   MIEGX940
      ELTRMX(2,J)=T(1)*T(1)+T(2)*T(2)                                   MIEGX950
      ELTRMX(3,J)=T(1)*T(3)+T(2)*T(4)                                   MIEGX960
      ELTRMX(4,J)=(T(2)*T(3)-T(4)*T(1))                                 MIEGX970
      PFNZRO= (ELTRMX(1,J)+ELTRMX(2,J))/2.0                             MIEGX980
      IF (IAPXCT.EQ.1) GO TO 20                                         MIEGX990
      P(J)=PFNZRO                                                       MIEGX000
   18 CONTINUE                                                          MIEGX010
C     ELTRMX(2,J)    IS THE VERTICAL COMPONENT SCATTERING I1 (EYE1)     MIEGX020
C     ELTRMX(1,J)    IS THE HORIZONTAL COMPONENT SCATTERING I2  (EYE2)  MIEGX030
C     ELTRMX(3,J)    IS EQUIVALENT TO EYE3                              MIEGX040
C     ELTRMX(4,J)    IS EQUIVALENT TO -1.0*EYE4                         MIEGX050
      T(1)=2.0*RX*RX                                                    MIEGX060
      SGT=QEXT*T(1)                                                     MIEGX070
      SGS=QSCAT*T(1)                                                    MIEGX080
      01STAR=3.0*01STAR/(X*X*SGT)                                       MIEGX090
      02STAR=4.0*02STAR/(X*X*SGT)                                       MIEGX100
```

```
      SGR=(SUMRR*SUMRR+SUMRI*SUMRI)*RX*RX                                    MIEGX110
      QTD=(SGT)                                                              MIEGX120
      QSD=(SGS)                                                              MIEGX130
      QRD=(SGR)                                                              MIEGX140
      O1STRD=(O1STAR)                                                        MIEGX150
      O2STRD=(O2STAR)                                                        MIEGX160
C     LOOP FOR CALCULATION OF PFN AT ZERO DEGREES - FOR GPHASX               MIEGX170
      IAPXCT=1                                                               MIEGX180
      IT=1                                                                   MIEGX190
      GO TO 21                                                               MIEGX200
   20 IT=ITT                                                                 MIEGX210
      RETURN                                                                 MIEGX220
 1000 FORMAT (52H V GT 20000 ERROR IN CONTINUED FRACTIONS MIE ROUTINE,       MIEGX230
     1    11H ** ALPHA =,E12.6,6H EM = ,E12.6,7H CAY = ,E12.6/,              MIEGX240
     2 1X,54H IT IS SUGGESTED THAT TOL=1.E-06 FOR SINGLE PRECISION. )        MIEGX250
      END                                                                    MIEGX260
```

```
      SUBROUTINE WATER(WAVE,EMT,CAYT,TMCHUR,RHODEN)                          WAT00010
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                                   WAT00020
      REAL LAMBDA(169),NSUBR(169),NSUBI(169)                                 WAT00030
      DIMENSION TEMP(7),DENS(7)                                              WAT00040
C   THIS PROGRAM SEARCHES HALE AND QUERRY TABLE FOR REFRACTION VS.           WAT00050
C   WAVELENGTH (APPLIED OPTICS, VOL. 12, NO. 3,MARCH 1973,PG 555)           WAT00060
C   AND THE DENSITY VS. TEMPERATURE (HNDBK OF CHEM AND PHYS).               WAT00070
C   IF THE INPUT VALUES (TMCHUR AND WAVE) ARE NOT IN THE TABLES, A          WAT00080
C   LINEAR INTERPOLATION IS COMPUTED. VALUES ACCURATE TO THREE PLACES.       WAT00090
      INTEGER P,POINT,H                                                      WAT00100
C           TABLES FOLLOW FOR 60 LINES                                       WAT00110
      DATA LAMBDA/0.200,0.225,0.250,0.275,0.300,0.325,0.350,0.375,0.400      WAT00120
     1,      0.425,0.450,0.475,0.500,0.525,0.550,0.575,0.600,0.625,0.650     WAT00130
     2,      0.675,0.700,0.725,0.750,0.775,0.800,0.825,0.850,0.875,0.900     WAT00140
     3,      0.925,0.950,0.975,1.000,1.200,1.400,1.600,1.800,2.000,2.200     WAT00150
     4,      2.400,2.600,2.650,2.700,2.750,2.800,2.850,2.900,2.950,3.000     WAT00160
     5,      3.050,3.100,3.150,3.200,3.250,3.300,3.350,3.400,3.450,3.500     WAT00170
     6,      3.600,3.700,3.800,3.900,4.000,4.100,4.200,4.300,4.400,4.500     WAT00180
     7,      4.600,4.700,4.800,4.900,5.000,5.100,5.200,5.300,5.400,5.500     WAT00190
     8,      5.600,5.700,5.800,5.900,6.000,6.100,6.200,6.300,6.400,6.500     WAT00200
     9,      6.600,6.700,6.800,6.900,7.000,7.100,7.200,7.300,7.400,7.500     WAT00210
     X,      7.600,7.700,7.800,7.900,8.000,8.200,8.400,8.600,8.800,9.000     WAT00220
     1,      9.200,9.400,9.600,9.800,10.00,10.50,11.00,11.50,12.00,12.50     WAT00230
     2,      13.00,13.50,14.00,14.50,15.00,15.50,16.00,16.50,17.00,17.50     WAT00240
     3,      18.00,18.50,19.00,19.50,20.00,21.00,22.00,23.00,24.00,25.00     WAT00250
     4,      26.00,27.00,28.00,29.00,30.00,32.00,34.00,36.00,38.00,40.00     WAT00260
     5,      42.00,44.00,46.00,48.00,50.00,60.00,70.00,80.00,90.00,100.0     WAT00270
     6,      110.0,120.0,130.0,140.0,150.0,160.0,170.0,180.0,190.0,200.0     WAT00280
     7,     /                                                                WAT00290
      DATA NSUBR/1.396,1.373,1.362,1.354,1.349,1.346,1.343,1.341,1.339       WAT00300
     1,      1.338,1.337,1.336,1.335,1.334,1.333,1.333,1.332,1.332,1.331     WAT00310
     2,      1.331,1.331,1.330,1.330,1.330,1.329,1.329,1.329,1.328,1.328     WAT00320
     3,      1.328,1.327,1.327,1.327,1.324,1.321,1.317,1.312,1.306,1.296     WAT00330
     4,      1.279,1.242,1.219,1.188,1.157,1.142,1.149,1.201,1.292,1.371     WAT00340
     5,      1.426,1.467,1.483,1.478,1.467,1.450,1.432,1.420,1.410,1.400     WAT00350
     6,      1.385,1.374,1.364,1.357,1.351,1.346,1.342,1.338,1.334,1.332     WAT00360
     7,      1.330,1.330,1.330,1.328,1.325,1.322,1.317,1.312,1.305,1.298     WAT00370
     8,      1.289,1.277,1.262,1.248,1.265,1.319,1.363,1.357,1.347,1.339     WAT00380
     9,      1.334,1.329,1.324,1.321,1.317,1.314,1.312,1.309,1.307,1.304     WAT00390
     X,      1.302,1.299,1.297,1.294,1.291,1.286,1.281,1.275,1.269,1.262     WAT00400
     1,      1.255,1.247,1.239,1.229,1.218,1.185,1.153,1.126,1.111,1.123     WAT00410
     2,      1.146,1.177,1.210,1.241,1.270,1.297,1.325,1.351,1.376,1.401     WAT00420
     3,      1.423,1.443,1.461,1.476,1.480,1.487,1.500,1.511,1.521,1.531     WAT00430
     4,      1.539,1.545,1.549,1.551,1.551,1.546,1.536,1.527,1.522,1.519     WAT00440
     5,      1.522,1.530,1.541,1.555,1.587,1.703,1.821,1.886,1.924,1.957     WAT00450
     6,      1.966,2.004,2.036,2.056,2.069,2.081,2.094,2.107,2.119,2.130     WAT00460
     7,     /                                                                WAT00470
C     DATA NSUBI/1.10E-7,4.90E-8,3.35E-8,2.35E-8,1.60E-8,1.08E-8,6.50E-9WAT00480
CC    1,  3.50E-9,1.86E-9,1.30E-9,1.02E-9,9.35E-10,1.00E-9,1.32E-9,1.96E-9WAT00490
CC    2,   3.60E-9,1.09E-8,1.39E-8,1.64E-8,2.23E-8,3.35E-8,9.15E-8,1.56E-7WAT00500
CC    3,   1.48E-7,1.25E-7,1.82E-7,2.93E-7,3.91E-7,4.86E-7,1.06E-6,2.93E-6WAT00510
CC    4,   3.48E-6,2.89E-6,9.89E-6,1.38E-4,8.55E-5,1.15E-4,1.10E-3,2.89E-4WAT00520
CC    5,   9.56E-4,3.17E-3,6.70E-3,1.90E-2,5.90E-2,1.15E-1,1.85E-1,2.68E-1WAT00530
CC    6,   2.98E-1,2.72E-1,2.40E-1,1.92E-1,1.35E-1,9.24E-2,6.10E-2,3.68E-2WAT00540
CC    7,   2.61E-2,1.95E-2,1.32E-2,9.40E-3,5.15E-3,3.60E-3,3.40E-3,3.80E-3WAT00550
CC    8,   4.60E-3,5.62E-3,6.88E-3,8.45E-3,1.03E-2,1.34E-2,1.47E-2,1.57E-2WAT00560
CC    9,   1.50E-2,1.37E-2,1.24E-2,1.11E-2,1.01E-2,9.80E-3,1.03E-2,1.16E-2WAT00570
CC    X,   1.42E-2,2.03E-2,3.30E-2,6.22E-2,1.07E-1,1.31E-1,8.80E-2,5.70E-2WAT00580
CC    1,   4.49E-2,3.92E-2,3.56E-2,3.37E-2,3.27E-2,3.22E-2,3.20E-2,3.20E-2WAT00590
CC    2,   3.21E-2,3.22E-2,3.24E-2,3.26E-2,3.28E-2,3.31E-2,3.35E-2,3.39E-2WAT00600
CC    3,   3.43E-2,3.51E-2,3.61E-2,3.72E-2,3.85E-2,3.99E-2,4.15E-2,4.33E-2WAT00610
CC    4,   4.54E-2,4.79E-2,5.08E-2,6.62E-2,9.68E-2,1.42E-1,1.99E-1,2.59E-1WAT00620
CC    5,   3.05E-1,3.43E-1,3.70E-1,3.88E-1,4.02E-1,4.14E-1,4.22E-1,4.28E-1WAT00630
CC    6,   4.29E-1,4.29E-1,4.26E-1,4.21E-1,4.14E-1,4.04E-1,3.93E-1,3.82E-1WAT00640
CC    7,   3.73E-1,3.67E-1,3.61E-1,3.56E-1,3.50E-1,3.44E-1,3.38E-1,3.33E-1WAT00650
CC    8,   3.28E-1,3.24E-1,3.29E-1,3.43E-1,3.61E-1,3.85E-1,4.09E-1,4.36E-1WAT00660
CC    9,   4.62E-1,4.88E-1,5.14E-1,5.87E-1,5.76E-1,5.47E-1,5.36E-1,5.32E-1WAT00670
CC    X,   5.31E-1,5.26E-1,5.14E-1,5.00E-1,4.95E-1,4.96E-1,4.97E-1,4.99E-1WAT00680
CC    1,   5.01E-1,5.04E-1/                                                  WAT00690
C     ALTERNATE FORM OF ABOVE DATA STMT DUE TO EXCESS CONTINUATION CARDS     WAT00700
```

```
      DATA (NSUBI(I),I=1,143)                                                  WAT00710
     +          /1.10E-7,4.90E-8,3.35E-8,2.35E-8,1.60E-8,1.08E-8,6.50E-9WAT00720
     1, 3.50E-9,1.86E-9,1.30E-9,1.02E-9,9.35E-10,1.00E-9,1.32E-9,1.96E-9WAT00730
     2,  3.60E-9,1.09E-8,1.39E-8,1.64E-8,2.23E-8,3.35E-8,9.15E-8,1.56E-7WAT00740
     3,  1.48E-7,1.25E-7,1.82E-7,2.93E-7,3.91E-7,4.86E-7,1.06E-6,2.93E-6WAT00750
     4,  3.48E-6,2.89E-6,9.89E-6,1.38E-4,8.55E-5,1.15E-4,1.10E-3,2.89E-4WAT00760
     5,  9.56E-4,3.17E-3,6.70E-3,1.90E-2,5.90E-2,1.15E-1,1.85E-1,2.68E-1WAT00770
     6,  2.98E-1,2.72E-1,2.40E-1,1.92E-1,1.35E-1,9.24E-2,6.10E-2,3.68E-2WAT00780
     7,  2.61E-2,1.95E-2,1.32E-2,9.40E-3,5.15E-3,3.60E-3,3.40E-3,3.80E-3WAT00790
     8,  4.60E-3,5.62E-3,6.88E-3,8.45E-3,1.03E-2,1.34E-2,1.47E-2,1.57E-2WAT00800
     9,  1.50E-2,1.37E-2,1.24E-2,1.11E-2,1.01E-2,9.80E-3,1.03E-2,1.16E-2WAT00810
     X,  1.42E-2,2.03E-2,3.30E-2,6.22E-2,1.07E-1,1.31E-1,8.80E-2,5.70E-2WAT00820
     1,  4.49E-2,3.92E-2,3.56E-2,3.37E-2,3.27E-2,3.22E-2,3.20E-2,3.20E-2WAT00830
     2,  3.21E-2,3.22E-2,3.24E-2,3.26E-2,3.28E-2,3.31E-2,3.35E-2,3.39E-2WAT00840
     3,  3.43E-2,3.51E-2,3.61E-2,3.72E-2,3.85E-2,3.99E-2,4.15E-2,4.33E-2WAT00850
     4,  4.54E-2,4.79E-2,5.08E-2,6.62E-2,9.68E-2,1.42E-1,1.99E-1,2.59E-1WAT00860
     5,  3.05E-1,3.43E-1,3.70E-1,3.88E-1,4.02E-1,4.14E-1,4.22E-1,4.28E-1WAT00870
     6,  4.29E-1,4.29E-1,4.26E-1,4.21E-1,4.14E-1,4.04E-1,3.93E-1,3.82E-1WAT00880
     7, 3.73E-1,3.67E-1,3.61E-1,3.56E-1,3.50E-1,3.44E-1,3.38E-1,3.33E-1/WAT00890
      DATA (NSUBI(I),I=144,169) /                                              WAT00900
     8    3.28E-1,3.24E-1,3.29E-1,3.43E-1,3.61E-1,3.85E-1,4.09E-1,4.36E-1WAT00910
     9,   4.62E-1,4.88E-1,5.14E-1,5.87E-1,5.76E-1,5.47E-1,5.36E-1,5.32E-1WAT00920
     X,   5.31E-1,5.26E-1,5.14E-1,5.00E-1,4.95E-1,4.96E-1,4.97E-1,4.99E-1WAT00930
     1,   5.01E-1,5.04E-1/                                                    WAT00940
      DATA TEMP/273.,278.,283.,288.,293.,298.,303./,DENS/                      WAT00950
     1      0.999841,0.999965,0.999700,0.999099,0.998203,0.997044,            WAT00960
     2      0.995994/                                                         WAT00970
      EMT=0.0                                                                  WAT00980
      CAYT=0.0                                                                 WAT00990
      POINT=0                                                                  WAT01000
      H=0                                                                      WAT01010
      IF (WAVE.LT.0.2.OR.WAVE.GT.200.0.OR.TMCHUR.LT.273.OR.                    WAT01020
     1    TMCHUR.GT.303.0) GO TO 11                                            WAT01030
C   BINARY SEARCH                                                            WAT01040
      L=1                                                                      WAT01050
      H=125                                                                    WAT01060
    1 POINT=((L+H)/2)                                                          WAT01070
      TEST=ABS(LAMBDA(POINT)-WAVE)                                             WAT01080
      IF (TEST.LE.0.0001) GO TO 4                                              WAT01090
      IF (WAVE.GT.LAMBDA(POINT)) GO TO 2                                       WAT01100
      H=POINT                                                                  WAT01110
      GO TO 3                                                                  WAT01120
    2 L=POINT+1                                                                WAT01130
    3 IF (L.NE.H) GO TO 1                                                      WAT01140
      L=L-1                                                                    WAT01150
C   INTERPOLATION ROUTINE                                                    WAT01160
      EMT=NSUBR(L)+(NSUBR(L+1)-NSUBR(L))*((WAVE-LAMBDA(L))/(LAMBDA(L+1)-WAT01170
     1       LAMBDA(L)))                                                     WAT01180
      CAYT=NSUBI(L)+(NSUBI(L+1)-NSUBI(L))*((WAVE-LAMBDA(L))/(LAMBDA(L+1)WAT01190
     1       -LAMBDA(L)))                                                    WAT01200
      GO TO 5                                                                  WAT01210
    4 CONTINUE                                                                 WAT01220
      EMT=NSUBR(POINT)                                                         WAT01230
      CAYT=NSUBI(POINT)                                                        WAT01240
C   SEARCH TEMP VS DENS                                                      WAT01250
    5 IF (TMCHUR.LT.273.0.OR.TMCHUR.GT.303.0) GO TO 11                         WAT01260
      L=1                                                                      WAT01270
      H=7                                                                      WAT01280
    6 P=((L+H)/2)                                                              WAT01290
      TESTT=ABS(TEMP(P)-TMCHUR)                                                WAT01300
      IF (TESTT.LE.0.0001) GO TO 9                                             WAT01310
      IF (TMCHUR.GT.TEMP(P)) GO TO 7                                           WAT01320
      H=P                                                                      WAT01330
      GO TO 8                                                                  WAT01340
    7 L=P+1                                                                    WAT01350
    8 IF (L.NE.H) GO TO 6                                                      WAT01360
      L=L-1                                                                    WAT01370
      RHODEN=DENS(L)+(DENS(L+1)-DENS(L))*(TMCHUR-TEMP(L))/(TEMP(L+1)-          WAT01380
     1       TEMP(L))                                                        WAT01390
      GO TO 10                                                                 WAT01400
```

```
 9 RHODEN=DENS(P)                                                        WAT 1410
10 CONTINUE                                                              WAT01420
   GO TO 13                                                              WAT01430
11 WRITE (IOUT,12) TMCHUR,WAVE                                           WAT01440
12 FORMAT (10H TEMP. OF ,E12.6,14H OR WAVEL. OF ,E12.6,14H BEYOND RANWAT01450
  1GE ,24HOF DATA IN WATER ROUTINE/22H EXECUTION TERMINATED )            WAT01460
   STOP                                                                  WAT01470
13 RETURN                                                                WAT01480
   END                                                                   WAT01490
```

```
      SUBROUTINE GAUS(IT)                                                GAUS0010
      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)        GAUS0020
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                               GAUS0030
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                        GAUS0040
      WRITE (IOUT,11)                                                    GAUS0050
C      INITIALIZE ARRAY PC( ) USED FOR RUNNING SUMMATION                 GAUS0060
      DO 1 I=1,IT                                                        GAUS0070
      OL(I)=0.                                                           GAUS0080
    1 PC(I)=0.E+00                                                       GAUS0090
C     LOOPS  2 AND 3 CALCULATE EXPANSION COEFS. FOR FUNCTION PSUM( )     GAUS0100
C     VIA GAUSS-LEGENDRE QUADRATURE. THE COEFS. GO INTO ARRAY OL( )      GAUS0110
      DO 3 I=1,IT                                                        GAUS0120
      COF=W(I)*PSUM(I)                                                   GAUS0130
      PLM1=C(I)                                                          GAUS0140
      PLM2=1.                                                            GAUS0150
      DO 2 LL=1,IT                                                       GAUS0160
      L=LL-1                                                             GAUS0170
      PL=PLM2                                                            GAUS0180
      IF (LL.EQ.2) PL=PLM1                                               GAUS0190
      IF (LL.LE.2) GO TO 2                                               GAUS0200
      PL=2.*C(I)*PLM1-PLM2-(C(I)*PLM1-PLM2)/FLOAT(L)                     GAUS0210
      PLM2=PLM1                                                          GAUS0220
      PLM1=PL                                                            GAUS0230
2     OL(LL)=OL(LL)+COF*PL*(FLOAT(L)+.5)                                 GAUS0240
3     CONTINUE                                                           GAUS0250
      DO 7 I=1,IT                                                        GAUS0260
      II=I-1                                                             GAUS0270
7     WRITE (IOUT,8) II,OL(I)                                            GAUS0280
    8 FORMAT (1H ,20X,I6,20X,1PE14.7)                                    GAUS0290
11    FORMAT (1H /,1X,25X,1HL,20X,16HL-TH COEFFICIENT)                   GAUS0300
      RETURN                                                             GAUS0310
      END                                                                GAUS0320
```

```
      SUBROUTINE DIMER(NGO)                                                 DIM00010
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                                  DIM00020
      GO TO (1,2,3),NGO                                                     DIM00030
1     WRITE(IOUT,101)                                                       DIM00040
101   FORMAT(1H ,'**** THE INPUT VALUE OF IT IS GREATER '                   DIM00050
     1,'THAN THE ARRAY DIMENSIONS',/,'CHANGE THE DIMENSIONS OF THE ',       DIM00060
     2'FOLLOWING ARRAYS IN SUBS AND COMMON',/,'COMMON BLOCK',10X,           DIM00070
     3'ARRAY(S)',/,'     AGXM',10X,'   C,H,OLT',/,'     PT2',11X,          DIM00080
     4'PC,OL,RMS,PSUM,PSUMT,P',/,'ALSO CHANGE THE VALUE OF JIMCK(1) '       DIM00090
     5,'IN THE DATA STMT TO AGREE WITH THE NEW',/,' DIMENSION LIMITS',      DIM00100
     6/,1H ,55HTHE SECOND INDEX ON ARRAYS PI, TAU, AND ELTRMX IN MIEGX      DIM00110
     7,1H ,56HMUST ALSO BE CHANGED AND ARRAYS P AND C CHANGED AS WELL.)     DIM00120
      STOP                                                                  DIM00130
2     WRITE (IOUT,102)                                                      DIM00140
102   FORMAT(1H ,'**** TOO MANY PARTICLE RADII FOR DIMENSION LIMITS:',      DIM00150
     1'IN SUBS AND COMMON CHANGE THE FOLLOWING ',                           DIM00160
     2'ARRAYS',/,'COMMON BLOCK',10X,'ARRAY(S)',/,'     PT1',11X,'F,R,FF',  DIM00170
     3/,'ARRAYS F AND R MUST BE CONSISTENT WITH THE FOLLOWING: ARRAY ',     DIM00180
     4'SIZE = 1 + 2**JDIMCK(2)',/,'ARRAY FF MUST BE DIMENSIONED TO ONE',    DIM00190
     5' MORE THAN ARRAYS F AND R',/,'ALSO CHANGE THE VALUE OF JDIMCK(2)'    DIM00200
     6,' IN THE DATA STATEMENT')                                            DIM00210
      STOP                                                                  DIM00220
3     WRITE (IOUT,103)                                                      DIM00230
103   FORMAT (1H ,' THE DIMENSIONS OF F AND'                                DIM00240
     1,'R DO NOT AGREE WITH THE FOLLOWING:',/,'SIZE=1+2**JDIMCK(2)  ',      DIM00250
     2'WHERE JDIMCK(2) APPEARS IN THE DATA STATEMENT',/,'ALSO DIMENSION'    DIM00260
     3,' ARRAY FF TO BE ONE MORE THAN ARRAYS F AND R')                      DIM00270
      STOP                                                                  DIM00280
      END                                                                   DIM00290
```

```
      BLOCK DATA                                                        BLK00010
C     IF YOU CHANGE THE DIMENSIONS MAKE SURE THAT YOU ALSO CHANGE THE   BLK00020
C     DATA STATEMENT CONTAINING JDIMCK();                               BLK0003
C     JDIMCK(1)=ORDER OF QUADRATURE (INPUT IT),I.E. MAX SIZE OF DIMENSIOBLK00040
C     JDIMCK(2):USED IN COMPUTATION OF SIZE OF ARRAYS F AND R,          BLK00050
C     SIZE=1+2**JDIMCK(2), ARRAY FF SHOULD BE ONE MORE THAN F AND R.    BLK00060
C     JIMCK(3) WILL BE CALCULATED.                                      BLK00070
C     THE OTHER ARRAYS ARE REUSED, SO                                   BLK00080
C     CHECK THE SUBROUTINE CALLS AND COMMON BLOCKS TO SEE IF            BLK00090
C     ARRAYS HAVE BEEN RENAMED WHEN U CHANGE DIMENSIONS.                BLK00100
      COMMON /AGXM/ C(65),W(65),OLT(65),JDIMCK(3)                       BLK00110
      COMMON /PT1/ F(513),R(513),DR(8),RR(9),FF(514)                    BLK00120
     +,NRADI,PI,IDSTP,NKG,NHALV,NI                                      BLK00130
      COMMON /PT2/ PC(65),OL(65),RMS(65),PSUM(65),PSUMT(65),P(65)       BLK00140
      COMMON /IO/ IOIN,IOUT,NUNIT,IEO,NEOU                              BLK00150
      DATA IOIN,IOUT,NUNIT,IEO,NEOU /5,6,3*0/                           BLK00160
      DATA JDIMCK /65,9,0/                                              BLK00170
C     EOSAEL OPTION:  65 PREDETERMINED ANGLES                           BLK00180
      DATA W                                                            BLK00190
     +     /  0.0,  0.5,  1.0,  2.0,  3.0,  4.0,  5.0,  6.0,  8.0, 10.0,BLK00200
     +       12.0, 14.0, 16.0, 18.0, 20.0, 24.0, 28.0, 32.0, 36.0, 40.0,BLK00210
     +       44.0, 48.0, 52.0, 56.0, 60.0, 64.0, 68.0, 72.0, 76.0, 80.0,BLK00220
     +       84.0, 88.0, 92.0, 96.0,100.0,104.0,108.0,112.0,116.0,120.0,BLK00230
     +      124.0,128.0,132.0,136.0,140.0,142.0,144.0,146.0,148.0,150.0,BLK00240
     +      152.0,154.0,156.0,158.0,160.0,162.0,164.0,166.0,168.0,170.0,BLK00250
     +      172.0,174.0,176.0,178.0,180.0/                             BLK00260
      END                                                               BLK00270
```

```
      PROGRAM FLASH1                                                        DRIV0010
      COMMON/IOUNT/IOIN,IOOUT                                               DRIV0020
      COMMON/CONST/PI                                                       DRIV0030
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,          DRIV0040
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,        DRIV0050
     *             TIME,ISRC,RCTSEE,RCTLOC                                  DRIV0060
      PI=3.14159                                                            DRIV0070
      IOIN=5                                                                DRIV0080
      IOOUT=6                                                               DRIV0090
C**** NOTE ICHK=-1 DEFAULTS USETR INPUT TO WAVE1                            DRIV0100
      ICHK=0                                                                DRIV0110
      WAVE1=.55                                                             DRIV0120
      CALL FLASH(WAVE1,ICHK)                                                DRIV0130
      WRITE (IOOUT,100) WAVE1,ICHK                                          DRIV0140
100   FORMAT (5X,13HEOSAEL OUTPUT,/,                                        DRIV0150
     *5X,13HWAVELENGTH = ,F6.1,1X,11HMICROMETERS,/,                         DRIV0160
     *5X,7HICHK = ,I4)                                                      DRIV0170
      STOP                                                                  DRIV0180
      END
```

```
      SUBROUTINE FLASH(WAVE1,ICHK)                                          FLASH010
C*************************************************************************FLASH020
C*                    PROGRAM FLASH                                       *FLASH030
C*                      EOSAEL80                                          *FLASH040
C*************************************************************************FLASH050
      DIMENSION FLASH1(3),FLASH2(3),IC1(3),IC2(3),IGUN(3)                   FLASH060
      COMMON/IOUNT/IOIN,IOOUT                                               FLASH070
      COMMON/CONST/PI                                                       FLASH080
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,          FLASH090
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,        FLASH100
     *             TIME,ISRC,RCTSEE,RCTLOC                                  FLASH110
      DATA FLASH1/3.85,2.55,0.0/                                            FLASH120
      DATA FLASH2/0.15,0.15,0.0/                                            FLASH130
      DATA IC1/2HHO,2HT-,2HUN/                                              FLASH140
      DATA IC2/2HW.,2H55,2HKN/                                              FLASH150
      DATA IGUN/105,100,0/                                                  FLASH160
   98 FORMAT(//,21X,30H*****PROGRAM FLASH OUTPUT*****,/,21X,40(2H--))       FLASH170
   99 FORMAT(21X,20H*****END OF RUN*****,/,21X,40(2H--))                    FLASH180
  100 FORMAT(1H1,20X,40(2H**),/,21X,1H*,34X,13HPROGRAM FLASH,31X,1H*,/,     FLASH190
     *21X,1H*,37X,8HEOSAEL80,33X,1H*,/,21X,40(2H**))                        FLASH200
  101 FORMAT(//,21X,15H*****INPUT*****,/,21X,40(2H--))                      FLASH210
  102 FORMAT(21X,14HSCENARIO DATA:,/,21X,14HREFERENCE TIME,1X,F8.3,         FLASH220
     *1X,3HSEC)                                                             FLASH230
  103 FORMAT(21X,9HOBSERVER:,18X,7HTARGET:,20X,7HSOURCE:)                   FLASH240
  104 FORMAT(24X,6HX(OBS),3X,F6.1,2H M,10X,6HX(TAR),3X,F6.1,2H M,10X,       FLASH250
     *6HX(SRC),3X,F6.1,2H M,/,                                              FLASH260
     *       24X,6HY(OBS),3X,F6.1,2H M,10X,6HY(TAR),3X,F6.1,2H M,10X,       FLASH270
     *6HY(SRC),3X,F6.1,2H M,/,                                              FLASH280
     *       24X,6HZ(OBS),3X,F6.1,2H M,10X,6HZ(TAR),3X,F6.1,2H M,10X,       FLASH290
     *6HZ(SRC),3X,F6.1,2H M)                                                FLASH300
  105 FORMAT(48X,11HORIENTATION,1X,F6.1,1X,3HDEG,8X,                        FLASH310
     *10HEVENT TIME,1X,F6.3,1X,3HSEC,/,67X,                                 FLASH320
     *10HCCW X-AXIS)                                                        FLASH330
  106 FORMAT(21X,25HDETECTOR CHARACTERISTICS:,/,                            FLASH340
     *       24X,13HFIELD OF VIEW,12X,F6.1,1X,7HDEGREES,/,                  FLASH350
     *       24X,10HWAVELENGTH,15X,F6.1,1X,11HMICROMETERS,/,                FLASH360
     *       24X,21HRESOLUTION CRITERIA--,/,                                FLASH370
     *       26X,17H(A) FOR DETECTION,6X,F6.3,1X,12HMILLIRADIANS,/,         FLASH380
     *       26X,15H(B) FOR LOCK ON,8X,F6.3,1X,12HMILLIRADIANS,/,           FLASH390
     *       24X,24HRECOVERY TIME (R=100M)--,/,                             FLASH400
     *       26X,17H(A) FOR DETECTION,6X,F6.1,1X,7HSECONDS,/,               FLASH410
     *       26X,15H(B) FOR LOCK ON,8X,F6.1,1X,7HSECONDS)                   FLASH420
  107 FORMAT(21X,23HTARGET CHARACTERISTICS:,/,                              FLASH430
     *       24X,6HLENGTH,19X,F6.1,1X,6HMETERS,/,                          FLASH440
     *       24X,5HWIDTH,20X,F6.1,1X,6HMETERS,/,                           FLASH450
     *       24X,6HHEIGHT,19X,F6.1,1X,6HMETERS,/,                          FLASH460
     *       24X,19HEXPOSURE CRITERIA--,/,                                 FLASH470
     *       26X,17H(A) FOR DETECTION,6X,F6.1,1X,7HPERCENT,/,              FLASH480
     *       26X,15H(B) FOR LOCK ON,8X,F6.1,1X,7HPERCENT)                  FLASH490
  108 FORMAT(21X,23HSOURCE CHARACTERISTICS:,/,                              FLASH500
     *       24X,4HTYPE,22X,I4,2HMM,1X,2A2,/,                              FLASH510
     *       24X,22HFLASH (VISIBLE) RADIUS,4X,F6.3,1X,6HMETERS,/,          FLASH520
     *       24X,24HFLASH (VISIBLE) DURATION,2X,F6.3,1X,7HSECONDS)         FLASH530
  109 FORMAT(24X,27H*****PROGRAM FLASH END*****,/,21X,40(2H--),1H1)         FLASH540
      IWRIT=1                                                               FLASH550
      IFLAG=0                                                               FLASH560
C*****READ IN DATA                                                          FLASH570
    1 WRITE(IOOUT,100)                                                      FLASH580
      CALL DATRD(IWRIT,IFLAG)                                               FLASH590
      IF(ICHK.EQ.-1)WAVE=WAVE1                                              FLASH600
      WAVE1=WAVE                                                            FLASH610
      IF(IFLAG.EQ.4)GO TO 9999                                              FLASH620
      JSRC=ISRC                                                             FLASH630
      IF(ISRC.LT.1.OR.ISRC.GT.2)JSRC=3                                      FLASH640
      FLASHR=FLASH1(JSRC)                                                   FLASH650
      FLASHT=FLASH2(JSRC)                                                   FLASH660
      WRITE(IOOUT,101)                                                      FLASH670
      WRITE(IOOUT,102)TIME                                                  FLASH680
      WRITE(IOOUT,103)                                                      FLASH690
      WRITE(IOOUT,104)(ROBS(I),RTAR(I),RSRC(I),I=1,3)                       FLASH700
```

```
      WRITE(IOOUT,105)TARDEG,TIMGUN                                     FLASH710
      WRITE(IOOUT,106)FOV,WAVE,RADSEE,RADLOC,RCTSEE,RCTLOC              FLASH720
      WRITE(IOOUT,107)TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC                FLASH730
      WRITE(IOOUT,108)IGUN(JSRC),IC1(JSRC),IC2(JSRC),FLASHR,FLASHT      FLASH740
      CALL GETIM(FLASHR,FLASHT,ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)           FLASH750
      WRITE(IOOUT,98)                                                   FLASH760
      CALL DATWT(ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)                         FLASH770
      WRITE(IOOUT,99)                                                   FLASH780
      GO TO 1                                                           FLASH790
 9999 WRITE(IOOUT,109)                                                  FLASH800
C*****DEFINE EOSAEL OUTPUT                                             FLASH810
      ICHK=ISEE+1                                                       FLASH820
      STOP                                                              FLASH830
      END                                                               FLASH840
```

```
      SUBROUTINE DATRD(IWRIT,IFLAG)                                     DTRD0010
C/*****************************************************************/DTRD0020
C/*                    SUBROUTINE DATRD                             */DTRD0030
C/*                      FLASH MODULE                               */DTRD0040
C/*                       EOSAEL80                                  */DTRD0050
C/*****************************************************************/DTRD0060
C*****THIS SUBROUTINE READS INPUT DATA IN EXACTLY THE SAME FORMAT AS    DTRD0070
C     THE SMOKE(EOSAEL) AND GRNAD(EOSAEL) MODULES                     DTRD0080
C     INPUTS                                                          DTRD0090
C       EACH CARD BEGINS WITH A 4 LETTER IDENTIFIER IN COL 1-4,       DTRD0100
C       FOLLOWED BY AS MANY (REAL) FIELDS AS NEEDED, 10 COL.          DTRD0110
C       PER FIELD BEGINNING IN COL 11.  THE CARDS ARE NOT ORDER       DTRD0120
C       DEPENDENT.                                                    DTRD0130
C       NAME                IGNORED                                   DTRD0140
C       SCEN                                                          DTRD0150
C                 TIME          SCENARIO REFERENCE TIME(SEC)          DTRD0160
C                 ISRC          SOURCE TYPE CODE (1=105MM 2=100MM)    DTRD0170
C                                                                     DTRD0180
C       OBSC                                                          DTRD0190
C                 ROBS(3)       OBSERVER COORDINATES (X,Y,Z METERS)   DTRD0200
C       TARC                                                          DTRD0210
C                 RTAR(3)       TARGET COORDINATES (X,Y,Z METERS)     DTRD0220
C                 TARDEG        TARGET ORIENTATION COUNTERCLOCKWISE TO DTRD0230
C                               POSITIVE X AXIS                       DTRD0240
C       SRCC                                                          DTRD0250
C                 RSRC(3)       CORDINATES OF FLASH CENTER (X,Y,Z METERS)DTRD0260
C                 TIMGUN        TIME OF INIATION OF GUNFLASH          DTRD0270
C       DCHR                                                          DTRD0280
C                 FOV           DETECTOR FIELD OF VIEW (DEGREES)      DTRD0290
C                 WAVE          DETECTOR WAVELENGTH (MICROMETERS)     DTRD0300
C                 RADSEE        ANGULAR RESOLUTION NEEDED TO DETECT   DTRD0310
C                               (MILLIRADIANS)                        DTRD0320
C                 RADLOC        ANGULAR RESOLUTION NEEDED TO LOCK ON  DTRD0330
C                               (MILLIRADIANS)                        DTRD0340
C                 RCTSEE        RECOVERY TIME AT 100 METERS FOR       DTRD0350
C                               DETECTION (SECONDS)                   DTRD0360
C                 RCTLOC        RECOVERY TIME AT 100 METERS FOR       DTRD0370
C                               LOCK ON (SECONDS)                     DTRD0380
C       TCHR                                                          DTRD0390
C                 TARLEN        TARGET LENGTH (METERS)                DTRD0400
C                 TARWID        TARGET WIDTH (METERS)                 DTRD0410
C                 TARHGT        TARGET HEIGHT (METERS)                DTRD0420
C                 PCTSEE        FRACTION OF EXPOSURE NEEDED FOR DETECTIONDTRD0430
C                               (PERCENT)                             DTRD0440
C                 PCTLOC        FRACTION OF EXPOSURE NEEDED FOR LOCK ON DTRD0450
C                               (PERCENT)                             DTRD0460
C       GO            SIGNIFIES END OF THIS RUN, BUT NOT END OF INPUT DTRD0470
C       DONE          END OF JOB.                                     DTRD0480
C*****************************************************************    DTRD0490
      COMMON/IOUNT/IOIN,IOOUT                                         DTRD0500
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,    DTRD0510
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,  DTRD0520
     *             TIME,ISRC,RCTSEE,RCTLOC                            DTRD0530
      DIMENSION IR(18),IR1(2),R1(7),INAME(35)                         DTRD0540
      DATA IR/2HNA,2HME,2HSC,2HEN,2HOB,2HSC,2HTA,2HRC,2HSR,2HCC,      DTRD0550
     *        2HDC,2HHR,2HTC,2HHR,2HGO,2H  ,2HDO,2HNE/                DTRD0560
  100 FORMAT(21X,20H*****CARD INPUT*****,/,21X,40(2H--))              DTRD0570
  101 FORMAT(2A2,6X,7F10.3)                                           DTRD0580
  102 FORMAT(21X,2A2,6X,7F10.3)                                       DTRD0590
  103 FORMAT(2A2,6X,35A2)                                             DTRD0600
  104 FORMAT(21X,2A2,6X,35A2)                                         DTRD0610
C*****                                                                DTRD0620
C***** BEGINNING OF READ LOOP                                         DTRD0630
C*****                                                                DTRD0640
      IF(IWRIT.EQ.0)GO TO 6                                           DTRD0650
      WRITE(IOOUT,100)                                                DTRD0660
    6 DO 10 I=1,9                                                     DTRD0670
      IF(I.EQ.9)GO TO 90                                              DTRD0680
      IF(IFLAG.GT.0)GO TO 4                                           DTRD0690
      IFLAG=1                                                         DTRD0700
      READ(IOIN,103)IR1(1),IR1(2),(INAME(J),J=1,35)
```

```
      IF(IWRIT.EQ.0)GO TO 4                                                  DTRD0710
      WRITE(IOOUT,104)IR1(1),IR1(2),(INAME(J),J=1,35)                        DTRD0720
    4 READ(IOIN,101) IR1(1),IR1(2),(R1(J),J=1,7)                             DTRD0730
      IF(IWRIT.EQ.0) GO TO 5                                                 DTRD0740
      WRITE(IOOUT,102) IR1(1),IR1(2),(R1(J),J=1,7)                           DTRD0750
    5 IF(IR1(1).EQ.IR(17).AND.IR1(2).EQ.IR(18)) GO TO 998                    DTRD0760
C*****                                                                       DTRD0770
C***  RELATING INPUT DATA TO VARIABLE NAMES.                                 DTRD0780
C*****                                                                       DTRD0790
          IF(IR1(1).EQ.IR(1).AND.IR1(2).EQ.IR(2))   GO TO 10                 DTRD0800
          IF(IR1(1).EQ.IR(3).AND.IR1(2).EQ.IR(4))   GO TO 20                 DTRD0810
          IF(IR1(1).EQ.IR(5).AND.IR1(2).EQ.IR(6))   GO TO 30                 DTRD0820
          IF(IR1(1).EQ.IR(7).AND.IR1(2).EQ.IR(8))   GO TO 40                 DTRD0830
          IF(IR1(1).EQ.IR(9).AND.IR1(2).EQ.IR(10))  GO TO 50                 DTRD0840
          IF(IR1(1).EQ.IR(11).AND.IR1(2).EQ.IR(12)) GO TO 60                 DTRD0850
          IF(IR1(1).EQ.IR(13).AND.IR1(2).EQ.IR(14)) GO TO 70                 DTRD0860
          IF(IR1(1).EQ.IR(15).AND.IR1(2).EQ.IR(16)) GO TO 9999               DTRD0870
C*****                                                                       DTRD0880
C     ERROR CAUTION FOR INVALID DATA CARD                                    DTRD0890
C*****                                                                       DTRD0900
      IFLAG=2                                                                DTRD0910
      WRITE(IOOUT,105)                                                       DTRD0920
  105 FORMAT(21X,35H*****CAUTION***** INVALID DATA CARD)                     DTRD0930
      GO TO 10                                                               DTRD0940
   20       TIME=R1(1)                                                      DTRD0950
            ISRC=IFIX(R1(2))                                                DTRD0960
      GO TO 10                                                               DTRD0970
   30       ROBS(1)=R1(1)                                                   DTRD0980
            ROBS(2)=R1(2)                                                   DTRD0990
            ROBS(3)=R1(3)                                                   DTRD1000
      GO TO 10                                                               DTRD1010
   40       RTAR(1)=R1(1)                                                   DTRD1020
            RTAR(2)=R1(2)                                                   DTRD1030
            RTAR(3)=R1(3)                                                   DTRD1040
            TARDEG=R1(4)                                                    DTRD1050
      GO TO 10                                                               DTRD1060
   50       RSRC(1)=R1(1)                                                   DTRD1070
            RSRC(2)=R1(2)                                                   DTRD1080
            RSRC(3)=R1(3)                                                   DTRD1090
            TIMGUN=R1(4)                                                    DTRD1100
      GO TO 10                                                               DTRD1110
   60       FOV=R1(1)                                                       DTRD1120
            WAVE=R1(2)                                                      DTRD1130
            RADSEE=R1(3)                                                    DTRD1140
            RADLOC=R1(4)                                                    DTRD1150
            RCTSEE=R1(5)                                                    DTRD1160
            RCTLOC=R1(6)                                                    DTRD1170
      GO TO 10                                                               DTRD1180
   70       TARLEN=R1(1)                                                    DTRD1190
            TARWID=R1(2)                                                    DTRD1200
            TARHGT=R1(3)                                                    DTRD1210
            PCTSEE=R1(4)                                                    DTRD1220
            PCTLOC=R1(5)                                                    DTRD1230
   10 CONTINUE                                                             DTRD1240
      GO TO 9999                                                           DTRD1250
C*****                                                                     DTRD1260
C*****CAUTION FOR TOO MANY CARDS                                           DTRD1270
C*****                                                                     DTRD1280
   90 WRITE(IOOUT,106)                                                     DTRD1290
      IFLAG=3                                                              DTRD1300
  106 FORMAT(21X,17H*****CAUTION*****,/,                                   DTRD1310
     *21X,56HMORE THAN 10 DATA CARDS ENTERED--REMAINING CARDS IGNORED)     DTRD1320
      GO TO 9999                                                           DTRD1330
  998 IFLAG=4                                                              DTRD1340
 9999 RETURN                                                               DTRD1350
      END                                                                  DTRD1360
```

```
      SUBROUTINE DATWT(ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)                    DAT00010
C*********************************************************************  DAT00020
C*                    SUBROUTINE DATWT                               *  DAT00030
C*                      FLASH MODULE                                 *  DAT00040
C*                      EOSAEL80                                     *  DAT00050
C*********************************************************************  DAT00060
      COMMON/IOUNT/IOIN,IOOUT                                            DAT00070
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,       DAT00080
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,     DAT00090
     *             TIME,ISRC,RCTSEE,RCTLOC                               DAT00100
      DIMENSION I1(3),I2(3)                                              DAT00110
      DATA I1/2HNO,2HYE,2HPA/                                            DAT00120
      DATA I2/2H  ,2HS ,2HRT/                                            DAT00130
      IANS1=ISEE+1                                                       DAT00140
      IF(ISEE.EQ.3)IANS1=3                                               DAT00150
      IF(ILOC.EQ.0)IANS2=2                                               DAT00160
      IF(ILOC.EQ.1)IANS2=1                                               DAT00170
      IF(ILOC.EQ.2)IANS2=3                                               DAT00180
      IF(ILOC.EQ.0)IANS2=2                                               DAT00190
      IF(ILOC.EQ.3)IANS2=3                                               DAT00200
   82 FORMAT(21X,36HDETECTION EXPOSURE CRITERIA DEFEATED)                DAT00210
   83 FORMAT(21X,38HDETECTION RESOLUTION CRITERIA DEFEATED)              DAT00220
   92 FORMAT(21X,34HLOCK ON EXPOSURE CRITERIA DEFEATED)                  DAT00230
   93 FORMAT(21X,36HLOCK ON RESOLUTION CRITERIA DEFEATED)                DAT00240
   97 FORMAT(21X,17HTARGET OBSCURED ?,1X,2A2,/,                          DAT00250
     *       21X,15HTARGET LOCKED ?,1X,2A2)                              DAT00260
   98 FORMAT(21X,10HTIME LEFT:,4X,F8.3,1X,7HSECONDS,/,                   DAT00270
     *       21X,13HTIME NO LOCK:,1X,F8.3,1X,7HSECONDS)                  DAT00280
   99 FORMAT(21X,25HPROGRAM FLASH--STOP CODE:,1X,3I1)                    DAT00290
  100 FORMAT(21X,11HSOURCE TYPE,1X,I2,1X,12HUNIDENTIFIED)                DAT00300
  101 FORMAT(21X,17HINPUT WAVELENGTH:,1X,F6.1,1X,11HMICROMETERS,         DAT00310
     *25H IS OUT OF RANGE OF MODEL)                                      DAT00320
  102 FORMAT(21X,37HGUNFLASH HAS NOT OCCURED YET: TIME = ,1X,F8.3,/,42X,DAT00330
     *16HTIME OF FLASH = ,1X,F8.3)                                       DAT00340
  103 FORMAT(21X,38HFLASH IS NOT IN DETECTOR FIELD OF VIEW)              DAT00350
  104 FORMAT(21X,24HFLASH IS BEHIND TARGET--,                            DAT00360
     *15H NO OBSCURATION)                                                DAT00370
  105 FORMAT(21X,24HNORMAL PROGRAM EXECUTION)                            DAT00380
      WRITE(IOOUT,99)ISTOP,ISEE,ILOC                                     DAT00390
      IF(ISTOP.GT.1)GO TO 1                                              DAT00400
      IF(ISTOP.GT.0)GO TO 10                                             DAT00410
      WRITE(IOOUT,100)ISRC                                               DAT00420
      GO TO 9999                                                         DAT00430
   10 WRITE(IOOUT,101)WAVE                                               DAT00440
      GO TO 9999                                                         DAT00450
    1 WRITE(IOOUT,97)I1(IANS1),I2(IANS1),I1(IANS2),I2(IANS2)             DAT00460
      IF(ISEE.LT.2.AND.ILOC.LT.2)GO TO 11                                DAT00470
      IF(ISEE.EQ.2)WRITE(IOOUT,82)                                       DAT00480
      IF(ISEE.EQ.3)WRITE(IOOUT,83)                                       DAT00490
      IF(ILOC.EQ.2)WRITE(IOOUT,92)                                       DAT00500
      IF(ILOC.EQ.3)WRITE(IOOUT,93)                                       DAT00510
      GO TO 9998                                                         DAT00520
   11 IGO=ISTOP-1                                                        DAT00530
      GO TO (2,3,4,5,6,7,8,9),IGO                                        DAT00540
    2 WRITE(IOOUT,102)TIME,TIMGUN                                        DAT00550
      GO TO 9999                                                         DAT00560
    3 WRITE(IOOUT,103)                                                   DAT00570
      GO TO 9999                                                         DAT00580
    4 WRITE(IOOUT,103)                                                   DAT00590
      GO TO 9999                                                         DAT00600
    5 WRITE(IOOUT,104)                                                   DAT00610
      GO TO 9999                                                         DAT00620
    6 WRITE(IOOUT,103)                                                   DAT00630
      GO TO 9999                                                         DAT00640
    7 WRITE(IOOUT,103)                                                   DAT00650
      GO TO 9999                                                         DAT00660
    8 WRITE(IOOUT,103)                                                   DAT00670
      GO TO 9999                                                         DAT00680
    9 WRITE(IOOUT,105)                                                   DAT00690
 9998 WRITE(IOOUT,98)TIMLEF,TIMNOL                                       DAT00700
```

```
 9999 RETURN                                                    DAT00710
      END                                                       DAT00720
```

```
      SUBROUTINE GETIM(FLASHR,FLASHT,ISTOP,ISEE,ILOC,TIMLEF,TIMNOL)     GTM00010
C*************************************************************************GTM00020
C*                    SUBROUTINE GETIM                                   *GTM00030
C*                       FLASH MODULE                                    *GTM00040
C*                       EOSAEL80                                        *GTM00050
C*************************************************************************GTM00060
      REAL LOCSTO,LOCFRA                                                 GTM00070
      COMMON/IDATA/ROBS(3),RTAR(3),TARDEG,RSRC(3),TIMGUN,FOV,WAVE,       GTM00080
     *             RADSEE,RADLOC,TARLEN,TARWID,TARHGT,PCTSEE,PCTLOC,     GTM00090
     *             TIME,ISRC,RCTSEE,RCTLOC                               GTM00100
      COMMON/CONST/PI                                                    GTM00110
C*****ALL DISTANCES IN METERS ANGLES CONVERTED TO RADIANS(CCW A-AXES      GTM00120
      DIMENSION DOTVEC(3),RHTVEC(3),RHFVEC(3),RFLASH(3)                  GTM00130
C*****SCALAR(A,B)=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)                         GTM00140
      SCALAR(A1,A2,A3,B1,B2,B3)=A1*B1+A2*B2+A3*B3                        GTM00150
C*****DEFAULT AND CONVERT INPUT                                         GTM00160
      ANGGUN=0.0                                                         GTM00170
      ELEGUN=0.0                                                         GTM00180
      GUNHT=0.0                                                          GTM00190
      GUNLEN=0.0                                                         GTM00200
      TARHHT=TARHGT/2.0                                                  GTM00210
      ANGTAR=TARDEG*(PI/180.0)                                           GTM00220
      HAFFOV=(FOV/2.0)*(PI/180.0)                                        GTM00230
      SEESTO=RADSEE/1000.0                                               GTM00240
      LOCSTO=RADLOC/1000.0                                               GTM00250
      SEEFRA=PCTSEE/100.0                                                GTM00260
      LOCFRA=PCTLOC/100.0                                                GTM00270
      RADFL=FLASHR                                                       GTM00280
      DURTIM=FLASHT                                                      GTM00290
      DURSEE=10.0*RCTSEE                                                 GTM00300
      DURLOC=10.0*RCTLOC                                                 GTM00310
      REACFA=SEEFRA-LOCFRA                                               GTM00320
C*****INITIALIZE FLAGS *****                                            GTM00330
      AGINSE=0.0                                                         GTM00340
      AGINLO=0.0                                                         GTM00350
      TIMLEF=0.0                                                         GTM00360
      TIMNOL=0.0                                                         GTM00370
      TIMGON=TIME-TIMGUN                                                 GTM00380
      ISEE=0                                                             GTM00390
      ILOC=0                                                             GTM00400
      ISTOP=0                                                            GTM00410
      IF(ISRC.LT.1.OR.ISRC.GT.2)GO TO 9999                               GTM00420
      ISTOP=1                                                            GTM00430
      IWAVE=0                                                            GTM00440
      IF(WAVE.GE.0.40.AND.WAVE.LE.0.70)IWAVE=1                           GTM00450
      IF(WAVE.GE.8.00.AND.WAVE.LE.12.0)IWAVE=2                           GTM00460
      IF(IWAVE.EQ.0)GO TO 9999                                           GTM00470
      ISTOP=2                                                            GTM00480
      IF(TIMGON.LT.0.)GO TO 9999                                         GTM00490
      TIMLEF=DURTIM-TIMGON                                               GTM00500
C*****TIMLEF IS DURATION LEFT OF FLASH                                  GTM00510
      TIMNOL =TIMLEF+REACFA*DURTIM                                       GTM00520
C*****TIMNOL IS DURATION OF LOCK                                        GTM00530
      FLSHIF=RADFL                                                       GTM00540
C*****CALCULATE FLASH COORDINATES FROM GIVEN COORDINATES                 GTM00550
      RFLASH(1)=RSRC(1)+FLSHIF*COS(ANGGUN)*COS(ELEGUN)                   GTM00560
      RFLASH(2)=RSRC(2)+FLSHIF*SIN(ANGGUN)*COS(ELEGUN)                   GTM00570
      RFLASH(3)=RSRC(3)+FLSHIF*SIN(ELEGUN) + GUNHT                       GTM00580
C*****DEFINE DIRECTION OF TARGET UNIT VECTOR                            GTM00590
      DOTVEC(1)=COS(ANGTAR)                                              GTM00600
      DOTVEC(2)=SIN(ANGTAR)                                              GTM00610
      DOTVEC(3)=0.                                                       GTM00620
C*****DEFINE OBSERVER-TARGET,OBSERVER-FLASH VECTORS                     GTM00630
      DO 10 I=1,3                                                        GTM00640
      RHTVEC(I)=RTAR(I)-ROBS(I)                                          GTM00650
      RHFVEC(I)=RFLASH(I)-ROBS(I)                                        GTM00660
   10 CONTINUE                                                           GTM00670
C*****ADD HALF TARGET HEIGHT                                            GTM00680
      RHTVEC(3)=RHTVEC(3)+TARHHT                                         GTM00690
C*****FIND LENGTHS                                                      GTM00700
```

```
      A1=RHTVEC(1)                                                          GTM00710
      A2=RHTVEC(2)                                                          GTM00720
      A3=RHTVEC(3)                                                          GTM00730
      B1=RHFVEC(1)                                                          GTM00740
      B2=RHFVEC(2)                                                          GTM00750
      B3=RHFVEC(3)                                                          GTM00760
      RHT=SQRT(SCALAR(A1,A2,A3,A1,A2,A3))                                   GTM00770
      RHF=SQRT(SCALAR(B1,B2,B3,B1,B2,B3))                                   GTM00780
C*****FIND COSINES BETWEEN VECTORS                                          GTM00790
      C1=DOTVEC(1)                                                          GTM00800
      C2=DOTVEC(2)                                                          GTM00810
      C3=DOTVEC(3)                                                          GTM00820
      GAMMAD= SCALAR(C1,C2,C3,B1,B2,B3)/RHF                                 GTM00830
      GAMMAT= SCALAR(A1,A2,A3,B1,B2,B3)/(RHT*RHF)                           GTM00840
      W     = SCALAR(C1,C2,C3,A1,A2,A3)/RHT                                 GTM00850
C*****FIND TRIG FCNS OF THETA=HALF-ANGLE OF FLASH CONE                      GTM00860
      SINTH=RADFL/RHF                                                       GTM00870
      COSTH=SQRT(1.0-SINTH*SINTH)                                           GTM00880
      COSSQ=COSTH*COSTH                                                     GTM00890
      COSINE=ABS(W)                                                         GTM00900
      SINE=SQRT(1.0-COSINE**2)                                              GTM00910
C*****CALCULATE CYLINDRICAL TARGET LONGEST DIMENSION AS SEEN IN              GTM00920
C*****PLANE PERPENDICULAR TO RHFVEC                                         GTM00930
      TARDIM=TARLEN*SINE+TARWID*COSINE                                      GTM00940
      HAFDIM=TARDIM/2.                                                      GTM00950
      THETA=ATAN2(SQRT(1.0-COSTH**2),COSTH)                                 GTM00960
      THETAT=ATAN2(SQRT(1.0-GAMMAT**2),GAMMAT)                              GTM00970
      DELTH=THETAT-THETA                                                    GTM00980
      ISTOP=3                                                               GTM00990
      IF(DELTH.GT.HAFFOV)GO TO 9999                                         GTM01000
C*****IF OUT OF FIELD OF VIEW,RETURN.                                       GTM01010
      DISPLC=RHT*SIN(DELTH)                                                 GTM01020
      PROJRH=RHT*GAMMAT                                                     GTM01030
      ISTOP=4                                                               GTM01040
      IF(DISPLC.GT.HAFDIM)GO TO 99                                          GTM01050
      ISTOP=5                                                               GTM01060
      IF(PROJRH.LT.RHF.AND.IWAVE.LT.2)GO TO 99                              GTM01070
C*****NO OBSCURATION IF TARGET IN FRONT OF FLASH SO RETURN                  GTM01080
C*****FROM HERE ON, GAMMAT NECESSARILY POSITIVE                             GTM01090
C*****FIND INTERSECTIONS OF FLASH CONE WITH DOTVEC EXTENDED FROM TARGET     GTM01100
C*****GET COEFFICIENT OF QUADRATIC EQUATION FOR DISTANCE ALONG DOTVEC       GTM01110
C*****FROM TARGET TO FLASH CONE/RHT                                         GTM01120
      A  =COSSQ  -GAMMAD**2                                                 GTM01130
      BD2=COSSQ*W-GAMMAD*GAMMAT                                             GTM01140
      C  =COSSQ  -GAMMAT**2                                                 GTM01150
      B=BD2*2.                                                              GTM01160
C*****IF DISCRIMINANT NEGATIVE, NO INTERSECTION                             GTM01170
      DISCRM=BD2*BD2-A*C                                                    GTM01180
      ISTOP=6                                                               GTM01190
      IF(DISCRM.LT.0.)GO TO 99                                              GTM01200
C*****IF A=0,QUADRATIC FORMULA BLOWS UP.ACTUALLY HAVE LINEAR EQN.           GTM01210
      IF(ABS(A).GT.1.E-30)GOTO20                                            GTM01220
      IF(ABS(B).LT.1.E-30)B=SIGN(1.E-30,B)                                  GTM01230
      SPL=-C/B*RHT                                                          GTM01240
      ZPL=SPL*GAMMAD+PROJRH                                                 GTM01250
      ISTOP=7                                                               GTM01260
      IF(ZPL.LE.0.) GO TO 99                                                GTM01270
C*****REJECT IF SOLE INTERSEC IS W NEG CONE SHEET.MEANINGLESS               GTM01280
      DPL=ABS(SPL)*SINE                                                     GTM01290
      IF(GAMMAT-COSTH)14,14,12                                              GTM01300
C*****IF TAR OUTSIDE FLASH CONE GO TO 14,OTHERWISE 12                       GTM01310
   12 OBSCUR=HAFDIM+AMIN1(DPL,HAFDIM)                                       GTM01320
      GO TO 30                                                              GTM01330
   14 OBSCUR=AMAX1(0.,HAFDIM-DPL)                                           GTM01340
      GO TO 30                                                              GTM01350
   20 ROOT=SQRT(DISCRM)                                                     GTM01360
      ROOT=SIGN(ROOT,A)                                                     GTM01370
C*****ABOVE NOT REALLY NEEDED BUT NICER TO HAVE SPL.GT.SMI-SEE BELOW        GTM01380
      SPL=(-BD2+ROOT)/A *RHT                                                GTM01390
      SMI=(-BD2-ROOT)/A *RHT                                                GTM01400
```

```
C*****SPL,SMI ARE 2 LENGTHS ALONG DOTVEC OF INTERSEC PTS W FLASH CONE    GTM01410
C*****ZPL,ZMI ARE Z COORDINATES OF INTERSECTIONS(ON FLASH CONE AXIS)     GTM01420
      ZPL=SPL*GAMMAD+PROJRH                                              GTM01430
      ZMI=SMI*GAMMAD+PROJRH                                              GTM01440
      ISTOP=8                                                            GTM01450
      IF(ZPL.LE.0..AND.ZMI.LE.0.)GO TO 99                                GTM01460
C*****REJECT IF BOTH INTERCEPTS IN NEGATIVE CONE                        GTM01470
C*****DPL,DMI ARE PROJECNS OF SPL,SMI PERPEN TO RHTVEC                  GTM01480
C*****THEIR MAGNITUDES ARE LIMITED TO HAFDIM                            GTM01490
      DPL=SPL*SINE                                                       GTM01500
      DPL=AMIN1(DPL,HAFDIM)                                              GTM01510
      DPL=AMAX1(DPL,-HAFDIM)                                             GTM01520
      DMI=SMI*SINE                                                       GTM01530
      DMI=AMIN1(DMI,HAFDIM)                                              GTM01540
      DMI=AMAX1(DMI,-HAFDIM)                                             GTM01550
      OBSCUR=ABS(DPL-DMI)                                                GTM01560
      IF(ZPL*ZMI.GT.0.)GOTO30                                            GTM01570
C*****SKIP AROUND  UNLESS BOTH SHEETS OF CONE INVOLVED                  GTM01580
      IF(ZPL.LE.0.)DPL=SIGN(HAFDIM,DPL)                                  GTM01590
      IF(ZMI.LE.0.)DMI=SIGN(HAFDIM,DMI)                                  GTM01600
      OBSCUR=TARDIM-ABS(DPL-DMI)                                         GTM01610
C*****ABOVE BRANCH RARE.INTERSEC W BOTH CONE SHEETS.NEG SHEET IGNORE.    GTM01620
   30 CONTINUE                                                           GTM01630
      SEEN=TARDIM-OBSCUR                                                 GTM01640
      ISTOP=9                                                            GTM01650
   98 IF(IWAVE.EQ.2)GO TO 1                                              GTM01660
      CALL VSBLC(ISEE,ILOC,SEEN,TARDIM,RHT,TIMLEF,SEESTO,LOCSTO,         GTM01670
     *           SEEFRA,LOCFRA)                                          GTM01680
      GO TO 99                                                           GTM01690
    1 CALL IRBLC(RHF,ISEE,ILOC,AGINSE,AGINLO,DURSEE,DURLOC,TIMGON)       GTM01700
   99 CONTINUE                                                           GTM01710
      TIMLEF =AMAX1(TIMLEF,AGINSE)                                       GTM01720
      TIMNOL=AMAX1(TIMNOL,AGINLO)                                        GTM01730
 9999 RETURN                                                             GTM01740
      END                                                                GTM01750
```

```
      SUBROUTINE IRBLC(RHF,ISEE,ILOC,AGINSE,AGINLO,DURSEE,DURLOC,TIMGON)IRBLC010
C*************************************************************************IRBLC020
C*                    SUBROUTINE IRBLC                                   *IRBLC030
C*                       FLASH MODULE                                    *IRBLC040
C*                       EOSAEL80                                        *IRBLC050
C*************************************************************************IRBLC060
      INTEGER LOCIRM                                                      IRBLC070
      SEEIRM=4.0                                                          IRBLC080
      LOCIRM=5.0                                                          IRBLC090
      SQRTRH=SQRT(RHF)                                                    IRBLC100
      AGINSE=DURSEE/SQRTRH                                                IRBLC110
      AGINLO=DURLOC/SQRTRH                                                IRBLC120
C*****ABOVE WHOLLY EMPIRICAL FROM FIT TO TV TAPES AT 2 RANGES              IRBLC130
      IF(AGINSE.GT.SEEIRM) AGINSE=SEEIRM                                  IRBLC140
      IF(AGINLO.GT.LOCIRM) AGINLO=LOCIRM                                  IRBLC150
      IF(AGINSE.GT.TIMGON) ISEE=1                                         IRBLC160
      IF(AGINLO.GT.TIMGON) ILOC=1                                         IRBLC170
      RETURN                                                              IRBLC180
      END                                                                 IRBLC190
```

```
      SUBROUTINE VSBLC(ISEE,ILOC,SEEN,TARDIM,RHT,TIMLEF,SEESTO,LOCSTO,   VSBLC010
     *                 SEEFRA,LOCFRA)                                    VSBLC020
C*************************************************************************VSBLC030
C*                    SUBROUTINE VSBLC                                   *VSBLC040
C*                      FLASH MODULE                                     *VSBLC050
C*                      EOSAEL80                                         *VSBLC060
C*************************************************************************VSBLC070
      REAL LOCCRI,LOCFRA,LOCSTO                                          VSBLC080
C*****SEEFRA IS FRACTION THAT MUST BE SEEN,LIKEWISE FOR LOCK              VSBLC090
C*****SEESTO IS MIN ANGLE THAT MUST BE SEEN,LIKEWISE FOR LOCK            VSBLC100
      SEEANG=SEEN/RHT                                                    VSBLC110
      TARANG=TARDIM/RHT                                                  VSBLC120
C*****DEFINE SEEN ANGLE AND TARGET ANGLE                                 VSBLC130
      SEECRI=AMIN1(TARANG,SEESTO)                                        VSBLC140
      LOCCRI=AMIN1(TARANG,LOCSTO)                                        VSBLC150
      IF(TIMLEF.LE.0.)GO TO 9999                                         VSBLC160
      IF(SEEANG.LT.SEEFRA*TARANG)GO TO 48                                VSBLC170
      ISEE=2                                                             VSBLC180
      GO TO 50                                                           VSBLC190
   48 IF(SEEANG.LT.SEECRI)GO TO 49                                       VSBLC200
      ISEE=1                                                             VSBLC210
      GO TO 50                                                           VSBLC220
   49 CONTINUE                                                           VSBLC230
      ISEE=3                                                             VSBLC240
   50 CONTINUE                                                           VSBLC250
      IF(SEEANG.LT.LOCFRA*TARANG)GO TO 98                                VSBLC260
      ILOC=2                                                             VSBLC270
      GO TO 9999                                                         VSBLC280
   98 IF(SEEANG.LT.LOCCRI)GO TO 99                                       VSBLC290
      ILOC=1                                                             VSBLC300
      GO TO 9999                                                         VSBLC310
   99 CONTINUE                                                           VSBLC320
      ILOC=3                                                             VSBLC330
 9999 RETURN                                                             VSBLC340
      END                                                                VSBLC350
```

```
EORUN   ,     4.0
WAVL    ;    10.6    10.6
VIS     ,     7.0
TURB
XSCALE  ,     3.0
SMOKE
DRTRAN
LZTRAN
CLTRAN  ,     3.0
SCREEN
OVRCST  ,     2.0
CLIMAT  ;     1.0     1.0     5.0     6.0    -1.0     1.0
GRNADE
GO
PARM          0.1016   1.3E-04       0.0     400.0       5.0     512.0
CN1              1.0   6.0E-13   1.0E-14   1.3E-14   2.7E-14   5.0E-14
V1               1.0      0.93      0.93      0.93      0.93      0.93
DVRV          0.400     500.0
CN2              1.0   6.0E-13   1.0E-14   1.3E-14   2.7E-14   5.0E-14
V2               1.0      0.93      0.93      0.93      0.93      0.93
GO
END
FOG      1.
HORZ    0.4
GO
FOG      2.
SLNH      0.133,56.3
GO
FOG      3.
HORZ    0.4
GO
END
MUNC        0.0      -50.0         0.
OBSC       200.        0.         2.0
TARC      -200.        0.         2.0
BART         5.      180.         5.        90.
OUTP         0.        0.         0.        0.
BURN        10.        0.         0.        0.        0.
GO
BURN         4.        0.         0.        0.        0.
BART         5.      250.         5.        90.
GO
DONE
END
MET1            4.       2.    286.0       2.      2.0     0.0
MET2            1.      53.
SOIL            2.     0.0     0.15
CHAR            3.     6.8      0.0
EXPL            1.      1.       1.      0.0     0.0     0.0     0.0     0.0     0.0
OBSC       200.0       0.0      2.0
TRNC      -200.0       0.0      2.0
RECC       200.0       0.0      2.0
TIMS         1.0      71.0      2.0
GO
DONE
END
              8.55       15.      0.4
END
SEEK,        0.2,       0.0,      0.6
TARG,       -0.2,       0.0,    0.002
CLST,       0.20,      0.40
GO
SEEK,      0.133,       0.0,      0.5
GO
SEEK,        0.0,       0.0,      0.3
GO
END
 1 1 1
TARV              1.0       2.0      0.24       2.3       81.       2.      .000
SENS              .99        8.        1.        0.        0.        1.
```

```
GO
SENS              .90          8.          1.          0.          0.          1.
GO
SENS              .75          8.          1.          0.          0.          1.
GO
SENS              .50          8.          1.          0.          0.          1.
GO
SENS              .10          8.          1.          0.          0.          1.
GO
DONE
SCRN         15.     400.      0.4          0.          90.         74.         2.
DONE
END
OPOS,    -0.0667,          0.0,          0.2
CLDS,        0.2,        200.0,         40.0,         0.7,         1.0
SPOS,       -0.2,          0.0,        0.002
BKGR,       50.0
GRND,       50.0
TEMP,        9.8
GO
OPOS,    -0.1333,          0.0,          0.1
GO
END
NAME
OUTP
OBSC       200.0       0.0          2.0
MUNC      -200.0         0.0        2.0          95.0        100.0        10.0
TARC      -200.0       +40.         2.0
BART         5.0       400.0        5.0          90.0
MUNT         1.0         0.793     14.3          1.0          0.0          4.7          0.07
METR        50.0         2.0       220.0         4.0         20.0          0.0          0.0
EXTC
BURN
MISC
GO
DONE
END
WAVL     ,     1.06     1.06
VIS      ,      5.0      5.0      4.0
BASCAT
FCLOUD
GO
PART,          1.,       5000.,          1.
SORC,       -0.2,          0.,       -0.098,         90.,          0.,         50.
DETR,        0.2,          0.,       -0.098,         90.,        180.,          1.,          1.
CLDS,         .1,          .2,           .1,          0.,          0.,          0.,          5.
GRND,       -0.1,         0.5
PULS,        .33,          0.
GO
END
CPOS,        0.0,          0.0,          0.1
RPOS,        0.2,          0.0,        0.002
SPOS,       -0.2,          0.0,        0.002
AXES,        0.1,          0.2,          0.1
CLDS,        5.0,         0.95,          4.0,         1.0,         9.8
ATMO,        2.0,          9.8
BKGR,        0.5,         50.0
SANG,       80.0,          0.0
LUND,        0.0
GO
END
FREQ     ,     35.0     35.0
NMMW     ,      2.0
GO
PATH,        0.4
ATMO,       15.0     1013.25        6.44
FOGD,        0.5
RAIN,        5.0
GO
ATMO,       -1.0     1015.2         7.7
```

```
RAIN,          0.0
SNOW,          5.0
GO
END
WVNUM        2010.   2710.   2.
LT4M
RESF
SPOT
GO
ENVR             3,          2,          2,          4,          1.          1.
EMIS              .100+01     .283+03     .950+00     .295+03
ATM               .650+02      0.0         .0
TARG              .400+00     .450+02     .900+02     .450+02
REFL            0.5         0.5           .0          .500-01     .0          0.
SENS              .200-02     .900+02     .270+03     .100+01
GO
I2
          3.5,        0.78
          3.6,        0.83
          3.7,        0.87
          3.8,        0.92
          3.9,        0.96
          4.0,        0.98
          4.1,        0.97
          4.3,        0.96
          4.5,        0.95
          4.7,        0.94
          4.9,        0.93
          5.0,        0.93
END
   4   2  1   0   1   0   0   0   0       0.000       0.000
       0.002       0.002       0.000       0.400       0.000       1.000
   4
   4   2  1   0   1   0   0   0   1       0.000       0.000
   0
END
```

```
**********************************
*                                *
*   ELECTRO-OPTICAL SYSTEMS      *
*                                *
*     ATMOSPHERIC EFFECTS        *
*                                *
*          LIBRARY               *
*                                *
**********************************
```

INDIVIDUAL MODULES SELECTED

TURB
XSCALE
SMOKE
DRTRAN
LZTRAN
CLTRAN
SCREEN
OVRCST
GRNADE
CLIMATE

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	943.396	943.396
WAVELENGTH(MICRONS)	10.600	10.600
FREQUENCY(GHZ)	28301.886	28301.886

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

CLASS 1 = FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS 2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLASS 3 = FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLASS 4 = FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS 5 = DUST WITH VIS LT 3 KM.
CLASS 6 = DUST WITH VIS GE 3 KM.
CLASS 7 = DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM.
CLASS 8 = DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM.
CLASS 9 = DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM.
CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM.
CLASS 11 = SNOW WITH VIS LT 1 KM.
CLASS 12 = SNOW WITH 1 LE VIS LT 3 KM.
CLASS 13 = SNOW WITH 3 LE VIS LT 7 KM.
CLASS 14 = SNOW WITH VIS GE 7 KM.
CLASS 15 = NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 16 = NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 17 = VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 18 = VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLASS 19 = CEILING HEIGHT LT 300 M.
CLASS 20 = CEILING HEIGHT LT 1000 M.
CLASS 21 = NO CEILING.
CLASS 22 = ALL CONDITIONS COMBINED.

EOSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS — DURING SPRING AT 03-09 (LST).

CLASS NO.	FREQCY CLASS (%)	MEAN TEMP (C)	MEAN DP (C)	MEAN AH (GM/CU.M)	MEAN RH (%)	MEAN VIS (KM)	MEAN PRESS (MB)	MEAN/STDEV WNDVEL (MPS)	MEAN CLDHT (KM)	MEAN/STDEV CLDCVR (%)	FREQCY A (%)	FREQCY B (%)	FREQCY C (%)	FREQCY D (%)	FREQCY E (%)	FREQCY F (%)
1	4.5	3.5	3.2	6.3	98.2	.365	1016.7	1.9/ 1.7	.929	74.4/ 39.2	.0	2.6	12.9	[illegible]	1.4	14.2
2	8.2	4.1	3.2	6.4	93.8	1.945	1016.1	2.3/ 1.7	1.818	60.0/ 40.4	.0	4.1	19.0	[illegible]	[illegible]	19.2
3	17.9	5.1	3.3	6.4	89.0	4.502	1016.4	2.8/ 2.1	2.124	57.8/ 39.4	.0	4.3	15.4	[illegible]	[illegible]	19.7
4	9.6	5.8	3.3	6.3	84.5	9.224	1016.6	3.4/ 2.2	1.911	56.2/ 38.6	.0	3.1	12.5	[illegible]	11.[illegible]	[illegible]
5	.0	.0	.0	.0	.0	.000	.0	.0/ .0	.000	.0/ .0	.0	.0	.0	[illegible]	[illegible]	[illegible]
6	.0	.8	-1.4	5.5	85.5	10.617	1012.3	4.2/ 3.7	1.279	64.0/ 38.2	.0	.0	5.9	[illegible]	[illegible]	[illegible]
7	.6	5.9	[illegible]	7.3	97.6	.466	1013.4	2.9/ 2.6	.804	96.5/ 10.3	.0	.0	.0	[illegible]	[illegible]	1.4
8	2.7	[illegible]	[illegible]	7.5	95.2	1.955	1011.8	3.6/ 2.9	.565	98.4/ 8.2	.0	1.1	1.9	[illegible]	1.[illegible]	1.[illegible]
9	5.8	[illegible]	[illegible]	7.3	92.4	4.590	1010.3	4.5/ 3.5	.872	98.0/ 7.3	.0	.2	2.2	[illegible]	[illegible]	1.[illegible]
10	7.1	6.9	-4.3	7.0	88.0	12.633	1008.8	5.6/ 3.0	1.340	92.7/ 14.5	.0	.1	1.8	[illegible]	[illegible]	[illegible]
11	.2	-1.0	-1.6	4.4	95.9	.522	1009.9	4.3/ 3.2	.428	96.9/ 13.5	.0	.0	2.7	[illegible]	[illegible]	[illegible]
12	.7	-1.1	-1.8	4.4	95.0	1.870	1009.9	4.4/ 2.7	.270	97.4/ 10.7	.0	.0	2.2	[illegible]	[illegible]	[illegible]
13	.8	-1.4	-2.7	4.1	91.2	4.436	1013.4	4.6/ 2.7	.824	94.1/ 14.4	.0	.0	2.7	[illegible]	[illegible]	[illegible]
14	.8	-1.5	-2.5	4.2	86.9	13.761	1010.9	5.6/ 3.7	1.359	83.8/ 22.9	.0	.0	1.7	[illegible]	[illegible]	[illegible]
15	40.3	5.2	2.3	5.9	82.3	14.756	1015.9	3.9/ 2.6	2.293	51.9/ 39.9	.0	2.1	10.1	[illegible]	1.[illegible]	[illegible]
16	3.8	14.7	12.5	11.1	87.3	12.319	1011.7	3.2/ 1.9	2.958	66.5/ 29.8	.0	6.9	17.7	[illegible]	[illegible]	[illegible]
17	3.6	4.0	3.8	6.6	98.4	.343	1015.5	2.4/ 1.9	.088	99.0/ 4.9	.0	.0	1.8	[illegible]	[illegible]	[illegible]
18	10.0	4.6	4.0	6.7	96.4	1.312	1014.2	3.1/ 2.2	.189	98.0/ 6.3	.0	.0	2.6	[illegible]	[illegible]	[illegible]
19	14.2	4.9	4.2	6.7	95.3	3.326	1013.7	3.9/ 2.6	.143	98.1/ 6.0	.0	.0	1.2	[illegible]	[illegible]	[illegible]
20	35.4	5.2	3.8	6.5	90.9	7.312	1012.9	4.4/ 2.8	.390	94.9/ 9.4	.0	.0	1.2	[illegible]	[illegible]	[illegible]
21	37.2	4.3	1.8	5.8	85.1	10.680	1010.1	4.9/ 2.3	1.878	16.4/ 17.2	.0	5.6	17.9	[illegible]	1.4	[illegible]
22	100.0	5.3	3.2	6.3	87.1	9.302	1015.0	3.6/ 2.6	1.885	63.2/ 38.7	.0	2.5	10.7	[illegible]	9.1	[illegible]

CLASS NO.	FREQCY CLASS (%)	FREQCY WNDDIR 015 (%)	FREQCY WNDDIR 045 (%)	FREQCY WNDDIR 075 (%)	FREQCY WNDDIR 105 (%)	FREQCY WNDDIR 135 (%)	FREQCY WNDDIR 165 (%)	FREQCY WNDDIR 195 (%)	FREQCY WNDDIR 225 (%)	FREQCY WNDDIR 255 (%)	FREQCY WNDDIR 285 (%)	FREQCY WNDDIR 315 (%)	FREQCY WNDDIR 345 (%)	FREQCY WNDDIR VBL (%)
1	4.5	8.6	11.1	8.3	4.0	2.8	4.6	5.2	6.9	8.2	7.0	4.7	7.2	21.3
2	8.2	8.5	12.0	10.0	5.9	4.6	5.7	5.7	8.7	8.5	5.8	3.9	5.9	14.8
3	17.9	7.6	10.1	11.4	8.3	6.7	7.0	7.7	10.1	8.4	5.3	3.6	5.3	8.5
4	9.6	6.3	7.7	9.8	7.8	7.4	8.6	9.5	11.3	11.[illegible]	6.8	3.9	4.7	5.1
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	23.5	11.8	.0	5.9	5.9	11.8	11.8	5.9	5.9	5.9	11.8
7	.6	8.9	11.1	10.0	3.9	1.4	2.5	4.4	12.2	12.2	10.6	7.2	[illegible]	6.1
8	2.7	6.5	7.8	6.9	4.2	3.3	4.9	8.3	15.6	14.5	9.7	6.8	5.3	6.0
9	5.8	5.3	6.6	4.7	3.6	4.0	6.8	11.3	19.5	14.9	9.5	6.1	5.1	2.5
10	7.1	4.2	4.1	2.6	2.3	3.6	7.3	12.3	21.7	16.9	11.5	7.4	5.0	1.1
11	.2	10.0	12.7	3.6	1.8	1.8	1.8	3.6	3.6	13.6	12.7	11.8	10.9	11.8
12	.7	11.1	12.9	7.3	3.8	3.1	2.9	4.5	8.2	13.8	10.7	7.8	12.0	1.8
13	.8	10.1	11.5	9.5	3.5	1.9	2.7	4.1	9.3	15.4	10.1	7.8	10.7	3.5
14	.8	7.0	6.8	7.4	3.2	.6	1.9	1.9	8.5	21.9	17.5	12.1	8.7	2.6
15	40.3	7.5	9.1	9.7	6.8	4.7	6.8	8.4	13.2	10.5	7.8	5.0	5.9	4.8
16	.8	3.0	4.5	4.3	10.2	10.8	20.7	11.4	14.0	7.3	2.8	2.8	1.8	3.3
17	3.6	10.0	12.9	8.2	3.8	2.5	3.3	5.0	8.8	10.0	8.8	6.1	8.6	10.1
18	10.0	9.3	12.0	8.0	3.6	2.5	3.5	5.4	11.2	12.6	9.6	6.4	7.6	3.3
19	14.2	9.2	10.9	7.5	2.9	1.7	2.7	5.4	12.7	14.8	10.7	7.9	8.3	3.3
20	35.4	7.5	8.2	6.2	2.9	2.1	4.0	7.6	16.1	16.1	11.6	7.3	7.2	2.2
21	37.2	8.1	10.8	12.7	8.7	6.3	7.2	7.1	8.4	6.9	5.0	3.6	5.2	9.9
22	100.0	7.2	8.9	9.0	6.3	5.0	6.8	8.3	12.7	10.9	7.6	4.9	5.7	6.6

VISIBILITY
9.30 KM

TUPB LASER MODULE

CALCULATION OF POWER SPECTRUM AND TURBULENCE INDUCED POINTING JITTER OF A LASER TARGET DESIGNATOR AND SEEKER

LASER WAVELENGTH (MICROMETERS)	10.6000
DESIG. APERTURE DIAMETER (METERS)	.101600
BEAMSPREAD ANGLE (RADIANS)	.000130
SEEKER APERTURE DIAMETER (METERS)	.400000
RANGE FROM TARGET TO SEEKER (METERS)	500.00
BEAM SLUE RATE (RAD/SEC)	.000000
DESIGNATION RANGE (METERS)	400.00
DURATION OF TEST (SECONDS)	5.0000
TOTAL DESIGNATOR PATH SEGMENTS	5
TOTAL SEEKER PATH SEGMENTS	5
TOTAL FREQUENCIES FOR WHICH POWER SPECTRUM IS TO BE CALCULATED	512

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN DESIGNATOR PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-013	.93
3	.130000-013	.93
4	.270000-013	.93
5	.500000-013	.93

VALUES OF REFRACTIVE INDEX STRUCTURE CONSTANT AND WIND SPEED IN SEEKER PATH

SEGMENT NO.	CN**2 (METER**(-2/3))	WINDSPEED (METER/SEC)
1	.600000-012	.93
2	.100000-013	.93
3	.130000-013	.93
4	.270000-013	.93
5	.500000-013	.93

DESIGNATOR TO TARGET

VIRTUAL POINT SOURCE TO APERTURE DISTANCE		781.53845 (METERS)
DISTANCE FROM VIRTUAL POINT SOURCE TO TARGET		1181.53845 (METERS)
INTEGRATED COHERENCE LENGTH		.335957 (METERS)
DIAMETER/INTEGRATED COHERENCE LENGTH		.302419
TRANSMITTER-INDUCED BEAM SPREAD	.13000-003 (RADIANS)	.052000 (METERS)
DIFFRACTION-LIMITED BEAM SPREAD	.11769-003 (RADIANS)	.047074 (METERS)
DIFFRACTION AND TURBULENCE BEAM SPREAD	.11907-003 (RADIANS)	.047629 (METERS)
TOTAL EFFECTIVE BEAM SIZE	.17629-003 (RADIANS)	.172116 (METERS)

SEGMENT NO.	COHERENCE LENGTH	REFERENCE FREQUENCY(HERTZ)
1	.341660	56.928500
2	5.124240	18.976166
3	6.129030	11.385700
4	6.588533	8.132643
5	13.656743	6.325389

TARGET TO SEEKER

INTEGRATED COHERENCE LENGTH		.934983 (METERS)
DIAMETER/INTEGRATED COHERENCE LENGTH		.184085
TRANSMITTER-INDUCED BEAM SPREAD	.13000-003 (RADIANS)	.065000 (METERS)
DIFFRACTION-LIMITED BEAM SPREAD	.69469-004 (RADIANS)	.034735 (METERS)
DIFFRACTION AND TURBULENCE BEAM SPREAD	.69828-004 (RADIANS)	.034914 (METERS)
TOTAL EFFECTIVE BEAM SIZE	.14757-003 (RADIANS)	.245900 (METERS)

SEGMENT NO.	COHERENCE LENGTH	REFERENCE FREQUENCY(HERTZ)
1	2.689629	7.400705
2	10.458297	2.466902
3	5.360999	1.480141
4	2.469819	1.057244
5	1.327262	.822301

THE VARIANCE OF THE POWER SPECTRUM IS .8090-011

OUTPUT FOR DESIGNATOR TO TARGET PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .92021-007, MEAN OF IMAG PART = .00000
VAR. OF REAL PART = .83284-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = -.52602-011, MEAN OF IMAG PART = .19630-016
VAR. OF REAL PART = .79425-017, VAR. OF IMAG PART = .13959-031

OUTPUT FOR TARGET TO SEEKER PATH

MEAN AND VARIANCE OF RANDOM ARRAY

MEAN OF REAL PART = .17265-006, MEAN OF IMAG PART = .00000
VAR. OF REAL PART = .91505-011, VAR. OF IMAG PART = .00000

MEAN AND VARIANCE OF TIME SEQUENCE

MEAN OF REAL PART = .24531-011, MEAN OF IMAG PART = .26733-016
VAR. OF REAL PART = .87592-017, VAR. OF IMAG PART = .12492-031

XSCALE HORIZONTAL-SLANT PATH EXTINCTION MODULE

OPTIONS CHOSEN
MARITIME ARTIC
HORIZONTAL PATH

EXTINCTION FROM 8.0 TO 12.0 MICRONS KM**-1	DISTANCE KM	TRANSMISSION
.178	.400	.931+000

OPTIONS CHOSEN
MARITIME POLAR
SLANT PATH FOR 10.600 MICRONS

WARNING FROM SLANT
THE VERTICAL DISTANCE EXCEEDS THE 500 METER UPPER LIMIT, OR
IS NOT AN INTEGER MULTIPLE OF 20 METERS
SLANT DISTANCE CHANGED FROM .2397 TO .2238 KM

SLANT EXTINCTION AT 10.60 MICRONS KM**-1	DISTANCE KM	TRANSMISSION	ANGLE
1.295	.224	.748+000	56.30

OPTIONS CHOSEN
CONTINENTAL POLAR
HORIZONTAL PATH

EXTINCTION FROM 8.0 TO 12.0 MICRONS KM**-1	DISTANCE KM	TRANSMISSION
.008	.400	.997+000

SMOKE MODEL MODULE

```
**********
*  SMOKE   *
**********
```

EXECUTION 1

SMOKE MUNITIONS
WHITE PHOSPHORUS (WP)
NO. ROUNDS 1
FILL WEIGHT 15.600 LB
BURN TIME 1.0 SEC
EFFICIENCY 100.0 PERCENT
YIELD FACTOR 5.84

METEOROLOGICAL CONDITIONS

WINDSPEED	3.6	M/S
WIND DIRECTION (USUAL		
MET CONVENTION AZIMUTH)	225.0	DEGREES
RELATIVE HUMIDITY	87.1	PERCENT
PASQUILL CATEGORY	D	
AIR TEMPERATURE	5.3	DEGREE C
TEMP. GRADIENT	.00	C DEG./M

EXTINCTION COEFFICIENTS

0.4-0.7 MICROMETERS	4.304	M**2/GM
0.7-1.2 MICROMETERS	2.166	M**2/GM
1.06 MICROMETERS	1.541	M**2/GM
3.0-5.0 MICROMETERS	.350	M**2/GM
8.0-12. MICROMETERS	.338	M**2/GM
10.6 MICROMETERS	.364	M**2/GM
94.0 GHZ	.001	M**2/GM

BURN RATE PROFILE = 1.0000

FIELD COORDINATES

	(X)	(Y)	(Z)	
MUNITION COORDINATES=	.00	-50.00	.00	METERS
OBSERVER COORDINATES=	200.00	.00	2.00	METERS
TARGET COORDINATES=	-200.00	.00	2.00	METERS

ROTATED COORD.(WIND X-AXIS,MUNITION ORIGIN)

(XW)	(YW)	(ZW)	
.00	.00	.00	METERS
176.78	-106.07	2.00	METERS
-106.07	176.78	2.00	METERS

ANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH = 90.00 DEG.
EVENT TIME = .0 SEC

TIME (SEC)	LENGTH (METERS)	WIDTH (METERS)	HEIGHT (METERS)	PATHLENGTH (METERS)	CL (GM/M**2)	TRANSMISSION SPECTRAL BANDS (MICROMETERS) 0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94.GHZ
5.	18.	13.	22.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10.	36.	16.	34.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15.	54.	20.	44.	.00	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20.	72.	23.	53.	18.16	4.01	.000	.000	.002	.245	.258	.232	.996
25.	90.	27.	61.	37.02	4.89	.000	.000	.001	.181	.191	.169	.995
30.	108.	31.	69.	40.38	3.48	.000	.001	.005	.296	.309	.282	.997
35.	126.	34.	76.	43.27	2.58	.000	.004	.019	.405	.418	.391	.997
40.	144.	38.	83.	45.86	1.99	.000	.013	.047	.498	.510	.485	.998
45.	162.	41.	90.	48.25	1.58	.001	.033	.088	.575	.586	.563	.998
50.	180.	45.	95.	50.49	1.29	.004	.062	.138	.638	.648	.626	.999
55.	198.	49.	101.	52.62	1.07	.010	.099	.193	.688	.697	.678	.999
60.	216.	52.	106.	54.64	.90	.021	.142	.250	.730	.738	.721	.999
65.	234.	56.	111.	56.58	.77	.036	.189	.305	.764	.771	.756	.999
70.	252.	59.	115.	58.45	.67	.057	.236	.358	.792	.798	.785	.999
75.	270.	63.	120.	60.26	.58	.081	.283	.407	.815	.821	.809	.999
80.	288.	67.	124.	62.02	.52	.109	.328	.452	.835	.840	.829	.999
85.	306.	70.	127.	63.72	.46	.139	.370	.493	.852	.856	.846	1.000
90.	324.	74.	131.	65.38	.41	.171	.411	.531	.866	.870	.861	1.000
95.	342.	77.	135.	67.00	.37	.203	.448	.565	.878	.882	.874	1.000
100.	360.	81.	138.	68.59	.34	.235	.483	.596	.889	.893	.885	1.000
105.	378.	85.	141.	70.13	.31	.267	.515	.624	.898	.902	.894	1.000
110.	396.	88.	144.	71.65	.28	.299	.545	.649	.906	.910	.903	1.000
115.	414.	92.	147.	73.14	.26	.330	.572	.672	.914	.917	.910	1.000
120.	432.	95.	150.	74.60	.24	.359	.597	.693	.920	.923	.917	1.000
125.	450.	99.	153.	76.04	.22	.387	.620	.712	.926	.928	.923	1.000
130.	468.	103.	155.	77.45	.20	.414	.642	.729	.931	.933	.928	1.000
135.	486.	106.	158.	78.83	.19	.440	.662	.745	.935	.938	.933	1.000
140.	504.	110.	160.	80.20	.18	.465	.680	.760	.940	.942	.937	1.000
145.	522.	113.	163.	81.55	.17	.488	.697	.773	.943	.945	.941	1.000
150.	540.	117.	165.	82.87	.16	.510	.712	.786	.947	.948	.945	1.000
155.	558.	121.	167.	84.18	.15	.531	.727	.797	.950	.951	.948	1.000
160.	576.	124.	170.	85.47	.14	.550	.740	.808	.953	.954	.951	1.000
165.	594.	128.	172.	86.75	.13	.569	.753	.817	.955	.957	.953	1.000
170.	612.	131.	174.	88.00	.12	.587	.765	.826	.958	.959	.956	1.000
175.	630.	135.	176.	89.25	.12	.603	.776	.835	.960	.961	.958	1.000
180.	648.	139.	178.	90.48	.11	.619	.786	.842	.962	.963	.960	1.000

```
                    ******************
                    *                *
                    *     SMOKE      *
                    *                *
                    ******************

                                                                    EXECUTION  2

  SMOKE MUNITIONS
MC SMOKE
NO. ROUNDS      1.
FILL WEIGHT   17.190 LB
BURN TIME    100.0   SEC
EFFICIENCY    40.0   PERCENT
YIELD FACTOR  2.39

                METEOROLOGICAL CONDITIONS
WINDSPEED                          3.6  M/S
WIND DIRECTION (USUAL              225.0  DEGREES
MET CONVENTION AZIMUTH)
RELATIVE HUMIDITY                  87.1  PERCENT
PASQUILL CATEGORY                   D
AIR TEMPERATURE                    5.3  DEGREE C
TEMP. GRADIENT                      .00 C DEG./M

                                    EXTINCTION COEFFICIENTS
                                0.4-0.7  MICROMETERS      4.579   M**2/GM
                                0.7-1.2  MICROMETERS      .186    M**2/GM
                                1.06     MICROMETERS      .040    M**2/GM
                                3.0-5.0  MICROMETERS      .190    M**2/GM
                                8.0-12.  MICROMETERS      .052    M**2/GM
                                10.6     MICROMETERS      .051    M**2/GM
                                94.0     GHZ              .001    M**2/GM

        BURN RATE PROFILE =    .5370+     .4760 (T/TBURN) +  4.7790 (T/TBURN)**2 + -5.4720 (T/TBURN)**3

                        FIELD COORDINATES                    ROTATED COORD.(WIND X-AXIS,MUNITION ORIGIN)
                    (X)          (Y)          (Z)                (XW)         (YW)         (ZW)
MUNITION COORDINATES=        .00       -50.00       .00  METERS        .00          .00          .00  METERS
OBSERVER COORDINATES=     200.00          .00      2.00  METERS     176.78      -106.07         2.00  METERS
TARGET   COORDINATES=    -200.00          .00      2.00  METERS    -106.07       176.78         2.00  METERS
ANGLE OF ORIGINAL X-AXIS, CLOCKWISE WRT NORTH =   90.00 DEG.
    EVENT TIME =      .0 SEC

   TIME   LENGTH   WIDTH   HEIGHT  PATHLENGTH     CL                                TRANSMISSION
  (SEC)  (METERS) (METERS) (METERS)  (METERS)  (GM/M**2)          SPECTRAL BANDS (MICROMETERS)
                                                                0.4-0.7 0.7-1.2   1.06  3.0-5.0 8.0-12.   10.6   94.GHZ
      5.     18.     13.      6.        .00        .00  1.000   1.000   1.000   1.000   1.000   1.000   1.000
     10.     36.     16.      8.        .00        .00  1.000   1.000   1.000   1.000   1.000   1.000   1.000
     15.     54.     20.     10.        .00        .01   .949    .975    .977    .998    .999    .999   1.000
     20.     72.     23.     11.      15.56       1.16   .635    .079    .094    .802    .941    .941    .999
     25.     90.     27.     12.      30.12       2.00   .000    .013    .017    .684    .901    .901    .998
     30.    108.     31.     13.      28.24       2.31   .000    .006    .009    .645    .887    .887    .998
     35.    126.     34.     14.      26.88       2.63   .000    .003    .005    .607    .872    .874    .997
     40.    144.     38.     16.      25.86       2.98   .000    .001    .002    .567    .856    .859    .997
     45.    162.     41.     17.      25.07       3.35   .000    .001    .001    .529    .840    .843    .996
     50.    180.     45.     18.      24.42       3.73   .000    .000    .000    .492    .823    .827    .996
     55.    198.     49.     19.      23.89       4.11   .000    .000    .000    .458    .807    .811    .995
     60.    216.     52.     20.      23.45       4.47   .000    .000    .000    .427    .792    .796    .995
     65.    234.     56.     21.      23.07       4.80   .000    .000    .000    .401    .779    .783    .995
     70.    252.     59.     23.      22.75       5.09   .000    .000    .000    .380    .768    .772    .995
     75.    270.     63.     24.      22.46       5.30   .000    .000    .000    .365    .759    .763    .995
     80.    288.     67.     25.      22.22       5.44   .000    .000    .000    .356    .754    .758    .995
     85.    306.     70.     26.      21.99       5.49   .000    .000    .000    .353    .752    .755    .995
     90.    324.     74.     27.      21.80       5.42   .000    .000    .000    .357    .755    .759    .995
     95.    342.     77.     29.      21.62       5.22   .000    .000    .000    .371    .762    .766    .995
    100.    360.     81.     30.      21.46       4.87   .000    .000    .000    .396    .776    .780    .995
    105.    378.     85.     31.      21.32       4.37   .000    .000    .000    .436    .797    .800    .996
    110.    396.     88.     32.      21.18       3.68   .000    .000    .001    .497    .826    .829    .996
    115.    414.     92.     33.      21.06       2.79   .000    .002    .003    .588    .865    .868    .996
    120.    432.     95.     35.      20.95       1.28   .003    .060    .073    .783    .935    .937    .998
    125.    450.     99.     36.      20.85        .62   .058    .257    .281    .889    .968    .969    .999
    130.    468.    103.     37.      20.75        .47   .116    .358    .383    .915    .976    .976   1.000
    135.    486.    106.     38.      20.67        .37   .182    .443    .468    .932    .981    .981   1.000
    140.    504.    110.     39.      20.58        .30   .251    .517    .540    .944    .984    .985   1.000
    145.    522.    113.     40.      20.51        .25   .319    .579    .601    .954    .987    .987   1.000
    150.    540.    117.     42.      20.44        .21   .384    .633    .652    .961    .989    .989   1.000
    155.    558.    121.     43.      20.37        .18   .443    .678    .696    .967    .991    .991   1.000
    160.    576.    124.     44.      20.30        .15   .498    .717    .733    .971    .992    .992   1.000
    165.    594.    128.     45.      20.25        .13   .547    .750    .764    .975    .993    .993   1.000
    170.    612.    131.     46.      20.19        .11   .591    .778    .791    .978    .994    .994   1.000
    175.    630.    135.     48.      20.14        .10   .630    .802    .814    .981    .995    .995   1.000
    180.    648.    139.     49.      20.09        .09   .665    .823    .834    .983    .995    .995   1.000
    185.    666.    142.     50.      20.04        .08   .696    .841    .851    .985    .996    .996   1.000
    190.    684.    146.     51.      19.99        .07   .723    .857    .866    .987    .996    .996   1.000
    195.    702.    149.     52.      19.95        .06   .748    .870    .879    .988    .997    .997   1.000
    200.    720.    153.     54.      19.91        .06   .770    .882    .890    .989    .997    .997   1.000
    205.    738.    157.     55.      19.87        .05   .789    .893    .900    .990    .997    .997   1.000
    210.    756.    160.     56.      19.83        .05   .806    .902    .909    .991    .998    .998   1.000
    215.    774.    164.     57.      19.80        .04   .822    .911    .916    .992    .998    .998   1.000
    220.    792.    167.     58.      19.77        .04   .836    .918    .923    .993    .998    .998   1.000
    225.    810.    171.     59.      19.73        .04   .849    .925    .930    .993    .998    .998   1.000
    230.    828.    175.     61.      19.70        .03   .860    .931    .935    .994    .998    .998   1.000
    235.    846.    178.     62.      19.67        .03   .871    .936    .940    .994    .998    .998   1.000
    240.    864.    182.     63.      19.64        .03   .880    .941    .945    .995    .999    .999   1.000
    245.    882.    185.     64.      19.62        .03   .888    .945    .949    .995    .999    .999   1.000
    250.    900.    189.     65.      19.59        .02   .896    .949    .952    .995    .999    .999   1.000
```

COMBINED EFFECT OF 2 EXECUTIONS IN SMOKE:

TIME (SEC)	CL (G/M**2)	TRANSMISSION 0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94.GHZ
5.	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10.	.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15.	.01	.949	.975	.977	.998	.999	.999	1.000
20.	5.17	.000	.000	.000	.374	.764	.768	.995
25.	6.89	.000	.000	.000	.270	.699	.704	.993
30.	5.78	.000	.000	.000	.333	.740	.745	.994
35.	5.21	.000	.000	.000	.371	.763	.767	.995
40.	4.97	.000	.000	.000	.389	.772	.776	.995
45.	4.93	.000	.000	.000	.392	.774	.778	.995
50.	5.02	.000	.000	.000	.385	.770	.774	.995
55.	5.18	.000	.000	.000	.374	.764	.768	.995
60.	5.37	.000	.000	.000	.360	.756	.760	.995
65.	5.57	.000	.000	.000	.347	.748	.753	.994
70.	5.75	.000	.000	.000	.335	.741	.746	.994
75.	5.89	.000	.000	.000	.327	.736	.741	.994
80.	5.96	.000	.000	.000	.322	.734	.738	.994
85.	5.94	.000	.000	.000	.323	.734	.739	.994
90.	5.83	.000	.000	.000	.331	.739	.743	.994
95.	5.59	.000	.000	.000	.346	.748	.752	.994
100.	5.21	.000	.000	.000	.372	.763	.767	.995
105.	4.67	.000	.000	.000	.412	.784	.788	.995
110.	3.96	.000	.000	.000	.471	.814	.817	.996
115.	3.05	.000	.001	.002	.560	.853	.856	.997
120.	1.52	.001	.036	.045	.749	.924	.925	.998
125.	.84	.021	.159	.179	.852	.957	.958	.999
130.	.67	.046	.229	.252	.880	.966	.966	.999
135.	.56	.076	.292	.317	.899	.971	.972	.999
140.	.48	.111	.350	.375	.913	.975	.976	1.000
145.	.42	.149	.402	.428	.924	.979	.979	1.000
150.	.37	.187	.449	.474	.933	.981	.982	1.000
155.	.32	.226	.492	.515	.940	.983	.984	1.000
160.	.29	.264	.529	.552	.946	.985	.985	1.000
165.	.26	.300	.563	.585	.951	.986	.987	1.000
170.	.24	.335	.593	.614	.956	.988	.988	1.000
175.	.22	.368	.621	.641	.959	.989	.989	1.000
180.	.20	.399	.645	.664	.963	.990	.990	1.000
185.	.08	.696	.841	.851	.985	.996	.996	1.000
190.	.07	.723	.857	.866	.987	.996	.996	1.000
195.	.06	.748	.870	.879	.988	.997	.997	1.000
200.	.06	.770	.882	.890	.989	.997	.997	1.000
205.	.05	.789	.893	.900	.990	.997	.997	1.000
210.	.05	.806	.902	.909	.991	.998	.998	1.000
215.	.04	.822	.911	.916	.992	.998	.998	1.000
220.	.04	.836	.918	.923	.993	.998	.998	1.000
225.	.04	.849	.925	.930	.993	.998	.998	1.000
230.	.03	.860	.931	.935	.994	.998	.998	1.000
235.	.03	.871	.936	.940	.994	.998	.999	1.000
240.	.03	.880	.941	.945	.995	.999	.999	1.000
245.	.03	.888	.945	.949	.995	.999	.999	1.000
250.	.02	.896	.949	.952	.995	.999	.999	1.000

***TRANSMISSION RETURNED TO MAIN FOR WAVELENGTH OF 10.600 MICROMETERS IS .999 AT TIME 250.

DIRT TRANSMISSION MODULE

DIRTRAN-2 DUST CLOUD INFRARED TRANSMISSION CALCULATION

*** NOTE -- ALL UNITS ARE MKS UNLESS OTHERWISE SPECIFIED ***

PASQUILL CATEGORY D
HT 2.00 TEMP 278.30 HT 2.00 WIND 3.60
WIND DIRECTION 45.00

LATITUDE 53.00
THE INVERSION LAYER HEIGHT IS GROWING

SOIL-2
SILT CONTENT .15
DEPTH OF SOD .00

30 DEGREE TILTED TIP AT 0.3 METER DEPTH
WEIGHT OF CHARGE 6.80 KG.
DETONATION DEPTH .00

SIMULTANEOUS BURST , UNIFORMLY DISTRIBUTED CHARGES IN A PARALLELOGRAM
TOTAL NUMBER OF CHARGES IS 1 WITH REFERENCE CHARGE AT (.00, .00)
1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (.00, .00)
1 CHARGES WITH DIRECTION AND SPACING GIVEN BY (.00, .00)

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 1.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .767-002

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 19.74 METERS
THE CENTROID COORDINATES ARE 1.34 10.55
THE WIDTH AT THE CENTROID IS 20.62 METERS
THE WIDTH AT 2.00 METERS IS 16.56 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
-5.670 2.000
-8.973 10.550
1.337 19.745
1.337 19.745
11.646 10.550
10.893 2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 3.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .620+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 21.89 METERS
THE CENTROID COORDINATES ARE 8.02 10.55
THE WIDTH AT THE CENTROID IS 31.26 METERS
THE WIDTH AT 2.00 METERS IS 21.88 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED
-2.947 2.000
-7.608 10.550

8.020	21.888
8.020	21.888
23.644	10.550
18.928	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 5.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .940+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00

THE HEIGHT OF THE CLOUD IS 23.56 METERS

THE CENTROID COORDINATES ARE 14.70 10.55

THE WIDTH AT THE CENTROID IS 38.28 METERS

THE WIDTH AT 2.00 METERS IS 29.37 METERS

6 CONTOUR POINTS HAVE BEEN DETERMINED

-.224	2.000
-4.436	10.550
14.704	23.558
14.704	23.558
33.844	10.550
29.151	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 7.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .987+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00

THE HEIGHT OF THE CLOUD IS 24.95 METERS

THE CENTROID COORDINATES ARE 21.39 10.55

THE WIDTH AT THE CENTROID IS 43.76 METERS

THE WIDTH AT 2.00 METERS IS 35.94 METERS

6 CONTOUR POINTS HAVE BEEN DETERMINED

3.436	2.000
-.491	10.550
21.398	24.948
21.388	24.948
43.267	10.550
39.374	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 9.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .997+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00

THE HEIGHT OF THE CLOUD IS 26.15 METERS

THE CENTROID COORDINATES ARE 28.07 10.55

THE WIDTH AT THE CENTROID IS 48.33 METERS

THE WIDTH AT 2.00 METERS IS 41.56 METERS

6 CONTOUR POINTS HAVE BEEN DETERMINED

7.097	2.000

8.906	10.550
8.072	28.148
8.072	28.148
8.906	10.550
48.659	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 11.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .999+000

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 27.21 METERS
THE CENTROID COORDINATES ARE 34.76 10.55
THE WIDTH AT THE CENTROID IS 52.30 METERS
THE WIDTH AT 2.00 METERS IS 46.56 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

11.382	2.000
8.607	10.550
34.755	27.213
34.755	27.213
60.904	10.550
57.944	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 13.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 28.17 METERS
THE CENTROID COORDINATES ARE 41.44 10.55
THE WIDTH AT THE CENTROID IS 55.82 METERS
THE WIDTH AT 2.00 METERS IS 50.94 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

15.667	2.000
13.529	10.550
41.439	28.173
41.439	28.173
69.349	10.550
66.605	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 15.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 29.05 METERS
THE CENTROID COORDINATES ARE 48.12 10.55
THE WIDTH AT THE CENTROID IS 59.00 METERS
THE WIDTH AT 2.00 METERS IS 55.00 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

20.265	2.000
18.622	10.550
48.123	29.050
48.123	29.050
77.624	10.550
75.265	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 17.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 29.86 METERS
THE CENTROID COORDINATES ARE 54.81 10.55
THE WIDTH AT THE CENTROID IS 61.91 METERS
THE WIDTH AT 2.00 METERS IS 59.38 METERS

6 CONTOUR POINTS HAVE BEEN DETERMINED

24.551	2.000
23.850	10.550
54.807	29.859
54.807	29.859
85.763	10.550
83.926	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 19.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 30.61 METERS
THE CENTROID COORDINATES ARE 61.49 10.55
THE WIDTH AT THE CENTROID IS 64.60 METERS
THE WIDTH AT 2.00 METERS IS 63.12 METERS

6 CONTOUR POINTS HAVE BEEN DETERMINED

29.148	2.000
29.190	10.550
61.490	30.611
61.490	30.611
93.791	10.550
92.273	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 21.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 31.32 METERS
THE CENTROID COORDINATES ARE 68.17 10.55
THE WIDTH AT THE CENTROID IS 67.10 METERS
THE WIDTH AT 2.00 METERS IS 66.87 METERS

6 CONTOUR POINTS HAVE BEEN DETERMINED

33.434	2.000
34.623	10.550
68.174	31.315
68.174	31.315
101.725	10.550
100.309	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 23.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 31.98 METERS
THE CENTROID COORDINATES ARE 74.86 10.55
THE WIDTH AT THE CENTROID IS 69.44 METERS
THE WIDTH AT 2.00 METERS IS 70.31 METERS

6 CONTOUR POINTS HAVE BEEN DETERMINED

38.032	2.000
40.135	10.550
74.858	31.977
74.858	31.977
109.580	10.550
108.344	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 25.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 32.60 METERS
THE CENTROID COORDINATES ARE 81.54 10.55
THE WIDTH AT THE CENTROID IS 71.65 METERS
THE WIDTH AT 2.00 METERS IS 73.75 METERS

6 CONTOUR POINTS HAVE BEEN DETERMINED

42.630	2.000
45.717	10.550
81.541	32.603
81.541	32.603
117.366	10.550
116.380	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 27.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 33.20 METERS
THE CENTROID COORDINATES ARE 88.23 10.55
THE WIDTH AT THE CENTROID IS 73.73 METERS

THE WIDTH AT 2.00 METERS IS 77.19 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

47.227	2.000
51.359	10.550
88.225	33.196
88.225	33.196
125.091	10.550
124.415	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 29.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 33.76 METERS
THE CENTROID COORDINATES ARE 94.91 10.55
THE WIDTH AT THE CENTROID IS 75.71 METERS
THE WIDTH AT 2.00 METERS IS 80.94 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

51.513	2.000
57.055	10.550
94.909	33.761
94.909	33.761
132.763	10.550
132.450	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 31.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 34.30 METERS
THE CENTROID COORDINATES ARE 101.59 10.55
THE WIDTH AT THE CENTROID IS 77.59 METERS
THE WIDTH AT 2.00 METERS IS 84.06 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

56.111	2.000
62.799	10.550
101.593	34.299
101.593	34.299
140.387	10.550
140.173	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 33.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 34.81 METERS
THE CENTROID COORDINATES ARE 108.28 10.55

THE WIDTH AT THE CENTROID IS 79.38 METERS
THE WIDTH AT 2.00 METERS IS 87.19 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

60.708	2.000
68.585	10.550
108.276	34.814
108.276	34.814
147.968	10.550
147.896	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 35.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 35.31 METERS
THE CENTROID COORDINATES ARE 114.96 10.55
THE WIDTH AT THE CENTROID IS 81.10 METERS
THE WIDTH AT 2.00 METERS IS 90.31 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

65.306	2.000
74.410	10.550
114.960	35.308
114.960	35.308
155.510	10.550
155.619	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 37.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 35.78 METERS
THE CENTROID COORDINATES ARE 121.64 10.55
THE WIDTH AT THE CENTROID IS 82.75 METERS
THE WIDTH AT 2.00 METERS IS 93.75 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

69.592	2.000
80.271	10.550
121.644	35.782
121.644	35.782
163.017	10.550
163.342	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 39.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 36.24 METERS

THE CENTROID COORDINATES ARE 128.33 10.55
THE WIDTH AT THE CENTROID IS 84.33 METERS
THE WIDTH AT 2.00 METERS IS 96.88 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

74.190	2.000
86.164	10.550
128.328	36.238
128.328	36.238
170.492	10.550
171.065	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 41.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 36.68 METERS
THE CENTROID COORDINATES ARE 135.01 10.55
THE WIDTH AT THE CENTROID IS 85.85 METERS
THE WIDTH AT 2.00 METERS IS 100.00 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

78.787	2.000
92.086	10.550
135.011	36.677
135.011	36.677
177.936	10.550
178.787	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 43.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 37.10 METERS
THE CENTROID COORDINATES ARE 141.70 10.55
THE WIDTH AT THE CENTROID IS 87.32 METERS
THE WIDTH AT 2.00 METERS IS 102.81 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

83.385	2.000
98.037	10.550
141.695	37.102
141.695	37.102
185.353	10.550
186.198	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 45.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00

THE HEIGHT OF THE CLOUD IS 37.51 METERS
THE CENTROID COORDINATES ARE 148.38 10.55
THE WIDTH AT THE CENTROID IS 88.73 METERS
THE WIDTH AT 2.00 METERS IS 105.94 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

87.983	2.000
104.013	10.550
148.379	37.512
148.379	37.512
192.745	10.550
193.921	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 47.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 37.91 METERS
THE CENTROID COORDINATES ARE 155.06 10.55
THE WIDTH AT THE CENTROID IS 90.10 METERS
THE WIDTH AT 2.00 METERS IS 108.75 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

92.581	2.000
110.012	10.550
155.063	37.909
155.063	37.909
200.113	10.550
201.331	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 49.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 38.29 METERS
THE CENTROID COORDINATES ARE 161.75 10.55
THE WIDTH AT THE CENTROID IS 91.43 METERS
THE WIDTH AT 2.00 METERS IS 111.87 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

97.175	2.000
116.034	10.550
161.746	38.293
161.746	38.293
207.459	10.550
209.054	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 51.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 38.67 METERS
THE CENTROID COORDINATES ARE 168.43 10.55
THE WIDTH AT THE CENTROID IS 92.71 METERS
THE WIDTH AT 2.00 METERS IS 114.69 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

101.777	2.000
122.076	10.550
168.430	38.665
168.430	38.665
214.784	10.550
216.464	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 53.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 39.03 METERS
THE CENTROID COORDINATES ARE 175.11 10.55
THE WIDTH AT THE CENTROID IS 93.95 METERS
THE WIDTH AT 2.00 METERS IS 117.50 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

106.375	2.000
128.138	10.550
175.114	39.027
175.114	39.027
222.090	10.550
223.875	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 55.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 39.38 METERS
THE CENTROID COORDINATES ARE 181.80 10.55
THE WIDTH AT THE CENTROID IS 95.16 METERS
THE WIDTH AT 2.00 METERS IS 120.00 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

111.285	2.000
134.217	10.550
181.797	39.378
181.797	39.378
229.377	10.550
231.285	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 57.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 39.72 METERS
THE CENTROID COORDINATES ARE 188.48 10.55
THE WIDTH AT THE CENTROID IS 96.33 METERS
THE WIDTH AT 2.00 METERS IS 122.81 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

115.883	2.000
140.315	10.550
188.481	39.720
188.481	39.720
236.648	10.550
238.695	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 59.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 40.05 METERS
THE CENTROID COORDINATES ARE 195.16 10.55
THE WIDTH AT THE CENTROID IS 97.47 METERS
THE WIDTH AT 2.00 METERS IS 125.62 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

120.481	2.000
146.428	10.550
195.165	40.052
195.165	40.052
243.902	10.550
246.106	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 61.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 40.38 METERS
THE CENTROID COORDINATES ARE 201.85 10.55
THE WIDTH AT THE CENTROID IS 98.58 METERS
THE WIDTH AT 2.00 METERS IS 128.12 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

125.391	2.000
152.557	10.550
201.849	40.376
201.849	40.376
251.140	10.550
253.516	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 63.00

WAVELENGTH 10.60 MICROMETERS

TRANSMITTER COORDINATES	-200.00	.00	2.00
RECEIVER COORDINATES	200.00	.00	2.00

TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 40.69 METERS
THE CENTROID COORDINATES ARE 208.53 10.55
THE WIDTH AT THE CENTROID IS 99.66 METERS
THE WIDTH AT 2.00 METERS IS 130.62 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

130.301	2.000
158.700	10.550
208.532	40.691
208.532	40.691
258.364	10.550
260.926	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 65.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 41.00 METERS
THE CENTROID COORDINATES ARE 215.22 10.55
THE WIDTH AT THE CENTROID IS 100.72 METERS
THE WIDTH AT 2.00 METERS IS 133.12 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

135.212	2.000
164.857	10.550
215.216	40.998
215.216	40.998
265.575	10.550
268.337	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 67.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 41.30 METERS
THE CENTROID COORDINATES ARE 221.90 10.55
THE WIDTH AT THE CENTROID IS 101.74 METERS
THE WIDTH AT 2.00 METERS IS 135.00 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

140.435	2.000
171.028	10.550
221.900	41.298
221.900	41.298
272.772	10.550
275.435	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 69.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 41.59 METERS
THE CENTROID COORDINATES ARE 228.58 10.55
THE WIDTH AT THE CENTROID IS 102.75 METERS
THE WIDTH AT 2.00 METERS IS 137.50 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

145.345	2.000
177.211	10.550
228.584	41.591
228.584	41.591
279.956	10.550
282.845	2.000

ESTIMATED INVERSION HEIGHT 935

TIME AFTER BLAST 71.00

WAVELENGTH 10.60 MICROMETERS
TRANSMITTER COORDINATES -200.00 .00 2.00
RECEIVER COORDINATES 200.00 .00 2.00
TRANSMITTANCE ALONG THE LINE OF SIGHT .100+001

AERODYNAMIC CLOUD DIMENSIONS

OBSERVER COORDINATES 200.00 .00
THE HEIGHT OF THE CLOUD IS 41.88 METERS
THE CENTROID COORDINATES ARE 235.27 10.55
THE WIDTH AT THE CENTROID IS 103.72 METERS
THE WIDTH AT 2.00 METERS IS 139.69 METERS
6 CONTOUR POINTS HAVE BEEN DETERMINED

150.568	2.000
183.406	10.550
235.267	41.877
235.267	41.877
287.129	10.550
290.255	2.000

LASER TRANSMITTANCE MODULE

WAVELENGTH (MICRONS)	H2O PRESSURE (TORR)	TEMPERATURE (ABS)	ABSORPTION COEFFICIENT (KM-1)	LINE	PATHLENGTH (KM)	TRANSMISSION
*** WARNING INPUT WAVELENGTH 10.600 CHANGED TO 10.591 NEAREST STANDARD WAVELENGTH ***						
10.591	5.821	278.46	.111+000	P(20)	.4000+000	.9566+000

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .719 KM
TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .481 KM
TOTAL OPTICAL DEPTH = 25.11

TRANSMITTANCE ALONG LINE-OF-SIGHT = .12449-010

SEEKER COORDINATES (KM) XSEEKER	YSEEKER	ZSEEKER	TARGET COORDINATES (KM) XTARGET	YTARGET	ZTARGET
.200	.000	.600	-.200	.000	.002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT INTERSECTION COORDINATES (KM) XUPPER	YUPPER	ZUPPER	XLOWER	YLOWER	ZLOWER
ST/ 1	.200	.000	.600	-.068	.000	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	25.11	.12449-010

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .599 KM
TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .361 KM
TOTAL OPTICAL DEPTH = 16.57

TRANSMITTANCE ALONG LINE-OF-SIGHT = .63594-007

SEEKER COORDINATES (KM) XSEEKER	YSEEKER	ZSEEKER	TARGET COORDINATES (KM) XTARGET	YTARGET	ZTARGET
.133	.000	.500	-.200	.000	.002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT INTERSECTION COORDINATES (KM) XUPPER	YUPPER	ZUPPER	XLOWER	YLOWER	ZLOWER
ST/ 1	.133	.000	.500	-.068	.000	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	16.57	.63594-007

CLOUD TRANSMITTANCE MODULE

WAVELENGTH = 10.60 MICRONS

LINE-OF-SIGHT SLANTS DOWNWARD

TOTAL LINE-OF-SIGHT LENGTH = .359 KM
TOTAL LINE-OF-SIGHT LENGTH INTERRUPTED BY CLOUD = .120 KM
TOTAL OPTICAL DEPTH = 3.02

TRANSMITTANCE ALONG LINE-OF-SIGHT = .48846-001

SEEKER COORDINATES (KM) XSEEKER	YSEEKER	ZSEEKER	TARGET COORDINATES (KM) XTARGET	YTARGET	ZTARGET
.000	.000	.300	-.200	.000	.002

CLOUD TYPE /ID NUMBER	LINE-OF-SIGHT INTERSECTION COORDINATES (KM) XUPPER	YUPPER	ZUPPER	XLOWER	YLOWER	ZLOWER
ST/ 1	.000	.000	.300	-.067	.000	.200

CLOUD TYPE /ID NUMBER	HEIGHT OF BASE (KM)	THICKNESS (KM)	RADIUS OF CLOUD (KM)	OPTICAL DEPTH ALONG L-O-S	TRANSMITTANCE ALONG L-O-S
ST/ 1	.200	.400	.000	3.02	.48846-001

CWIC MUNITION EXPENDITURES / INVERSE STATIC TARGET DETECTION MODULE

****
* *
* INVERSE STATIC TARGET DETECTION MODEL *
* *
****

DEVICE NUMBER 8
FOV TYPE - WIDE
FOV (DEG) 10.620
MAGNIFICATION 1.000 (ARBITRARY DEFAULT)

TARGET INTRINSIC CONTRAST OR TEMPERATURE DIFFERENCE 1.000
MINIMUM TARGET DIMENSION (M) 2.300
ACQUISITION LEVEL (50 PCNT) 1.000
SEARCH ZONE (DEGREES**2) 81.000

FOR NO GREATER INPUT DETECTION PROB.	FOR NO GREATER INPUT DETECTION TIME(SEC)	UNDER INPUT AMB. ILLUM. (FT CDLS)	AND INPUT SKY/GROUND RATIO	AT INPUT RANGE (KM)	REQUIRES (TO DEFEAT DEVICE) AT MOST: COMPUTED RESOLVABLE CYCLES, RC	CONTRAST OR TEMP. DIFF. AT DEVICE	COMPUTED TOTAL PATH TRANSMITTANCE	COMMENTS
.990	1.32	10.000000	2.000	.24	.773	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.
.900	1.58	10.000000	2.000	.24	.567	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.
.750	2.16	10.000000	2.000	.24	.446	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.
.500	3.89	10.000000	2.000	.24	.313	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.

** CONTINUED ON NEXT PAGE **

****
* *
* INVERSE STATIC TARGET DETECTION MODEL *
* *
****

DEVICE NUMBER 8
FOV TYPE - WIDE
FOV (DEG) 10.620
MAGNIFICATION 1.000 (ARBITRARY DEFAULT)

TARGET INTRINSIC CONTRAST OR TEMPERATURE DIFFERENCE 1.000
MINIMUM TARGET DIMENSION (M) 2.300
ACQUISITION LEVEL (50 PCNT) 1.000
SEARCH ZONE (DEGREES**2) 81.000

FOR NO GREATER INPUT DETECTION PROB.	FOR NO GREATER INPUT DETECTION TIME(SEC)	UNDER INPUT AMB. ILLUM. (FT CDLS)	AND INPUT SKY/GROUND RATIO	AT INPUT RANGE (KM)	REQUIRES (TO DEFEAT DEVICE) AT MOST: COMPUTED RESOLVABLE CYCLES, RC	REQUIRES (TO DEFEAT DEVICE) AT MOST: CONTRAST OR TEMP. DIFF. AT DEVICE	REQUIRES (TO DEFEAT DEVICE) AT MOST: COMPUTED TOTAL PATH TRANSMITTANCE	COMMENTS
.100	24.64	10.000000	2.000	.24	.059	.037	.037	NOTE- INPUT DETECTION PROBABILITY REQUIRES CONTRAST (OR TEMP. DIFF.) BELOW THRESHOLD. VALUES ASSUMED ARE 99 PERCENT OF THRESHOLD. ADDITIONAL OBSCURANT WILL NOT DECREASE DETECTION PROBABILITY.

*** FINAL TOTAL TRANSMISSION FROM ITAM = .037

****
*
* CWIC MUNITION EXPENDITURES
*
****

MUNITION EXPENDITURES
FOR HC AND WP SMOKE

SLANT RANGE OBS-TGT	- KM	=	.400
ELEVATION OF TARGET	- DEG	=	.00
AZIMUTH OF TARGET	- DEG	=	90.00
AVG ROUGHNESS ELEMENT	- CM	=	74.0

ATMOSPHERIC EXTINCTION CORRECTIONS		
CORRECTED FOR VISIBILITY	-	YES
MARITIME ARCTIC AIR MASS	-	YES
MARITIME POLAR AIR MASS	-	NO
CONTINENTAL POLAR AIR MASS	-	NO
CORRECTED FOR RAIN	-	NO
CORRECTED FOR SNOW	-	NO

METEOROLOGICAL INPUTS			
WINDSPEED	- M/SEC	=	3.60
WIND DIRECTION	- DEG	=	225.00
PASQUILL CATEGORY	-	=	D
VISIBILITY	- KM	=	9.302
RELATIVE HUMIDITY	- PERCENT	=	87.1

TRANSMISSION THRESHOLDS	TOTAL	SMOKE
VISIBLE:	.037	.043
NEAR IR:	.037	.041
MID IR:	.037	.044
FAR IR:	.037	.039

- VISIBLE: -

SCREEN	LENGTH METERS	DURATION MINUTES
	400.	15.00

HC SMOKE SCREEN

105MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	8.		51.	
SUSTAINING:	8.	.5	51.	60.

155MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	3.		162.	
SUSTAINING:	2.	.5	328.	16.

WP SMOKE SCREEN

105MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	6.		82.	
SUSTAINING:	6.	.8	82.	129.

155MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	3.		246.	
SUSTAINING:	3.	.8	246.	31.

- NEAR IR: -

SCREEN	LENGTH METERS	DURATION MINUTES
	400.	15.00

HC SMOKE SCREEN

105MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	23.		18.	
SUSTAINING:	23.	.5	18.	173.

155MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	4.		114.	
SUSTAINING:	4.	.5	114.	30.

WP SMOKE SCREEN

105MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	8.		66.	
SUSTAINING:	8.	.8	66.	172.

155MM HOWITZER

VOLLEY	GUNS	RATE /MIN	SPACING METERS	ROUNDS
INITIAL:	4.		196.	
SUSTAINING:	3.	.8	196.	41.

- MID IR: -

	LENGTH METERS	DURATION MINUTES
SCREEN	400.	15.00

WP SMOKE SCREEN

	ROUNDS/ 60 METERS	RATE/ MINUTE	TOTAL ROUNDS
105MM:	8.	1.	1319.
155MM:	3.	1.	495.

- FAR IR: -

	LENGTH METERS	DURATION MINUTES
SCREEN	400.	15.00

WP SMOKE SCREEN

	ROUNDS/ 60 METERS	RATE/ MINUTE	TOTAL ROUNDS
105MM:	6.	1.	989.
155MM:	2.	1.	330.

OVERCAST SKY RADIATIVE TRANSFER MODULE

-- RADIATION UNDER OVERCAST SKY --

XO	=	-.067	(KM)	XT	=	-.200 (KM)
YO	=	.000		YT	=	.000
ZO	=	.200		ZT	=	.002
ZC	=	.200		LC	=	2.0000+002 (W/M2-SR-MU)
LAMBDA	=	10.600	(MU)	LG	=	5.0000+001
TEMP	=	9.8	(DEG.C)	LBO	=	5.0000+001
KAPPA	=	4.0000+001	(KM-1)	WO	=	1.000
ETA	=	.700				

**THERMAL CALCULATION OF PATH RADIANCE

BBTEMP= 7.4082+000 W/M2-SR-MU

PATH LENGTH (KM)	TRANSMITTANCE	PATH RADIANCE (W/M2-SR-MU)	CONTRAST TRANSMITTANCE
.239	.00007	2.409+001	.00015

-- RADIATION UNDER OVERCAST SKY --

XO	=	-.133	(KM)	XT	=	-.200 (KM)
YO	=	.000		YT	=	.000
ZO	=	.100		ZT	=	.002
ZC	=	.200		LC	=	2.0000+002 (W/M2-SR-MU)
LAMBDA	=	10.600	(MU)	LG	=	5.0000+001
TEMP	=	9.8	(DEG.C)	LBO	=	5.0000+001
KAPPA	=	4.0000+001	(KM-1)	WO	=	1.000
ETA	=	.700				

**THERMAL CALCULATION OF PATH RADIANCE

BBTEMP= 7.4082+000 W/M2-SR-MU

PATH LENGTH (KM)	TRANSMITTANCE	PATH RADIANCE (W/M2-SR-MU)	CONTRAST TRANSMITTANCE
.119	.00872	9.826-001	.30742

SELF-SCREENING SMOKE GRENADE MODULE

```
*****************************************************************************
*                                PROGRAM GRNADE                             *
*                                  EOSAEL80                                 *
*****************************************************************************
```

*****CARD INPUT*****

NAME							
OUTP	.000	.000	.000	.000	.000	.000	.000
OBSC	200.000	.000	2.000	.000	.000	.000	.000
MUNC	-200.000	.000	2.000	95.000	100.000	10.000	.000
TARC	-200.000	40.000	2.000	.000	.000	.000	.000
BART	5.000	400.000	5.000	90.000	.000	.000	.000
MUNT	1.000	.793	14.300	1.000	.000	4.700	.070
METR	50.000	2.000	220.000	4.000	20.000	.000	.000
EXTC	.000	.000	.000	.000	.000	.000	.000
BURN	.000	.000	.000	.000	.000	.000	.000
MISC	.000	.000	.000	.000	.000	.000	.000
GO	.000	.000	.000	.000	.000	.000	.000

*****INPUT*****ALL LENGTHS IN METERS

METEOROLOGICAL:
- WIND SPEED 3.6 M/S
- WIND DIRECTION 225.0 DEG
- PASQUILL CATEGORY 4
- RELATIVE HUMIDITY 87.1 %
- NOTE: X AXIS HEADING: 90.0 DEG CLOCKWISE FROM NORTH (DCWFN)

OBSERVER/TARGET:

X(OBS)	200.0
Y(OBS)	.0
Z(OBS)	2.0
X(TAR)	-200.0
Y(TAR)	40.0
Z(TAR)	2.0

EXTINCTION COEFFICIENTS:

MICRONS	M**2/GM
0.4-0.7	4.304
0.7-1.2	2.166
1.06	1.541
3.0-5.0	.350
8.0-12.	.338
10.6	.364
94.0 GHZ	.001

TANK/MUNITION DATA:

X(TANK)	-200.0
Y(TANK)	.0
Z(TANK)	2.0
HEADING(DCWFN)	95.0
RANGE	100.0
NO GRNADES	1.0
SMOKE MASS(GM)	359.7
LINE LENGTH	10.0
BURN CONSTANT	.070 1/S
EFFICIENCY	62.0

DIFFUSION PARAMETERS:

SIGZ(XREF)	7.2
XREF	100.0
MIXING HEIGHT(HM)	300.0
SCAVENGING COEFF(HK)	.002
REFLECTION COEFF(RC)	.700
SETTLING VELOCITY(VS)	.021 CM/S
VERTICAL WIND EXPONENT(WPOWR)	.000
VERTICAL DIFF CONSTANT(ZDIFF)	1.400
CROSSWIND DIFF CONSTANT(YDIFF)	.355
YIELD FACTOR	4.7

*****OUTPUT*****

TIME	CL	TRANSMISSION						
(SEC)	(GM/M**2)	0.4-0.7	0.7-1.2	1.06	3.0-5.0	8.0-12.	10.6	94GHZ
5.0	.014	.940	.969	.978	.995	.995	.995	1.000
10.0	4.659	.000	.000	.001	.196	.207	.183	.995
15.0	4.320	.000	.000	.001	.220	.232	.208	.996
20.0	3.819	.000	.000	.003	.263	.275	.249	.996
25.0	3.384	.000	.001	.005	.306	.319	.292	.997
30.0	3.016	.000	.001	.010	.348	.361	.334	.997
35.0	2.705	.000	.003	.015	.388	.401	.374	.997
40.0	2.442	.000	.005	.023	.425	.438	.411	.998
45.0	2.218	.000	.008	.033	.460	.472	.446	.998
50.0	2.028	.000	.012	.044	.492	.504	.478	.998
55.0	1.864	.000	.018	.057	.521	.533	.507	.998
60.0	1.722	.001	.024	.070	.547	.559	.534	.998
65.0	1.599	.001	.031	.085	.571	.582	.559	.998
70.0	1.491	.002	.040	.100	.593	.604	.581	.999
75.0	1.396	.002	.049	.116	.613	.624	.602	.999
80.0	1.312	.004	.058	.132	.632	.642	.620	.999
85.0	1.237	.005	.069	.149	.649	.658	.637	.999
90.0	1.170	.006	.079	.165	.664	.673	.653	.999
95.0	1.110	.008	.090	.181	.678	.687	.668	.999
100.0	1.055	.011	.102	.197	.691	.700	.681	.999
105.0	1.006	.013	.113	.212	.703	.712	.693	.999
110.0	.961	.016	.125	.227	.714	.723	.705	.999
115.0	.920	.019	.136	.242	.725	.733	.716	.999
120.0	.882	.022	.148	.257	.734	.742	.725	.999
125.0	.847	.026	.160	.271	.744	.751	.735	.999
130.0	.814	.030	.171	.285	.752	.759	.743	.999
135.0	.785	.034	.183	.298	.760	.767	.752	.999
140.0	.757	.038	.194	.312	.767	.774	.759	.999
145.0	.731	.043	.205	.324	.774	.781	.766	.999
150.0	.707	.048	.216	.337	.781	.788	.773	.999
155.0	.684	.053	.227	.348	.787	.794	.780	.999
160.0	.663	.058	.238	.360	.793	.799	.786	.999
165.0	.643	.063	.248	.371	.799	.805	.791	.999
170.0	.624	.068	.259	.382	.804	.810	.797	.999
175.0	.606	.074	.269	.393	.809	.815	.802	.999
180.0	.590	.079	.279	.403	.814	.819	.807	.999
185.0	.574	.085	.289	.413	.818	.824	.812	.999
190.0	.559	.090	.298	.423	.822	.828	.816	.999
195.0	.545	.096	.307	.432	.826	.832	.820	.999
200.0	.531	.102	.317	.441	.830	.836	.824	.999
205.0	.518	.108	.326	.450	.834	.839	.828	.999
210.0	.506	.113	.334	.459	.838	.843	.832	.999
215.0	.494	.119	.343	.467	.841	.846	.835	1.000
220.0	.483	.125	.351	.475	.844	.849	.839	1.000
225.0	.472	.131	.360	.483	.848	.852	.842	1.000
230.0	.462	.137	.368	.491	.851	.855	.845	1.000
235.0	.452	.143	.375	.498	.854	.858	.848	1.000
240.0	.443	.149	.383	.505	.856	.861	.851	1.000
245.0	.434	.155	.391	.512	.859	.864	.854	1.000
250.0	.425	.160	.398	.519	.862	.866	.857	1.000
255.0	.417	.166	.405	.526	.864	.869	.859	1.000
260.0	.409	.172	.412	.532	.867	.871	.862	1.000
265.0	.401	.178	.419	.539	.869	.873	.864	1.000
270.0	.394	.184	.426	.545	.871	.875	.866	1.000
275.0	.387	.189	.433	.551	.873	.877	.869	1.000
280.0	.380	.195	.439	.557	.876	.880	.871	1.000
285.0	.373	.201	.446	.563	.878	.881	.873	1.000
290.0	.367	.206	.452	.568	.880	.883	.875	1.000
295.0	.361	.212	.458	.574	.881	.885	.877	1.000
300.0	.355	.217	.464	.579	.883	.887	.879	1.000
305.0	.349	.223	.470	.584	.885	.889	.881	1.000
310.0	.343	.228	.475	.589	.887	.890	.883	1.000
315.0	.338	.234	.481	.594	.888	.892	.884	1.000
320.0	.333	.239	.487	.599	.890	.894	.886	1.000
325.0	.327	.244	.492	.604	.892	.895	.888	1.000
330.0	.322	.250	.497	.608	.893	.897	.889	1.000
335.0	.318	.255	.503	.613	.895	.898	.891	1.000
340.0	.313	.260	.508	.617	.896	.900	.892	1.000
345.0	.308	.265	.513	.622	.898	.901	.894	1.000
350.0	.304	.270	.518	.626	.899	.902	.895	1.000
355.0	.300	.275	.522	.630	.900	.904	.897	1.000
360.0	.296	.280	.527	.634	.902	.905	.898	1.000
365.0	.292	.285	.532	.638	.903	.906	.899	1.000
370.0	.288	.290	.536	.642	.904	.907	.901	1.000
375.0	.284	.295	.541	.646	.905	.909	.902	1.000
380.0	.280	.299	.545	.649	.907	.910	.903	1.000

385.0	.277	.304	.549	.653	.908	.911	.904	1.000
390.0	.273	.309	.554	.657	.909	.912	.905	1.000
395.0	.270	.313	.558	.660	.910	.913	.907	1.000
400.0	.266	.318	.562	.664	.911	.914	.908	1.000

*****CARD INPUT*****

DONE .000 .000 .000 .000 .000 .000
*****PROGRAM GRNADE END*****

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .2331-019

RUN NUMBER 2

INDIVIDUAL MODULES SELECTED
BASCAT
FCLOUD

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	9433.962	9433.962
WAVELENGTH(MICRONS)	1.060	1.060
FREQUENCY(GHZ)	283018.863	283018.863

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

CLASS 1 = FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS 2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLASS 3 = FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLASS 4 = FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS 5 = DUST WITH VIS LT 3 KM.
CLASS 6 = DUST WITH VIS GE 3 KM.
CLASS 7 = DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM.
CLASS 8 = DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM.
CLASS 9 = DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM.
CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM.
CLASS 11 = SNOW WITH VIS LT 1 KM.
CLASS 12 = SNOW WITH 1 LE VIS LT 3 KM.
CLASS 13 = SNOW WITH 3 LE VIS LT 7 KM.
CLASS 14 = SNOW WITH VIS GE 7 KM.
CLASS 15 = NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 16 = NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 17 = VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 18 = VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLASS 19 = CEILING HEIGHT LT 300 M.
CLASS 20 = CEILING HEIGHT LT 1000 M.
CLASS 21 = NO CEILING.
CLASS 22 = ALL CONDITIONS COMBINED.

EOSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS DURING SPRING AT 03-09 (LST).

CLASS NO.	FREQCY CLASS (%)	MEAN TEMP (C)	MEAN DP (C)	MEAN AH (GM/CU.M)	MEAN RH (%)	MEAN VIS (KM)	MEAN PRESS (MB)	MEAN/STDEV WNDVEL (MPS)	MEAN CLDHT (KM)	MEAN/STDEV CLDCVR (%)	FREQCY A (%)	FREQCY B (%)	FREQCY C (%)	FREQCY D (%)	FREQCY E (%)	FREQCY F (%)
1	4.5	3.5	3.2	6.3	98.2	.365	1016.7	1.9/ 1.7	.929	74.4/ 39.2	.0	2.6	12.9	69.2	1.4	14.0
2	8.2	4.1	3.2	6.4	93.8	1.945	1016.1	2.3/ 1.7	1.818	60.0/ 40.4	.0	4.1	19.0	52.6	3.5	19.8
3	17.9	5.1	3.3	6.4	89.0	4.502	1016.4	2.8/ 2.1	2.124	57.8/ 39.4	.0	4.3	15.4	52.4	3.0	19.8
4	9.6	5.8	3.3	6.3	84.5	9.224	1016.6	3.4/ 2.2	1.911	56.2/ 38.6	.0	3.1	12.5	53.1	11.5	19.8
5	.0	.0	.0	.0	.0	.000	.0	.0/ .0	.000	.0/ .0	.0	.0	.0	.0	.0	.0
6	.0	.8	-1.4	5.5	85.5	10.617	1012.3	4.2/ 3.7	1.279	64.0/ 38.2	.0	.0	5.9	64.7	.0	29.4
7	.6	5.9	5.5	7.3	97.6	.460	1013.4	2.9/ 2.6	.204	98.5/ 10.3	.0	.0	.0	97.8	1.3	1.7
8	2.7	6.6	5.8	7.5	95.2	1.955	1011.8	3.6/ 2.3	.565	98.4/ 8.7	.0	1.1	1.9	95.4	1.1	1.7
9	5.8	6.8	5.6	7.3	92.4	4.590	1010.3	4.5/ 2.7	.872	98.0/ 7.3	.0	.2	2.2	94.5	1.1	1.5
10	7.1	6.3	4.6	7.0	88.0	12.633	1008.8	5.6/ 3.0	1.340	92.7/ 14.5	.0	.1	1.8	92.6	.1	1.8
11	.2	-1.0	-1.6	4.4	95.9	.522	1009.9	4.3/ 2.5	.428	96.9/ 13.5	.0	.9	2.7	95.5	.0	.9
12	.7	-1.1	-1.8	4.4	95.0	1.870	1009.9	4.4/ 2.7	.270	97.4/ 10.7	.0	.2	2.5	96.1	.0	.2
13	.8	-1.4	-2.7	4.1	91.2	4.436	1013.4	4.6/ 2.7	.824	94.1/ 14.4	.0	.0	2.7	93.1	.0	1.1
14	.8	-1.5	-2.5	4.2	86.9	13.761	1010.9	5.6/ 3.7	1.359	83.8/ 22.9	.0	.0	1.7	91.1	.0	7.2
15	40.3	5.2	2.3	5.9	82.3	14.756	1015.9	3.9/ 2.6	2.293	51.9/ 37.9	.0	.1	10.1	54.1	1.1	34.6
16	.8	14.7	12.5	11.1	87.3	12.319	1011.7	3.2/ 1.9	2.938	66.5/ 29.8	.0	6.9	17.7	47.6	1.1	26.8
17	3.6	4.0	3.8	6.6	98.4	.343	1015.5	2.4/ 1.9	.888	99.0/ 4.9	.0	.0	1.8	97.1	.0	1.1
18	10.0	4.6	4.0	6.7	96.4	1.312	1014.2	3.1/ 2.2	.189	98.0/ 6.3	.0	.0	2.6	94.9	.1	2.4
19	14.2	4.9	4.2	6.7	95.3	3.326	1013.7	3.9/ 2.6	.143	98.1/ 6.0	.0	.0	1.2	97.1	.1	1.7
20	35.4	5.2	3.8	6.5	90.9	7.312	1012.9	4.4/ 2.8	.399	94.9/ 9.4	.0	.0	2.2	95.8	.1	1.9
21	37.2	4.2	1.8	5.8	85.1	10.680	1018.1	3.9/ 2.3	1.878	16.4/ 17.2	.0	2.6	17.9	28.5	1.1	49.9
22	100.0	5.3	3.2	6.3	87.1	9.302	1015.0	3.6/ 2.6	1.885	63.2/ 38.7	.0	2.5	10.7	61.9	1.1	23.8

CLASS NO.	FREQCY CLASS (%)	FREQCY WNDDIR *015* (%)	FREQCY WNDDIR *045* (%)	FREQCY WNDDIR *075* (%)	FREQCY WNDDIR *105* (%)	FREQCY WNDDIR 135 (%)	FREQCY WNDDIR 165 (%)	FREQCY WNDDIR 195 (%)	FREQCY WNDDIR 225 (%)	FREQCY WNDDIR 255 (%)	FREQCY WNDDIR 285 (%)	FREQCY WNDDIR 315 (%)	FREQCY WNDDIR 345 (%)	FREQCY WNDDIR VBL (%)
1	4.5	8.6	11.1	8.3	4.0	2.8	4.6	5.2	6.9	8.2	7.0	4.7	7.2	21.3
2	8.2	8.5	12.0	10.0	5.9	4.6	5.7	5.7	8.7	8.5	5.8	3.9	5.9	14.4
3	17.9	7.6	10.1	11.4	8.3	6.7	7.0	7.7	10.1	8.4	5.3	3.6	5.3	8.5
4	9.6	6.3	7.7	9.8	7.8	7.4	8.6	9.5	11.3	11.1	6.8	3.9	4.7	5.1
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	23.5	11.8	.0	5.9	5.9	11.8	11.8	5.9	5.9	5.9	11.8
7	.6	8.9	11.1	10.0	3.9	1.4	2.5	4.4	12.2	12.2	10.6	5.2	4.4	6.1
8	2.7	6.5	7.8	6.9	4.2	3.3	4.9	8.3	15.6	14.5	9.7	6.8	5.3	9.0
9	5.8	5.3	6.6	4.7	3.6	4.0	6.8	11.3	19.5	14.9	9.5	6.1	3.1	2.5
10	7.1	4.2	4.1	2.6	2.3	3.6	7.3	12.3	21.7	16.9	11.5	7.4	3.0	1.1
11	.2	10.0	12.7	3.6	1.8	1.8	1.8	3.6	3.6	13.6	12.7	11.8	10.9	11.8
12	.7	11.1	12.9	7.3	3.8	3.1	2.9	4.5	8.2	13.8	10.7	7.8	12.0	1.8
13	.8	10.1	11.5	9.5	3.5	1.9	2.7	4.1	9.3	15.4	10.1	7.8	10.7	3.5
14	.8	7.0	6.8	7.4	3.2	.6	1.9	1.9	8.5	21.9	17.5	12.1	8.7	2.6
15	40.3	7.5	9.1	9.7	6.8	4.7	6.8	8.4	13.2	10.5	7.8	5.0	5.9	4.8
16	.8	3.0	4.5	4.3	10.2	10.8	20.7	11.4	14.0	7.3	2.8	2.8	1.8	6.3
17	3.6	10.0	12.3	8.2	3.8	2.5	3.3	5.0	8.8	10.0	8.8	6.1	8.6	12.1
18	10.0	9.3	12.0	8.0	3.6	2.5	3.5	5.4	11.2	12.6	9.6	6.4	6.6	8.8
19	14.2	9.2	10.9	7.5	2.9	1.7	2.7	5.4	12.7	14.8	10.7	5.9	5.3	5.3
20	35.4	7.5	8.2	6.2	2.9	2.1	4.0	7.6	16.1	16.1	11.6	5.3	3.5	3.9
21	37.2	8.1	10.8	12.7	8.7	6.3	7.2	7.1	8.4	6.9	5.0	3.6	5.5	9.6
22	100.0	7.2	8.9	9.0	6.3	5.0	6.8	8.3	12.7	10.9	7.6	4.9	5.7	6.6

VISIBILITY
9.30 KM

BASCAT LASER SCATTERING MODULE

****
*
* MONTE CARLO MULTIPLE SCATTERING
* AEROSOL SCATTERING
*
****

PARAMETERS FOR THIS RUN

WHITE PHOSPHORUS
WAVELENGTH= 1.060 MICROMETERS ALBEDO= .999
AEROSOL EXTINCTION COEFFICIENT= .4000+001 KM**-1

ELLIPSOIDAL AEROSOL CLOUD
COORDINATE ORIGIN AT CENTER OF CLOUD
Z-AXIS VERTICAL, X-AXIS EAST, Y-AXIS NORTH

SOURCE PARAMETERS
SOURCE XYZ COORDINATES(KM)= -.2000 .0000 -.0980
SOURCE AXIS POLAR ANGLE = 90.000 DEGREES
SOURCE AXIS AZIMUTH ANGLE= .000 DEGREES
SOURCE APERTURE RADIUS(MM)= 50.000
SOURCE BEAM SPREAD ANGLE = .1293-004 RADIANS

DETECTOR PARAMETERS
CONE OF VIEW HALF-ANGLE = 1.000 DEGREES
DETECTOR APERTURE RADIUS = 1.000 CM
DETECTOR XYZ COORDINATES(KM)= .2000 .0000 -.0980
DETECTOR AXIS POLAR ANGLE = 90.000 DEGREES
DETECTOR AXIS AZIMUTH ANGLE=180.000 DEGREES

GROUND PLANE PARAMETERS
ISOTROPIC REFLECTION FROM GROUND PLANE
GROUND PLANE Z-COORDINATE ZG(KM)= -.100
GROUND PLANE ALBEDO, ALBG, = .500

CLOUD PARAMETERS

ELLIPSOID PRINCIPAL XYZ HALF-AXES(KM) =	.1000	.2000	.1000
EULER ANGLES THE,PHE,PSE OF ELLIPSOID(DEG)=	.0000	.0000	.0000
OPTICAL DEPTHS ALONG ELLIPSOID XYZ AXES =	.8000	1.6000	.8000

STEADY STATE POWER TO DETECTOR, FOR UNIT SOURCE POWER

ORDER	STEADY STATE POWER	NUMBER OF PHOTONS
0	.27973-001	.500000+004
1	.66398-009	.500000+004
2	.90441-011	.499900+004
3	.18337-012	.500000+004
4	.12166-013	.500000+004
5	.58910-015	.500000+004
TOTAL	.27973-001	

POWER INTO DETECTOR FOR 1 PULSE(S) OF DIFFERENT LENGTH

PULSE NUMBER 1 HAS LENGTH .2667+000 MICROSECONDS
DETECTOR RESPONSE CUTOFF TIME FOR PULSE NUMBER 1 IS .7067+001 MICROSECONDS

DETECTOR RESPONSE, POWER AS A FUNCTION OF TIME, FOR UNIT PULSE POWER

		POWER FROM EACH ORDER					
TIME	TOTAL POWER	0	1	2	3	4	5
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.1333+000	.2797-001	.2797-001	.6640-009	.9044-011	.1828-012	.1188-013	.2368-015
.2667+000	.2797-001	.2797-001	.6640-009	.9044-011	.1833-012	.1206-013	.4930-015
.4000+000	.1323-014	.0000	.0000	.1887-015	.5639-015	.2577-015	.3125-015
.5333+000	.2141-015	.0000	.0000	.7290-017	.4161-016	.9302-016	.7222-016
.6667+000	.6756-016	.0000	.0000	.0000	.1688-016	.2560-016	.2508-016
.8000+000	.3742-016	.0000	.0000	.0000	.1106-016	.1301-016	.1334-016

FINITE CLOUD RADIATIVE TRANSFER MODULE

-- RADIATIVE TRANSFER THROUGH FINITE CLOUD --

(XC,YC,ZC) = (.0000, .0000, .1000) KILOMETERS
(AE,BE,CE) = (.1000, .2000, .1000)
(XR,YR,ZR) = (.2000, .0000, .0020)
(XS,YS,ZS) = (-.2000, .0000, .0020)

INDEXP =	5	ETA =	.950
LAMBDA =	1.060 (MU)	KAPPA =	4.0000+000 (KM-1)
OMEGA_0 =	1.000	TAUBAR =	2.0000+000
THETA0 =	80.0	PHI0 =	.0 (DEGREES)
RHO =	.500	LB0 =	50.000 (W/M2-SR-MU)
TMPA =	9.8 (DEG.C)	TMPC =	9.8
LD =	0		

**THERMAL CALCULATION OF PATH RADIANCE
BB(LAMBDA,TMPC) = 1.3071-013 W/M2-SR-MU
BB(LAMBDA,TMPA) = 1.3071-013 W/M2-SR-MU

PATH LENGTH (IN CLOUD)	TRANSMITTANCE	PATH RADIANCE (W/M2-SR-MU)	CONTRAST TRANSMITTANCE
.040	8.528-001	1.129-014	1.000+000

RUN NUMBER 3

**** EOSAEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

INDIVIDUAL MODULES SELECTED
NMMW

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	1.167	1.167
WAVELENGTH(MICRONS)	8571.428	8571.428
FREQUENCY(GHZ)	35.000	35.000

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

CLASS 1 = FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS 2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLASS 3 = FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLASS 4 = FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS 5 = DUST WITH VIS LT 3 KM.
CLASS 6 = DUST WITH VIS GE 3 KM.
CLASS 7 = DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM.
CLASS 8 = DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM.
CLASS 9 = DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM.
CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM.
CLASS 11 = SNOW WITH VIS LT 1 KM.
CLASS 12 = SNOW WITH 1 LE VIS LT 3 KM.
CLASS 13 = SNOW WITH 3 LE VIS LT 7 KM.
CLASS 14 = SNOW WITH VIS GE 7 KM.
CLASS 15 = NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 16 = NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 17 = VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 18 = VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLASS 19 = CEILING HEIGHT LT 300 M.
CLASS 20 = CEILING HEIGHT LT 1000 M.
CLASS 21 = NO CEILING.
CLASS 22 = ALL CONDITIONS COMBINED.

EOSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS DURING SPRING AT 03-09 (LST).

CLASS NO.	FREQCY CLASS (%)	MEAN TEMP (C)	MEAN DP (C)	MEAN AH (GM/CU.M)	MEAN RH (%)	MEAN VIS (KM)	MEAN PRESS (MB)	MEAN/STDEV WNDVEL (MPS)	MEAN CLDHT (KM)	MEAN/STDEV CLDCVR (%)	FREQCY A (%)	FREQCY B (%)	FREQCY C (%)	FREQCY D (%)	FREQCY E (%)	FREQCY F (%)
1	4.5	3.5	3.2	6.3	98.2	.365	1016.7	1.9/ 1.7	.929	74.4/ 39.2	.0	2.6	12.9	69.2	1.4	14.0
2	18.2	4.1	3.2	6.4	93.8	1.945	1016.1	2.3/ 1.7	1.818	60.0/ 40.4	.0	4.1	19.0	52.6	4.1	19.2
3	17.9	5.1	3.3	6.4	89.0	4.502	1016.4	2.8/ 2.1	2.124	57.8/ 39.4	.0	4.3	15.4	50.4	8.5	19.2
4	9.6	5.8	3.3	6.3	84.5	9.224	1016.0	3.4/ 2.2	1.911	56.2/ 38.6	.0	3.1	12.5	53.1	11.9	19.2
5	.0	.0	.0	.0	.0	.000	.0	.0/ .0	.000	.0/ .0	.0	.0	.0	.0	.0	.0
6	.0	.8	-1.4	5.5	85.5	10.617	1012.3	4.2/ 3.7	1.279	64.0/ 38.2	.0	.0	5.9	64.7	.0	29.4
7	.6	5.9	5.5	7.3	97.6	.460	1013.4	2.9/ 2.6	.204	98.5/ 10.3	.0	.0	.0	97.8	.3	1.7
8	2.7	6.6	5.8	7.5	95.2	1.955	1011.8	3.6/ 3.3	.565	98.4/ 8.7	.0	.1	1.9	95.4	1.6	1.0
9	5.8	6.8	5.6	7.3	92.4	4.590	1010.3	4.5/ 3.7	.872	98.0/ 7.3	.0	.2	2.0	94.5	1.1	1.5
10	7.1	6.9	4.9	7.0	88.0	12.633	1008.8	5.6/ 3.8	1.340	92.7/ 14.5	.0	.1	1.8	92.5	3.5	1.8
11	.2	-1.0	-1.6	4.4	95.9	.522	1009.9	4.3/ 3.2	.428	96.9/ 13.5	.0	.0	.9	98.2	.0	.9
12	.7	-1.1	-1.8	4.4	95.0	1.870	1009.9	4.4/ 3.2	.270	97.4/ 10.7	.0	.0	.2	99.6	.0	.2
13	.8	-1.4	-2.7	4.1	91.2	4.436	1013.4	4.6/ 3.7	.824	94.1/ 14.4	.0	.0	.8	98.3	.0	1.1
14	.8	.5	-1.5	4.2	86.9	13.761	1010.9	5.6/ 3.7	1.559	83.8/ 22.9	.0	.0	1.7	91.1	1.4	5.8
15	40.3	5.5	3.5	5.9	82.3	14.756	1015.9	3.9/ 2.6	2.695	51.9/ 37.9	.0	.2	10.1	54.8	13.4	19.5
16	.8	14.7	1.5	11.1	87.3	12.319	1011.7	3.2/ 1.9	2.938	66.5/ 29.8	.0	6.9	17.3	47.0	12.8	15.9
17	3.6	4.0	3.8	6.6	98.4	.343	1015.5	2.4/ 1.9	.088	99.0/ 4.9	.0	.0	1.8	97.1	.4	.7
18	10.0	4.6	4.0	6.7	96.4	1.312	1014.2	3.1/ 2.2	.189	98.0/ 6.3	.0	.0	2.6	94.9	1.2	1.4
19	14.2	4.9	4.2	6.7	95.3	3.326	1013.7	3.9/ 2.6	.143	98.1/ 6.0	.0	.0	1.2	97.0	1.0	.7
20	35.4	5.2	3.8	6.5	90.9	7.312	1012.9	4.4/ 2.8	.390	94.9/ 9.4	.0	.0	2.0	93.8	2.7	1.6
21	37.2	4.2	1.8	5.8	85.1	10.680	1018.1	2.9/ 2.3	1.878	16.4/ 17.2	.0	5.6	17.9	28.5	13.4	34.6
22	100.0	5.3	3.2	6.3	87.1	9.302	1015.0	3.6/ 2.6	1.885	63.2/ 38.7	.0	2.5	10.7	61.9	9.1	15.8

CLASS NO.	FREQCY CLASS (%)	FREQCY WNDDIR 015 (%)	FREQCY WNDDIR 045 (%)	FREQCY WNDDIR 075 (%)	FREQCY WNDDIR 105 (%)	FREQCY WNDDIR 135 (%)	FREQCY WNDDIR 165 (%)	FREQCY WNDDIR 195 (%)	FREQCY WNDDIR 225 (%)	FREQCY WNDDIR 255 (%)	FREQCY WNDDIR 285 (%)	FREQCY WNDDIR 315 (%)	FREQCY WNDDIR 345 (%)	FREQCY WNDDIR VBL (%)
1	4.5	8.6	11.1	8.3	4.0	2.8	4.6	5.2	6.9	8.2	7.0	4.7	7.2	21.3
2	18.2	8.5	12.0	10.0	5.9	4.6	5.7	5.7	8.7	8.5	5.8	3.9	5.9	14.8
3	17.9	7.6	10.1	11.4	8.3	6.7	7.0	7.7	10.1	8.4	5.3	3.6	5.3	8.5
4	9.6	6.3	7.7	9.8	7.8	7.4	8.6	9.5	11.3	11.1	6.8	3.9	4.7	5.1
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	23.5	11.8	.0	5.9	5.9	11.8	11.8	5.9	5.9	5.9	11.8
7	.6	8.9	11.1	10.8	3.9	1.4	2.5	4.4	12.2	12.2	10.6	7.2	9.4	6.1
8	2.7	6.5	7.8	6.9	4.2	3.3	4.9	8.3	15.6	14.5	9.7	6.8	5.3	6.9
9	5.8	5.3	6.6	4.7	3.6	4.0	6.8	11.3	19.5	14.9	9.5	6.1	5.1	2.5
10	7.1	4.2	4.1	2.6	2.3	3.6	7.3	12.8	21.7	16.9	11.5	7.4	5.6	1.1
11	.2	10.0	12.7	9.6	1.8	1.8	1.8	3.6	3.6	13.6	12.7	11.8	10.9	11.8
12	.7	11.1	12.9	7.3	3.8	3.1	2.9	4.5	8.2	13.8	10.7	7.8	12.0	1.8
13	.8	10.1	11.5	9.5	4.5	1.9	2.7	4.1	9.3	15.4	10.1	7.8	10.7	3.5
14	.8	7.0	6.8	7.4	3.2	.6	1.9	1.9	8.5	21.9	17.5	12.1	8.7	2.6
15	40.3	7.5	9.1	9.7	6.8	4.7	6.8	8.4	13.2	10.5	7.8	5.0	5.9	4.8
16	.8	3.0	4.5	4.3	10.2	10.8	20.7	11.4	14.0	7.3	2.8	1.8	1.8	6.3
17	3.6	10.0	12.9	8.2	3.8	2.5	3.3	5.0	8.8	10.0	8.8	6.1	8.6	12.1
18	10.0	9.3	12.0	8.0	3.6	2.5	3.5	5.4	11.2	12.6	9.6	6.4	8.3	8.3
19	14.2	9.2	10.9	7.5	3.9	1.7	2.7	5.4	12.7	14.8	10.7	7.9	8.3	5.3
20	35.4	7.5	8.2	6.2	3.9	2.1	4.8	7.6	16.1	16.1	11.6	7.3	7.2	3.3
21	37.2	8.1	10.8	10.7	6.9	6.3	7.2	7.1	8.4	6.9	5.0	3.6	5.2	9.6
22	100.0	7.2	8.9	9.0	6.3	5.0	6.8	8.3	12.7	10.9	7.6	4.9	5.7	6.6

VISIBILITY
9.30 KM

NEAR MILLIMETER WAVE MODULE

TEMPERATURE	5.300	DEGREES C
PRESSURE	1015.000	MB
ABSOLUTE HUMIDITY	6.300	G/M**3
FOG DENSITY	.500	G/M**3
RAIN RATE	5.000	MM/HR
SNOW RATE	.000	MM/HR
FREQUENCY	35.000	GHZ
PATH LENGTH	.400	KM
GAS ABSORPTION	.6838-001	DB/KM
FOG EXTINCTION	.4565+000	DB/KM
RAIN EXTINCTION	.1352+001	DB/KM
SNOW EXTINCTION	.0000	DB/KM
TRANSMISSION	.8413+000	
FOG BACKSCATTER	.5724-010	M**2/M**3
RAIN BACKSCATTER	.1026-003	M**2/M**3
SNOW BACKSCATTER	.0000	M**2/M**3
TOTAL BACKSCATTER	.1026-003	M**2/M**3

TEMPERATURE	5.300	DEGREES C
PRESSURE	1015.000	MB
ABSOLUTE HUMIDITY	6.300	G/M**3
FOG DENSITY	.500	G/M**3
RAIN RATE	.000	MM/HR
SNOW RATE	5.000	MM/HR
FREQUENCY	35.000	GHZ
PATH LENGTH	.400	KM
GAS ABSORPTION	.6838-001	DB/KM
FOG EXTINCTION	.4565+000	DB/KM
RAIN EXTINCTION	.0000	DB/KM
SNOW EXTINCTION	.4708+001	DB/KM
TRANSMISSION	.6175+000	
FOG BACKSCATTER	.5724-010	M**2/M**3
RAIN BACKSCATTER	.0000	M**2/M**3
SNOW BACKSCATTER	.3277-003	M**2/M**3
TOTAL BACKSCATTER	.3277-003	M**2/M**3

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .5195+000

RUN NUMBER 4

**** EOSAEL WARNING ****
VISIBILITY AND EXTINCTION = 0.0, VISIBILITY CHANGED TO 10.0 KM

INDIVIDUAL MODULES SELECTED
SPOT
LT4M

	BEGINNING	ENDING
WAVENUMBER(CM**-1)	2010.000	2710.000
WAVELENGTH(MICRONS)	3.690	4.975
FREQUENCY(GHZ)	60300.000	81300.000

CLIMATOLOGY MODEL

DEFINITIONS OF METEOROLOGICAL CLASSES

CLASS 1 = FOG, HAZE AND MIST WITH VIS LT 1 KM.
CLASS 2 = FOG, HAZE AND MIST WITH 1 LE VIS LT 3 KM.
CLASS 3 = FOG, HAZE AND MIST WITH 3 LE VIS LT 7 KM.
CLASS 4 = FOG, HAZE AND MIST WITH VIS GE 7 KM.
CLASS 5 = DUST WITH VIS LT 3 KM.
CLASS 6 = DUST WITH VIS GE 3 KM.
CLASS 7 = DRIZZLE, RAIN AND TSTMS WITH VIS LT 1 KM.
CLASS 8 = DRIZZLE, RAIN AND TSTMS WITH 1 LE VIS LT 3 KM.
CLASS 9 = DRIZZLE, RAIN AND TSTMS WITH 3 LE VIS LT 7 KM.
CLASS 10 = DRIZZLE, RAIN AND TSTMS WITH VIS GE 7 KM.
CLASS 11 = SNOW WITH VIS LT 1 KM.
CLASS 12 = SNOW WITH 1 LE VIS LT 3 KM.
CLASS 13 = SNOW WITH 3 LE VIS LT 7 KM.
CLASS 14 = SNOW WITH VIS GE 7 KM.
CLASS 15 = NO WEATHER AND ABSOLUTE HUMIDITY LT 10 GM/CU M.
CLASS 16 = NO WEATHER AND ABSOLUTE HUMIDITY GE 10 GM/CU M.
CLASS 17 = VIS LT 1 KM AND CEILING HEIGHT LT 300 M.
CLASS 18 = VIS LT 3 KM AND CEILING HEIGHT LT 1000 M.
CLASS 19 = CEILING HEIGHT LT 300 M.
CLASS 20 = CEILING HEIGHT LT 1000 M.
CLASS 21 = NO CEILING.
CLASS 22 = ALL CONDITIONS COMBINED.

EOSAEL CLIMATOLOGY FOR EUROPEAN LOWLANDS DURING SPRING AT 03-09 (LST).

CLASS NO.	FREQCY CLASS (%)	MEAN TEMP (C)	MEAN DP (C)	MEAN AH (GM/CU.M)	MEAN RH (%)	MEAN VIS (KM)	MEAN PRESS (MB)	MEAN/STDEV WNDVEL (MPS)	MEAN CLDHT (KM)	MEAN/STDEV CLDCVR (%)	FREQCY A (%)	FREQCY B (%)	FREQCY C (%)	FREQCY D (%)	FREQCY E (%)	FREQCY F (%)
1	4.5	3.5	3.2	6.3	98.2	.365	1016.7	1.9/ 1.7	.929	74.4/ 39.2	.0	2.6	12.9	69.2	1.4	14.0
2	8.2	4.1	3.2	6.4	93.8	1.945	1016.1	2.3/ 1.7	1.818	60.0/ 40.4	.0	4.1	19.0	52.6	5.1	19.2
3	17.9	5.1	3.3	6.4	89.0	4.502	1016.4	2.8/ 2.1	2.124	57.8/ 39.4	.0	4.3	15.4	52.4	8.9	19.0
4	9.6	5.8	3.3	6.3	84.5	9.224	1016.0	3.4/ 2.2	1.911	56.2/ 38.6	.0	3.1	12.5	53.1	11.5	19.7
5	.0	.0	.0	.0	.0	.000	.0	.0/ .0	.000	.0/ .0	.0	.0	.0	.0	.0	.0
6	.0	.8	-1.4	5.5	85.5	10.617	1012.3	4.2/ 3.7	1.279	64.0/ 38.2	.0	.0	5.9	64.7	.0	29.4
7	.6	5.9	5.5	7.3	97.6	.460	1013.4	2.9/ 2.6	.204	98.5/ 10.3	.0	.0	.0	98.8	.0	1.7
8	.7	6.6	5.8	7.5	95.2	1.955	1011.8	3.6/ 3.3	.565	98.4/ 8.7	.0	.1	1.9	95.4	1.9	1.9
9	.8	6.8	5.6	7.3	92.4	4.590	1010.3	4.5/ 3.7	.872	98.0/ 7.3	.0	.2	2.2	94.5	1.6	1.5
10	2.1	6.9	4.9	7.0	88.0	12.633	1008.8	5.6/ 3.6	1.340	92.7/ 14.5	.0	.1	1.8	92.8	3.5	1.8
11	.2	-1.0	-1.6	4.4	95.9	.522	1009.9	4.3/ 3.2	.428	96.9/ 13.5	.0	.9	2.7	95.5	.0	.9
12	.7	-1.1	-1.8	4.4	95.0	1.870	1009.9	4.4/ 3.7	.270	97.4/ 10.7	.0	.2	2.2	96.7	.0	.9
13	.8	-1.4	-2.7	4.1	91.2	4.436	1013.4	4.6/ 3.7	.824	94.1/ 14.4	.0	.0	2.7	93.0	2.4	1.9
14	.8	-1.5	-2.5	4.2	86.9	13.761	1010.9	5.6/ 3.7	1.359	83.8/ 22.9	.0	.0	1.7	91.1	14.5	2.6
15	40.3	1.2	-2.3	5.9	82.3	14.756	1015.9	3.9/ 2.6	2.293	51.9/ 37.9	.0	2.1	10.1	54.8	13.4	19.5
16	.8	14.7	12.5	11.1	87.3	12.319	1011.7	3.2/ 1.9	2.938	66.5/ 29.8	.0	6.9	17.7	47.0	12.8	15.7
17	3.6	4.0	3.8	6.6	98.4	.343	1015.5	2.4/ 1.9	.088	99.0/ 4.9	.0	.0	1.8	97.1	.4	.7
18	10.0	4.6	4.2	6.7	96.4	1.312	1014.2	3.1/ 2.2	.189	98.0/ 6.3	.0	.0	2.6	94.9	1.2	1.4
19	14.2	4.9	4.2	6.7	95.3	3.326	1013.7	3.9/ 2.6	.143	98.1/ 6.0	.0	.0	1.2	97.0	1.0	.7
20	35.4	5.2	3.8	6.5	90.9	7.312	1012.9	4.4/ 3.2	.390	94.9/ 9.4	.0	.0	2.0	93.8	2.7	1.6
21	37.2	4.8	1.8	5.8	85.1	10.680	1018.1	2.9/ 2.6	1.878	16.4/ 17.2	.0	5.6	17.9	68.5	13.4	34.6
22	100.0	5.3	3.2	6.3	87.1	9.302	1015.0	3.6/ 2.6	1.885	63.2/ 38.7	.0	2.5	10.7	61.9	9.1	15.8

CLASS NO.	FREQCY CLASS (%)	FREQCY WNDDIR 015 (%)	FREQCY WNDDIR 045 (%)	FREQCY WNDDIR 075 (%)	FREQCY WNDDIR 105 (%)	FREQCY WNDDIR 135 (%)	FREQCY WNDDIR 165 (%)	FREQCY WNDDIR 195 (%)	FREQCY WNDDIR 225 (%)	FREQCY WNDDIR 255 (%)	FREQCY WNDDIR 285 (%)	FREQCY WNDDIR 315 (%)	FREQCY WNDDIR 345 (%)	FREQCY WNDDIR VBL (%)
1	4.5	8.6	11.1	8.3	4.0	2.8	4.6	5.2	6.9	8.2	7.0	4.7	7.2	21.3
2	8.2	8.5	12.0	10.0	5.9	4.6	5.7	5.7	8.7	8.5	5.8	3.9	5.9	14.8
3	17.9	7.6	10.1	11.4	8.3	6.7	7.0	7.7	10.1	8.4	5.3	3.6	5.3	8.5
4	9.6	6.3	7.7	9.8	7.8	7.4	8.6	9.5	11.3	11.1	6.8	3.9	4.7	5.1
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	23.5	11.8	.0	5.9	5.9	11.8	11.8	5.9	5.9	5.9	11.8
7	.6	8.9	11.1	10.0	3.9	1.4	2.5	4.4	12.2	12.2	10.6	7.2	3.4	6.1
8	.7	6.9	7.8	6.9	4.2	3.3	4.9	8.3	15.6	14.5	9.7	6.8	5.3	6.0
9	.8	5.3	6.6	4.7	3.6	4.0	6.8	11.3	19.5	14.9	9.5	6.1	5.1	2.5
10	2.1	4.2	4.1	2.6	2.3	3.6	7.3	12.3	21.7	16.9	11.5	7.4	5.0	1.1
11	.2	10.0	12.7	3.6	1.8	1.8	1.8	3.6	3.6	13.6	12.7	11.8	10.9	11.8
12	.7	11.1	12.9	7.3	3.8	3.1	2.9	4.5	8.2	13.8	10.7	7.8	12.9	1.8
13	.8	10.1	11.5	9.5	3.5	1.9	2.7	4.1	9.3	15.4	10.1	7.8	10.7	3.5
14	.8	7.0	6.8	7.4	3.2	.6	1.9	1.9	8.5	21.9	17.5	12.1	8.7	2.6
15	40.3	7.5	9.1	9.7	6.8	4.7	6.8	8.4	13.2	10.5	7.8	5.0	5.9	4.8
16	.8	14.0	14.5	14.3	10.2	10.8	20.7	11.4	14.0	7.3	2.8	2.8	1.8	6.3
17	3.6	10.0	12.9	8.2	3.8	2.5	3.3	5.0	8.8	10.0	8.8	6.1	8.6	12.1
18	10.0	9.3	13.0	8.0	3.6	2.5	3.5	5.4	11.5	12.6	9.6	6.4	7.6	5.3
19	14.2	9.2	10.9	7.5	3.9	1.7	2.7	5.4	12.5	14.8	10.9	7.9	8.3	3.3
20	35.4	7.5	8.2	6.2	2.9	2.1	4.0	7.6	16.1	16.1	11.6	7.3	7.2	3.2
21	37.2	8.1	10.8	12.7	8.7	6.3	7.2	7.1	8.4	6.9	5.0	3.6	5.2	9.9
22	100.0	7.2	8.9	9.0	6.3	5.0	6.8	8.3	12.7	10.9	7.6	4.9	5.7	6.6

VISIBILITY
9.30 KM

SPOT CONTRAST MODULE

SPOT CONTROL CARDS READ FOR THIS RUN:

ENVR	.3000+001	.2000+001	.2000+001	.4000+001	.1000+001	.1000+001
EMIS	.1000+001	.2830+003	.9500+000	.2950+003	.0000	.0000
ATM	.6500+002	.0000	.0000	.0000	.0000	.0000
TARG	.4000+000	.4500+002	.9000+002	.4500+002	.0000	.0000
REFL	.5000+000	.5000+000	.0000	.5000-001	.0000	.0000
SENS	.2000-002	.9000+002	.2700+003	.1000+001	.0000	.0000
GO	.0000	.0000	.0000	.0000	.0000	.0000

DIVISION LIMITS CHANGED FROM 40.000 TO 60.000

INPUT RESPONSE FUNCTION

WAVELENGTH	R FUNCTION
.3500+001	.7800+000
.3600+001	.8300+000
.3700+001	.8700+000
.3800+001	.9200+000
.3900+001	.9600+000
.4000+001	.9800+000
.4100+001	.9700+000
.4300+001	.9600+000
.4500+001	.9500+000
.4700+001	.9400+000
.4900+001	.9300+000
.5000+001	.9300+000

****

SPOT DIAGNOSTIC MESSAGES FOLLOW:

1. NO DIRECT SUNLIGHT INCIDENT WITHIN RECEIVER'S FIELD OF VIEW.

****

DEFINITION OF CONTROL PARAMETERS FOLLOWS:

PARAMETER	VALUE	DESCRIPTION
ISORC	3	SUNLIGHT AND EMISSION
ITARG	2	TARGET REFLECTANCE / EMISSION
MODEL	4	SUBARCTIC SUMMER
IHAZE	2	

CLOUD BOTTOM HEIGHT = .000 KM

SOURCE INTENSITIES

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	SUNLIGHT SOURCE STRENGTH (WATTS M-2 MICRON-1)	TARGET SOURCE STRENGTH (WATTS M-2 MICRON-1 SR-1)	GROUND SOURCE STRENGTH (WATTS M-2 MICRON-1 SR-1)
4.9751+000	2010	3.8787+000	2.0519+000	.0000
4.8309+000	2070	4.3600+000	1.7739+000	.0000
4.6948+000	2130	4.8846+000	1.5271+000	.0000
4.5662+000	2190	5.4551+000	1.3095+000	.0000
4.4444+000	2250	6.0741+000	1.1187+000	.0000
4.3290+000	2310	6.7442+000	9.5228-001	.0000
4.2194+000	2370	7.4682+000	8.0789-001	.0000
4.1152+000	2430	8.2488+000	6.8321-001	.0000
4.0161+000	2490	9.0890+000	5.7601-001	.0000
3.9216+000	2550	9.9916+000	4.8422-001	.0000
3.8314+000	2610	1.0960+001	4.0593-001	.0000
3.7453+000	2670	1.1996+001	3.3940-001	.0000

COMPONENTS FOR RADIANCE FROM TARGET

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET EMISSION	TARGET REFLECTANCE	PARTIAL ATMOSPHERIC EMISSION	PARTIAL PATH RADIANCE	TOTAL TARGET RADIANCE
4.9751+000	2010	8.3677-001	3.5950-003	6.9555-001	2.6453-004	1.5362+000
4.8309+000	2070	7.0848-001	1.4220-003	6.1320-001	1.0494-004	1.3232+000
4.6948+000	2130	7.8964-001	7.6302-003	3.2337-001	4.8794-004	1.1211+000
4.5662+000	2190	6.8010-001	8.1827-003	2.7173-001	4.8895-004	9.6050-001
4.4444+000	2250	2.2254-001	6.0905-006	6.2320-001	4.6230-007	8.4575-001
4.3290+000	2310	6.5140-004	1.6137-015	7.3146-001	1.0483-024	7.3211-001
4.2194+000	2370	8.7626-003	2.4889-011	6.0666-001	3.2369-016	6.1542-001
4.1152+000	2430	4.3506-001	3.1147-002	5.1388-002	1.2642-003	5.1886-001
4.0161+000	2490	3.7393-001	4.9777-002	3.5402-002	1.9074-003	4.6102-001
3.9216+000	2550	3.1054-001	6.7674-002	3.3520-002	2.6148-003	4.1435-001
3.8314+000	2610	2.6043-001	8.2765-002	2.7776-002	3.2195-003	3.7419-001
3.7453+000	2670	2.1124-001	7.8440-002	2.9776-002	3.1049-003	3.2256-001

COMPONENTS FOR BACKGROUND RADIANCE

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	GROUND EMISSION	GROUND REFLECTANCE	TOTAL ATMOSPHERIC EMISSION	TOTAL PATH RADIANCE	TOTAL BACKGROUND RADIANCE
4.9751+000	2010	.0000	.0000	1.6433+000	1.3950-003	1.6447+000
4.8309+000	2070	.0000	.0000	1.4091+000	3.3501-004	1.4094+000
4.6948+000	2130	.0000	.0000	1.2032+000	2.6254-003	1.2059+000
4.5662+000	2190	.0000	.0000	1.0234+000	4.0099-003	1.0274+000
4.4444+000	2250	.0000	.0000	8.6716-001	1.0520-006	8.6716-001
4.3290+000	2310	.0000	.0000	7.3217-001	1.0655-024	7.3217-001
4.2194+000	2370	.0000	.0000	6.1611-001	3.5637-016	6.1611-001
4.1152+000	2430	.0000	.0000	5.1679-001	1.3371-002	5.3016-001
4.0161+000	2490	.0000	.0000	4.3216-001	2.4559-002	4.5672-001
3.9216+000	2550	.0000	.0000	3.6035-001	3.6939-002	3.9728-001
3.8314+000	2610	.0000	.0000	2.9963-001	4.6921-002	3.4655-001
3.7453+000	2670	.0000	.0000	2.4849-001	4.1007-002	2.8950-001

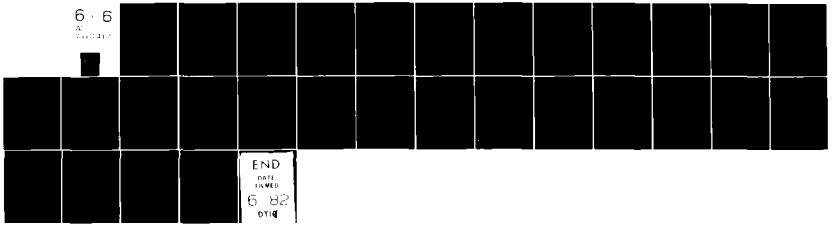
DIRECT SUNLIGHT

(WATTS M-2 MICRON-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	SUNLIGHT SOURCE STRENGTH	SUNLIGHT FLUX
4.9751+000	2010	3.8787+000	.0000
4.8309+000	2070	4.3600+000	.0000
4.6948+000	2130	4.8846+000	.0000
4.5662+000	2190	5.4551+000	.0000
4.4444+000	2250	6.0741+000	.0000
4.3290+000	2310	6.7442+000	.0000
4.2194+000	2370	7.4682+000	.0000
4.1152+000	2430	8.2488+000	.0000
4.0161+000	2490	9.0890+000	.0000
3.9216+000	2550	9.9916+000	.0000
3.8314+000	2610	1.0960+001	.0000
3.7453+000	2670	1.1996+001	.0000

AD-A114 417 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND WS--ETC F/G 4/1
PROGRAM LISTINGS FOR EOSAEL 80-B AND ANCILLARY CODES AGAUS AND --ETC(U)
FEB 82 R G STEINHOFF

UNCLASSIFIED ERADCOM/ASL-TR-0107-V2-SU NL



END
DATE FILMED
6 82
DTIC

TOTAL RADIANCE

(WATTS M-2 MICRON-1 SR-1)

WAVELENGTH (MICRONS)	WAVENUMBER (CM-1)	TARGET	BACKGROUND	CONTRAST RATIO
4.9751+000	2010	1.5362+000	1.6447+000	-6.5959-002
4.8309+000	2070	1.3232+000	1.4094+000	-6.1179-002
4.6948+000	2130	1.1211+000	1.2059+000	-7.0270-002
4.5662+000	2190	9.6050-001	1.0274+000	-6.5111-002
4.4444+000	2250	8.4575-001	8.6716-001	-2.4691-002
4.3290+000	2310	7.3211-001	7.3217-001	-7.7684-005
4.2194+000	2370	6.1542-001	6.1611-001	-1.1165-003
4.1152+000	2430	5.1886-001	5.3016-001	-2.1316-002
4.0161+000	2490	4.6102-001	4.5672-001	9.4083-003
3.9216+000	2550	4.1435-001	3.9728-001	4.2959-002
3.8314+000	2610	3.7419-001	3.4655-001	7.9745-002
3.7453+000	2670	3.2256-001	2.8950-001	1.1420-001

DETECTOR-RESPONSE WAVELENGTH-INTEGRATED

(WATTS M-2 SR-1)

TARGET EMISSION	4.9117-002
TARGET REFLECTANCE	1.8157-003
PARTIAL ATMOSPHERIC EMISSION	4.3029-002
PARTIAL PATH RADIANCE	7.7395-005
TOTAL TARGET RADIANCE	9.4039-002
GROUND EMISSION	.0000
GROUND REFLECTANCE	.0000
TOTAL ATMOSPHERIC EMISSION	9.7352-002
TOTAL PATH RADIANCE	9.1328-004
TOTAL BACKGROUND RADIANCE	9.8265-002

****

CONTRAST -4.3004-002

****

DIRECT SUNLIGHT .0000
(WATTS M-2)

LT4M ATMOSPHERIC TRANSMISSION MODULE

PROGRAM WILL BE EXECUTED IN THE TRANSMISSION MODE

HORIZONTAL PATH, ALTITUDE = .002 KM,RANGE = .400 KM

MODEL ATMOSPHERE 4 = SUB-ARCTIC SUMMER

FREQUENCY RANGE V1= 2010.0 CM-1 TO V2= 2710.0 CM-1 FOR DV = 60.0 CM-1 (3.69 - 4.98 MICRONS)

EQUIVALENT SEA LEVEL ABSORBER AMOUNTS

	WATER VAPOUR GM CM-2	CO2 ETC. KM	OZONE ATM CM	NITROGEN (CONT) KM	H2O (CONT) GM CM-2	MOL SCAT KM	OZONE(U-V) ATM CM
W(1-6,8)=	.355+000	.372+000	.905-003	.295+000	.595-002	.380+000	.915-003
W(10)=					.546-001		

	NITRIC ACID	SO2	NH3	NO2		
W(11-16)=	.000	.834-002	.779-002	.337-002	.433-002	.707-004

FREQ CM-1	WAVELENGTH MICRONS	H2O TRANS	CO2+ TRANS	OZONE TRANS	N2 C TRANS	H2O C TRANS	MOL S TRANS	NITRIC TRANS	SO2 TRANS	HNO3 TRANS	NO2 TRANS	INTEGRATED ABSORPTION	TOTAL TRANS	AEROSOL TRANS
2010	4.9751	.6090	.9987	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	12.6982	.5767	.94820
2070	4.8309	.7608	.9529	.9996	1.0000	.8220	1.0000	1.0000	1.0000	1.0000	1.0000	38.8084	.5648	.94820
2130	4.6948	.8649	.9853	.9990	.9990	.9066	1.0000	1.0000	1.0000	1.0000	1.0000	54.9334	.7313	.94820
2190	4.5662	.9398	.8313	1.0000	.9915	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	70.8646	.7345	.94820
2250	4.4444	.9367	.3230	1.0000	.9807	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	113.9851	.2813	.94820
2310	4.3290	.9769	.0011	1.0000	.9652	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	173.9270	.0010	.94820
2370	4.2194	.9925	.0172	1.0000	.9675	.9791	1.0000	1.0000	1.0000	1.0000	1.0000	233.0067	.0153	.94820
2430	4.1152	.9977	.9941	1.0000	.9723	.9848	1.0000	1.0000	.9971	1.0000	1.0000	238.9729	.9006	.94820
2490	4.0161	.9986	.9941	1.0000	.9890	.9890	1.0000	1.0000	1.0000	1.0000	1.0000	243.8880	.9181	.94820
2550	3.9216	.9903	.9768	1.0000	.9972	.9916	1.0000	1.0000	1.0000	1.0000	1.0000	249.4692	.9070	.94820
2610	3.8314	.9663	.9985	1.0000	.9993	.9925	1.0000	1.0000	1.0000	1.0000	1.0000	255.0312	.9073	.94820
2670	3.7453	.9383	.9981	1.0000	.9996	.9916	1.0000	1.0000	1.0000	1.0000	1.0000	262.2208	.8802	.94820

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

INTEGRATED ASORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358

LT4M ATMOSPHERIC TRANSMISSION MODULE

PROGRAM WILL BE EXECUTED IN THE EMISSION MODE

HORIZONTAL PATH, ALTITUDE = .002 KM,RANGE = .400 KM

MODEL ATMOSPHERE 4 = SUB-ARCTIC SUMMER

FREQUENCY RANGE V1= 2010.0 CM-1 TO V2= 2710.0 CM-1 FOR DV = 60.0 CM-1 (3.69 - 4.98 MICRONS)

EQUIVALENT SEA LEVEL ABSORBER AMOUNTS

	WATER VAPOUR GM CM-2	CO2 ETC. KM	OZONE ATM CM	NITROGEN (CONT) KM	H2O (CONT) GM CM-2	MOL SCAT KM	OZONE(U-V) ATM CM
W(1-6,8)=	.355+000	.372+000	.905-003	.295+000	.595-002	.380+000	.915-003
W(10)=					.546-001		

	NITRIC ACID	SO2	NH3	NO2			
W(11-16)=	.000	.834-002	.779-002	.337-002	.433-002	.707-004	

CUMULATIVE ABSORBER AMOUNTS FOR THE ATMOSPHERIC PATH

	H2O	CO2+	O3	N2	H2O C	MOL S	O3 UV	H2O C	HNO3	SO2	NH3	NO2	TAVE
1	.355+000	.372+000	.905-003	.295+000	.595-002	.380+000	.915-003	.546-001	.000	.834-002	.779-002	.337-002	287.000

FR(CM-1)	WVL(MICRON)	RADIANCE(WATTS/CM2-STER-XXX) PER CM-1	PER MICRON	INTEGRAL	TRANS	AERO TRAN EXTN	ABS
2010.0	4.975124	.17216-006	.69555-004	.51648-005	.576727	.94820	.98942
2070.0	4.830918	.14311-006	.61320-004	.13751-004	.564830	.94820	.98942
2130.0	4.694836	.71276-007	.32337-004	.18028-004	.731250	.94820	.98942
2190.0	4.566210	.56657-007	.27173-004	.21427-004	.734479	.94820	.98942
2250.0	4.444444	.12310-006	.62320-004	.28813-004	.281325	.94820	.98942
2310.0	4.329004	.13708-006	.73146-004	.37038-004	.000967	.94820	.98942
2370.0	4.219409	.10801-006	.60666-004	.43518-004	.015339	.94820	.98942
2430.0	4.115226	.87026-008	.51388-005	.44040-004	.900563	.94820	.98942
2490.0	4.016064	.57098-008	.35402-005	.44383-004	.918083	.94820	.98942
2550.0	3.921569	.51549-008	.33520-005	.44692-004	.906979	.94820	.98942
2610.0	3.831418	.40774-008	.27776-005	.44937-004	.907301	.94820	.98942
2670.0	3.745318	.41768-008	.29776-005	.45188-004	.880173	.94820	.98942

WAVELENGTH AND SENSOR INTEGRATED TRANSMISSION = .4711-001

RADMIN 2610.000 .40774-008
RADMAX 2010.000 .17216-006

INTEGRATED ASORPTION FROM 2010 TO 2730 CM-1 = 262.22, AVERAGE TRANSMITTANCE = .6358
INTEGRATED RADIANCE = .45188-004 WATT CM -2 SR

COMBINED TRANSMISSION FOR THE SELECTED MODULES = .4711-001

END EOSAEL RUN

ELECTRO-OPTICS DISTRIBUTION LIST

Commander
US Army Aviation School
Fort Rucker, AL 36362

Commander
US Army Aviation Center
ATTN: ATZQ-D-MA (Mr. Oliver N. Heath)
Fort Rucker, AL 36362

Commander
US Army Aviation Center
ATTN: ATZQ-D-MS (Mr. Donald Wagner)
Fort Rucker, AL 36362

NASA/Marshall Space Flight Center
ATTN: ES-83 (Otha H. Vaughan, Jr.)
Huntsville, AL 35812

NASA/Marshall Space Flight Center
Atmospheric Sciences Division
ATTN: Code ES-81 (Dr. William W. Vaughan)
Huntsville, AL 35812

Nichols Research Corporation
ATTN: Dr. Lary W. Pinkley
4040 South Memorial Parkway
Huntsville, AL 35802

John M. Hobbie
c/o Kentron International
2003 Byrd Spring Road
Huntsville, AL 35802

Mr. Ray Baker
Lockheed-Missile & Space Company
4800 Bradford Blvd
Huntsville, AL 35807

Commander
US Army Missile Command
ATTN: DRSMI-OG (Mr. Donald R. Peterson)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-OGA (Dr. Bruce W. Fowler)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REL (Dr. George Emmons)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Huey F. Anderson)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Mr. Maxwell W. Harper)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-REO (Mr. Gene Widenhofer)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RHC (Dr. Julius Q. Lilly)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
Redstone Scientific Information Center
ATTN: DRSMI-RPRD (Documents Section)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRA (Dr. Oskar Essenwanger)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRO (Mr. Charles Christensen)
Redstone Arsenal, AL 35809

Commander
US Army Missile Command
ATTN: DRSMI-RRO (Dr. George A. Tanton)
Redstone Arsenal, AL 35809

Commander
US Army Communications Command
ATTN: CC-OPS-PP
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-CS (Mr. Richard G. Cundy)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CD-MD (Mr. Harry Wilder)
Fort Huachuca, AZ 85613

Commander
US Army Intelligence Center & School
ATTN: ATSI-CS-C (2LT Coffman)
Fort Huachuca, AZ 85613

Commander
US Army Yuma Proving Ground
ATTN: STEYP-MSA-TL
Bldg 2105
Yuma, AZ 85364

Northrop Corporation
Electro-Mechanical Division
ATTN: Dr. Richard D. Tooley
500 East Orangethorpe Avenue
Anaheim, CA 92801

Commander
Naval Weapons Center
ATTN: Code 3918 (Dr. Alexis Shlanta)
China Lake, CA 93555

Hughes Helicopters
Army Advanced Attack Helicopter Weapons
ATTN: Mr. Charles R. Hill
Centinela and Teale Streets
Bldg 305, MS T-73A
Culter City, CA 90230

Commander
US Army Combat Developments
Experimentation Command
ATTN: ATEC-PL-M (Mr. Gary G. Love)
Fort Ord, CA 93941

SRI International
ATTN: K2060/Dr. Edward E. Uthe
333 Ravenswood Avenue
Menlo Park, CA 94025

SRI International
ATTN: Mr. J. E. Van der Laan
333 Ravenswood Avenue
Menlo Park, CA 94025

Joane May
Naval Environmental Prediction
Research Facility (NEPRF)
ATTN: Library
Monterey, CA 93940

Sylvania Systems Group,
Western Division
GTE Products Corporation
ATTN: Technical Reports Library
P.O. Box 205
Mountain View, CA 94042

Sylvania Systems Group
Western Division
GTE Products Corporation
ATTN: Mr. Lee W. Carrier
P.O. Box 188
Mountain View, CA 94042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3250-3 (R. de Violini)
Point Mugu, CA 93042

Pacific Missile Test Center
Geophysics Division
ATTN: Code 3253 (Terry E. Battalino)
Point Mugu, CA 93042

Effects Technology Inc.
ATTN: Mr. John D. Carlyle
5383 Hollister Avenue
Santa Barbara, CA 93111

Commander
Naval Ocean Systems Center
ATTN: Code 532 (Dr. Juergen Richter)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 5322 (Mr. Herbert G. Hughes)
San Diego, CA 92152

Commander
Naval Ocean Systems Center
ATTN: Code 4473 (Tech Library)
San Diego, CA 92152

The RAND Corporation
ATTN: Ralph Huschke
1700 Main Street
Santa Monica, CA 90406

Particle Measuring Systems, Inc.
ATTN: Dr. Robert G. Knollenberg
1855 South 57th Court
Boulder, CO 80301

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: Library, R-51, Technical Reports
325 Broadway
Boulder, CO 80303

US Department of Commerce
National Oceanic and Atmospheric Admin
Environmental Research Laboratories
ATTN: R45X3 (Dr. Vernon E. Derr)
Boulder, CO 80303

US Department of Commerce
National Telecommunications and
Information Administration
Institute for Telecommunication Sciences
ATTN: Code 1-3426 (Dr. Hans J. Liebe)
Boulder, CO 80303

AFATL/DLODL
Technical Library
Eglin AFB, FL 32542

Commanding Officer
Naval Training Equipment Center
ATTN: Technical Information Center
Orlando, FL 32813

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. Robert W. McMillan
Atlanta, GA 30332

Georgia Institute of Technology
Engineering Experiment Station
ATTN: Dr. James C. Wiltse
Atlanta, GA 30332

Commandant
US Army Infantry Center
ATTN: ATSH-CD-MS-E (Mr. Robert McKenna)
Fort Benning, GA 31805

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-CS
Fort Gordon, GA 30905

Commander
US Army Signal Center & Fort Gordon
ATTN: ATZHCD-O
Fort Gordon, GA 30905

USAFETAC/DNE
ATTN: Mr. Charles Glauber
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DNDP (LTC Kit G. Cottrell)
Scott AFB, IL 62225

Commander
Air Weather Service
ATTN: AWS/DOOF (MAJ Robert Wright)
Scott AFB, IL 62225

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-CAA-Q (Mr. H. Kent Pickett)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Robert DeKinder, Jr.)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-SAN (Mr. Kent I. Johnson)
Fort Leavenworth, KS 66027

Commander
US Army Combined Arms Center
& Ft. Leavenworth
ATTN: ATZLCA-WE (LTC Darrell Holland)
Fort Leavenworth, KS 66027

President
USAARENBD
ATTN: ATZK-AE-TA (Dr. Charles R. Leake)
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-MS
Fort Knox, KY 40121

Commander
US Army Armor Center and Fort Knox
ATTN: ATZK-CD-SD
Fort Knox, KY 40121

Aerodyne Research Inc.
ATTN: Dr. John F. Ebersole
Crosby Drive
Bedford, MA 01730

Commander
Air Force Geophysics Laboratory
ATTN: OPA (Dr. Robert W. Fenn)
Hanscom AFB, MA 01731

Commander
Air Force Geophysics Laboratory
ATTN: OPI (Dr. Robert A. McClatchey)
Hanscom AFB, MA 01731

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. T. J. Goblick, B-370
P.O. Box 73
Lexington, MA 02173

Massachusetts Institute of Technology
Lincoln Laboratory
ATTN: Dr. Michael Gruber
P.O. Box 73
Lexington, MA 02173

Raytheon Company
Equipment Division
ATTN: Dr. Charles M. Sonnenschein
430 Boston Post Road
Wayland, MA 01778

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-BLB (Mr. Richard McGee)
Aberdeen Proving Ground, MD 21005

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Dr. Edward Stuebing)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLB-PS (Mr. Joseph Vervier)
Aberdeen Proving Ground, MD 21010

Commander/Director
Chemical Systems Laboratory
US Army Armament Research
& Development Command
ATTN: DRDAR-CLY-A (Mr. Ronnald Pennsyle)
Aberdeen Proving Ground, MD 21010

Commander
US Army Ballistic Research Laboratory/
ARRADCOM
ATTN: DRDAR-TSB-S (STINFO)
Aberdeen Proving Ground, MD 21005

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CCM (W. H. Pepper)
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CG/DRDEL-DC/DRDEL-CS
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-CT
2800 Powder Mill Road
Adelphi, MD 20783

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-PAO (M. Singleton)
2800 Powder Mill Road
Adelphi, MD 20783

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK
(Dr. Anthony Van de Wal, Jr.)
Aberdeen Proving Ground, MD 21005

Project Manager
Smoke/Obscurants
ATTN: DRDPM-SMK-T (Mr. Sidney Gerard)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSTE-AD-M (Mr. Warren M. Baity)
Aberdeen Proving Ground, MD 21005

Commander
US Army Test & Evaluation Command
ATTN: DRSTE-AD-M (Dr. Norman E. Pentz)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-AAM (Mr. William Smith)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-CS (Mr. Philip H. Beavers)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GB (Wilbur L. Warfield)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GP (Mr. Fred Campbell)
Aberdeen Proving Ground, MD 21005

Director
US Army Material Systems Analysis Activity
ATTN: DRXSY-GP (H. Stamper)
Aberdeen Proving Grounds, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GS
(Mr. Michael Starks/Mr. Julian Chernick)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-J (Mr James F. O'Bryon)
Aberdeen Proving Ground, MD 21005

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-LM (Mr. Robert M. Marchetti)
Aberdeen Proving Ground, MD 21005

Commander
Harry Diamond Laboratories
ATTN: Dr. William W. Carter
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM (Mr. Robert McCoskey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Robert Humphrey)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Z. G. Sztankay)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Harry Diamond Laboratories
ATTN: DELHD-R-CM-NM (Dr. Joseph Nemarich)
2800 Powder Mill Road
Adelphi, MD 20783

Commander
Air Force Systems Command
ATTN: WER (Mr. Richard F. Picanso)
Andrews AFB, MD 20334

Martin Marietta Laboratories
ATTN: Jar Mo Chen
1450 South Rolling Road
Baltimore, MD 21227

Commander
US Army Concepts Analysis Agency
ATTN: CSCA-SMC (Mr. Hal E. Hock)
8120 Woodmont Avenue
Bethesda, MD 20014

Director
National Security Agency
ATTN: R52/Dr. Douglas Woods
Fort George G. Meade, MD 20755

Chief
Intelligence Materiel Development
& Support Office
US Army Electronic Warfare Laboratory
ATTN: DELEW-I (LTC Kenneth E. Thomas)
Fort George G. Meade, MD 20755

The Johns Hopkins University
Applied Physics Laboratory
ATTN: Dr. Michael J. Lun
John Hopkins Road
Laurell, MD 20810

Dr. Stephen T. Hanley
1720 Rhodesia Avenue
Oxon Hill, MD 20022

Science Applications Inc.
ATTN: Mr. G. D. Currie
15 Research Drive
Ann Arbor, MI 48103

Science Applications Inc.
ATTN: Dr. Robert E. Turner
15 Research Drive
Ann Arbor, MI 48103

Commander
US Army Tank-Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Harry Young)
Warren, MI 48090

Commander
US Army Tank Automotive Research
& Development Command
ATTN: DRDTA-ZSC (Mr. Wallace Mick, Jr.)
Warren, MI 48090

Dr. A. D. Belmont
Research Division
Control Data Corporation
P.O. Box 1249
Minneapolis, MN 55440

Director
US Army Engr Waterways Experiment Station
ATTN: WESEN (Mr. James Mason)
P.O. Box 631
Vicksburg, MS 39180

Dr. Jerry Davis
Department of Marine, Earth
and Atmospheric Sciences
North Carolina State University
Raleigh, NC 27650

Commander
US Army Research Office
ATTN: DRXRO-GS (Dr. Leo Alpert)
P.O. Box 12211
Research Triangle Park, NC 27709

Commander
US Army Research Office
ATTN: DRXRO-PP (Brenda Mann)
P.O. Box 12211
Research Triangle Park, NC 27709

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RD (Dr. K. F. Sterrett)
Hanover, NH 03755

Commander/Director
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. George Aitken)
Hanover, NH 03755

Commander
US Army Cold Regions Research
& Engineering Laboratory
ATTN: CRREL-RG (Mr. Roger H. Berger)
Hanover, NH 03755

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-AC (Mr. James Greenfield)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRDAR-TSS (Bldg #59)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. Peteris Jansons)
Dover, NJ 07801

Commander
US Army Armament Research
& Development Command
ATTN: DRCPM-CAWS-EI (Mr. G. H. Waldron)
Dover, NJ 07801

Deputy Joint Project Manager
for Navy/USMC SAL GP
ATTN: DRCPM-CAWS-NV (CPT Joseph Miceli)
Dover, NJ 07801

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-I (Mr. David Longinotti)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-PE (Mr. Ben A. Di Campli)
Fort Monmouth, NJ 07703

Commander/Director
US Army Combat Surveillance & Target
Acquisition Laboratory
ATTN: DELCS-R-S (Mr. Donald L. Foiani)
Fort Monmouth, NJ 07703

Director
US Army Electronics Technology &
Devices Laboratory
ATTN: DELET-DD (S. Danko)
Fort Monmouth, NJ 07703

Project Manager
FIREFINDER/REMBASS
ATTN: DRCPM-FFR-TM (Mr. John M. Bialo)
Fort Monmouth, NJ 07703

Commander
US Army Electronics Research
& Development Command
ATTN: DRDEL-SA (Dr. Walter S. McAfee)
Fort Monmouth, NJ 07703

OLA, 2WS (MAC)
Holloman AFB, NM 88330
Commander
Air Force Weapons Laboratory
ATTN: AFWL/WE (MAJ John R. Elrick)
Kirtland, AFB, NM 87117

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-SL (Dolores Anguiano)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. Louie Dominguez)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TDB (Mr. William J. Leach)
White Sands Missile Range, NM 88002

Director
USA TRADOC Systems Analysis Activity
ATTN: ATAA-TGP (Mr. Roger F. Willis)
White Sands Missile Range, NM 88002

Director
Office of Missile Electronic Warfare
ATTN: DELEW-M-STO (Dr. Steven Kovel)
White Sands Missile Range, NM 88002

Office of the Test Director
Joint Services EO GW CM Test Program
ATTN: DRXDE-TD (Mr. Weldon Findley)
White Sands Missile Range, NM 88002

Commander
US Army White Sands Missile Range
ATTN: STEWS-PT-AL (Laurel B. Saunders)
White Sands Missile Range, NM 88002

Commander
US Army R&D Coordinator
US Embassy - Bonn
Box 165
APO New York 09080

Grumman Aerospace Corporation
Research Department - MS A08-35
ATTN: John E. A. Selby
Bethpage, NY 11714

Rome Air Development Center
ATTN: Documents Library
TSLD (Bette Smith)
Griffiss AFB, NY 13441

Dr. Roberto Vaglio-Laurin
Faculty of Arts and Science
Dept. of Applied Science
26-36 Stuyvesant Street
New York, NY 10003

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (Mr. Harold Geltmacher)
Wright-Patterson AFB, OH 45433

Air Force Wright Aeronautical Laboratories/
Avionics Laboratory
ATTN: AFWAL/AARI-3 (CPT William C. Smith)
Wright-Patterson AFB, OH 45433

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R (CPT James M. Watson)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CD-MS
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: ATSF-CF-R
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
ATTN: NOAA Liaison Officer
(CDR Jeffrey G. Carlen)
Fort Sill, OK 73503

Commandant
US Army Field Artillery School
Morris Swett Library
ATTN: Reference Librarian
Fort Sill, OK 73503

Commander
Naval Air Development Center
ATTN: Code 301 (Mr. George F. Eck)
Warminster, PA 18974

The University of Texas at El Paso
Electrical Engineering Department
ATTN: Dr. Joseph H. Pierluissi
El Paso, TX 79968

Commandant
US Army Air Defense School
ATTN: ATSA-CD-SC-A (CPT Charles T. Thorn)
Fort Bliss, TX 79916

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-OP-Q (CPT Henry C. Cobb, Jr.)
Fort Hood, TX 76544

Commander
HQ, TRADOC Combined Arms Test Activity
ATTN: ATCAT-SCI (Dr. Darrell W. Collier)
Fort Hood, TX 76544

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-L
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-M (Mr. Paul E. Carlson)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. John Trethewey)
Dugway, UT 84022

Commander
US Army Dugway Proving Ground
ATTN: STEDP-MT-DA-T (Mr. William Peterson)
Dugway, UT 84022

Defense Documentation Center
ATTN: DDC-TCA
Cameron Station Bldg 5
Alexandria, VA 22314
12

Ballistic Missile Defense Program Office
ATTN: DACS-BMT (Colonel Harry F. Ennis)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Technical Information Center
ATTN: DDA-2 (Mr. James E. Shafer)
Cameron Station, Bldg 5
Alexandria, VA 22314

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCBSI-EE (Mr. Albert Giambalvo)
5001 Eisenhower Avenue
Alexandria, VA 22333

Commander
US Army Materiel Development
& Readiness Command
ATTN: DRCLDC (Mr. James Bender)
5001 Eisenhower Avenue
Alexandria, VA 22333

Defense Advanced Rsch Projects Agency
ATTN: Steve Zakanyez
1400 Wilson Blvd
Arlington, VA 22209

Defense Advanced Rsch Projects Agency
ATTN: Dr. James Tegnelia
1400 Wilson Blvd
Arlington, VA 22209

Institute for Defense Analyses
ATTN: Mr. Lucien M. Biberman
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Ernest Bauer
400 Army-Navy Drive
Arlington, VA 22202

Institute for Defense Analyses
ATTN: Dr. Hans G. Wolfhard
400 Army-Navy Drive
Arlington, VA 22202

System Planning Corporation
ATTN: Mr. Daniel Friedman
1500 Wilson Boulevard
Arlington, VA 22209

System Planning Corporation
ATTN: COL Hank Shelton
1500 Wilson Boulevard
Arlington, VA 22209

US Army Intelligence & Security Command
ATTN: Edwin Speakman, Scientific Advisor
Arlington Hall Station
Arlington, VA 22212

Commander
US Army Operational Test
& Evaluation Agency
ATTN: CSTE-ED (Mr. Floyd I. Hill)
5600 Columbia Pike
Falls Church, VA 22041

Commander and Director
US Army Engineer Topographic Laboratories
ATTN: ETL-GS-A (Mr. Thomas Neidringhaus)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Rudolf G. Buser)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-L (Dr. Robert S. Rodhe)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Joseph R. Moulton)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Luanne P. Obert)
Fort Belvoir, VA 22060

Director
US Army Night Vision
& Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Thomas W. Cassidy)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Mr. Richard J. Bergemann)
Fort Belvoir, VA 22060

Director
US Army Night Vision &
Electro-Optics Laboratory
ATTN: DELNV-VI (Dr. James A. Ratches)
Fort Belvoir, VA 22060

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-AN-M
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-F-A (Mr. Chris O'Connor, Jr.)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-IE-R (Mr. David M. Ingram)
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATCD-M-I/ATCD-M-A
Fort Monroe, VA 23651

Commander
US Army Training & Doctrine Command
ATTN: ATDOC-TA (Dr. Marvin P. Pastel)
Fort Monroe, VA 23651

Department of the Air Force
OL-I, AWS
Fort Monroe, VA 23651

Department of the Air Force
HQS 5 Weather Wing (MAC)
ATTN: 5 WW/DN
Langley Air Force Base, VA 23655

Commander
US Army INSCOM/Quest Research Corporation
ATTN: Mr. Donald Wilmot
6845 Elm Street, Suite 407
McLean, VA 22101

General Research Corporation
ATTN: Dr. Ralph Zirkind
7655 Old Springhouse Road
McLean, VA 22102

Science Applications, Inc.
8400 Westpark Drive
ATTN: Dr. John E. Cockayne
McLean, VA 22102

US Army Nuclear & Chemical Agency
ATTN: MONA-WE (Dr. John A. Berberet)
7500 Backlick Road, Bldg 2073
Springfield, VA 22150

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-EA (Mr. Douglas Harkleroad)
Vint Hill Farms Station
Warrenton, VA 22186

Director
US Army Signals Warfare Laboratory
ATTN: DELSW-OS (Dr. Royal H. Burkhardt)
Vint Hill Farms Station
Warrenton, VA 22186

Commander
US Army Cold Regions Test Center
ATTN: STECR-TD (Mr. Jerold Barger)
APO Seattle, WA 98733

HQDA (SAUS-OR/Hunter M. Woodall, Jr./
Dr. Herbert K. Fallin)
Rm 2E 614, Pentagon
Washington, DC 20301

COL Elbert W. Friday, Jr.
OUSDRE
Rm 3D 129, Pentagon
Washington, DC 20301

Defense Communications Agency
Technical Library Center
Code 222
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: Technical Library (Mrs. Betty Fox)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: RAAE (Dr. Carl Fitz)
Washington, DC 20305

Director
Defense Nuclear Agency
ATTN: SPAS (Mr. Donald J. Kohler)
Washington, DC 20305

Defense Intelligence Agency
ATTN: DT/AC (LTC Robert Poplawski)
Washington, DC 20301

HQDA (DAMA-ARZ-D/Dr. Verderame)
Washington, DC 20310

HQDA (DAMI-ISP/Mr. Beck)
Washington, DC 20310

Department of the Army
Deputy Chief of Staff for
Operations and Plans
ATTN: DAMO-RQ
Washington, DC 20310

Department of the Army
Director of Telecommunications and
Command and Control
ATTN: DAMO-TCZ
Washington, DC 20310

Department of the Army
Assistant Chief of Staff for Intelligence
ATTN: DAMI-TS
Washington, DC 20310

HQDA (DAEN-RDM/Dr. de Percin)
Casimir Pulaski Building
20 Massachusetts Avenue
Room 6203
Washington, DC 20314

National Science Foundation
Division of Atmospheric Sciences
ATTN: Dr. Eugene W. Bierly
1800 G. Street, N.W.
Washington, DC 20550

Director
Naval Research Laboratory
ATTN: Code 4320 (Dr. Lothar H. Ruhnke)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6009 (Dr. John MacCallum, Jr.)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6530 (Mr. Raymond A. Patten)
Washington, DC 20375

Commanding Officer
Naval Research Laboratory
ATTN: Code 6533 (Dr. James A. Dowling)
Washington, DC 20375

ATMOSPHERIC SCIENCES RESEARCH REPORTS

1. Lindberg, J. D. "An Improvement to a Method for Measuring the Absorption Coefficient of Atmospheric Dust and other Strongly Absorbing Powders," ECOM-5565, July 1975.

2. Avara, Elton P., "Mesoscale Wind Shears Derived from Thermal Winds," ECOM-5566, July 1975.

3. Gomez, Richard B., and Joseph H. Pierluissi, "Incomplete Gamma Function Approximation for King's Strong-Line Transmittance Model," ECOM-5567, July 1975.

4. Blanco, A. J., and B. F. Engebos, "Ballistic Wind Weighting Functions for Tank Projectiles," ECOM-5568, August 1975.

5. Taylor, Fredrick J., Jack Smith, and Thomas H. Pries, "Crosswind Measurements through Pattern Recognition Techniques," ECOM-5569, July 1975.

6. Walters, D. L., "Crosswind Weighting Functions for Direct-Fire Projectiles," ECOM-5570, August 1975.

7. Duncan, Louis D., "An Improved Algorithm for the Iterated Minimal Information Solution for Remote Sounding of Temperature," ECOM-5571, August 1975.

8. Robbiani, Raymond L., "Tactical Field Demonstration of Mobile Weather Radar Set AN/TPS-41 at Fort Rucker, Alabama," ECOM-5572, August 1975.

9. Miers, B., G. Blackman, D. Langer, and N. Lorimier, "Analysis of SMS/GOES Film Data," ECOM-5573, September 1975.

10. Manquero, Carlos, Louis Duncan, and Rufus Bruce, "An Indication from Satellite Measurements of Atmospheric CO_2 Variability," ECOM-5574, September 1975.

11. Petracca, Carmine, and James D. Lindberg, "Installation and Operation of an Atmospheric Particulate Collector," ECOM-5575, September 1975.

12. Avara, Elton P., and George Alexander, "Empirical Investigation of Three Iterative Methods for Inverting the Radiative Transfer Equation," ECOM-5576, October 1975.

13. Alexander, George D., "A Digital Data Acquisition Interface for the SMS Direct Readout Ground Station - Concept and Preliminary Design," ECOM-5577, October 1975.

14. Cantor, Israel, "Enhancement of Point Source Thermal Radiation Under Clouds in a Nonattenuating Medium," ECOM-5578, October 1975.

15. Norton, Colburn, and Glenn Hoidale, "The Diurnal Variation of Mixing Height by Month over White Sands Missile Range, NM," ECOM-5579, November 1975.

16. Avara, Elton P., "On the Spectrum Analysis of Binary Data," ECOM-5580, November 1975.

17. Taylor, Fredrick J., Thomas H. Pries, and Chao-Huan Huang, "Optimal Wind Velocity Estimation," ECOM-5581, December 1975.

18. Avara, Elton P., "Some Effects of Autocorrelated and Cross-Correlated Noise on the Analysis of Variance," ECOM-5582, December 1975.

19. Gillespie, Patti S., R. L. Armstrong, and Kenneth O. White, "The Spectral Characteristics and Atmospheric CO_2 Absorption of the Ho^{+3}:YLF Laser at 2.05μm," ECOM-5583, December 1975.

20. Novlan, David J., "An Empirical Method of Forecasting Thunderstorms for the White Sands Missile Range," ECOM-5584, February 1976.

21. Avara, Elton P., "Randomization Effects in Hypothesis Testing with Autocorrelated Noise," ECOM-5585, February 1976.

22. Watkins, Wendell R., "Improvements in Long Path Absorption Cell Measurement," ECOM-5586, March 1976.

23. Thomas, Joe, George D. Alexander, and Marvin Dubbin, "SATTEL - An Army Dedicated Meteorological Telemetry System," ECOM-5587, March 1976.

24. Kennedy, Bruce W., and Delbert Bynum, "Army User Test Program for the RDT&E-XM-75 Meteorological Rocket," ECOM-5588, April 1976.

25. Barnett, Kenneth M., "A Description of the Artillery Meteorological Comparisons at White Sands Missile Range, October 1974 - December 1974 ('PASS' - Prototype Artillery [Meteorological] Subsystem)," ECOM-5589, April 1976.

26. Miller, Walter B., "Preliminary Analysis of Fall-of-Shot From Project 'PASS'," ECOM-5590, April 1976.

27. Avara, Elton P., "Error Analysis of Minimum Information and Smith's Direct Methods for Inverting the Radiative Transfer Equation," ECOM-5591, April 1976.

28. Yee, Young P., James D. Horn, and George Alexander, "Synoptic Thermal Wind Calculations from Radiosonde Observations Over the Southwestern United States," ECOM-5592, May 1976.

29. Duncan, Louis D., and Mary Ann Seagraves, "Applications of Empirical Corrections to NOAA-4 VTPR Observations," ECOM-5593, May 1976.

30. Miers, Bruce T., and Steve Weaver, "Applications of Meteorological Satellite Data to Weather Sensitive Army Operations," ECOM-5594, May 1976.

31. Sharenow, Moses, "Redesign and Improvement of Balloon ML-566," ECOM-5595, June 1976.

32. Hansen, Frank V., "The Depth of the Surface Boundary Layer," ECOM-5596, June 1976.

33. Pinnick, R. G., and E. B. Stenmark, "Response Calculations for a Commerical Light-Scattering Aerosol Counter," ECOM-5597, July 1976.

34. Mason, J., and G. B. Hoidale, "Visibility as an Estimator of Infrared Transmittance," ECOM-5598, July 1976.

35. Bruce, Rufus E., Louis D. Duncan, and Joseph H. Pierluissi, "Experimental Study of the Relationship Between Radiosonde Temperatures and Radiometric-Area Temperatures," ECOM-5599, August 1976.

36. Duncan, Louis D., "Stratospheric Wind Shear Computed from Satellite Thermal Sounder Measurements," ECOM-5800, September 1976.

37. Taylor, F., P. Mohan, P. Joseph, and T. Pries, "An All Digital Automated Wind Measurement System," ECOM-5801, September 1976.

38. Bruce, Charles, "Development of Spectrophones for CW and Pulsed Radiation Sources," ECOM-5802, September 1976.

39. Duncan, Louis D., and Mary Ann Seagraves, "Another Method for Estimating Clear Column Radiances," ECOM-5803, October 1976.

40. Blanco, Abel J., and Larry E. Taylor, "Artillery Meteorological Analysis of Project Pass," ECOM-5804, October 1976.

41. Miller, Walter, and Bernard Engebos, "A Mathematical Structure for Refinement of Sound Ranging Estimates," ECOM-5805, November 1976.

42. Gillespie, James B., and James D. Lindberg, "A Method to Obtain Diffuse Reflectance Measurements from 1.0 and 3.0μm Using a Cary 17I Spectrophotometer," ECOM-5806, November 1976.

43. Rubio, Roberto, and Robert O. Olsen, "A Study of the Effects of Temperature Variations on Radio Wave Absorption," ECOM-5807, November 1976.

44. Ballard, Harold N., "Temperature Measurements in the Stratosphere from Balloon-Borne Instrument Platforms, 1968-1975," ECOM-5808, December 1976.

45. Monahan, H. H., "An Approach to the Short-Range Prediction of Early Morning Radiation Fog," ECOM-5809, January 1977.

46. Engebos, Bernard Francis, "Introduction to Multiple State Multiple Action Decision Theory and Its Relation to Mixing Structures," ECOM-5810, January 1977.

47. Low, Richard D. H., "Effects of Cloud Particles on Remote Sensing from Space in the 10-Micrometer Infrared Region," ECOM-5811, January 1977.

48. Bonner, Robert S., and R. Newton, "Application of the AN/GVS-5 Laser Rangefinder to Cloud Base Height Measurements," ECOM-5812, February 1977.

49. Rubio, Roberto, "Lidar Detection of Subvisible Reentry Vehicle Erosive Atmospheric Material," ECOM-5813, March 1977.

50. Low, Richard D. H., and J. D. Horn, "Mesoscale Determination of Cloud-Top Height: Problems and Solutions," ECOM-5814, March 1977.

51. Duncan, Louis D., and Mary Ann Seagraves, "Evaluation of the NOAA-4 VTPR Thermal Winds for Nuclear Fallout Predictions," ECOM-5815, March 1977.

52. Randhawa, Jagir S., M. Izquierdo, Carlos McDonald, and Zvi Salpeter, "Stratospheric Ozone Density as Measured by a Chemiluminescent Sensor During the Stratcom VI-A Flight," ECOM-5816, April 1977.

53. Rubio, Roberto, and Mike Izquierdo, "Measurements of Net Atmospheric Irradiance in the 0.7- to 2.8-Micrometer Infrared Region," ECOM-5817, May 1977.

54. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, Consultant for Chemical Kinetics, "Calculation of Selected Atmospheric Composition Parameters for the Mid-Latitude, September Stratosphere," ECOM-5818, May 1977.

55. Mitchell, J. D., R. S. Sagar, and R. O. Olsen, "Positive Ions in the Middle Atmosphere During Sunrise Conditions," ECOM-5819, May 1977.

56. White, Kenneth O., Wendell R. Watkins, Stuart A. Schleusener, and Ronald L. Johnson, "Solid-State Laser Wavelength Identification Using a Reference Absorber," ECOM-5820, June 1977.

57. Watkins, Wendell R., and Richard G. Dixon, "Automation of Long-Path Absorption Cell Measurements," ECOM-5821, June 1977.

58. Taylor, S. E., J. M. Davis, and J. B. Mason, "Analysis of Observed Soil Skin Moisture Effects on Reflectance," ECOM-5822, June 1977.

59. Duncan, Louis D., and Mary Ann Seagraves, "Fallout Predictions Computed from Satellite Derived Winds," ECOM-5823, June 1977.

60. Snider, D. E., D. G. Murcray, F. H. Murcray, and W. J. Williams, "Investigation of High-Altitude Enhanced Infrared Backround Emissions," (U), SECRET, ECOM-5824, June 1977.

51. Dubbin, Marvin H., and Dennis Hall, "Synchronous Meteorological Satellite Direct Readout Ground System Digital Video Electronics," ECOM-5825, June 1977.

62. Miller, W., and B. Engebos, "A Preliminary Analysis of Two Sound Ranging Algorithms," ECOM-5826, July 1977.

63. Kennedy, Bruce W., and James K. Luers, "Ballistic Sphere Techniques for Measuring Atmospheric Parameters," ECOM-5827, July 1977.

64. Duncan, Louis D., "Zenith Angle Variation of Satellite Thermal Sounder Measurements," ECOM-5828, August 1977.

65. Hansen, Frank V., "The Critical Richardson Number," ECOM-5829, September 1977.

66. Ballard, Harold N., and Frank P. Hudson (Compilers), "Stratospheric Composition Balloon-Borne Experiment," ECOM-5830, October 1977.

67. Barr, William C., and Arnold C. Peterson, "Wind Measuring Accuracy Test of Meteorological Systems," ECOM-5831, November 1977.

68. Ethridge, G. A., and F. V. Hansen, "Atmospheric Diffusion: Similarity Theory and Empirical Derivations for Use in Boundary Layer Diffusion Problems," ECOM-5832, November 1977.

69. Low, Richard D. H., "The Internal Cloud Radiation Field and a Technique for Determining Cloud Blackness," ECOM-5833, December 1977.

70. Watkins, Wendell R., Kenneth O. White, Charles W. Bruce, Donald L. Walters, and James D. Lindberg, "Measurements Required for Prediction of High Energy Laser Transmission," ECOM-5834, December 1977.

71. Rubio, Robert, "Investigation of Abrupt Decreases in Atmospherically Backscattered Laser Energy," ECOM-5835, December 1977.

72. Monahan, H. H., and R. M. Cionco, "An Interpretative Review of Existing Capabilities for Measuring and Forecasting Selected Weather Variables (Emphasizing Remote Means)," ASL-TR-0001, January 1978.

73. Heaps, Melvin G., "The 1979 Solar Eclipse and Validation of D-Region Models," ASL-TR-0002, March 1978.

74. Jennings, S. G., and J. B. Gillespie, "M.I.E. Theory Sensitivity Studies - The Effects of Aerosol Complex Refractive Index and Size Distribution Variations on Extinction and Absorption Coefficients, Part II: Analysis of the Computational Results," ASL-TR-0003, March 1978.

75. White, Kenneth O., et al, "Water Vapor Continuum Absorption in the 3.5μm to 4.0μm Region," ASL-TR-0004, March 1978.

76. Olsen, Robert O., and Bruce W. Kennedy, "ABRES Pretest Atmospheric Measurements," ASL-TR-0005, April 1978.

77. Ballard, Harold N., Jose M. Serna, and Frank P. Hudson, "Calculation of Atmospheric Composition in the High Latitude September Stratosphere," ASL-TR-0006, May 1978.

78. Watkins, Wendell R., et al, "Water Vapor Absorption Coefficients at HF Laser Wavelengths," ASL-TR-0007, May 1978.

79. Hansen, Frank V., "The Growth and Prediction of Nocturnal Inversions," ASL-TR-0008, May 1978.

80. Samuel, Christine, Charles Bruce, and Ralph Brewer, "Spectrophone Analysis of Gas Samples Obtained at Field Site," ASL-TR-0009, June 1978.

81. Pinnick, R. G., et al., "Vertical Structure in Atmospheric Fog and Haze and its Effects on IR Extinction," ASL-TR-0010, July 1978.

82. Low, Richard D. H., Louis D. Duncan, and Richard B. Gomez, "The Microphysical Basis of Fog Optical Characterization," ASL-TR-0011, August 1978.

83. Heaps, Melvin G., "The Effect of a Solar Proton Event on the Minor Neutral Constituents of the Summer Polar Mesosphere," ASL-TR-0012, August 1978.

84. Mason, James B., "Light Attenuation in Falling Snow," ASL-TR-0013, August 1978.

85. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' Meteorological Application," ASL-TR-0014, September 1978.

86. Heaps, M. G., and F. E. Niles, "Modeling of Ion Chemistry of the D-Region: A Case Study Based Upon the 1966 Total Solar Eclipse," ASL-TR-0015, September 1978.

87. Jennings, S. G., and R. G. Pinnick, "Effects of Particulate Complex Refractive Index and Particle Size Distribution Variations on Atmospheric Extinction and Absorption for Visible Through Middle-Infrared Wavelengths," ASL-TR-0016, September 1978.

88. Watkins, Wendell R., Kenneth O. White, Lanny R. Bower, and Brian Z. Sojka, "Pressure Dependence of the Water Vapor Continuum Absorption in the 3.5- to 4.0-Micrometer Region," ASL-TR-0017, September 1978.

89. Miller, W. B., and B. F. Engebos, "Behavior of Four Sound Ranging Techniques in an Idealized Physical Environment," ASL-TR-0018, September 1978.

90. Gomez, Richard G., "Effectiveness Studies of the CBU-88/B Bomb, Cluster, Smoke Weapon," (U), CONFIDENTIAL ASL-TR-0019, September 1978.

91. Miller, August, Richard C. Shirkey, and Mary Ann Seagraves, "Calculation of Thermal Emission from Aerosols Using the Doubling Technique," ASL-TR-0020, November 1978.

92. Lindberg, James D., et al, "Measured Effects of Battlefield Dust and Smoke on Visible, Infrared, and Millimeter Wavelengths Propagation: A Preliminary Report on Dusty Infrared Test-I (DIRT-I)," ASL-TR-0021, January 1979.

93. Kennedy, Bruce W., Arthur Kinghorn, and B. R. Hixon, "Engineering Flight Tests of Range Meteorological Sounding System Radiosonde," ASL-TR-0022, February 1979.

94. Rubio, Roberto, and Don Hoock, "Microwave Effective Earth Radius Factor Variability at Wiesbaden and Balboa," ASL-TR-0023, February 1979.

95. Low, Richard D. H., "A Theoretical Investigation of Cloud/Fog Optical Properties and Their Spectral Correlations, "ASL-TR-0024, February 1979.

96. Pinnick, R. G., and H. J. Auvermann, "Response Characteristics of Knollenberg Light-Scattering Aerosol Counters," ASL-TR-0025, February 1979.

97. Heaps, Melvin G., Robert O. Olsen, and Warren W. Berning, "Solar Eclipse 1979, Atmospheric Sciences Laboratory Program Overview," ASL-TR-0026, February 1979.

98. Blanco, Abel J., "Long-Range Artillery Sound Ranging: 'PASS' GR-8 Sound Ranging Data," ASL-TR-0027, March 1979.

99. Kennedy, Bruce W., and Jose M. Serna, "Meteorological Rocket Network System Reliability," ASL-TR-0028, March 1979.

100. Swingle, Donald M., "Effects of Arrival Time Errors in Weighted Range Equation Solutions for Linear Base Sound Ranging," ASL-TR-0029, April 1979.

101. Umstead, Robert K., Ricardo Pena, and Frank V. Hansen, "KWIK: An Algorithm for Calculating Munition Expenditures for Smoke Screening/Obscuration in Tactical Situations," ASL-TR-0030, April 1979.

102. D'Arcy, Edward M., "Accuracy Validation of the Modified Nike Hercules Radar," ASL-TR-0031, May 1979.

103. Rodriguez, Ruben, "Evaluation of the Passive Remote Crosswind Sensor," ASL-TR-0032, May 1979.

104. Barber, T. L., and R. Rodriguez, "Transit Time Lidar Measurement of Near-Surface Winds in the Atmosphere," ASL-TR-0033, May 1979.

105. Low, Richard D. H., Louis D. Duncan, and Y. Y. Roger R. Hsiao, "Microphysical and Optical Properties of California Coastal Fogs at Fort Ord," ASL-TR-0034, June 1979.

106. Rodriguez, Ruben, and William J. Vechione, "Evaluation of the Saturation Resistant Crosswind Sensor," ASL-TR-0035, July 1979.

107. Ohmstede, William D., "The Dynamics of Material Layers," ASL-TR-0036, July 1979.

108. Pinnick, R. G., S. G. Jennings, Petr Chylek, and H. J. Auvermann, "Relationships between IR Extinction Absorption, and Liquid Water Content of Fogs," ASL-TR-0037, August 1979.

109. Rodriguez, Ruben, and William J. Vechione, "Performance Evaluation of the Optical Crosswind Profiler," ASL-TR-0038, August 1979.

110. Miers, Bruce T., "Precipitation Estimation Using Satellite Data," ASL-TR-0039, September 1979.

111. Dickson, David H., and Charles M. Sonnenschein, "Helicopter Remote Wind Sensor System Description," ASL-TR-0040, September 1979.

112. Heaps, Melvin G., and Joseph M. Heimerl, "Validation of the Dairchem Code, I: Quiet Midlatitude Conditions," ASL-TR-0041, September 1979.

113. Bonner, Robert S., and William J. Lentz, "The Visioceilometer: A Portable Cloud Height and Visibility Indicator," ASL-TR-0042, October 1979.

114. Cohn, Stephen L., "The Role of Atmospheric Sulfates in Battlefield Obscurations," ASL-TR-0043, October 1979.

115. Fawbush, E. J., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF), White Sands Missile Range, New Mexico, Part I, 24 March to 8 April 1977," ASL-TR-0044, November 1979.

116. Barber, Ted L., "Short-Time Mass Variation in Natural Atmospheric Dust," ASL-TR-0045, November 1979.

117. Low, Richard D. H., "Fog Evolution in the Visible and Infrared Spectral Regions and its Meaning in Optical Modeling," ASL-TR-0046, December 1979.

118. Duncan, Louis D., et al, "The Electro-Optical Systems Atmospheric Effects Library, Volume I: Technical Documentation," ASL-TR-0047, December 1979.

119. Shirkey, R. C., et al, "Interim E-O SAEL, Volume II, Users Manual," ASL-TR-0048, December 1979.

120. Kobayashi, H. K., "Atmospheric Effects on Millimeter Radio Waves," ASL-TR-0049, January 1980.

121. Seagraves, Mary Ann, and Louis D. Duncan, "An Analysis of Transmittances Measured Through Battlefield Dust Clouds," ASL-TR-0050, February 1980.

122. Dickson, David H., and Jon E. Ottesen, "Helicopter Remote Wind Sensor Flight Test," ASL-TR-0051, February 1980.

123. Pinnick, R. G., and S. G. Jennings, "Relationships Between Radiative Properties and Mass Content of Phosphoric Acid, HC, Petroleum Oil, and Sulfuric Acid Military Smokes," ASL-TR-0052, April 1980.

124. Hinds, B. D., and J. B. Gillespie, "Optical Characterization of Atmospheric Particulates on San Nicolas Island, California," ASL-TR-0053, April 1980.

125. Miers, Bruce T., "Precipitation Estimation for Military Hydrology," ASL-TR-0054, April 1980.

126. Stenmark, Ernest B., "Objective Quality Control of Artillery Computer Meteorological Messages," ASL-TR-0055, April 1980.

127. Duncan, Louis D., and Richard D. H. Low, "Bimodal Size Distribution Models for Fogs at Meppen, Germany," ASL-TR-0056, April 1980.

128. Olsen, Robert O., and Jagir S. Randhawa, "The Influence of Atmospheric Dynamics on Ozone and Temperature Structure," ASL-TR-0057, May 1980.

129. Kennedy, Bruce W., et al, "Dusty Infrared Test-II (DIRT-II) Program," ASL-TR-0058, May 1980.

130. Heaps, Melvin G., Robert O. Olsen, Warren Berning, John Cross, and Arthur Gilcrease, "1979 Solar Eclipse, Part I - Atmospheric Sciences Laboratory Field Program Summary," ASL-TR-0059, May 1980

131. Miller, Walter B., "User's Guide for Passive Target Acquisition Program Two (PTAP-2)," ASL-TR-0060, June 1980.

132. Holt, E. H., editor, "Atmospheric Data Requirements for Battlefield Obscuration Applications," ASL-TR-0061, June 1980.

133. Shirkey, Richard C., August Miller, George H. Goedecke, and Yugal Behl, "Single Scattering Code AGAUSX: Theory, Applications, Comparisons, and Listing," ASL-TR-0062, July 1980.

134. Sojka, Brian Z., and Kenneth O. White, "Evaluation of Specialized Photoacoustic Absorption Chambers for Near-Millimeter Wave (NMMW) Propagation Measurements," ASL-TR-0063, August 1980.

135. Bruce, Charles W., Young Paul Yee, and S. G. Jennings, "In Situ Measurement of the Ratio of Aerosol Absorption to Extinction Coefficient," ASL-TR-0064, August 1980.

136. Yee, Young Paul, Charles W. Bruce, and Ralph J. Brewer, "Gaseous/Particulate Absorption Studies at WSMR using Laser Sourced Spectrophones," ASL-TR-0065, June 1980.

137. Lindberg, James D., Radon B. Loveland, Melvin Heaps, James B. Gillespie, and Andrew F. Lewis, "Battlefield Dust and Atmospheric Characterization Measurements During West German Summertime Conditions in Support of Grafenwohr Tests," ASL-TR-0066, September 1980.

138. Vechione, W. J., "Evaluation of the Environmental Instruments, Incorporated Series 200 Dual Component Wind Set," ASL-TR-0067, September 1980.

139. Bruce, C. W., Y. P. Yee, B. D. Hinds, R. G. Pinnick, R. J. Brewer, and J. Minjares, "Initial Field Measurements of Atmospheric Absorption at 9μm to 11μm Wavelengths," ASL-TR-0068, October 1980.

140. Heaps, M. G., R. O. Olsen, K. D. Baker, D. A. Burt, L. C. Howlett, L. L. Jensen, E. F. Pound, and G. D. Allred, "1979 Solar Eclipse: Part II Initial Results for Ionization Sources, Electron Density, and Minor Neutral Constituents," ASL-TR-0069, October 1980.

141. Low, Richard D. H., "One-Dimensional Cloud Microphysical Models for Central Europe and their Optical Properties," ASL-TR-0070, October 1980.

142. Duncan, Louis D., James D. Lindberg, and Radon B. Loveland, "An Empirical Model of the Vertical Structure of German Fogs," ASL-TR-0071, November 1980.

143. Duncan, Louis D., "EOSAEL 80, Volume I, Technical Documentation," ASL-TR-0072, January 1981.

144. Shirkey, R. C., and S. G. O'Brien, "EOSAEL 80, Volume II, Users Manual," ASL-TR-0073, January 1981.

145. Bruce, C. W., "Characterization of Aerosol Nonlinear Effects on a High-Power CO_2 Laser Beam," ASL-TR-0074, February 1981.

146. Duncan, Louis D., and James D. Lindberg, "Air Mass Considerations in Fog Optical Modeling," ASL-TR-0075, February 1981.

147. Kunkel, Kenneth E., "Evaluation of a Tethered Kite Anemometer," ASL-TR-0076, February 1981.

148. Kunkel, K. E., et al, "Characterization of Atmospheric Conditions at the High Energy Laser System Test Facility (HELSTF) White Sands Missile Range, New Mexico, August 1977 to October 1978, Part II, Optical Turbulence, Wind, Water Vapor Pressure, Temperature," ASL-TR-0077, February 1981.

149. Miers, Bruce T., "Weather Scenarios for Central Germany," ASL-TR-0078, February 1981.

150. Cogan, James L., "Sensitivity Analysis of a Mesoscale Moisture Model," ASL-TR-0079, March 1981.

151. Brewer, R. J., C. W. Bruce, and J. L. Mater, "Optoacoustic Spectroscopy of C_2H_4 at the 9μm and 10μm $C^{12}O_2^{16}$ Laser Wavelengths," ASL-TR-0080, March 1981.

152. Swingle, Donald M., "Reducible Errors in the Artillery Sound Ranging Solution, Part I: The Curvature Correction" (U), SECRET, ASL-TR-0081, April 1981.

153. Miller, Walter B., "The Existence and Implications of a Fundamental System of Linear Equations in Sound Ranging" (U), SECRET, ASL-TR-0082, April 1981.

154. Bruce, Dorothy, Charles W. Bruce, and Young Paul Yee, "Experimentally Determined Relationship Between Extinction and Liquid Water Content," ASL-TR-0083, April 1981.

155. Seagraves, Mary Ann, "Visible and Infrared Obscuration Effects of Ice Fog," ASL-TR-0084, May 1981.

156. Watkins, Wendell R., and Kenneth O. White, "Wedge Absorption Remote Sensor," ASL-TR-0085, May 1981.

157. Watkins, Wendell R., Kenneth O. White, and Laura J. Crow, "Turbulence Effects on Open Air Multipaths," ASL-TR-0086, May 1981.

158. Blanco, Abel J., "Extending Application of the Artillery Computer Meteorological Message," ASL-TR-0087, May 1981.

159. Heaps, M. G., D. W. Hoock, R. O. Olsen, B. F. Engebos, and R. Rubio, "High Frequency Position Location: An Assessment of Limitations and Potential Improvements," ASL-TR-0088, May 1981.

160. Watkins, Wendell R., and Kenneth O. White, "Laboratory Facility for Measurement of Hot Gaseous Plume Radiative Transfer," ASL-TR-0089, June 1981.

161. Heaps, M. G., "Dust Cloud Models: Sensitivity of Calculated Transmittances to Variations in Input Parameters," ASL-TR-0090, June 1981.

162. Seagraves, Mary Ann, "Some Optical Properties of Blowing Snow," ASL-TR-0091, June 1981.

163. Kobayashi, Herbert K., "Effect of Hail, Snow, and Melting Hydrometeors on Millimeter Radio Waves," ASL-TR-0092, July 1981.

164. Cogan, James L., "Techniques for the Computation of Wind, Ceiling, and Extinction Coefficient Using Currently Acquired RPV Data," ASL-TR-0093, July 1981.

165. Miller, Walter B., and Bernard F. Engebos, "On the Possibility of Improved Estimates for Effective Wind and Temperature," (U), SECRET, ASL-TR-0094, August 1981.

166. Heaps, Melvin G., "The Effect of Ionospheric Variability on the Accuracy of High Frequency Position Location," ASL-TR-0095, August 1981.

167. Sutherland, Robert A., Donald W. Hoock, and Richard B. Gomez, "An Objective Summary of US Army Electro-Optical Modeling and Field Testing in an Obscuring Environment," ASL-TR-0096, October 1981.

168. Pinnick, R. G., et al, "Backscatter and Extinction in Water Clouds," ASL-TR-0097, October 1981.

169. Cole, Henry P., and Melvin G. Heaps, "Properties of Dust as an Electron and Ion Attachment Site for Use in D Region Ion Chemistry," ASL-TR-0098, October 1981.

170. Spellicy, Robert L., Laura J. Crow, and Kenneth O. White, "Water Vapor Absorption Coefficients at HF Laser Wavelengths Part II: Development of the Measurement System and Measurements at Simulated Altitudes to 10 KM," ASL-TR-0099, November 1981.

171. Cohn, Stephen L., "Transport and Diffusion Solutions for Obscuration Using the XM-825 Smoke Munition," ASL-TR-0100, November 1981.

172. Pinnick, R. G., D. M. Garvey, and L. D. Duncan, "Calibration of Knollenberg FSSP Light-Scattering Counters for Measurement of Cloud Droplets," ASL-TR-0101, December 1981.

173. Cohn, Stephen L. and Ricardo Pena, "Munition Expenditure Model Verification: KWIK Phase I," ASL-TR-0102, December 1981.

174. Blackman, George R., "Cloud Geometry Analysis of the Smoke Week III Obscuration Trials," ASL-TR-0103, January 1982.

175. Sutherland, R. A., and D. W. Hoock, "An Improved Smoke Obscuration Model ACT II: Part 1 Theory," ASL-TR-0104, January 1982.

176. Lentz, W. J., "The Visioceilometer: A Portable Visibility and Cloud Ceiling Height Lidar," ASL-TR-0105, January 1982

177. Kennedy, Bruce W., "An Analysis of Craters Produced by Artillery Munitions During Dusty Infrared Test Series," ASL-TR-0106, January 1982.

178. Steinhoff, R. G., "Program Listings for EOSAEL 80-B and Ancillary Codes AGAUS and FLASH," ASL-TR-0107, February 1982.


DATE
ILME