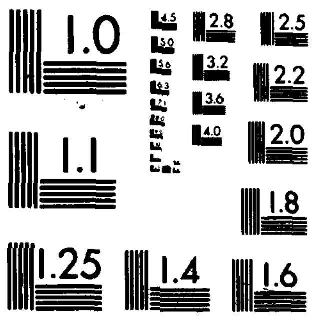
AD-A081 795 AERODYNE RESEARCH INC BEDFORD MA F/G 4/1
DIRTRAN-I USER'S MANUAL.(U)
JAN 80 D DVORE DAAK20-79-Q-0043
UNCLASSIFIED ARI-RR-178.1 ERADCOM/ASL-CR-80-0043-1 NL

END
DATE
FILMED
4-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ASL-CR-80-0043-1

AD

Reports Control Symbol
OSD 1366

DIRTRAN-I USER'S MANUAL

JANUARY 1980

Prepared by

DAVID DVORE

Aerodyne Research, Inc.
Crosby Drive
Bedford, MA 01730

Under Contract DAAK20-79-Q-0043
CONTRACT MONITOR: LOUIS DUNCAN

Approved for public release; distribution unlimited

US Army Electronics Research and Development Command
ATMOSPHERIC SCIENCES LABORATORY
White Sands Missile Range, NM 88002

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

Destory 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-CR-80-0043-1	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): DIRTRAN-I User's Manual	5. TYPE OF REPORT & PERIOD COVERED: Technical Report, June-September 1979
	6. PERFORMING ORG. REPORT NUMBER: ARI-RR-178
7. AUTHOR(s): David Dvore	8. CONTRACT OR GRANT NUMBER(s): DAAK20-79-Q-0043
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Aerodyne Research, Inc., Crosby Drive, Bedford, MA 01730	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102A- 1L161102B53A13
11. CONTROLLING OFFICE NAME AND ADDRESS: Mr. George Haber, Electronics Warfare Laboratory, Ft. Monmouth, NJ 07703	12. REPORT DATE: January 1980
	13. NUMBER OF PAGES: 82
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Dr. Louis Duncan, Atmospheric Sciences Laboratory, White Sands Missile Range, NM	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

Contract Monitor: Louis Duncan

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

Dust	Obscuration	MM-Wave	Gravitational Settling
Aerosol	Transmission	Atmospheric Diffusion	DIRT I Test
Explosions	Visible	Wind Shear	GRAF II Test
Artillery	Infrared	Buoyancy	

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

The DIRTRAN-I Code is a computer-implemented model for predicting the optical effects of an explosion-produced dust cloud as it disperses in the lower atmosphere. This model is based on first principles of fluid dynamics, atmospherics, and optics. The model has been validated using cloud dimension and line-of-sight optical transmission data from the DIRT I and Graf II-Winter Army dust obscuration field trials. The DIRTRAN-I Code exploits information available about crater sizes produced by explosions in conjunction

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

S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) with distinct models for coupling of energy to the ground for artillery projectiles versus bare charges. The model recognizes that dust ejecta are partitioned into a buoyantly rising fireball and a non-buoyant "dust skirt" which is subject to diffusion in the vertically sheared wind field. The DIRTRAN-I Code solves separately for these two clouds. The solutions are based on atmospheric diffusion theory and take into account the effects of wind and temperature profiles in the constant shear stress layer of the lower atmosphere for different atmospheric stability categories. Separate treatment is given to particles of different sizes, the larger ones being allowed to settle out. Outputs of the code include dust cloud displacement and dimensions for both the non-buoyant wind-dominated skirt and the initial buoyant fireball as it is wind blown and eventually also becomes subject to wind diffusion. Line-of-sight transmittances at several wavelength bands (visible: 0.4 - 0.7 μm; infrared: 0.8 - 1.1, 3.5 - 4.0, and 8.5 - 12 μm; mm wave: 94 - 140 GHz) are also output options. Because of the analytic solutions derived from first-principle physics, the DIRTRAN-I Code has been designed to keep both storage and computation time to a minimum. In order to be machine transportable, the code has been written in ANSI FORTRAN IV with no system specific enhancements.

TABLE OF CONTENTS

Section	Description	Page
1.	INTRODUCTION	5
2.	FUNCTIONS AND SUBROUTINES	6
3.	USAGE	11
	3.1 Common Blocks Used to Pass Information to DIRTRAN-I	11
	3.1.1 COMMON /IOUNIT/ IOIN, IOOUT, ISPTPF, LOUNIT, DIRTU, NBSCAT	11
	3.1.2 COMMON /CLYMAT/ TEMP, PRESS, RH, AH, DP, VIS, CLDAMT, CLDHYT, FOGPRB, IPASCT, WNDVEL, WNDDIR	12
	3.2 Input from FORTRAN Logical Unit IOIN	12
	3.3 Example	14
4.	DIRTRAN-I GEOMETRY	20
	4.1 User Coordinates	20
	4.2 Wind Geometry	20
	4.3 Transmission Geometry	20
	4.4 Observer Geometry	20
5.	LISTING OF DIRTRAN-I CODE	23

1. INTRODUCTION

The DIRTRAN-I Code is a computer-implemented model for predicting the optical effects of an explosion-produced dust cloud as it disperses in the lower atmosphere. This model is based on first principles of fluid dynamics, atmospherics, and optics. The model has been validated using cloud dimension and line-of-sight optical transmission data from the DIRT I and Graf II-Winter Army dust obscuration field trials. The DIRTRAN-I Code exploits information available about crater sizes produced by explosions in conjunction with distinct models for coupling of energy to the ground for artillery projectiles versus bare charges. The model recognizes that dust ejecta are partitioned into a buoyantly rising fireball and a non-buoyant "dust skirt" which is subject to diffusion in the vertically sheared wind field. The DIRTRAN-I Code solves separately for these two clouds. The solutions are based on atmospheric diffusion theory and take into account the effects of wind and temperature profiles in the constant shear stress layer of the lower atmosphere for different atmospheric stability categories. Separate treatment is given to particles of different sizes, the larger ones being allowed to settle out. Outputs of the code include dust cloud displacement and dimensions for both the non-buoyant wind-dominated skirt and the initial buoyant fireball as it is wind blown and eventually also becomes subject to wind diffusion. Line-of-sight transmittances at several wavelength bands (visible: 0.4 - 0.7 μm; infrared: 0.8 - 1.1, 3.5 - 4.0, and 8.5 - 12 μm; mm wave: 94 - 140 GHz) are also output options.

Because of the analytic solutions derived from first-principle physics, the DIRTRAN-I Code has been designed to keep both storage and computation time to a minimum. In order to be machine transportable, the code has been written in ANSI FORTRAN IV with no system specific enhancements.

2. FUNCTIONS AND SUBROUTINES

There is a great deal of internal documentation in the code listed in Section 5. In Figs. 2.1 through 2.5 are box diagrams illustrating the structure of the code.

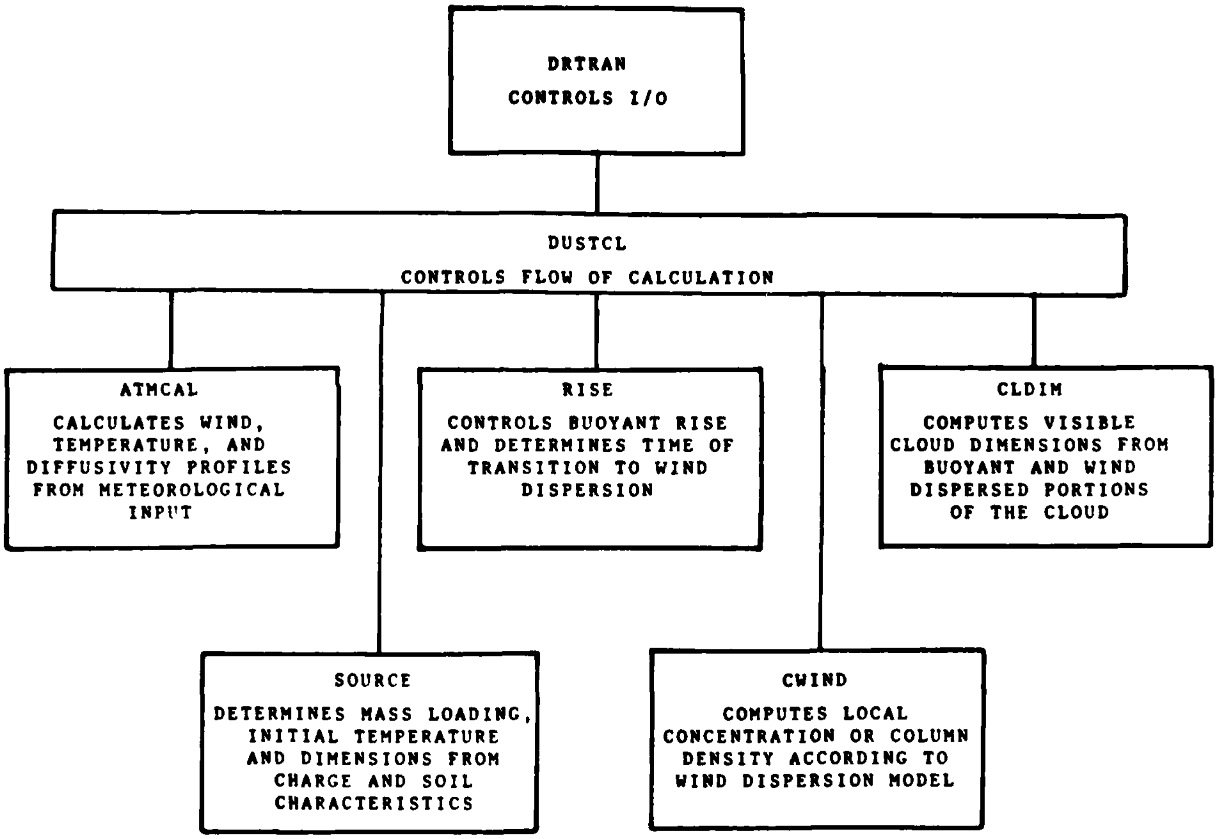


Figure 2.1 Controlling Routines.

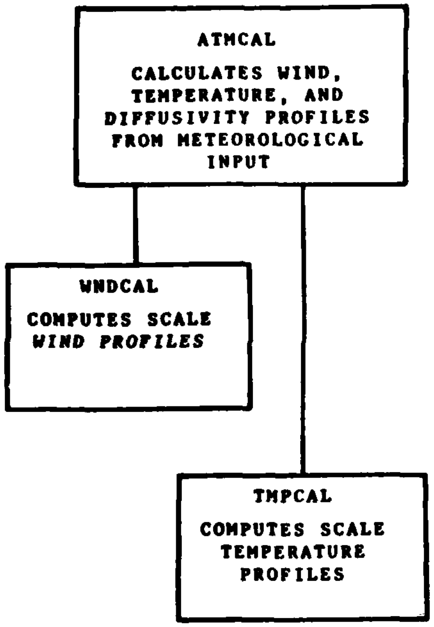


Figure 2.2 Routines for Interpreting Meteorological Data.

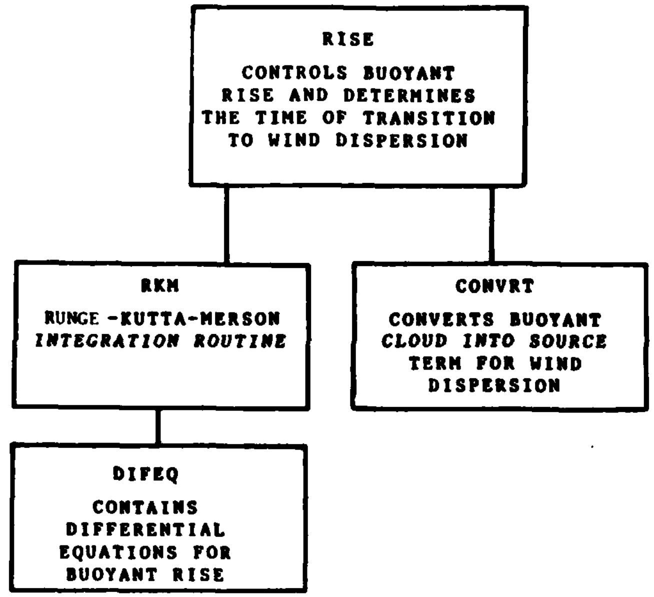


Figure 2.3 Routines for Buoyant Rise Model

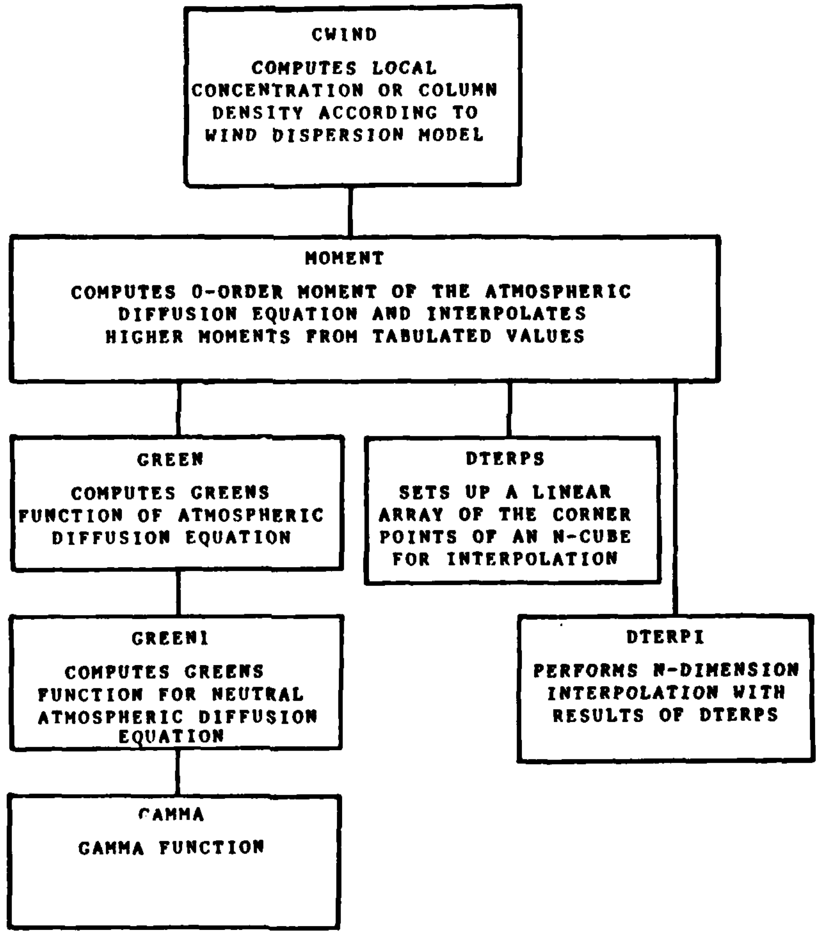


Figure 2.4 Routines for Wind Dispersion Model.

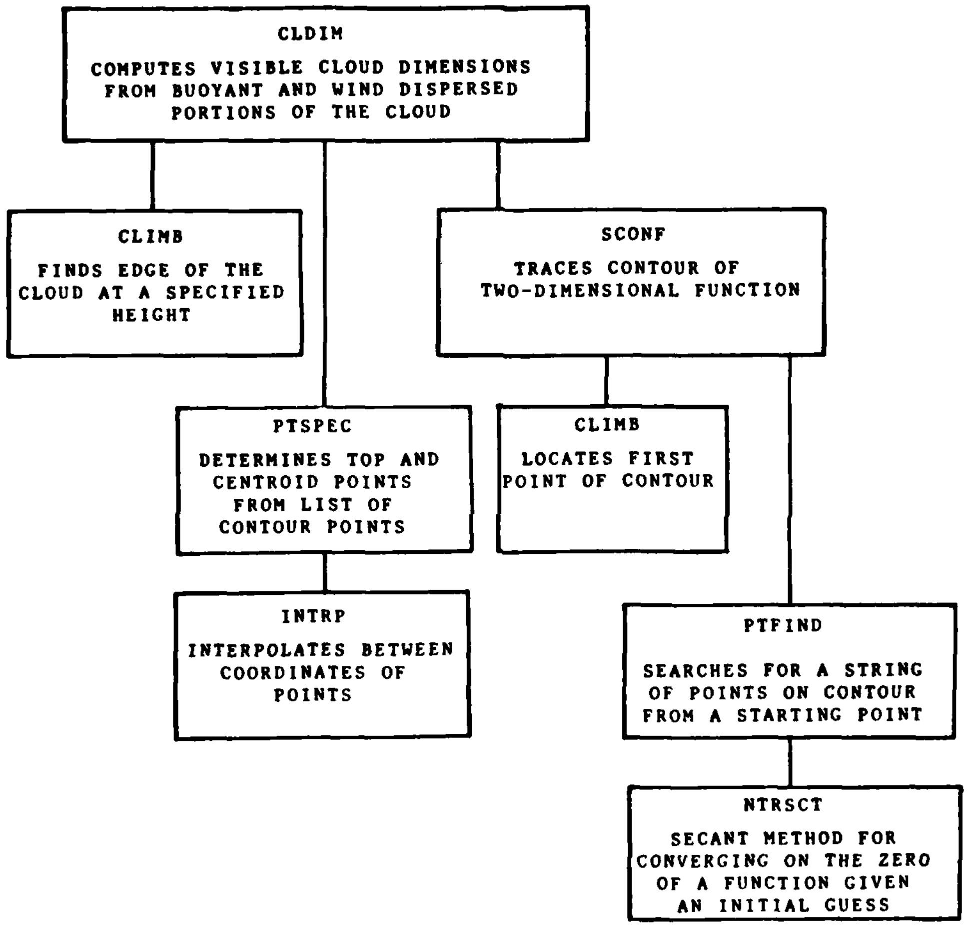


Figure 2.5 Routines for Determining Cloud Dimensions.

Miscellaneous Routines

FUNCT Two-dimensional function which provides optically weighted column densities for visible contour tracing.

GRAD2 Computes the two-dimensional gradient vector of a function.

GFUN The restriction of a function of two variables to a line specified by a base point and direction vector.

PERP Computes a vector rotated 90° counterclockwise from a given vector.

UNIT Computes the length of a vector and a unit vector in the same direction as a given vector.

VSUM Adds a given scaler multiple of one given vector to another.

3. USAGE

The user accesses the DIRTRAN-I Code via Subroutine DRTRAN.

CALL DRTRAN (WAVE1, ICLMAT, TRNLOS, IERR)

where:

WAVE1 is the wavelength in micrometers. Valid ranges are:

0.4 -	0.7 micrometers
0.8 -	1.1 micrometers
3.5 -	4.0 micrometers
8.5 -	12.0 micrometers
2100.0 -	3200.0 micrometers

ICLMAT is an integer index which is 0 if meteorological data is read from an external unit with other inputs and is 1 if passed in COMMON /CLYMAT/.

TRNLOS is the returned value of the transmittance along the line-of-sight between the transmitter and the reciever.

IERR is an integer error code which is returned one if a fatal error occurs and zero otherwise.

3.1 Common Blocks Used to Pass Information to DIRTRAN-I

There are two common blocks which communicate information to DIRTRAN-I: /IOUNIT/ and /CLYMAT/.

3.1.1 COMMON /IOUNIT/ IOIN, IOOUT, ISPTPF, LOUNIT, NDIRTU, NBSCAT

Of the variables in /IOUNIT/ only IOIN, IOOUT, and NDIRTU are used by DIRTRAN-I.

IOIN The Fortran logical unit from which DIRTRAN-I reads input data to be described in Subsection 3.2.

IOOUT The Fortran logical unit onto which DIRTRAN-I writes output and error messages.

NDIRTU The Fortran logical unit from which DIRTRAN-I reads the data file containing tabulated values for the moments of the atmospheric diffusion equation.

3.1.2 COMMON /CLYMAT/ TEMP, PRESS, RH, AH, DP, VIS, CLDAMT, CLDHYT, FOGPRB, IPASCT, WNDVEL, WNDDIR

DIRTRAN-I uses only TEMP, IPASCT, WNDVEL, and WNDDIR.

TEMP Temperature in degrees C taken at approximately two meters above ground.

IPASCT Integer 1 - 6 corresponding to Pasquill Categories A - F.

WNDVEL The wind speed in meters per second measured at approximately two meters above ground.

WNDDIR The wind direction (in degrees clockwise from true north) from which the wind is blowing. With this option the user's coordinate system must have the positive x-axis pointing east and positive y-axis pointing north. (Thus 0° corresponds to a wind blowing from the north to the south; 90° is a wind blowing from the east to the west; etc.)

3.2 Input from FORTRAN Logical Unit IOIN

The input read from logical unit, IOIN, provides the user with a variety of ways to use DIRTRAN-I. There are seven types of records to be distinguished on the input file. They are listed here with the names of variables contained on each record and the format type.

RECORD

RECORD												
1	NAME	NEWATM	NEWSRC	LOSTRN	EDGE	NEWTIM						
	FORMAT	L1	L1	L1	L1	L1						
2	NAME	NATMOS	ZTMP	TMPMES	ZWND	WNDMES	ZTMP	TEMPMES	ZWND	WNDMES	ZINV	THWND
	FORMAT	I1,1X	F7.2	F7.2	F7.2	F7.2	F7.2	F7.2	F7.2	F7.2	F7.2	F7.2
3	NAME	NSOIL	NCHRG	CHWT	DETDEP	DSOD	SRCCOR	SRCCOR				
	FORMAT	I1,1X	I1,1X	F7.2	F7.2	F7.2	F7.2	F7.2				
4	NAME	TRNCOR	TRNCOR	TRNCOR	RECCOR	RECCOR	RECCOR					
	FORMAT	F7.2	F7.2	F7.2	F7.2	F7.2	F7.2					
5	NAME	OBSCOR	OBSCOR	SPCHT								
	FORMAT	F7.2	F7.2	F7.2								
6	NAME	TIME										
	FORMAT	F7.2										
7	NAME	CNTNU										
	FORMAT	L1										

The description of the variables is in the comments in subroutine DRTRAN and appears on Pages 5-1 - 5-6 in this manual. Additional descriptions for some of the variables appears in Section 4.

The input file contains one or several sequences of these records each of which must begin with record 1, end with record 7, and contain a subset of records 2 - 6 corresponding to the entries in record 1. In each sequence, each of records 2 through 6 must appear if and only if the corresponding control variable entered in record 1 is .TRUE..

If NEWATM is .TRUE. then record 2 must appear.
If NEWSRC is .TRUE. then record 3 must appear.
If LOSTRN is .TRUE. then record 4 must appear.
If EDGE is .TRUE. then record 5 must appear.
If NEWTIM is .TRUE. then record 6 must appear.

On the first record of the input file NEWATM, NEWSRC, and NEWTIM must all be .TRUE.. This initializes the meteorological conditions, the charge and soil characteristics, and time after blast for the first observation. After that, DIRTRAN-I assumes that:

- Meteorological data are unchanged until NEWATM is .TRUE. again.
- Charge and Soil Characteristics are unchanged until NEWSRC is .TRUE. again.
- Time after blast is unchanged until NEWTIM is .TRUE. again.

The first time that NEWATM is .TRUE., if ICLMAT = 1 (see arguments of DRTRAN in Section 2), then record 2 should not appear following record 1. This is the only time that a control variable in record 1 may be .TRUE. without the corresponding record 2 - 6 following.

Record 7 contains the control variable, CNTNU, which is .TRUE. if another sequence of records 1 - 7 is to be read on this call of DRTRAN and .FALSE. is DRTRAN is to return control to the calling routine. An example follows.

3.3 Example

```
TTTTT
3        1.  271.30       1.   2.30
1 1       15.       0.      .15   000.       0.                              0.    090.
 175.00-1148.0     1.8 -173.0 1135.0      1.8
  175.0-1148.0     1.8
     1.
T
FFTTT
 175.00-1148.9     1.8 -173.0 1135.0      1.8
  175.0-1148.0     1.8
     2.
T
FFTTT
 175.00-1148.9     1.8 -173.0 1135.0      1.8
  175.0-1148.0     1.8
     5.
T
FFTTT
 175.00-1148.9     1.8 -173.0 1135.0      1.8
  175.0-1148.0     1.8
    10.
T
FFTTT
 175.00-1148.9     1.8 -173.0 1135.0      1.8
  175.0-1148.0     1.8
    20.
T
FFTTT
 175.00-1148.9     1.8 -173.0 1135.0      1.8
  175.0-1148.0     1.8
    30.
T
FFTTT
 175.00-1148.9     1.8 -173.0 1135.0      1.8
  175.0-1148.0     1.8
    40.
T
FFTTT
 175.00-1148.9     1.8 -173.0 1135.0      1.8
  175.0-1148.0     1.8
    60.
F
```

The results from the above input file follow.

```
     DIRTRAN-I DUST CLOUD INFRARED TRANSMISSION CALCULATION

ALL UNITS ARE MKS UNLESS OTHERWISE SPECIFIED.


PASQUILL CATEGORY                    3
    HT     1.00  TEMP   271.30      HT    1.00  WIND      2.30
WIND DIRECTION             90.00

SOIL INDEX       1
CHARGE INDEX     1
WEIGHT OF CHARGE                   15.00
DETONATION DEPTH                    0.00
DEPTH OF SOD                        0.15
SOURCE COORDINATES                      0.00        0.00



TIME AFTER BLAST                    1.00

WAVELENGTH                          0.55 MICROMETERS
TRANSMITTER COORDINATES              175.00  -1148.00        1.80
RECEIVER    COORDINATES             -173.00   1135.00        1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT   0.149E-02

OBSERVER    COORDINATES              175.00  -1148.00
THE HEIGHT OF THE CLOUD IS                     8.39 METERS
THE CENTROID COORDINATES ARE                   0.41        4.79
THE WIDTH AT THE CENTROID IS                   7.19 METERS
THE WIDTH AT      1.80 METERS IS              14.06 METERS
  5 CONTOUR POINTS HAVE BEEN DETERMINED
     -6.654        1.800
     -3.184        4.794
      0.410        8.388
      4.003        4.794
      7.408        1.800


TIME AFTER BLAST                    2.00

WAVELENGTH                          0.55 MICROMETERS
TRANSMITTER COORDINATES              175.00  -1148.90        1.80
RECEIVER    COORDINATES             -173.00   1135.00        1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT   0.296E-02

OBSERVER    COORDINATES              175.00  -1148.00
THE HEIGHT OF THE CLOUD IS                    12.83 METERS
THE CENTROID COORDINATES ARE                   0.83        8.34
```

```
THE WIDTH AT THE CENTROID IS                    8.97 METERS
THE WIDTH AT       1.80 METERS IS              14.69 METERS
  5 CONTOUR POINTS HAVE BEEN DETERMINED
    -6.590       1.800
    -3.650       8.345
     0.832      12.827
     5.315       8.345
     8.098       1.800


TIME AFTER BLAST                    5.00

WAVELENGTH                          0.55 MICROMETERS
TRANSMITTER COORDINATES              175.00  -1148.90        1.80
RECEIVER    COORDINATES             -173.00   1135.00        1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT   0.156E-01

OBSERVER    COORDINATES              175.00  -1148.00
THE HEIGHT OF THE CLOUD IS                     23.46 METERS
THE CENTROID COORDINATES ARE                    2.21      16.85
THE WIDTH AT THE CENTROID IS                   13.23 METERS
THE WIDTH AT       1.80 METERS IS              16.56 METERS
  5 CONTOUR POINTS HAVE BEEN DETERMINED
    -6.396       1.800
    -4.409      16.845
     2.208      23.462
     8.825      16.845
    10.166       1.800


TIME AFTER BLAST                   10.00

WAVELENGTH                          0.55 MICROMETERS
TRANSMITTER COORDINATES              175.00  -1148.90        1.80
RECEIVER    COORDINATES             -173.00   1135.00        1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT   0.107E 00

OBSERVER    COORDINATES              175.00  -1148.00
THE HEIGHT OF THE CLOUD IS                     36.10 METERS
THE CENTROID COORDINATES ARE                    4.71      26.94
THE WIDTH AT THE CENTROID IS                   18.32 METERS
THE WIDTH AT       1.80 METERS IS              19.37 METERS
  5 CONTOUR POINTS HAVE BEEN DETERMINED
    -5.762       1.800
    -4.453      26.943
     4.709      36.105
    13.871      26.943
    13.613       1.800
```

```
TIME AFTER BLAST                        20.00

WAVELENGTH                               0.55 MICROMETERS
TRANSMITTER COORDINATES                   175.00  -1148.90        1.80
RECEIVER    COORDINATES                  -173.00   1135.00        1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT   0.508E 00

OBSERVER    COORDINATES                   175.00  -1148.00
THE HEIGHT OF THE CLOUD IS                         59.23 METERS
THE CENTROID COORDINATES ARE                        9.98      40.06
THE WIDTH AT THE CENTROID IS                       42.01 METERS
THE WIDTH AT      1.80 METERS IS                   24.06 METERS
  5 CONTOUR POINTS HAVE BEEN DETERMINED
    -4.179        1.800
   -11.030       40.057
    10.032       59.232
    30.983       40.057
    19.883        1.800


TIME AFTER BLAST                        30.00

WAVELENGTH                               0.55 MICROMETERS
TRANSMITTER COORDINATES                   175.00  -1148.90        1.80
RECEIVER    COORDINATES                  -173.00   1135.00        1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT   0.789E 00

OBSERVER    COORDINATES                   175.00  -1148.00
THE HEIGHT OF THE CLOUD IS                         67.78 METERS
THE CENTROID COORDINATES ARE                       15.89      40.18
THE WIDTH AT THE CENTROID IS                       69.05 METERS
THE WIDTH AT      1.80 METERS IS                   27.81 METERS
  5 CONTOUR POINTS HAVE BEEN DETERMINED
    -2.285        1.800
   -18.638       40.178
    17.039       67.779
    50.413       40.178
    25.528        1.800


TIME AFTER BLAST                        40.00

WAVELENGTH                               0.55 MICROMETERS
TRANSMITTER COORDINATES                   175.00  -1148.90        1.80
RECEIVER    COORDINATES                  -173.00   1135.00        1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT   0.909E 00
```

```
OBSERVER    COORDINATES              175.00  -1148.00
THE HEIGHT OF THE CLOUD IS                 72.35 METERS
THE CENTROID COORDINATES ARE               21.67      40.70
THE WIDTH AT THE CENTROID IS               83.14 METERS
THE WIDTH AT      1.80 METERS IS           30.63 METERS
  5 CONTOUR POINTS HAVE BEEN DETERMINED
     0.235       1.800
   -19.906      40.703
    20.352      72.355
    63.238      40.703
    30.860       1.800


TIME AFTER BLAST                   60.00

WAVELENGTH                          0.55 MICROMETERS
TRANSMITTER COORDINATES             175.00  -1148.90       1.80
RECEIVER    COORDINATES            -173.00   1135.00       1.80
TRANSMITTANCE ALONG THE LINE-OF-SIGHT   0.977E 00

OBSERVER    COORDINATES              175.00  -1148.00
THE HEIGHT OF THE CLOUD IS                 77.53 METERS
THE CENTROID COORDINATES ARE               33.99      48.28
THE WIDTH AT THE CENTROID IS               99.13 METERS
THE WIDTH AT      1.80 METERS IS           36.56 METERS
  5 CONTOUR POINTS HAVE BEEN DETERMINED
     4.962       1.800
   -15.571      48.284
    34.569      77.535
    83.556      48.284
    41.524       1.800
```

4. DIRTRAN-I GEOMETRY

4.1 User Coordinates

A top view of the (x,y) user coordinate plane for the DIRTRAN-I geometry is shown in Fig. 4.1. All inputs are in this user coordinate plane. Negative coordinates are allowed.

4.2 Wind Geometry

THWND is the angle in degrees that the wind velocity vector makes with the user's positive x-axis and is measured counter-clockwise. If the x-axis points east, then the following relationship holds between THWND and WINDIR (which is the standard meteorological wind direction as discussed in Sub-Section 3.1.2):

$$THWND = 270 - WINDIR$$

4.3 Transmission Geometry

The height above ground of the transmitter and receiver, TRNCOR(3) and RECCOR(3), must be equal and must be between 0 and 5 meters. The transmitter height will be used as a default for the receiver height.

4.4 Observer Geometry

A local coordinate system, x', y', z' is used for viewing geometry. The x'-axis is aligned from the detonation point with coordinates, SRCCOR(I), to the observer position with coordinates, OBSCOR(I). The y'-axis is rotated 90° counterclockwise from the x'-axis, and the z'-axis is vertical so as to make a right-handed coordinate system. The visible cloud is projected along the observer's line-of-sight, the x'-axis. The visible cloud contour thus appears on the y'-z' coordinate plane, shown in Fig. 4.2. Five contour points are reported by DIRTRAN-I:

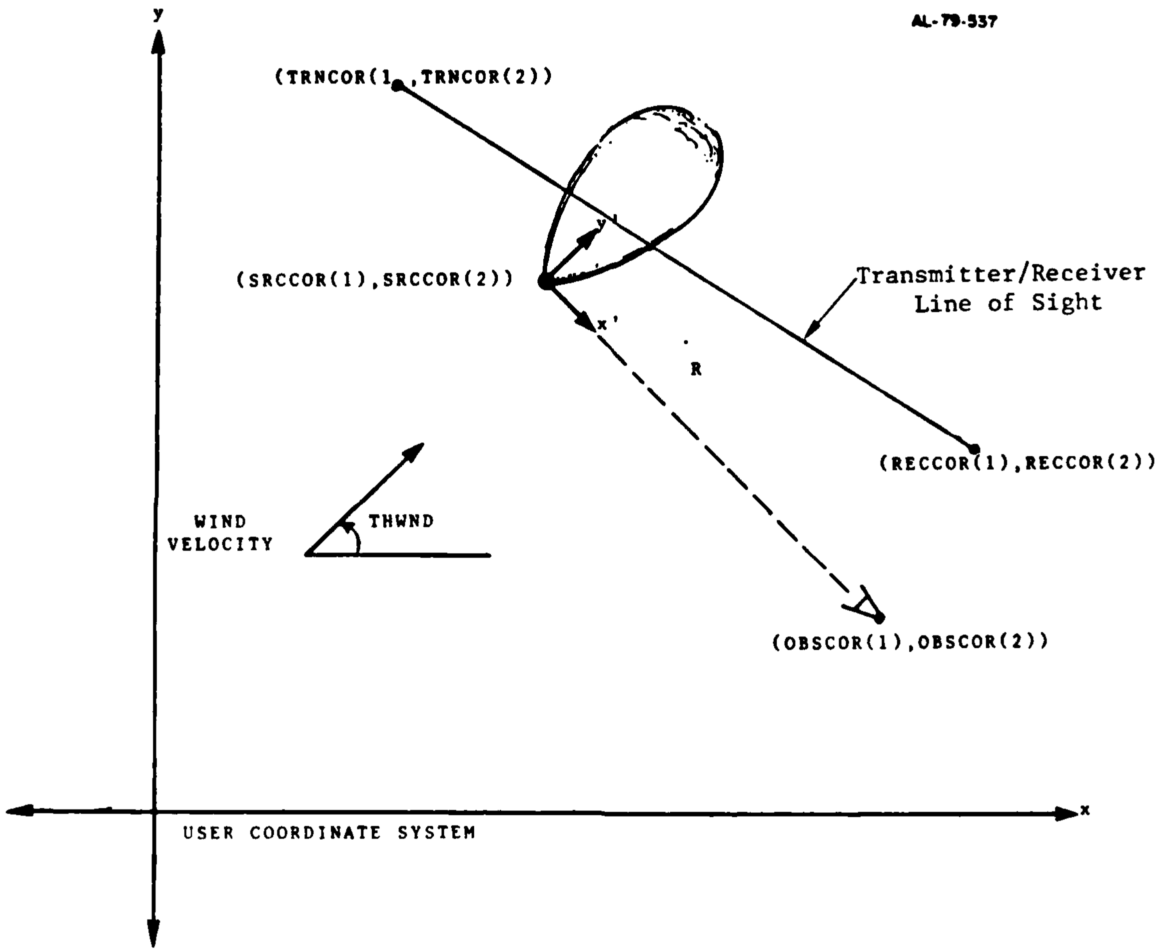


Figure 4.1 Top View of DIRTRAN-I Geometry.

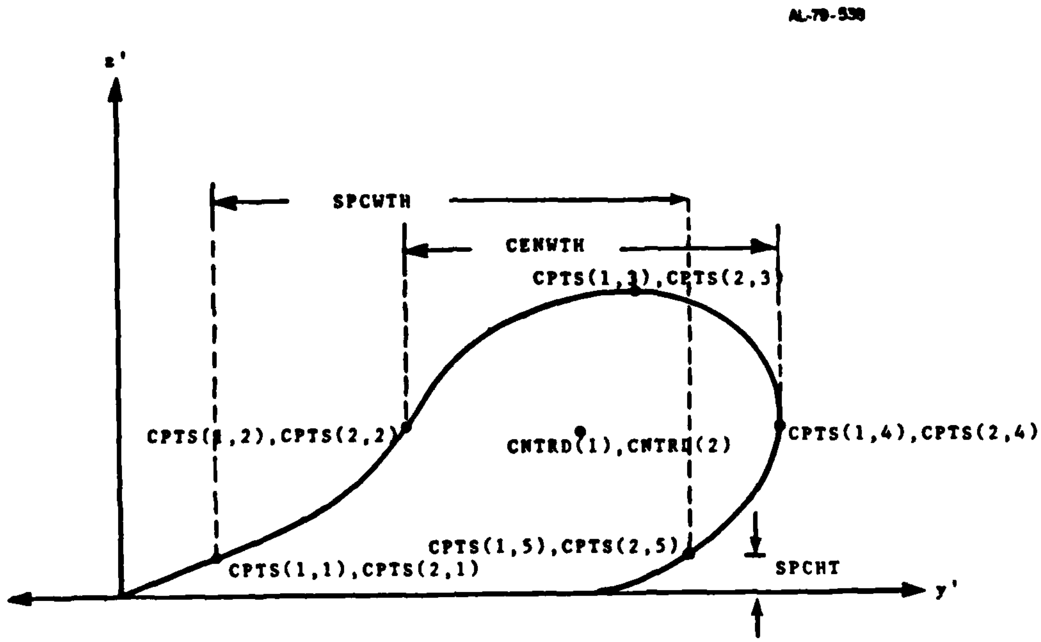


Figure 4.2 Observer's View.

- The left and right edges of the cloud at a specified height, SPCHT,
- The left and right edges of the cloud at the centroid height, and
- The top of the cloud.

The centroid is determined by first locating the leading edge of the cloud which is (CPTS(1,4), CPTS(2,4)) in Fig. 4.2. A horizontal line is drawn across the cloud to determine (CPTS(1,2), CPTS(2,2)) in Fig. 4.2. The centroid is defined to be the midpoint of these two.

5. LISTING OF DIRTRAN-I CODE

```
C           DIRTRAN-I CODE
C
C     TEST CALLING ROUTINE
C
      COMMON /IOUNIT/ IOIN,IOOUT,ISPTPF,LOUNIT,NDIRTU,NBSCAT
      DATA IOIN,IOOUT,NDIRTU/5,1,7/
      WAVE1=.55
      ICLMAT=0
      CALL DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)
      STOP
      END
C     UTILITY ROUTINE FOR INTERFACING DIRTRAN-I WITH EC-SAEL
C
C ******************  SUBFILE 1  ******************************
C
      SUBROUTINE DRTRAN(WAVE1,ICLMAT,TRNLOS,IERR)
      IMPLICIT INTEGER*4 (I-N)
      LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM,CNTNU,CLMRED
      DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2),SRCCOR(2),TRNCOR(3)
     1  ,RECCOR(3),CPTS(2,200),CNTRD(2),OBSCOR(2)
      COMMON /IOUNIT/ IOIN,IOOUT,ISPTPF,LOUNIT,NDIRTU,NBSCAT
      COMMON /CLYMAT/ TEMP,PRESS,RH,AH,DP,VIS,CLDAMT,CLDHYT,
     1 FOGPRB,IPASCT,WNDVEL,WINDIR
C ********************************************************************
C
C     DAVID DVORE
C     AERODYNE RESEARCH, INC.
C     (617)275-9400 X127
C
C     PURPOSE
C
C          DRTRAN CALCULATES DUST CLOUD DIMENSIONS AND TRANSMITTANCES
C     THROUGH DUST CLOUDS FOR GIVEN METEOROLOGICAL DATA, SOIL TYPE,
C     BOMB CHARACTERISTICS, AND WAVELENGTH.
C
C
C     INPUT
C
C     NEWATM      A LOGICAL VARIABLE WHICH IS .TRUE. IF NEW ATMOSPHERIC
C                 CONDITIONS ARE TO BE SPECIFIED AND .FALSE. IF PREVIOUS
C                 VALUES ARE TO BE ASSUMED. IF .TRUE. THEN NATMOS, ZTMP,
C                 TMPMES, ZWND, WNDMES, HTINV, AND THWND MUST BE
C                 SPECIFIED
```

```
C
C      NATMOS      INTEGER WITH VALUES 1 TO 6 CORRESPONDING TO PASQUILL
C                  CATEGORIES A TO F.  IF WIND AND TEMPERATURE
C                  MEASUREMENTS ARE AVAILABLE AT TWO HEIGHTS THEN NATMOS
C                  SHOULD BE SPECIFIED  0 .
C
C      ZTMP        A SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS,
C                  MEASURED IN METERS, AT WHICH TEMPERATURE MEASUREMENTS
C                  ARE AVAILABLE.  THE HEIGHTS SHOULD BE IN INCREASING
C                  ORDER. IF ONLY ONE IS AVAILABLE, THE PASQUILL CATEGORY
C                  MUST BE SPECIFIED IN NATMOS.  VALID RANGE: 0.5-100. M.
C
C      TMPMES      A SINGLY DIMENSIONED ARRAY CONTAINING ONE OR TWO
C                  TEMPERATURE MEASUREMENTS IN DEGREES KELVIN TAKEN AT
C                  HEIGHTS ZTMP.  VALID RANGE: 273.-315. K.
C
C      ZWND        A SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS,
C                  MEASURED IN METERS, AT WHICH WIND MEASUREMENTS ARE
C                  AVAILABLE.  THE HEIGHTS SHOULD BE IN INCREASING ORDER.
C                  IF ONLY ONE IS AVAILABLE, THE PASQUILL CATEGORY MUST
C                  BE SPECIFIED IN NATMOS.  VALID RANGE: 0.5-100.M.
C
C      WNDMES      A SINGLY DIMENSIONED ARRAY CONTAINING WIND SPEEDS
C                  MEASURED IN METERS/SEC, AT HEIGHTS ZWND.
C                  VALID RANGE: 0.1 - 20. M/S.
C
C      HTINV       THE INVERSION HEIGHT MEASURED IN METERS.  DEFAULTS
C                  TO 200 METERS IF A NUMBER LESS THAN 10.0 IS GIVEN.
C                  VALID RANGE: 10.0 - 200.0 M.
C
C      THWND       THE ANGLE THAT THE WIND VELOCITY VECTOR MAKES
C                  WITH THE POSITIVE X AXIS MEASURED IN DEGREES
C                  COUNTERCLOCKWISE.
C                  VALID RANGE:  -360.0 - 360.0 DEGREES.
C
C      NEWSRC      A LOGICAL VARIBLE WHICH IS .TRUE. IF A NEW SOURCE
C                  IS TO BE SPECIFIED AND .FALSE. IF FURTHER RESULTS
C                  ARE DESIRED FROM THE PRESENT SOURCE.  IF .TRUE.
C                  THEN CHWT, NCHRG, DETDEP, NSOIL, DSOD, NWL, AND SRCCOR
C                  MUST BE SPECIFIED
C
C      CHWT        THE WEIGHT OF THE CHARGE IN LBS-TNT.
C                  VALID RANGE:  1.0 - 150.0 LBS-TNT.
C
C      NCHRG       CHARGE TYPE INDEX WITH FOLLOWING VALUES
C                       1    SURFACE - LIVE FIRE OR 30 DEGREE TILTED
C                            STATIC, TIP ON GROUND
C                       2    BARE CHARGE ON SURFACE
```

```
C                     3    30 DEGREE TILTED TIP AT 0.3 METER DEPTH
C                     4    30 DEGREE TILTED TIP AT 0.6 METER DEPTH
C                     5    HORIZONTAL PROJECTILE ON SURFACE
C                  DEFAULT VALUE IS 1 IF NCHRG IS NOT BETWEEN 1 AND 5.
C
C
C     DETDEP      THE DEPTH OF DETONATION IN METERS.
C                 VALID RANGE:  0.0 - 2.0 M.
C
C     NSOIL       INTEGER INDEX OF SOIL TYPE.  NSOIL IS
C                       1  FOR EUROPEAN SOIL
C                       2  FOR DESERT SOIL
C
C     DSOD        DEPTH OF SOD IN METERS
C                 VALID RANGE: 0.0 - 1.0 M.
C
C     WAVE1       WAVELENGTH IN MICROMETERS.  USED TO DETERMINE NWL.
C                 VALID RANGES:
C
C     NWL         INTEGER INDEX FOR WAVELENGTH
C                       1  FOR  0.4 - 0.7 MICROMETER  (VISIBLE)
C                       2  FOR  0.8 - 1.1 MICROMETER
C                       3  FOR  3.5 - 4.0 MICROMETER
C                       4  FOR  8.5 - 12.0 MICROMETER
C                       5  FOR  2100 - 3200 MICROMETER
C
C     SRCCOR      A SINGLY DIMENSIONED ARRAY CONTAINING THE X AND
C                 Y COORDINATES OF THE DETONATION POSITION.
C                 VALID RANGE: -9999.9 - 9999.99 M.
C
C     LOSTRN      A LOGICAL VARIABLE WHICH IS .TRUE. IF THE
C                 TRANSMITTANCE ALONG A LINE-OF-SIGHT IS DESIRED
C                 AND .FALSE. IF NOT.  IF .TRUE. THEN TRNCOR AND RECCOR
C                 MUST BE SPECIFIED.
C
C     TRNCOR      A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE
C                 COORDINATES OF THE TRANSMITTER.  THE COORDINATE SYSTEM
C                 MUST BE IN METERS.  THE THIRD COORDINATE IS RESTRICTED
C                 TO BE BETWEEN 1 AND 5 METERS (HEIGHT).
C                 VALID RANGE:  -9999.9  -  9999.99 M.
C
C     RECCOR      A SINGLY DIMENSIONED ARRAY CONTAINING THE THREE
C                 COORDINATES OF THE RECEIVER.  (METERS)
C                 THE THIRD COORDINATE MUST BE THE SAME AS THE THIRD
C                 COORDINATE OF TRNCOR.
C                 VALID RANGE: -9999.9  -  9999.99 M.
C
C     EDGE        A LOGICAL VARIABLE WHICH .TRUE. IF CLOUD DIMENSIONS
```

```
C                 ARE TO BE CALCULATED AND .FALSE. IF NOT.  IF .TRUE.
C                 THEN OBSCOR AND SPCHT MUST BE SPECIFIED.
C
C     OBSCOR      A SINGLY DIMENSIONED ARRAY CONTAINING THE X AND Y
C                 COORDINATES, RESP., OF THE OBSERVER. (METERS)
C                 VALID RANGE:    -9999.9  -  9999.99
C
C     SPCHT       A SPECIFIED HEIGHT IN METERS AT WHICH THE WIDTH OF
C                 THE CLOUD AS VIEWED FROM POSITION OBSCOR IS DESIRED.
C                 MUST BE BETWEEN 1 AND 5 METERS.
C
C     NEWTIM      A LOGICAL VARIABLE WHICH IS .TRUE. IF ONE WISHES TO
C                 ADVANCE THE TIME AND FALSE IF MORE RESULTS ARE
C                 REQUIRED AT THE CURRENT TIME.  IF .TRUE. THEN TIME
C                 MUST BE SPECIFIED.
C
C     TIME        TIME MEASURED IN SECONDS AFTER DETONATION.
C                 SUCCESSIVE CALLS OF DRTRAN MAY KEEP THE TIME UNCHANGED
C                 FROM PREVIOUS CALL (NEWTIM = .FALSE.) OR SPECIFY A
C                 NEW TIME GREATER THAN IN THE PREVIOUS CALL.  WHEN
C                 A NEW SOURCE IS SPECIFIED (NEWSRC=.TRUE.) A NEW TIME
C                 MUST ALSO BE SPECIFIED (NEWTIM=.TRUE.) AND ONLY IN
C                 THIS CASE MAY A TIME LESS THAN THE PREVIOUS CALL
C                 BE SPECIFIED.
C                 VALID RANGE: 0.5 - 1000.0 SECONDS.
C
C
C
C     OUTPUT
C
C     TRNLOS      THE TRANSMITTANCE ALONG THE LINE-OF-SIGHT BETWEEN
C                 THE TRANSMITTER AND THE RECEIVER.
C
C     IERR        INTEGER ERROR CODE WHICH EQUALS 1 IF A FATAL ERROR
C                 OCCURS AND 0 OTHERWISE
C
C     CNTRD       A SINGLY DIMENSIONED ARRAY CONTAINING THE HORIZONTAL
C                 COORDINATE AND THE VERTICAL COORDINATE OF THE
C                 CENTROID OF THE CLOUD.
C
C     HEIGHT      THE HEIGHT OF THE CLOUD IN METERS.
C
C     CENWTH      THE WIDTH OF THE CLOUD IN METERS AT THE CENTROID
C                 HEIGHT
C
C     SPCWTH      THE WIDTH OF THE CLOUD IN METERS AT THE SPECIFIED
C                 HEIGHT
C
```

```
C      NCPTS       THE NUMBER OF POINTS DETERMINED ON THE EDGE OF THE
C                  CLOUD.
C
C      CPTS        A DOUBLY DIMENSIONED ARRAY CONTAINING THE COORDINATES
C                  OF POINTS ON THE EDGE OF THE CLOUD.  CPTS(1,J)
C                  IS THE HORIZONTAL COORDINATE OF THE J-TH POINT
C                  AND CPTS(2,J)IS THE VERTICAL COORDINATE OF THE
C                  J-TH POINT.  THE FIRST INDEX MUST BE DIMENSIONED
C                  TO 2.
C
C      NERR        AN INTEGER ERROR CODE WITH VALUES
C                       -1   CPTS NOT LARGE ENOUGH FOR THE NUMBER OF
C                            POINTS CALCULATED
C                        0   NO ERRORS
C                        1   NO CONTOUR FOUND BY THE CONTOUR TRACING
C                            ROUTINE.  THIS MEANS THAT THE CLOUD HAS
C                            DISSIPATED.
C                        2   CONVERGENCE WAS NOT ACHIEVED IN SEARCHING
C                            FOR A POINT ON THE CONTOUR.  MAY REFER TO
C                            A DISCONTINUITY IN THE CLOUD EDGE.
C                        3   CONTOUR TOO SMALL TO TRACE I.E. LESS THAN
C                            A METER IN DIAMETER.
C                        4   LOSTRN AND EDGE WERE BOTH SPECIFIED FALSE
C                            SO NO RESULTS WERE CALCULATED
C                        5   NEWSRC WAS FALSE WHEN NEWATM WAS TRUE.
C                            A NEW ATMOSPHERE CANNOT BE SPECIFIED DURING
C                            THE EVOLUTION OF ONE DUST CLOUD.
C                        6   THE SPECIFIED HEIGHT FOR THE WIDTH OF THE
C                            CLOUD WAS ABOVE THE CLOUD
C                        7   THE CALCULATION OF THE ATMOSPHERIC
C                            PARAMETERS DID NOT CONVERGE.  THE MEASURED
C                            DATA MAY BE INCONSISTENT.
C
C
C
C      SUBROUTINES CALLED
C
C      ATMCAL      ACCEPTS METEOROLOGICAL DATA AS ARGUMENTS AND COMPUTES
C                  NECESSARY PARAMETERS IN COMMON /WNDPRM/ AND /TMPPRM/
C
C      SOURCE      ACCEPTS SOIL, CHARGE, AND WAVELENGTH SPECIFICATIONS
C                  AS INPUT AND COMPUTES NECESSARY PARAMETERS AND INITIAL
C                  VALUES IN COMMON /PRTINF/ AND /BUOYCL/
C
C      RISE        GIVEN CLOUD DIMENSIONS DURING BUOYANT RISE DEVELOPMENT
C                  OF CLOUD, RISE CALCULATES THE DIMENSIONS AT A LATER
C                  TIME
C
```

```
C      CONBYN      CONVERTS CLOUD DIMENSIONS AS SPECIFIED BY VARIABLES
C                  USED IN RISE TO OUTPUT VARIABLES DESCRIBING CLOUD
C                  DIMENSIONS
C
C      WIND        COMPUTES WIND VELOCITY AS A FUNCTION OF HEIGHT
C
C      SCONF       DETERMINES A SET OF POINTS DEFINING THE OBSERVABLE
C                  CONTOUR OF THE CLOUD AS VIEWED FROM A DISTANCE AT A
C                  GIVEN ANGLE TO THE WIND VELOCITY.
C
C      PTSPEC      USES THE SET OF COORDINATE POINTS DETERMINED BY SCONF
C                  TO CALCULATE OUTPUT VARIABLES DESCRIBING CLOUD
C                  DIMENSIONS
C
C
C      FUNCTIONS CALLED
C
C      CWIND       COMPUTES THE OPTICALLY WEIGHTED CONCENTRATION
C                  FOR TIMES AFTER THE TRANSITION FROM BUOYANT RISE
C                  TO WIND BLOWN
C
C ************************************************************************
      IERR=0
      CLMRED=.FALSE.
      WRITE(IOOUT,900)
  900 FORMAT(49H1     DIRTRAN-I DUST CLOUD INFRARED TRANSMISSION ,
     1        12HCALCULATION //
     2        46H ALL UNITS ARE MKS UNLESS OTHERWISE SPECIFIED.//)
  100 READ(IOIN,701)NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM
      IF(.NOT.NEWATM) GO TO 10
      IF(ICLMAT.EQ.1.AND..NOT.CLMRED)GO TO 5
      READ(IOIN,702)NATMOS,(ZTMP(I),TMPMES(I),ZWND(I),
     +WNDMES(I),I=1,2),ZINV,THWND
      GO TO 6
C
C      IPASCT    PASQUILL CATEGORY
C      WNDVEL    WIND VELOCITY IN M/S MEASURED AT 2 METERS ABOVE GROUND
C      WINDIR    WIND DIRECTION IN DEGREES CLOCKWISE FROM TRUE NORTH
C      TEMP      TEMPERATURE IN DEGREES C MEASURED AT 2 METERS HEIGHT
C
    5 NATMOS=IPASCT
      ZWND(1)=2.
      ZTMP(1)=2.
      WNDMES(1)=WNDVEL
      TMPMES(1)=TEMP+273.0
C
C      NOTE THAT POSITIVE X AXIS IS ASSUMED TO BE EAST.
C      NOTE THAT WINDIR IS THE DIRECTION OF ORIGIN OF THE WIND.
```

```
C
      THWND=270.-WINDIR
      ZINV=150.0
      CLMRED=.TRUE.
    6 CONTINUE
 950  FORMAT(1X)
      WRITE(IOOUT,950)
      WRITE(IOOUT,951)NATMOS
      NIO=1
      IF(NATMOS.EQ.0)NIO=2
      WRITE(IOOUT,952)(ZTMP(I),TMPMES(I),ZWND(I),WNDMES(I),I=1,NIO)
      WRITE(IOOUT,953)THWND
  951 FORMAT(30H PASQUILL CATEGORY             ,I2)
  952 FORMAT(8H     HT ,F7.2,7H  TEMP ,F7.2,7H     HT,F7.2,7H  WIND ,
     1 F7.2)
  953 FORMAT(22H WIND DIRECTION        ,F7.2)
   10 CONTINUE
      IF(.NOT.NEWSRC)GO TO 20
      READ(IOIN,703)NSOIL,NCHRG,CHWT,DETDEP,DSOD,SRCCOR(1),
     1 SRCCOR(2)
      WRITE(IOOUT,950)
      WRITE(IOOUT,955)NSOIL
      WRITE(IOOUT,954)NCHRG
      WRITE(IOOUT,956)CHWT
      WRITE(IOOUT,957)DETDEP
      WRITE(IOOUT,958)DSOD
      WRITE(IOOUT,959)(SRCCOR(I),I=1,2)
      WRITE(IOOUT,950)
  954 FORMAT(15H CHARGE INDEX    ,I2)
  955 FORMAT(15H SOIL INDEX      ,I2)
  964 FORMAT(30H WAVELENGTH                    ,F7.2,13H MICROMETERS )
  956 FORMAT(30H WEIGHT OF CHARGE              ,F7.2)
  957 FORMAT(30H DETONATION DEPTH              ,F7.2)
  958 FORMAT(30H DEPTH OF SOD                  ,F7.2)
  959 FORMAT(30H SOURCE COORDINATES            ,2F10.2)
   20 CONTINUE
      IF(.NOT.LOSTRN)GO TO 30
      READ (IOIN,704)(TRNCOR(I),I=1,3),(RECCOR(J),J=1,3)
      IF(WAVE1.LT.0.4)GO TO 29
      IF(WAVE1.GT.0.7)GO TO 21
      NWL=1
      GO TO 30
   21 IF(WAVE1.LT.0.8)GO TO 29
      IF(WAVE1.GT.1.1)GO TO 22
      NWL=2
      GO TO 30
   22 IF(WAVE1.LT.3.5)GO TO 29
      IF(WAVE1.GT.4.0)GO TO 23
```

```
      NWL=3
      GO TO 30
   23 IF(WAVE1.LT.8.5)GO TO 29
      IF(WAVE1.GT.12.0)GO TO 24
      NWL=4
      GO TO 30
   24 IF(WAVE1.LT.2100)GO TO 29
      IF(WAVE1.GT.3200)GO TO 29
      NWL=5
      GO TO 30
   29 WRITE(IOOUT,25)
   25 FORMAT(42H *** DIRTRAN-I ERROR - WAVE1 OUT OF RANGE    )
      IERR=1
      GO TO 999
   30 IF (EDGE) READ (IOIN,704) (OBSCOR(IO),IO=1,2),SPCHT
      TIMTST=TIME
      IF (NEWTIM) READ (IOIN,704) TIME
      IF(.NOT.NEWSRC.AND.TIME.LT.TIMTST)GO TO 997
      WRITE(IOOUT,960)TIME
  960 FORMAT(//30H TIME AFTER BLAST                ,F7.2)
      NERR=0
      CALL DUSTCL (NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,ZINV,THWND,
     1  NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,NWL,SRCCOR,
     2  LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,TIME,
     3  TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS,NERR)
C       IF (NERR.EQ.0)GO TO 600
C       WRITE(IOOUT,969)NERR
C 969   FORMAT(20H ***** ERROR NUMBER ,I2)
C       GO TO 800
 600  IF(.NOT.LOSTRN)GO TO 700
      WRITE(IOOUT,950)
      WRITE(IOOUT,964)WAVE1
      WRITE(IOOUT,961)(TRNCOR(I),I=1,3)
      WRITE(IOOUT,962)(RECCOR(I),I=1,3)
  961 FORMAT(30H TRANSMITTER COORDINATES       ,3F10.2)
  962 FORMAT(30H RECEIVER    COORDINATES       ,3F10.2)
      WRITE(IOOUT,970)TRNLOS
 970  FORMAT(40H TRANSMITTANCE ALONG THE LINE-OF-SIGHT     ,E10.3)
 700  IF(.NOT.EDGE)GO TO 800
      WRITE(IOOUT,950)
      WRITE(IOOUT,963)(OBSCOR(I),I=1,2)
  963 FORMAT(30H OBSERVER    COORDINATES       ,2F10.2)
      WRITE(IOOUT,971)HEIGHT
 971  FORMAT(28H THE HEIGHT OF THE CLOUD IS ,10X,F10.2,7H METERS)
C
  990 FORMAT((1X,2F10.2))
C
      WRITE(IOOUT,972)(CNTRD(IO),IO=1,2)
```

```
 972  FORMAT(30H THE CENTROID COORDINATES ARE ,8X,2F10.2)
      WRITE(IOOUT,973)CENWTH
 973  FORMAT(30H THE WIDTH AT THE CENTROID IS ,8XF10.2, 7H METERS)
      WRITE(IOOUT,974)SPCHT,SPCWTH
 974  FORMAT(14H THE WIDTH AT ,F8.2,11H METERS IS ,5X,F10.2,7H METERS)
      WRITE(IOOUT,975)NCPTS
 975  FORMAT(1X,I3,37H CONTOUR POINTS HAVE BEEN DETERMINED )
      WRITE(IOOUT,708)((CPTS(IO,IPT),IO=1,2),IPT=1,NCPTS)
800   READ(IOIN,701)CNTNU
      IF(CNTNU)GO TO 100
  701 FORMAT (5L1)
  702 FORMAT (I1,1X,10F7.2)
  703 FORMAT (2(I1,1X),5F7.2)
  704 FORMAT (6F7.2)
  705 FORMAT (I2)
  708 FORMAT ((1X,2(F10.3,2X)))
  709 FORMAT(2(I2,1X),2F7.2)
      GO TO 999
  997 WRITE(IOOUT,998)
  998 FORMAT(54H *** DIRTRAN ERROR - TIMES ARE NOT IN INCREASING ORDER)
      IERR=1
  999 RETURN
      END
C              CONTROLLING ROUTINE FOR DIRTRAN-I CODE
C
C ***********************  SUBFILE 2  ***************************
C
      SUBROUTINE DUSTCL(NEWATM,NATMOS,ZTMP,TMPMES,ZWND,WNDMES,HTINV,
     1  THWND,NEWSRC,CHWT,NCHRG,DETDEP,NSOIL,DSOD,NWL,SRCCOR,
     2  LOSTRN,TRNCOR,RECCOR,EDGE,OBSCOR,SPCHT,NEWTIM,
     3  TIME,TRNLOS,CNTRD,HEIGHT,CENWTH,SPCWTH,NCPTS,CPTS,NERR)
      IMPLICIT INTEGER*4 (I-N)
      LOGICAL NEWATM,NEWSRC,LOSTRN,EDGE,NEWTIM,HORIZ,ERR
      DIMENSION ZTMP(2),TMPMES(2),ZWND(2),WNDMES(2),SRCCOR(2),TRNCOR(3)
     1  ,RECCOR(3),CPTS(2,200),ORIG(2),TRNFRM(2,2),TRN(3),REC(3)
     2  ,PNT(3),DELS(3),CNTRD(2),OBSCOR(2),DIR(2),SCRN(2)
      DIMENSION OWF(5,2)
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI
      COMMON /MODE/ HORIZ
      COMMON /WNDPRM/ DXZO,DYXO,DZO,UO,UM,DN,ZINV
      COMMON /CLOCK/ FTIME,TWIND
      COMMON /SEPRTN/ SEP1(2),SEP2(2),PRSEP1,PRSEP2,NUM1,NUM2
      COMMON /IOUNIT/ IOIN,IOOUT,ISPTPF,LOUNIT,NDIRTU,NBSCAT
      EXTERNAL FUNCT
      DATA PI/3.1415927/,VISEXT,ZMIN,RES,TANT/.1,0.,1.,.1/
      DATA ONEM/-1./,RTPI/1.77245/
      DATA OWF/1.,.93,.52,.44,7.E-4,1.,.93,.66,.52,7.E-4/
C ****************************************************************
C
```

```
C      DAVID DVORE
C      AERODYNE RESEARCH, INC.
C      (617)275-9400 X127
C
C      PURPOSE
C
C           DUSTCL CALCULATES DUST CLOUD DIMENSIONS AND TRANSMITTANCES
C      THROUGH DUST CLOUDS FOR GIVEN METEOROLOGICAL DATA, SOIL TYPE,
C      BOMB CHARACTERISTICS, AND WAVELENGTH.
C
C      SEE COMMENTS IN DRTRAN FOR DETAILS.
C
C *******************************************************************
       IF(LOSTRN.OR.EDGE)GO TO 100
       NERR=4
       GO TO 999
   100 IF(.NOT.NEWATM) GO TO 200
       IF(NEWSRC) GO TO 150
       NERR =5
       GO TO 999
   150 CONTINUE
       ZINV=HTINV
       CALL ATMCAL(NATMOS,ZTMP,TMPMES,ZWND,WNDMES,ERR)
       IF(.NOT.ERR)GO TO 155
       NERR=7
       GO TO 999
   155 CONTINUE
C
C      COMPUTE THE ROTATION TRANSFORMATION MATRIX TO CONVERT USER
C      DEFINED COORDINATES INTO LOCAL COORDINATES WITH X AXIS IN
C      THE WIND DIRECTION.
C
       THETAX=THWND*PI/180.
       TRNFRM(1,1)=COS(THETAX)
       TRNFRM(2,2)=TRNFRM(1,1)
       TRNFRM(1,2)=SIN(THETAX)
       TRNFRM(2,1)=-TRNFRM(1,2)
   200 CONTINUE
       IF(.NOT.NEWSRC) GO TO 300
       TWIND=1.E5
       TPRES=0.
       DEL=.001
       CALL SOURCE(CHWT,NCHRG,DETDEP,NSOIL,DSOD)
       DO 250 I=1,2
       ORIG(I)=SRCCOR(I)
   250 CONTINUE
       OVRLAP=SPACNG/2.
       SEP1(1)=SPACNG*COS(THARRY-THETAX)
```

```
      SEP1(2)=SPACNG*SIN(THARRY-THETAX)
      CALL PERP(SEP1,SEP2)
  300 CONTINUE
      IF(.NOT.LOSTRN) GO TO 400
C
C     CONVERT TRNCOR AND RECCOR TO LOCAL COORDINATES WITH ORIGIN AT
C     DETONATION POINT AND X AXIS IN WIND DIRECTION.
C
      TRN(3)=TRNCOR(3)
      REC(3)=RECCOR(3)
      DO 320 I=1,2
      TRN(I)=0.
      REC(I)=0.
      DO 310 J=1,2
      TRN(I)=TRN(I)+TRNFRM(I,J)*(TRNCOR(J)-ORIG(J))
      REC(I)=REC(I)+TRNFRM(I,J)*(RECCOR(J)-ORIG(J))
  310 CONTINUE
  320 CONTINUE
  400 CONTINUE
      IF(.NOT.EDGE) GO TO 500
C
C     COMPUTE SIN AND COS OF THE ANGLE BETWEEN THE OBSERVERS VIEWING
C     ANGLE AND THE LOCAL POSITIVE X AXIS WHICH IS IN THE WIND
C     DIRECTION.
C
      CALL VSUM(ORIG,OBSCOR,ONEM,DIR)
      CALL UNIT(DIR,DIR,RANGE)
      COSTH=0.
      SINTH=0.
      DO 410 J=1,2
      COSTH=COSTH+TRNFRM(1,J)*DIR(J)
      SINTH=SINTH+TRNFRM(2,J)*DIR(J)
  410 CONTINUE
      SINTH2=SINTH*SINTH
      COSTH2=COSTH**2
      SCRN(1)=SINTH
      SCRN(2)=-COSTH
  500 CONTINUE
      IF(NEWTIM) CALL RISE(TPRES,TIME,DEL)
  600 IF(.NOT.EDGE) GO TO 650
      FTIME=TIME
      CALL CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS,
     1 NERR)
      IF(.NOT.ERR)GO TO 650
      NERR=6
      GO TO 999
  650 CONTINUE
      IF(.NOT.LOSTRN)GO TO 999
```

```
C
C     DETERMINE THE OPTICALLY WEIGHTED CL VALUE
C     ALONG THE LINE CONNECTING THE TRANSMITTER AND RECIEVER
C
C
      CALL VSUM(REC,TRN,ONEM,DIR)
      CALL UNIT(DIR,DIR,RANGE)
      COSTH=DIR(1)
      SINTH=DIR(2)
      SINTH2=SINTH*SINTH
      COSTH2=COSTH**2
      SCRN(1)=SINTH
      SCRN(2)=-COSTH
      X=DOTPRD(SCRN,TRN)
      HORIZ=.TRUE.
      TRNLOS=EXP(-CWIND(X,Y,TRN(3),TIME)*OWF(NWL,NSOIL))
  999 RETURN
      END
C        CALCULATION OF ATMOSPHERIC PARAMETERS FOR DIRTRAN-I CODE
C
C *****************  SUBFILE 3  *********************
C
      SUBROUTINE ATMCAL(NATM,ZT,TMES,ZU,UMES,ERR)
      IMPLICIT INTEGER*4 (I-N)
      REAL M,N
      LOGICAL ERR
      DIMENSION ZT(2),TMES(2),ZU(2),UMES(2),ZLO(6)
      COMMON /WNDPRM/ DXZO,DYXO,DZO,UO,M,N,ZINV
      COMMON /TMPPRM/TO,T1,TM
      COMMON /IOUNIT/ IOIN,IOOUT,ISPTPF,LOUNIT,NDIRTU,NBSCAT
      DATA ZLO/-2.5,-4.5,-13.5,1000.,55.,20./
C ***********************************************************
C
C
C     PURPOSE
C
C          TO FIT THE BEST POWER-LAW PROFILES OF WINDSPEED,
C          DIFFUSIVITY, AND TEMPERATURE CONSISTENT WITH KNOWN RELATIONS
C          GOVERNING THE CONSTANT SHEAR STRESS LAYER TO GIVEN
C          MEASUREMENTS AT TWO HEIGHTS.
C
C
C     INPUTS
C
C          NATM      INTEGER WHICH IS 0 IF WINDSPEED AND TEMPERATURE
C                    ARE AVAILABLE AT TWO HEIGHTS AND EQUAL TO THE
C                    PASQUILL CATEGORY OTHERWISE.
C
```

```
C          ZT          SINGLY DIMENSIONED ARRAY CONTAINING TWO HEIGHTS
C                      (IN METERS) AT WHICH TEMPERATURES WILL BE GIVEN.
C                      MUST BE IN ASCENDING ORDER.
C
C          TMES        SINGLY DIMENSIONED ARRAY CONTAINING THE TWO
C                      TEMPERATURE MEASUREMENTS IN DEGREES KELVIN
C                      AT HEIGHTS ZT.
C
C          ZU          SINGLY DIMENSIONED ARRAY CONTAINING ONE OR TWO
C                      HEIGHTS (METERS) AT WHICH WIND SPEEDS WILL BE
C                      GIVEN.  MUST BE IN ASCENDING ORDER.
C
C          UMES        SINGLY DIMENSIONED ARRAY CONTAINING THE ONE OR
C                      TWO WIND SPEED MEASUREMENTS (M/S) AT HEIGHTS UMES.
C
C          ZINV        INVERSION HEIGHT IN METERS.
C
C
C
C
C     OUTPUTS
C
C          ERR          A LOGICAL WHICH IS TRUE IF AN ERROR IS INCURED
C                       DURING THE CALCULATION.
C
C          DXZO         THE RATIO OF THE DIFFUSIVITY IN THE X DIRECTION
C                       TO THE DIFFUSIVITY IN THE Z DIRECTION. RETURNED
C                       IN COMMON /WNDPRM/.
C
C          DYXO         THE RATIO OF THE DIFFUSIVITY IN THE Y DIRECTION
C                       TO THE DIFFUSIVITY IN THE X DIRECTION. RETURNED
C                       IN COMMON /WNDPRM/.
C
C          DZO          THE COEFICIENT OF Z**N IN THE VERTICAL PROFILE
C                       OF VERTICAL DIFFUSIVITY.  RETURNED IN COMMON
C                       /WNDPRM/.
C
C          UO           THE COEFICIENT OF Z**M IN THE VERTICAL PROFILE
C                       OF HORIZONTAL WIND SPEED.  RETURNED IN COMMON
C                       /WNDPRM/.
C
C          M            THE EXPONENT OF Z IN THE HORIZONTAL WIND SPEED
C                       PROFILE.  RETURNED IN COMMON /WNDPRM/.
C
C          N            THE EXPONENT OF Z IN THE VERTICAL DIFFUSIVITY
C                       PROFILE.  RETURNED IN COMMON /WNDPRM/.
C
C          TO,T1,TM     CONSTANTS FOR THE TEMPERATURE PROFILE SUCH THAT
```

```
C                       T=TO+T1*Z**TM.  RETURNED IN COMMON /TMPPRM/.
C
C
C
C      CALLED FROM DUSTCL
C
C      NEEDED FUNCTIONS AND SUBROUTINES
C
C            TMPCAL    CALCULATES SCALED TEMPERATURE PROFILES
C
C            WNDCAL    CALCULATES SCALED WIND SPEED PROFILES
C
C ******************************************************************
C
       ERR=.FALSE.
C
C      DELTH IS THE DIFFERENCE IN POTENTIAL TEMPERATURE BETWEEN THE
C      TWO HEIGHTS WHERE TEMPERATURE IS GIVEN.
C
       Z0=0.02
       IF(NATM.EQ.0)GO TO 100
C
C      ASSIGN ATMOSPHERIC PROFILE ACCORDING TO SPECIFIED PASQUILL
C      CATEGORY
C
C      Z0      FRICTION HEIGHT
C      ZL      MOMIN-OBUKOV LENGTH
C      USTAR   THE FRICTION VELOCITY
C      TSTAR   THE SCALING TEMPERATURE
C
       ZL=ZLO(NATM)
       NP=IFIX(SIGN(1.,ZL))
       USTAR=UMES(1)/WNDCAL(Z0,ZL,ZU(1))
       TSTAR=TMES(1)*USTAR**2/1.568/ZL
       IF(NATM-4)200,300,210
  100  CONTINUE
C
C      USE ITERATIVE PROCEDURE TO CONVERGE ON BEST ATMOSPHERIC PROFILE
C      TO MATCH DATA AT TWO HEIGHTS
C
       DELTH=TMES(2)-TMES(1)+.0098*(ZT(2)-ZT(1))
       NP=SIGN(1.,DELTH)
       DELU=UMES(2)-UMES(1)
       ZULOG=ALOG(ZU(2)/ZU(1))
       ZTLOG=ALOG(ZT(2)/ZT(1))
       USTAR=(UMES(2)-UMES(1))/ZULOG
       TSTAR=DELTH/ZTLOG
       ZL=.638*TMES(1)*USTAR**2/TSTAR
```

```
      IF(ABS(ZL).GE.1000.)GO TO 300
      DO 110 ITER=1,100
      USTAR=DELU/(WNDCAL(Z0,ZL,ZU(2))-WNDCAL(Z0,ZL,ZU(1)))
      TSTAR=DELTH/(TMPCAL(Z0,ZL,ZT(2))-TMPCAL(Z0,ZL,ZT(1)))
      ZLP=ZL
      ZL=.638*TMES(1)*USTAR**2/TSTAR
      IF(ABS((ZL-ZLP)/ZLP).LT..01)GO TO 120
  110 CONTINUE
      ERR=.TRUE.
      GO TO 999
  120 CONTINUE
      Z0=EXP(.4*(WNDCAL(Z0,ZL,ZU(1))-UMES(1)/USTAR))*Z0
      IF(NP)200,300,210
  200 CONTINUE
C
C     UNSTABLE ATMOSPHERE
C
      DZ0=3.15*.4*USTAR/ABS(ZL)**(1./3.)
      DXZ0=2.
      M=1./7.
      N=4./3.
      GO TO 430
  210 CONTINUE
C
C     STABLE ATMOSPHERE
C
      DXZ0=6.
      M=1./7.
      N=1.
      DZ0=.4*USTAR/(1.+47./ZL)
      GO TO 430
  300 CONTINUE
C
C     NEUTRAL ATMOSPHERE
C
      DZ0=.4*USTAR
      DXZ0=5.
      NP=0
      N=1.
      M=1./7.
  430 CONTINUE
C
C     COMMON CALCULATION TO UNSTABLE, NEUTRAL, AND STABLE ATMOSPHERES
C
      IF(ZINV.LE.10.)ZINV=200.
      U0=UMES(1)/ZU(1)**M
      DYX0=1.
      IF(NATM.EQ.0)U0=(U0+UMES(2)/ZU(2)**M)/2.
```

```
      IF(NATM.EQ.0)GO TO 450
      ZT(2)=2.*ZT(1)
      DELTH=TSTAR*(TMPCAL(Z0,ZL,ZT(2))-TMPCAL(Z0,ZL,ZT(1)))
      TMES(2)=TMES(1)+DELTH
  450 T1=DELTH/(ZT(2)**M-ZT(1)**M)
      TM=M
      T0=TMES(1)-T1*ZT(1)**M
      DZ0=.4*USTAR
      N=1.
  999 RETURN
      END
C
C *****************  SUBFILE 4  ********************************
C
      FUNCTION WNDCAL(Z0,ZL,Z)
      IMPLICIT INTEGER*4 (I-N)
C ****************************************************************
C
C
C
C     PURPOSE
C
C     TO CALCULATE THE WIND SPEED, U/U*, SCALED BY THE FRICTION
C     VELOCITY FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV LENGTH AT A
C     SPECIFIED HEIGHT.
C
C
C     INPUTS
C
C     Z0       THE FRICTION HEIGHT IN METERS.
C     ZL       THE MONIN-OBUKHOV LENGTH IN METERS.
C     Z        THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED
C              IN METERS
C
C
C     RETURNS VELOCITY SCALED BY FRICTION VELOCITY
C
C
C     CALLED BY ATMCAL
C
C ****************************************************************
      LOGICAL LOW
      PSIM(S)=-(ALOG((S+1.)**2*(S*S+1)/8.)+2.*(ATAN(1.)-ATAN(S)))
      PSIMS(Z)=-7.*Z
      PHIM(Z)=(1.-16.*Z)**(-.25)
C
C     PHIM       THE SHEAR OF MOMENTUM
C     PSIM       THE UNIVERSAL FUNCTION FOR THE DEVIATION FROM
```

```
C               LOGARITHMIC WIND VELOCITY BOUNDARY LAYER PROFILE IN AN
C               UNSTABLE ATMOSPHERE
C     PSIMS     THE SAME AS PSIM FOR A STABLE ATMOSPHERE
C
      IF(ABS(ZL).LE.1.E3)GO TO 100
      WNDCAL=ALOG(1.+Z/Z0)
      GO TO 999
  100 CONTINUE
      P=SIGN(1.,ZL)
      LOW=.TRUE.
      S=Z/ZL
      IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10
      S=AMIN1(S,1.5)
      S=AMAX1(S,-2.)
      LOW=.FALSE.
   10 CONTINUE
      IF(P)120,130,130
  120 S=1./PHIM(S)
      PSI=PSIM(S)
      GO TO 52
  130 CONTINUE
      PSI=PSIMS(S)
   52 CONTINUE
      WNDCAL=ALOG(1.+Z/Z0)-PSI
      IF(LOW)GO TO 999
      IF(P)53,53,54
   53 WNDCAL=WNDCAL+.75-.95*(-ZL/Z)**(1./3.)
      GO TO 999
   54 WNDCAL=WNDCAL-15.+10.*Z/ZL
  999 WNDCAL=WNDCAL/.4
      RETURN
      END
C
C ******************  SUBFILE 5  ************************
C
      FUNCTION TMPCAL(Z0,ZL,Z)
      IMPLICIT INTEGER*4 (I-N)
C ***************************************************************
C
C
C
C     PURPOSE
C
C     TO CALCULATE THE POTENTIAL TEMPERATURE SCALED BY THE SCALE
C     TEMPERATURE, T*, FROM GIVEN FRICTION HEIGHT AND MONIN-OBUKHOV
C     LENGTH AT A SPECIFIED HEIGHT.
C
C
```

```
C       INPUTS
C
C       ZO      THE FRICTION HEIGHT IN METERS.
C       ZL      THE MONIN-OBUKHOV LENGTH IN METERS.
C       Z       THE HEIGHT AT WHICH THE SCALED VELOCITY IS DESIRED
C               IN METERS
C
C
C       RETURNS SCALED TEMPERATURE
C
C
C       CALLED BY ATMCAL
C
C ********************************************************************
        LOGICAL LOW
        PHIM(Z)=(1.-16.*Z)**(-.25)
        PSIH(S)=-2.*ALOG((S*S+1.)/2.)
        PSIHS(Z)=-11.*Z
C
C
C       PHIM     THE SHEAR OF MOMENTUM
C       PSIH     THE UNIVERSAL FUNCTION FOR DEVIATION FROM LOGARITHMIC
C                POTENTIAL TEMPERATURE PROFILE IN THE BOUNDARY LAYER
C                OF AN UNSTABLE ATMOSPHERE
C       PHIHS    THE SAME AS PHIH EXCEPT FOR STABLE ATMOSPHERE
C
        IF(ABS(ZL).LE.1.E3)GO TO 100
        TMPCAL=ALOG(1.+Z/ZO)
        GO TO 999
    100 CONTINUE
        P=SIGN(1.,ZL)
        LOW=.TRUE.
        S=Z/ZL
        IF(S.LE.1.5.AND.S.GE.-2.)GO TO 10
        S=AMIN1(S,1.5)
        S=AMAX1(S,-2.)
        LOW=.FALSE.
     10 CONTINUE
        IF(P)120,130,130
    120 S=1./PHIM(S)
        PSI=PSIH(S)
        GO TO 52
    130 CONTINUE'
        PSI=PSIHS(S)
     52 CONTINUE
        TMPCAL=ALOG(1.+Z/ZO)-PSI
        IF(LOW)GO TO 999
        IF(P)53,53,54
```

```
   53 TMPCAL=TMPCAL+.75-.95*(-ZL/Z)**(1./3.)
      GO TO 999
   54 TMPCAL=TMPCAL-15.+10.*Z/ZL
  999 RETURN
      END
C                 SOURCE CALCULATION FOR DIRTRAN-I CODE
C
C ********************  SUBFILE 6  ***********************
C
C
C
      SUBROUTINE SOURCE(W,NCHRG,DD,NSOIL,DSOD)
      IMPLICIT INTEGER*4 (I-N)
C *****************************************************
C
C
C     PURPOSE
C
C     TO CALCULATE EXPLOSIVE DUST SOURCE TERM FOR THE
C              DIRTRAN-I CODE
C
C     INPUT
C
C       W        THE WEIGHT OF THE CHARGE IN LBS. TNT
C       DD       DETONATION DEPTH IN METERS
C       NSOIL    INTEGER SOIL INDEX
C       DSOD     DEPTH OF SOD IN METERS
C
C     OUTPUT  (RETURNED IN COMMON /PRTINF/ AND /BUOYCL/ )
C
C       RO       INITIAL CLOUD RADIUS IN METERS
C       VGRAV    SINGLY DIMENSIONED ARRAY CONTAINING OPTICALLY WEIGHTED
C                AVERAGE SETTLING VELOCITIES FOR EACH SIZE RANGE IN
C                THE PARTICLE DISTRIBUTION  (METERS/SEC)
C       NPRTS    THE NUMBER OF SIZE RANGES IN THE PARTITIONING OF THE
C                PARTICLE SIZE SPECTRUM
C       RSPH     THE INITIAL RADIUS OF THE CLOUD IN METERS
C       DELT     THE INITIAL DIFFERENCE IN TEMPERATURE BETWEEN THE CLOUD
C                AND SURROUNDINGS  (DEGREES KELVIN)
C       VXSPH    THE INITIAL HORIZONTAL VELOCITY OF THE CLOUD (M/S)
C       VZSPH    THE INITIAL VERTICAL VELOCITY OF THE CLOUD (M/S)
C       XCMSPH   INITIAL HORIZONTAL POSITION OF THE CLOUD (METERS)
C       ZCMSPH   INITIAL HEIGHT OF THE CLOUD (METERS)
C       RISTIM   TIME LAPSED SINCE DETONATION IN SECONDS
C
C
C     CALLED BY DUSTCL
C
```

```
C **********************************************************************
      LOGICAL HORIZ
      DIMENSION CR(5,4),CD(5,4),OWML(3,4),OWSV(3,4),PRTTN(4)
      DIMENSION S(3),BURHTR(5),WTRAT(5)
      COMMON/PRTINF/ RO,VGRAV(3),NPRTS
      COMMON/BUOYCL/ RSPH,DELT,VXSPH,VZSPH,XCMSPH,ZCMSPH,
     * SPHNS(3),RISTIM
      COMMON /TMPPRM/ TO,T1,TM
      COMMON /WNDPRM/ DXZO,DYXO,DZO,UO,UM,DN,ZINV
      COMMON /BURST/ ACCEL,TBURST
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI
      COMMON /MODE/ HORIZ
      COMMON/DISCS/NDSCS,TDSC(10),XDSC(10),ZDSC(10),R2DSC(10),
     1 QDSC(10,3)
C
C   CR IS THE CRATER RADIUS INDEXED BY COEFFICIENT AND SOIL TYPE
      DATA CR/.483,-.328,.0319,.0536,0.,.386,-.291,.0543,.057,11*0./
C   CD IS THE CRATER DIAMETER INDEXED BY COEFFICIENT AND SOIL TYPE
      DATA CD/.261,-.453,.0681,.224,.055,.198,-.355,.0582,.215,.0586,
     1 10*0./
C   OWML IS THE OPTICALLY WEIGHTED MASS LOADING COEFFICIENT INDEXED BY
C        BIN SIZE AND SOIL TYPE
      DATA OWML/1.3E4,2*0.,2.61E4,8*0./
C   OWSV IS THE OPTICALLY WEIGHTED PARTICLE SETTLING VELOCITY (CM/SEC)
C        INDEXED BY BIN SIZE AND SOIL TYPE
      DATA OWSV/12*0./
C   PRTTN IS THE PARTITIONING RATIO INDEXED ON SOIL TYPE
      DATA PRTTN/4*.9/
C    BURHTR IS THE RATIO OF BURST HEIGHT TO INITIAL RADIUS AND WTRAT
C    IS THE FRACTION OF THE TOTAL WEIGHT WHICH IS EFFECTIVE IN THE CLOUD
      DATA BURHTR/0.,4.,2.,4.,3./,WTRAT/.6,1.,.8,1.,.6/
C
C
      RISTM=0.
      XCMSPH=0.
      NPRTS=1
      W3=(W*WTRAT(NCHRG))**.3333333
      RO=1.535*W3
      TAMB=TO+T1*RO**TM
      DELT=.57*TAMB
      RSPH=RO
      ZCMSPH=RO
      VXSPH=UO*ZCMSPH**UM
      BURHT=BURHTR(NCHRG)*RO
      BURVZ=1.3*SQRT(RO)
      TBURST=.15*RO
      VZSPH= 2.*BURHT/TBURST-BURVZ
      ACCEL=(BURVZ-VZSPH)/TBURST
```

```
      CLAM=DD/W3
C
C  CALCULATE CRATER RADIUS AND DEPTH
C
      RC=CR(1,NSOIL)
      DC=CD(1,NSOIL)
      IF (CLAM.LT.1.E-30) GO TO 98
      TERM=1.
      DO 100 I=2,5
      TERM=TERM*CLAM
      RC=RC + CR(I,NSOIL)*TERM
      DC=DC + CD(I,NSOIL)*TERM
100   CONTINUE
   98 CONTINUE
      RC=RC*W3
      DC=DC*(W*WTRAT(NCHRG))**.3
C
C  GET CRATER VOLUME
C
      VC=(3.1415926/3.) * (RC/DC)**2 * (DC - DSOD)**3
C
C  CALCULATE OPTICALLY WEIGHTED PARAMETERS
C
      NDSCS=MAX0(5,IFIX(5.*W3/2.4))
      DO 101 L=1,NPRTS
      S(L)=OWML(L,NSOIL) * VC
      VGRAV(L)=CWSV(L,NSOIL)
      SPHNS(L)=PRTTN(NSOIL) * S(L)
      QDSC(1,L) =(1.-PRTTN(NSOIL)) * S(L)/FLOAT(NDSCS)
101   CONTINUE
      DELH=2.*RO/FLOAT(NDSCS)
      Z=-DELH/2.
      DO 200 I=1,NDSCS
      Z=Z+DELH
      ZDSC(I)=Z
      DO 201 J=1,NPRTS
      QDSC(I,J)=QDSC(1,J)
  201 CONTINUE
      CON=ALOG(QDSC(I,1)/VISEXT/DELH/(2.*RO)/3.14159)
      IF(CON.GT.1.)GO TO 210
      D=1.
      GO TO 230
  210 D=CON
      DO 220 IT=1,5
      D=(CON-1.+ALOG(D))*D/(D-1)
  220 CONTINUE
  230 R2DSC(I)=4.*RO*RO/D
      TDSC(I)=-DELH*DELH/D/(DZO*Z**DN)/4.
```

```
      SIGZ=DELH*DELH/D
C      WRITE(1,777)SIGZ
      XDSC(I)=U0*Z**UM * TDSC(I)
C      WRITE(1,777)TDSC(I),XDSC(I),ZDSC(I),R2DSC(I),QDSC(I,1)
  777 FORMAT(1X,5(1PE10.3))
  200 CONTINUE
      RETURN
      END
C           WIND DISPERSION OF DUST CLOUD FOR DIRTRAN-I CODE
C
C *****************  SUBFILE 7  ******************************
C
      FUNCTION FUNCT(X,Z)
      IMPLICIT INTEGER*4 (I-N)
      COMMON /CLOCK/ TIME,TWIND
C ***********************************************
C
C
C     PURPOSE
C
C       TO SUPPLY A TRANSMITTANCE FUNCTION FOR THE CONTOUR TRACING
C       ROUTINE IN ORDER TO DETERMINE THE CLOUD EDGE.
C
C     INPUT
C
C       X   THE HORIZONTAL COORDINATE IN METERS
C       Z   THE VERTICAL COORDINATE IN METERS
C
C     OUTPUT
C
C       RETURNS THE LOG OF THE OPTICALLY WEIGHTED CL VALUE FOR THE
C       LINE-OF-SIGHT SPECIFIED BY X,Z
C
C     FUNCTIONS CALLED
C
C       CWIND
C
C     CALLED BY GFUN, CLIMB, GRAD2
C
C **********************************************************
      Y=0.
C      WRITE(1,1)X,Z,TIME
    1 FORMAT(1X,3(1PE10.2))
      IF(Z.LE.0.)GO TO 100
      EXT=CWIND(X,Y,Z,TIME)
      IF(EXT.LE.1.E-30)GO TO 100
      FUNCT=ALOG(EXT)
      GO TO 999
```

```
  100 FUNCT=-30.
  999 CONTINUE
C        WRITE(1,2)FUNCT
    2 FORMAT(1H+,30X,1PE10.2)
      RETURN
      END
C
C
C **********************  SUBFILE 8  *****************************
C
      FUNCTION CWIND(X,Y,Z,T)
      IMPLICIT INTEGER*4 (I-N)
      REAL M,N
      LOGICAL HORIZ
      DIMENSION REF(2),REF0(2)
      COMMON/PRTINF/ R0,VGRAV(3),NPRTS
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI
      COMMON /MODE/ HORIZ
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
      COMMON/DISCS/NDSCS,TDSC(10),XDSC(10),ZDSC(10),R2DSC(10),
     1 QDSC(10,3)
      COMMON /SEPRTN/ SEP1(2),SEP2(2),PRSEP1,PRSEP2,NUM1,NUM2
C ***************************************************************
C
C
C      PURPOSE
C
C        TO COMPUTE THE CONCENTRATION AT A POINT OR INTEGRATED ALONG
C        A HORIZONTAL LINE
C
C      INPUT
C
C        X,Y,Z     COORDINATES IN METERS.  IF LINE INTEGRAL IS DESIRED,
C                  Y IS IGNORED AND LINE IS SPECIFIED BY X AND Z.
C
C        T         THE TIME IN SECONDS AFTER DETONATION
C
C      OUTPUT
C
C        RETURNS THE CONCENTRATION AT X,Y,Z,T IF HORIZ IS .FALSE. AND
C        THE LINE INTEGRAL OF CONCENTRATION IF HORIZ IS .TRUE.
C
C      SUBROUTINES CALLED
C
C        MOMENT    COMPUTES ZERO ORDER MOMENT AND INTERPOLATES FROM
C                  TABLE OF HIGHER ORDER MOMENTS.
C
C        CALLED BY FUNCT,DUSTCL
```

```
C
C ******************************************************************
C
C     COMMON /PRTINF/
C
C       RO       INITIAL RADIUS OF THE CLOUD IN METERS
C       VGRAV    SINGLY DIMENSIONED ARRAY.  VGRAV(I) IS THE OPTICALLY
C                WEIGHTED AVERAGE SETTLING VELOCITY FOR PARTICLES IN THE
C                I SIZE RANGE
C       NPRTS    THE NUMBER OF PARTICLE SIZE RANGES
C
C     COMMON /DISCS/
C
C       NDSCS    THE NUMBER OF DISC SOURCES
C       TDSC     SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF RELEASE
C                OF THE DISC SOURCES
C       XDSC     SINGLY DIMENSIONED ARRAY CONTAINING THE X COORDINATE
C                OF THE CENTER OF THE DISC SOURCES
C       ZDSC     SINGLY DIMENSIONED ARRAY CONTAINING THE Z COORDINATE
C                OF THE CENTER OF THE DISC SOURCES
C       R2DSC    SINGLY DIMENSIONED ARRAY CONTAINING THE SQUARE OF THE
C                RADII OF THE DISC SOURCES
C       QDSC     DOUBLY DIMENSIONED ARRAY.   QDSC(I,J) IS THE NUMBER OF
C                PARTICLES OF THE J SIZE RANGE IN THE I DISC.
C
C
C      SUM THE CONTRIBUTIONS OF THE DISC SOURCES TO THE
C      OPTICALLY WEIGHTED CONCENTRATION AT X,Y,Z,T
C
       CWIND=0.
       DO 211 I=1,NDSCS
       TOF=T-TDSC(I)
       REFO(1)=XDSC(I)
       REFO(2)=0.
       ROH2=R2DSC(I)
       H=ZDSC(I)
       IF(HORIZ) REFO(1)=REFO(1)*SINTH
       DO 210 J=1,NPRTS
C
C      DETERMINE MOMENTS FOR CURRENT SOURCE DISC AT Z
C
       CALL MOMENT(VGRAV(J),Z,H,TOF,Q,XBAR,SIGW2,SIGP2)
       IF(Q.GT.1.E-20) GO TO 113
C
C      IF Q IS TOO SMALL, ITS CONTRIBUTION IS INGORED
C
       CWNDSC=0.
       GO TO 210
```

```
  113 CONTINUE
      RX2=ROH2+2.*SIGW2
      RY2=ROH2+2.*SIGP2
      IF(HORIZ)GO TO 120
C
C  *****  COMPUTE CONCENTRATION AT (X,Y,Z,T)
C
      ARG=-(X-REFO(1)-XBAR)**2/RX2
      IF(ABS' RG).GT.30.)GO TO 150
      CWNDSC (Q/RTPI/SQRT(RX2))*EXP(ARG)
      ARG=-(Y-REFO(2))**2/RY2
      IF(ABS(ARG).GT.30.)GO TO 150
      CY=EXP(ARG          )/RTPI/SQRT(RY2)
      CWNDSC=QDSC(I,J)*CWNDSC*CY
      GO TO 150
C
C     COMPUTE CONCENTRATION ALONG LINE-OF-SIGHT SPECIFIED BY X,Z
C
  120 CONTINUE
      REF(1)=REFO(1)
      REFF2=RX2*SINTH2+RY2*COSTH2
      ARG=-(X-REF(1)-XBAR*SINTH)**2/REFF2
      IF(ABS(ARG).GT.30.)GO TO 150
      CWNDSC=EXP(ARG)                   /SQRT(REFF2)/RTPI
      CWNDSC=CWNDSC*Q*QDSC(I,J)
  150 CONTINUE
      CWIND=CWIND+CWNDSC
  210 CONTINUE
  211 CONTINUE
      RETURN
      END
C
C ********************** SUBFILE 9 ****************************
C
      SUBROUTINE CONVRT(T)
      IMPLICIT INTEGER*4 (I-N)
      REAL M,N
      LOGICAL HORIZ
      COMMON/BUOYCL/RSPH,DELT,VXSPH,VZSPH,XCMSPH,ZCMSPH,
     1 SPHNS(3),RISTIM
      COMMON/PRTINF/ RO,VGRAV( ),NPRTS
      COMMON /WNDPRM/DXZO,DYXO,DZO,UO,M,N,ZINV
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI
      COMMON /MODE/ HORIZ
      COMMON/DISCS/NDSCS,TDSC(10),XDSC(10),ZDSC(10),R2DSC(10),
     1 QDSC(10,3)
      COMMON /CLOCK/ TIME,TWIND
C ****************************************************************
```

```
C
C
C      PURPOSE
C
C        TO CONVERT THE CURRENT BUOYANT CLOUD INTO DISC SOURCES FOR
C        USE BY THE WIND DISPERSION MODEL
C
C      INPUT
C
C        T         THE TIME IN SECONDS AFTER DETONATION
C
C      OUTPUT
C
C
C      SUBROUTINES CALLED
C
C        MOMENT    COMPUTES ZERO ORDER MOMENT AND INTERPOLATES FROM
C                  TABLE OF HIGHER ORDER MOMENTS.
C
C        CALLED BY FUNCT,DUSTCL
C
C *****************************************************************
C
C
C     COMMON /BUOYCL/
C
C       RSPH     RADUIS OF THE SPHERICAL BUOYANT CLOUD IN METERS
C       DELT     TEMPERATURE EXCESS OF CLOUD OVER AMBIENT ATMOSPHERE
C                AT SAME HEIGHT IN DEGREES KELVIN
C       VXSPH    X COMPONENT OF CLOUD VELOCITY IN METERS/SEC
C       VZSPH    Z COMPONENT OF CLOUD VELOCITY IN METERS/SEC
C       XCMSPH   X COORDINATE OF CENTER OF SPHERE IN METERS
C       ZCMSPH   Z COORDINATE OF CENTER OF SPHERE IN METERS
C       SPHNS    SINGLY DIMENSIONED ARRAY.  SPHNS(I) IS THE NUMBER
C                OF PARTICLES OF THE I SIZE RANGE IN THE SPHERE.
C       RISTIM   THE TIME IN SECONDS AFTER DETONATION THAT THE CLOUD
C                HAS RISEN BUOYANTLY
C
C     COMMON /PRTINF/
C
C       RO       INITIAL RADIUS OF THE CLOUD IN METERS
C       VGRAV    SINGLY DIMENSIONED ARRAY.  VGRAV(I) IS THE OPTICALLY
C                WEIGHTED AVERAGE SETTLING VELOCITY FOR PARTICLES IN THE
C                I SIZE RANGE
C       NPRTS    THE NUMBER OF PARTICLE SIZE RANGES
C
C     COMMON /DISCS/
C
```

```
C       NDSCS   THE NUMBER OF DISC SOURCES
C       TDSC    SINGLY DIMENSIONED ARRAY CONTAINING THE TIME OF RELEASE
C               OF THE DISC SOURCES
C       XDSC    SINGLY DIMENSIONED ARRAY CONTAINING THE X COORDINATE
C               OF THE CENTER OF THE DISC SOURCES
C       ZDSC    SINGLY DIMENSIONED ARRAY CONTAINING THE Z COORDINATE
C               OF THE CENTER OF THE DISC SOURCES
C       R2DSC   SINGLY DIMENSIONED ARRAY CONTAINING THE SQUARE OF THE
C               RADII OF THE DISC SOURCES
C       QDSC    DOUBLY DIMENSIONED ARRAY.   QDSC(I,J) IS THE NUMBER OF
C               PARTICLES OF THE J SIZE RANGE IN THE I DISC.
C
      NSPH=5
      DEL=2./FLOAT(NSPH)
      A2=3./FLOAT(NSPH)**2
      A3=2./FLOAT(NSPH)**3
      DELZ=DEL*RSPH
      HREF=ZCMSPH-RSPH-.5*DELZ
      R2=RSPH*RSPH
      DO 50 I=1,NSPH
      IDSC=NDSCS+I
      ZDSC(IDSC)=HREF+FLOAT(I)*DELZ
      HMZ=ZDSC(IDSC)-ZCMSPH
      RD2=R2-HMZ*HMZ
      VFRAC=A2*(FLOAT(2*I-1))-A3*FLOAT(3*I*(I-1)+1)
      DO 5 IPRT=1,NPRTS
      QDSC(IDSC,IPRT)=VFRAC*SPHNS(IPRT)
    5 CONTINUE
      A=ALOG(QDSC(IDSC,1)/VISEXT/DELZ/SQRT(RD2)/RTPI**2)
      IF(A.GT.1.)GO TO 210
      D=1.
      GO TO 230
  210 D=A
      DO 220 IT=1,5
      D=(A-1.+ALOG(D))*D/(D-1)
  220 CONTINUE
  230 R2DSC(IDSC)=RD2/D
      GAPTIM=-DELZ*DELZ/D/4./(DZ0*ZDSC(IDSC)**N)
      XDSC(IDSC)=U0*ZDSC(IDSC)**M * GAPTIM +XCMSPH
      TDSC(IDSC)=T+GAPTIM
   50 CONTINUE
      NDSCS=NDSCS+NSPH
      TWIND=T
      RETURN
      END
C           BUOYANTLY RISING DUST CLOUD FOR DIRTRAN-I CODE
C
C ********************* SUBFILE 10 ******************************
```

```
C
      SUBROUTINE RISE(TPRES,TNEXT,DEL)
      IMPLICIT INTEGER*4 (I-N)
      REAL M,NDIF
      DIMENSION WK(12,6)
      COMMON/BUOYCL/ Y(6),SPHNS(3),RISTIM
      COMMON /WNDPRM/ DXZ0,DYX0,DZ0,U0,M,NDIF,ZINV
      COMMON /CLOCK/ TIME,TWIND
      COMMON /IOUNIT/ IOIN,IOOUT,ISPTPF,LOUNIT,NDIRTU,NBSCAT
      DATA HMIN,ACCURC,WK,N,ND/.001,.001,72*0.,12,12/
C ***********************************************************************
C
C
C     PURPOSE
C
C     THIS ROUTINE CALLS A RUNGE KUTTA ROUTINE TO INTEGRATE IN TIME
C     THE EQUATIONS FOR THE RISE OF A BUOYANT CLOUD BEGINNING AT TPRES
C     ENDING AT TNEXT UNLESS THE CONDITION FOR SWITCHING TO THE WIND
C     DISPERSION MODEL IS ENCOUNTERED IN WHICH CONVRT IS CALLED.
C     SEE SUBROUTINE DIFEQ FOR THE DEFINITIONS OF Y(I).
C
C
C      ARGUMENTS
C
C      TPRES     AS INPUT TPRES IS THE INITIAL TIME OF THIS SEGMENT OF
C                INTEGRATION AND IS RETURNED WITH THE VALUE OF THE LAST
C                SUCCESSFUL INTEGRATION STEP.
C      TNEXT     THE ENDTPOINT OF THE TIME INTERVAL WHICH IS INPUT.
C
C     REQUIRED SUBROUTINES
C
C          RKM       A RUNGE-KUTTA-MERSON INTEGRATION ROUTINE
C
C          CONVRT  A SUBROUTINE WHICH CONVERTS THE CURRENT BUOYANT
C                  DUST CLOUD TO A NUMBER OF DISC SOURCES FOR THE
C                  WIND DISPERSION MODEL.  A GAP TIME DURING WHICH THE
C                  BUOYANT MODEL IS CONTINUED IS COMPUTED.
C     CALLED BY DUSTCL
C
C ***********************************************************************
      IF(TNEXT.GT.TWIND)GO TO 999
      T2=TPRES
C
C     PERFORM INTEGRATION IN SEGMENTS OF TIME
C
   10 DO 20 NT=1,40
      T1=T2
      T2=1.2*T1
```

```
      IF(T2.LE.0.)T2=.5
      IF(T2.GT.TNEXT)T2=TNEXT
      IF(Y(1)+Y(6).GE.ZINV)GO TO 200
      IF (DEL.LT.HMIN)DEL=HMIN
      CALL RKM(N,T1,T2,Y,HMIN,DEL,ACCURC,WK,ND)
C
C     CHECK TO SEE IF CLOUD GROWTH IS DOMINATED BY WIND DIFFUSION
C     OVER BUOYANT RISE BY COMPARING WIND DIFFUSIVITY, DIFW, TO
C     THE EFFECTIVE BUOYANT DIFFUSIVITY, DIFB.
C
    5 DIFB=ABS(.25*Y(1)*Y(4))
      DIFW=DZ0*(Y(6)+Y(1))**NDIF
      IF(DIFB.GT.DIFW)GO TO 15
      CALL CONVRT(T2)
      GO TO 200
   15 CONTINUE
      IF(T2.GE.TNEXT)GO TO 200
      IF(T2.GT.300.)GO TO 99
   20 CONTINUE
   99 WRITE(IOOUT,98)
   98 FORMAT(55H *** DIRTRAN-I ERROR - 5 MINUTE CUT-OFF ON BUOYANT RISE
     1)
      STOP
  200 TPRES=T2
      RISTIM=TPRES
  999 RETURN
      END
C
C ********************  SUBFILE 11  ***********************************
C
      SUBROUTINE DIFEQ(N,T,Y,YP)
      IMPLICIT INTEGER*4 (I-N)
      DIMENSION Y(N),YP(N)
      COMMON/PRTINF/ROCL,VGRAV(3),NPRTS
      COMMON /TMPPRM/T0,T1,TM
      COMMON/WNDPRM/DXZ0,DYX0,DZ0,U0,UM,DN,ZINV
      COMMON /BURST/ ACCEL,TBURST
      DATA ALPHAK/.25/
C *********************************************************************
C
C
C     PURPOSE
C
C     DIFEQ CONTAINS THE PARTIAL DIFFERENTIAL EQUATIONS FOR THE
C     RISE OF A BUOYANT CLOUD WHICH ARE USED BY SUBROUTINE RKM.
C
C
C     INPUT
```

```
C
C     N              THE NUMBER OF DEPENDENT VARIABLES
C     T              THE INDEPENDENT VARIABLE, I.E. TIME
C    Y(1)            RADIUS OF CLOUD
C    Y(2)            CLOUD TEMPERATURE MINUS SURROUNDING TEMPERATURE
C    Y(3)            CLOUD VELOCITY IN WIND DIRECTION
C    Y(4)            VERTICAL VELOCITY OF CLOUD
C    Y(5)            X-COORDINATE OF CENTER OF MASS FOR THE CLOUD
C    Y(6)            THE HEIGHT OF THE CLOUD C.O.M.
C
C
C     OUTPUT
C
C     YP   AN ARRAY CONTAINING COMPUTED DERIVATIVES OF THE DEPENDENT
C          VARIABLES WITH RESPECT TO THE INDEPENDENT VARIABLE.
C
C
C
C     REQUIRED FUNCTIONS
C
C                NONE
C
C
C     CALLED BY RKM
C
C ********************************************************************
      IF(T.LT.TBURST)GO TO 200
      DELT=T1*Y(6)**TM
      TA=T0+DELT
      DTADZ=TM*DELT/Y(6)
      VW=U0*Y(6)**UM
      VR=SQRT((Y(3)-VW)*(Y(3)-VW)+Y(4)*Y(4))
C
C     TA         THE AMBIENT ATMOSPHERIC TEMPERATURE AT CLOUD HEIGHT
C     DTADZ      THE TEMPERATURE GRADIENT AT CLOUD HEIGHT
C     VW         THE WIND VELOCITY AT CLOUD HEIGHT
C     VR         THE RELATIVE VELOCITY OF THE CLOUD WITH RESPECT TO WIND
C     TR         THE RATIO OF CLOUD TEMPERATURE TO AMBIENT TEMPERATURE
C
      TR=Y(2)/TA
C
C     CALCULATE ARVOL, THE SURFACE AREA TO VOLUME RATIO
C     WHICH DEPENDS ON THE NUMBER AND PLACEMENT OF CHARGES
C
      ARVOL=3./Y(1)
   10 CONTINUE
      YP(1)=ALPHAK*VR
      YP(2)=-(1.+TR)*ARVOL*Y(2)*YP(1)-Y(4)*(DTADZ)
```

```
      YP(3)=ARVOL*(VW-Y(3))*YP(1)
      YP(4)=9.8*TR-ARVOL*Y(4)*YP(1)
      YP(5)=Y(3)
      YP(6)=Y(4)
      GO TO 999
  200 N=6
      DO 210 I=1,N
      YP(I)=0.
  210 CONTINUE
      YP(4)=ACCEL
      YP(5)=Y(3)
      YP(6)=Y(4)
  999 RETURN
      END
C
C *******************  SUBFILE 12  ******************************
C
      SUBROUTINE CLDIM(CNTRD,HEIGHT,CENWTH,SPCHT,SPCWTH,NCPTS,CPTS5,
     1 NERR)
      IMPLICIT INTEGER*4 (I-N)
C ********************************************************************
C
C
C     PURPOSE
C
C     CLDIM CALCULATES FIVE CONTOUR POINTS AND CLOUD DIMENSIONS AS
C     SEEN FROM THE SPECIFIED OBSERVER POSITION.  CLDIM REQUIRES CLOUD
C     PARAMETERS FROM THE BUOYANT RISE STAGE OF CLOUD DEVELOPMENT WHICH
C     ARE SUPPLIED IN COMMON STORAGE /BUOYCL/ AND /PRTINF/ AS WELL AS
C     VIEWING GEOMETRY WHICH IS SUPPLIED IN COMMON /GEOM/.  SPCHT IS
C     REQUIRED INPUT IN THE ARGUMENTS.  ALL OUTPUTS ARE ARGUMENTS.
C
C
C     INPUT
C
C     SPCHT      THE SPECIFIED HEIGHT AT WHICH THE WIDTH OF THE CLOUD
C                IS DESIRED. (METERS)
C
C
C
C     OUTPUT
C
C     CNTRD      A SINGLY DIMENSIONED ARRAY OF LENGTH 2 WHICH CONTAINS
C                THE HORIZONTAL AND VERTICAL COORDINATES, RESP., OF THE
C                CLOUD CENTROID.  (METERS)
C     HEIGHT     THE HEIGHT OF THE CLOUD IN METERS
C     CENWTH     THE WIDTH OF THE CLOUD AT THE CENTROID HEIGHT IN METERS
C     SPCWTH     THE WIDTH OF THE CLOUD AT THE SPECIFIED HEIGHT (METERS)
```

```
C     NCPTS     THE NUMBER OF CONTOUR POINTS (=5)
C     CPTS      A DOUBLY DIMENSIONED ARRAY OF SIZE (2,N),N.GE.5, WHICH
C               CONTAINS THE HORIZONTAL AND VERTICAL COORDINATES OF
C               THE FIVE CONTOUR POINTS. (METERS)
C
C
C     REQUIRED SUBROUTINES
C
C        SCONF    MAIN ROUTINE FOR CONTOUR TRACING ALGORITHM.  USED
C                 WITH WIND MODEL.
C
C     CALLED BY DUSTCL
C
C ***********************************************************************
      DIMENSION CNTRD(2),CPTS(2,200),CPTS5(2,5)
      LOGICAL HORIZ,NOCONT
      COMMON /BUOYCL/RSPH,DELT,VX,VZ,XCM,ZCM,SPHNS(3),TIM
      COMMON /PRTINF/RO,VGRAV(3),NPRTS
      COMMON /GEOM/COSTH2,SINTH,SINTH2,VISEXT,RTPI
      COMMON /MODE/ HORIZ
      COMMON /CLOCK/ T,TWIND
      COMMON/WNDPRM/DXZO,DYXO,DZO,UO,UM,DN,ZINV
      COMMON/DISCS/NDSCS,TDSC(10),XDSC(10),ZDSC(10),R2DSC(10),
     1 QDSC(10,3)
      COMMON /SPECS/ RES,STEP,TANT,CON
      EXTERNAL FUNCT,GFUN
      DATA      VISEXT,ZMIN,RES,TANT /.1,0.,.4,.1/
      HORIZ=.TRUE.
      CON=ALOG(VISEXT)
      CPTS5(2,1)=SPCHT
      CPTS5(2,5)=SPCHT
      U=U0*CPTS5(2,1)**UM
      CPTS5(1,1)=T*U*SINTH
      CPTS5(1,5)=CPTS5(1,1)
      NSERCH=-1
      STEP=20.
      CALL CLIMB(FUNCT,GFUN,CPTS5,FP1,NSERCH,NOCONT)
      NSERCH=1
      STEP=20.
      CALL CLIMB(FUNCT,GFUN,CPTS5(1,5),FP1,NSERCH,NOCONT)
      SPCWTH=CPTS5(1,5)-CPTS5(1,1)
      NCPTS=5
      IF(T.LE.TWIND)GO TO 50
      IND=NDSCS-2
      CPTS(2,1)=ZDSC(IND)
      CPTS(1,1)=(XDSC(IND)+(T-TDSC(IND))*U0*ZDSC(IND)**UM)*SINTH
      STEP=20.
      CALL SCONF(FUNCT,ZMIN,NCPTS,CPTS,NERR)
```

```
C        IF(NERR.NE.0) GO TO 999
        CALL PTSPEC(NCPTS,CPTS,XHIGH,HEIGHT,XC1,CNTRD,XC2,CENWTH)
        GO TO 100
   50 CNTRD(1)=XCM*SINTH
        CNTRD(2)=ZCM
        XHIGH=CNTRD(1)
        HEIGHT=ZCM+RSPH
        CENWTH=2.*(RSPH)
  100 CPTS5(1,2)=CNTRD(1)-CENWTH/2.
        CPTS5(1,4)=CNTRD(1)+CENWTH/2.
        CPTS5(1,3)=XHIGH
        CPTS5(2,2)=CNTRD(2)
        CPTS5(2,3)=HEIGHT
        CPTS5(2,4)=CNTRD(2)
        NCPTS=5
  999 RETURN
        END
C       ROUTINE FOR TRACING A CONTOUR OF A FUNCION OF TWO VARIABLES
C
C ******************  SUBFILE 13  **********************************
C
C
C        SCONF IS THE CONTROLLING ROUTINE FOR THE CONTOUR TRACING
C    ALGORITHM.  THE FUNCTION WHOSE CONTOUR IS DESIRED MUST BE A
C    CONTINUOUS, REAL VALUED FUNCTION OF TWO VARIABLES.
C        THE CONTOUR FOUND IS ONE CONTINUOUS CLOSED CONTOUR OR
C    ONE CONTINOUS CURVE BEGINNING AND ENDING ON THE SPECIFIED
C    BOUNDARY. THERE MAY BE OTHER PIECES TO THE CONTOUR.
C
C   INPUT
C
C         FUNCT    THE FUNCTION WHOSE CONTOUR IS TO BE FOUND
C         CON      THE CONTOUR LEVEL.
C         YMN      LOWER BOUND FOR THE SECOND COORDINATE
C         RES      THE RESOLUTION LENGTH.
C         DELTA    THE INITIAL STEP SIZE WHEN LOOKING ALONG THE
C                  GRADIENT TO FIND A POINT ON THE CONTOUR.
C         THETAN   THE TANGENT OF THE MAXIMUM ALLOWABLE ROTATION ANGLE
C                  BETWEEN SUCCESSIVE LINE SEGMENTS OF THE POLYGONAL
C                  CONTOUR.
C         MAXDIM   THE NUMBER OF POINTS FOR WHICH STORAGE HAS BEEN
C                  ALLOCATED IN THE ARRAY CP.
C         CP       A DOUBLY DIMENSIONED ARRAY TO BE FILLED WITH THE
C                  COORDINATES OF THE CONTOUR POINTS.  CP(I,J) IS THE
C                  I-TH COORDINATE, I=1,2 , OF THE J-TH POINT OF THE
C                  CONTOUR.  UPON CALLING SCONF, CP(I,1),I=1,2 , SHOULD
C                  BE THE COORDINATES OF THE BEST GUESS AS TO WHERE THE
```

```
C               CONTOUR IS LOCATED.
C
C
C  OUTPUT
C
C       CP      A DOUBLY DIMENSIONED ARRAY FILLED WITH THE
C               COORDINATES OF THE CONTOUR POINTS.  CP(I,J) IS THE
C               I-TH COORDINATE, I=1,2 , OF THE J-TH POINT OF THE
C               CONTOUR.
C       MAXDIM  THE NUMBER OF POINTS REPORTED IN CP.
C       ERROR   A NUMERICAL FLAG TELLING WHAT TYPE OF ERROR
C               OCCURRED WHILE RUNNING, IF ANY.
C               MEANING OF ERROR CODE-
C                -1=>ARRAY FILLED COMPLETELY
C                 0=>CONTOUR CLOSED OR MET BOUNDARIES, NO PROBLEM.
C                 1=>NO CONTOUR FOUND
C                 2=>PTFIND UNABLE TO FIND NEXT POINT.  CHECK FOR
C                    KINK OR DISCONTINUITY OF THE FUNCTION.
C
C
C  CALLED SUBROUTINES.
C
C       CLIMB   LOCATES FIRST POINT OF CONTOUR OR THAT CONTOUR
C               DOESN'T EXIST.
C       PTFIND  FINDS ADDITIONAL POINTS ON THE CONTOUR GIVEN AT LEAST
C               ONE.
C
C
C  LOCAL VARIABLES
C
C       LEFT    A LOGICAL WHICH IS .TRUE. FOR COUNTERCLOCKWISE CONTOUR
C               SEARCH AND .FALSE. FOR CLOCKWISE SEARCH
C       TMP     TEMPORARY STORAGE USED FOR SWAPPING POSITIONS IN
C               THE ARRAY OF CONTOUR POINTS.
C       NOCONT  A LOGICAL ERROR FLAG MEANING EITHER THE CONTOUR
C               DOES NOT EXIST OR IS TOO SMALL TO BE TRACED
C       ENDCON  A NUMERICAL FLAG INDICATING THE STATUS OF THE
C               CONTOUR POINTS TRACED WITH VALUES:
C                   -1     THE ARRAY CP IS FILLED
C                    0     THE CONTOUR IS CLOSED
C                    1     THE CONTOUR HAS MET A BOUNDARY
C                    2     THE REQUIRED STEP SIZE HAS BECOME LESS
C                          THAN THE RESOLUTION
C
C *********************************************************************
      SUBROUTINE SCONF(FUNCT,YMN,MAXDIM,CP,ERROR)
      IMPLICIT INTEGER*4 (I-N)
      EXTERNAL FUNCT,GFUN
```

```
      INTEGER ENDCON,ERROR
      LOGICAL NOCONT,ERR,LEFT
      DIMENSION                  TMP(2)
      DIMENSION CP(2,200)
      COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
      COMMON/LIMIT/YMIN,FMIN
      COMMON /IOUNIT/ IOIN,IOOUT,ISPTPF,LOUNIT,NDIRTU,NBSCAT
      DATA FMIN/-29./
      MAXDIM=200
      ERROR=0
      YMIN=YMN
C
C FINDING THE POINT ON THE CONTOUR,OR IF A CONTOUR EVEN EXISTS.
C
      NSERCH=0
      CALL CLIMB(FUNCT,GFUN,CP,FCP,NSERCH,NOCONT)
C *** APPROPRIATE ACTION IF THE CONTOUR DOES NOT EXIST,
      IF(NOCONT)GO TO 99
      L=1
      LEFT=.TRUE.
      CALL PTFIND(LEFT,GFUN,MAXDIM,CP,L,ENDCON)
C
C ** THE NEXT TWO DO LOOPS ARE FOR SWITCHING THE POSITIONS
C     OF THE POINTS IN THE ARRAY AROUND FOR EASE IN
C     WORKING WITH LATER ON.
C
      MID=L/2
      DO 17 I=1,MID
      J=L-I+1
      DO 19 K=1,2
      TMP(K)=CP(K,I)
      CP(K,I)=CP(K,J)
      CP(K,J)=TMP(K)
   19 CONTINUE
   17 CONTINUE
C ** IS THE CONTOUR CLOSED OR IS THE ARRAY FILLED
      IF(ENDCON)97,76,27
   27 IF(ENDCON.EQ.2)ERROR=2
      IF(ENDCON.EQ.2)WRITE(IOOUT,796)
      LEFT=.FALSE.
      CALL PTFIND(LEFT,GFUN,MAXDIM,CP,L,ENDCON)
      MAXDIM=L
      IF(ENDCON)97,76,95
   76 CONTINUE
      MAXDIM=L+1
      CP(1,MAXDIM)=CP(1,1)
      CP(2,MAXDIM)=CP(2,1)
      GO TO 98
```

```
   95 IF(ENDCON.EQ.1)GO TO 98
   96 ERROR=2
      MAXDIM=L
      WRITE(IOOUT,796)
  796 FORMAT(53H *** DIRTRAN-I ERROR - CONTOUR TRACING ROUTINE  STUCK /
     1     56H                          CHECK FOR DISCONTINUITY OF FUNCTION )
      GO TO 999
   97 ERROR=-1
      WRITE(IOOUT,797)
  797 FORMAT(51H *** DIRTRAN-I ERROR - ARRAY CPTS NOT LARGE ENOUGH   )
      GO TO 999
   98 ERROR=0
      GO TO 999
   99 ERROR=1
      WRITE(IOOUT,799)
  799 FORMAT(51H *** DIRTRAN-I WARNING - CLOUD CONTOUR NOT FOUND     /
     1        49H                           MAY HAVE DISSIPATED         )
  999 RETURN
      END
C
C ***********************  SUBFILE 14  *****************************
C
      FUNCTION GFUN(S)
      IMPLICIT INTEGER*4 (I-N)
C
C       GFUN IS THE RESTRICTION OF THE TWO DIMENSIONAL FUNCTION, F, TO
C               A LINE. I.E. FORM G(S)=F(X,Y), WHERE (X,Y)=BASE+S*DIR.
C
      EXTERNAL FUNCT
      DIMENSION                 P(2)
      COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
      CALL VSUM(BASE,DIR,S,P)
      GFUN=FUNCT(P(1),P(2))
      RETURN
      END
C
C **************************  SUBFILE 15  ************************
C
C      THIS MODULE IS A SUBROUTINE THAT FINDS A POINT ON A CONTOUR
C BY FINDING THE GRADIENT VECTOR AT THAT POINT AND MARCHING ALONG
C IT UNTIL IT FINDS ITSELF IN A REGION GREATER THAN THE CONTOUR LEVEL.
C AT WHICH POINT IT MARCHES HORIZONTALLY, HALVING THE STEP SIZE
C UNTIL THE CONTOUR IS REACHED WITHIN SPECIFIED RESOLUTION.
C     IN ADDITION IT WILL DETERMINE IF A CONTOUR EXISTS.
C
C
C    ARGUMENTS PASSED.
C
```

```
C     INPUT
C         FUNCT-THE FUNCTION(X,Y) ALSO GIVEN IN EXTERNAL.
C         P1-THE STARTING POINT.
C
C
C    OUTPUT
C
C         P1     - THE POINT ON THE CONTOUR OR THE POINT AT WHICH
C                  THE FUNCTION REACHES A MAXIMUM BELOW THE CONTOUR
C                  LEVEL
C         FP1    - THE VALUE OF THE FUNCTION AT P
C         NOCONT-THE ERROR FLAG.
C                  F-NO PROBLEM
C                  T-NO CONTOUR FOUND.
C         ERR-ERROR FLAG RETURNED BY 'NTRSCT'
C             F-NO ERROR
C             T-ITERATION DIVERGED OR MAXIMUM SEARCH AREA EXCEDED
C
C     IN ADDITION,IN COMMON ARE..,
C
C         YMIN-THE LOWER LIMIT ON Y.
C         DELTA- THE STEP SIZE,MODIFIED IN THIS SUBROUTINE.
C         CON-THE CONTOUR LEVEL.
C         RES-THE RESOLUTION LENGTH
C
C     OTHER VARIABLES INCLUDE
C         GRAD-THE GRADIENT VECTOR
C         P0-THE CURRENT POINT ON THE GRADIENT.
C         P1-THE POINT ON THE GRADIENT BEING TESTED
C            TO SEE ABOUT CONTOUR EXISTENCE.
C         FP0,FP1-THE FUNCTION VALUES OF P0 AND P1.
C
C     CALLED SUBROUTINES
C
C         GRAD2-FINDS THE GRADIENT VECTOR OF A FUNCTION AT
C                A POINT AND THE SLOPE THERE.
C         UNIT-CALCULATES THE NORM AND MAGNITUDE OF A 2 VECTOR.
C         VSUM-VECTOR SUM OF THE FORM C=A+SB WHERE S IS SCALAR
C               MULTIPLIER OF B.
C
      SUBROUTINE CLIMB(FUNCT,GFUN,P1,FP1,NSERCH,NOCONT)
      IMPLICIT INTEGER*4 (I-N)
      EXTERNAL FUNCT,GFUN
      LOGICAL NOCONT
      DIMENSION GRAD(2),P0(2),P1(2),P(2)
      COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
      COMMON/LIMIT/YMIN,FMIN
      NOCONT=.FALSE.
```

```
      ONEM=-1.0
      IF (NSERCH.EQ.0)GO TO 7
      DELTA=SIGN(DELTA,FLOAT(NSERCH))
      FP1=FUNCT(P1(1),P1(2))
      IF(FP1.LT.CON)GO TO 25
      GO TO 22
    3 CONTINUE
      P0(1)=P1(1)
      P0(2)=P1(2)
      FP0=FP1
C ** FINDING THE UNIT GRADIENT AND THE NEXT POINT ALONG IT.
    4 CALL GRAD2(P0,FUNCT,RES,GRAD,DFDS)
    5 CALL VSUM(P0,GRAD,DELTA,P1)
C ** IS THE POINT HEADING BELOW YMIN **
      IF(P1(2).GE.YMIN)GO TO 7
      P1(2)=YMIN
      CALL VSUM(P1,P0,ONEM,GRAD)
      CALL UNIT(GRAD,GRAD,DELTA)
      IF(ABS(DELTA).LT.RES)GO TO 25
    7 FP1=FUNCT(P1(1),P1(2))
C ** HAS THE CONTOUR BEEN CROSSED **
    8 IF(FP1.GE.CON)GO TO 22
      IF(FP1.GT.FP0)GO TO 3
      DELTA=DELTA/2.
      IF(ABS(DELTA).LT.RES)GO TO 25
      GO TO 5
   25 NOCONT=.TRUE.
      GO TO 99
   22 CONTINUE
C     BEGIN HORIZONTAL SEARCH
      P0(2)=P1(2)
   31 P0(1)=P1(1)
      FP0=FP1
   40 P1(1)=P0(1)+DELTA
      FP1=FUNCT(P1(1),P1(2))
      IF(ABS(DELTA).LT.RES/2.)GO TO 99
      IF(FP1.GE.CON)GO TO 31
      DELTA=DELTA/2.
      GO TO 40
   99 CONTINUE
      RETURN
      END
C *****************************  SUBFILE 16  **********************
C
C
C
C
C  THIS SUBROUTINE CALCULATES THE UNIT GRADIENT VECTOR
```

```
C  OF A FUNCTION AT THE GIVEN POINT USING THE FORMULA
C  PARTIAL DF/DX = (F(X+R,Y)-F(X,Y))/R WHERE R IS
C  SMALL, SIMILARLY FOR PARTIAL DF/DY. IT THEN
C  NORMALIZES THE RESULTANT VECTOR FOR THE UNIT GRADIENT
C  AND FINDS THE SLOPE,WHICH IS THE MAGNITUDE OF
C  OF THE REGULAR GRADIENT.
C
C  ARGUMENTS PASSED
C     PT-THE POINT AT WHICH THE UNIT GRADIENT IS FOUND.
C     FUNCT-THE FUNCTION(X,Y)
C     RES-R
C     GRAD-THE UNIT GRADIENT
C     SLOPE-SLOPE AT PT
C
C  OTHER VARIABLES
C     C00,C10,C01-THE FUNCTION AT THE POINTS F(X,Y),F(X+R,Y),
C         AND F(X,Y+R) RESPECTIVELY.
C
C  SUBROUTINES CALLED
C     UNIT-NORMALIZES A VECTOR AND FINDS ITS MAGNITUDE.
C
      SUBROUTINE GRAD2(PT,FUNCT,RES,GRAD,SLOPE)
      IMPLICIT INTEGER*4 (I-N)
      DIMENSION PT(2),GRAD(2)
      C00=FUNCT(PT(1),PT(2))
      C10=FUNCT(PT(1)+RES,PT(2))
      C01=FUNCT(PT(1),PT(2)+RES)
      GRAD(1)=(C10-C00)/RES
      GRAD(2)=(C01-C00)/RES
      CALL UNIT(GRAD,GRAD,SLOPE)
      RETURN
      END
C
C ********************************  SUBFILE 17  ********************
C
C THIS SUBROUTINE FINDS THE INTERSECTION OF A LINE SPECIFIED BY
C A BASE POINT, BASE, AND A DIRECTON VECTOR, DIR. THE ROUTINE
C DETERMINES A REAL NUMBER, S, SUCH THAT BASE+S*DIR IS THE
C POINT OF INTERSECTION WITH A CONTOUR THROUGH NEWTON-RAPHSON
C ITERATION.
C
C
C INPUT TERMS.
C
C       SLIM-THE LIMITING VALUE OF S, TO PREVENT THE INTERPOLATION
C            FROM BREAKING DOWN IN UNUSUAL CASES AND GOING INTO AN
C            INFINITE LOOP
C        GFUN-FUNCTION OF S EQUIVALENT TO FUNCTION(X,Y).
```

```
C
C
C OUTPUT TERMS.
C
C       S-A SCALAR PARAMETER SPECIFYING POINTS ON THE LINE.
C       P-THE POINT OF INTERSECTION.
C       ERR- AN ERROR FLAG.
C            F-NO PROBLEM
C            T-NO INTERSECTION FOUND WITHIN THE SET LIMITATIONS.
C
C TERMS IN COMMON.
C      BASE- A 2-VECTOR CONTAINING THE COORDINATES OF THE BASE
C            POINT OF THE LINE. BASE MUST BE INITIALIZED BEFORE
C            THE CALL TO NTRSCT.
C      DIR- A UNIT DIRECTION 2-VECTOR FOR THE LINE. DIR ALSO MUST
C           BE INITIALIZED PREVIOUS TO THE CALL.
C      CON-THE CONTOUR LEVEL TO BE FOUND.
C      DFDS- DF/DS WITH F=FUNCTION(X,Y), USED AS AN INITIALIZER
C            FOR THE INTERPOLATION. FIRST ONE MUST BE INITIALIZED
C            BEFORE THE CALL.
C      RES-THE RESOLUTION
C
C  CALLED SUBROUTINES
C
C       VSUM-VECTOR ADDITION SUBROUTINE OF FORM VSUM(A,B,S,C)=
C            C=A+S*B WHERE A,B,C ARE 2-VECTORS AND S SCALAR.
C
C
      SUBROUTINE NTRSCT(S,P,GFUN,SLIM,ERR)
      IMPLICIT INTEGER*4 (I-N)
      EXTERNAL GFUN,FUNCT
      LOGICAL ERR
      DIMENSION P(2)
      COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
      COMMON/LIMIT/YMIN,FMIN
      SOLD=0.0
      ERR=.FALSE.
C
C *** ITER IS THE NUMBER OF ITERATIONS.***
      ITER=0
      FSOLD=GFUN(SOLD)
      IF(FSOLD.LE.FMIN)GO TO 999
      DS=(CON-FSOLD)/DFDS
      S=SOLD+DS
      IF(ABS(S).LT.SLIM)GO TO 110
      DS=SIGN(SLIM/2.,DS)
      S=SOLD+DS
  110 ITER=ITER+1
```

```
      FS=GFUN(S)
      IF(FS.LE.FMIN)GO TO 999
    5 DFDS=(FS-FSOLD)/(S-SOLD)
      DS=   (CON-FS)/DFDS
      SOLD=S
      FSOLD=FS
      S=S+DS
      IF(ABS(DS).LT.RES)GO TO 998
      IF(ABS(S).GT.SLIM)GO TO 999
      IF(ITER.GT.9)GO TO 999
      GO TO 110
  998 CALL VSUM(BASE,DIR,S,P)
      GO TO 1000
  999 ERR=.TRUE.
 1000 RETURN
      END
C
C **************************  SUBFILE 18  *************************
C
C        PTFIND DETERMINES A STRING OF POINTS ON THE CONTOUR MARCHING
C     EITHER CLOCKWISE (.NOT.LEFT) OR COUNTERCLOCKWISE (LEFT) UNTIL
C     THE CONTOUR CLOSES, A BOUNDARY IS MET, THE ARRAY CPTS IS FILLED,
C     OR AN ERROR OCCURS.  A STARTING POINT MUST BE PROVIDED.  PTFIND
C     BEGINS BY CALCULATING THE TANGENT TO THE CONTOUR WHICH IS
C     PERPENDICULAR TO THE GRADIENT.  SUBSEQUENT POINTS ARE FOUND BY
C     EXTENDING THE TANGENT THROUGH THE TWO MOST RECENTLY DETERMINED
C     POINTS AND SEARCHING IN THE PERPENDUCULAR DIRECTION.  IN CASE
C     OF ERROR, ONE RESTART IS ALLOWED.
C        THE STEP SIZE, DELTA, IS HALVED WHEN THE EXTRAPOLATED POINT
C     EXCEDES BOUNDARY OR THE SEARCH IN THE PERPENDICULAR DIRECTION
C     DID NOT YIELD A POINT WITHIN LIMITS DETERMINED BY CURVATURE.
C     DELTA IS LENGTHENED INVERSELY AS THE LOCAL CURVATURE OF THE CURVE.
C
C  INPUT TERMS
C      GFUN-THE ONE VARIABLE EQUIVALENT TO THE FUNCTION
C           RESTRICTED TO A LINE THROUGH POINT, BASE, IN DIRECTION,DIR
C      MAXDIM-THE MAXIMUM NUMBER OF POINTS IN THE ARRAY.
C      CP-THE ARRAY OF CONTOUR POINTS AS IT INITIALLY IS.
C      L-THE INDEX OF HOW MANY POINTS HAD BEEN FOUND BEFORE.
C
C
C  OUTPUT TERMS
C      CP-CURRENT ARRAY OF CONTOUR POINTS
C       L-CURRENT INDEX OF POINTS. ON RETURN, L IS
C          THE NUMBER OF POINTS FOUND.
C      ENDCON-A NUMERICAL FLAG THAT TELLS
C             WHY THE SUBROUTINE IS ENDING THE SEARCH.
C        ENDCON EQUALS.
```

```
C                 -1=>THE ARRAY CP IS FILLED.
C                  0=>THE CONTOUR IS CLOSED.
C                  1=>THE CONTOUR HAS MET A BOUNDARY.
C                  2=>REQUIRED STEP SIZE BECAME LESS THAN RESOLUTION
C
C  OTHER PERTINENT VARIABLES IN COMMON...
C     CON-THE CONTOUR LEVEL
C     BASE-THE POINT ALONG THE TANGENT.
C     DFDS-THE SLOPE OF THE CURVE.  MODIFIED IN 'NTRSCT'
C     RES-THE RESOLUTION LENGTH
C     DELTA-THE STEP SIZE ALONG THE TANGENT,MODIFIED HERE.
C     THETAN-THE TANGENT OF THETA, THE ANGLESPREAD OF PERMISSIBLE
C            SEARCH AREA. THE SMALLER THETAN IS, THE SMOOTHER THE
C            POLYGON OF CALCULATED POINTS WILL BE.
C     YMIN-LOWER BOUND FOR SECOND COORDINATE
C
C  OTHER VARIABLES
C     P1-CURRENT POINT ON CONTOUR
C     PNEXT-NEXT POINT ON CONTOUR,NOT YET APPROVED BY 'ENDTST'.
C     FULTAN-THE APPROXIMATION OF THE TANGENT AT PNEXT BY
C            THE VECTOR FROM P1 TO PNEXT.
C     TAN-THE UNIT VECTOR OF FULTAN
C     DIST-THE MAGNITUDE OF FULTAN,USED IN 'ENDTST'.
C     S-THE DISTANCE ALONG THE GRADIENT FROM BASE TO
C       A POINT ON THE CONTOUR.
C     SMAX-THE LARGEST ALLOWABLE VALUE OF S.
C
C
C
C  SUBROUTINES CALLED.
C
C       VSUM(A,B,S,C)-VECTOR ADDITION OF THE FORM C=A+S*B WHERE
C          A,B,C ARE 2-VECTORS AND S IS SCALAR.
C       PERP(A,B)-A ,B ARE 2-VECTORS AND B IS A VECTOR ROTATED
C       90 DEGREES COUNTERCLOCKWISE.
C       NTRSCT(S,P,GFUN,SLIM,IC,ERR)-FINDS THE INTERSECTION OF A
C          LINE DRAWN FROM BASE(IN COMMON) ALONG DIR,THE DIRECTION
C          VECTOR(IN COMMON), WITH THE CONTOUR LEVEL (IN COMMON).
C          S IS THE DISTANCE FROM BASE TO THE CONTOUR,P IS THE
C          POINT OF INTERSECTION, GFUN IS GFUN(S) AND IS A ONE
C          VARIABLE EQUIVALENT OF THE FUNCTION WHOSE CONTOUR
C          IS BEING SOUGHT, SLIM IS THE MAXIMUM ALLOWABLE
C          VALUE OF S, AND ERR IS THE ERROR FLAG(LOGICAL).
C          IC IS A LOGICAL VARIABLE WHERE T MEANS FOR
C          NTRSCT TO STOP AFTER 5 ITERATIONS.
C       ENDTST-F(P0,P,TAN,DIST,SMAX,I,M,ENDCON)-FINDS IF
C          THE CONTOUR IS CLOSED, HAS RUN INTO THE Y BOUNDARY
C          OR IF THE ARRAY OF CONTOUR POINTS IS FILLED.
```

```
C      UNIT(A,B,S)-A,B ARE 2-VECTORS AND B IS THE NNORM OF A
C         AND S IS THE MAGNITUDE OF A.
C
C
C
       SUBROUTINE PTFIND(LEFT,GFUN,MAXDIM,CP,L,ENDCON)
       IMPLICIT INTEGER*4 (I-N)
       EXTERNAL GFUN,FUNCT
       INTEGER ENDCON
       LOGICAL ERR,ONCE,LEFT
       DIMENSION     CHORD(2),TAN(2)
       DIMENSION CP(2,MAXDIM)
       COMMON/LINE/BASE(2),DIR(2),DFDS/SPECS/RES,DELTA,THETAN,CON
       COMMON/LIMIT/YMIN,FMIN
       ERR=.FALSE.
       RES2=RES/2.
C
C      INITIALIZE SEARCH FOR CONTOUR POINTS MAKING USE OF THE FACT
C      THAT THE TANGENT TO THE CONTOUR IS PERPENDICULAR TO THE
C      GRADIENT OF THE FUNCTION.
C
     9 ONCE=.TRUE.
       CALL GRAD2(CP(1,L),FUNCT,5.*RES,DIR,DFDS)
       CALL PERP(DIR,TAN)
       IF(.NOT.LEFT)GO TO 31
       TAN(1)=-TAN(1)
       TAN(2)=-TAN(2)
       DFDS  =-DFDS
    31 DELTA=10.
       TANTH=1.
C *** STEPPING DELTA ALONG THE TANGENT ***
    10 CALL VSUM(CP(1,L),TAN,DELTA,BASE)
C *** IS BASE BELOW THE MINIMUM PERMISSIBLE Y***
       IF(BASE(2).GT.YMIN)GO TO 29
       DELTA=DELTA/2.
       IF(ABS(DELTA).LT.RES2)GO TO 91
       GO TO 10
C
C      DETERMINE THE RANGE OF SEARCH PERPENDICULAR TO THE EXTENDED
C      TANGENT, SMAX, AND CALL NTRSCT TO LOCATE THE CONTOUR POINT
C
    29 SMAX=AMIN1(ABS(DELTA*TANTH+RES),ABS((BASE(2)-YMIN)/DIR(2)))
       DFDS0=DFDS
    30 CALL NTRSCT(S,CP(1,L+1),GFUN,SMAX,ERR)
C *** HAS THE CONTOUR BEEN FOUND WITHIN LIMITING CONDITIONS**
       IF(.NOT.ERR)GO TO 32
       DFDS=DFDS0
       DELTA=DELTA/2.
```

```
C
C      CHECK TO SEE IF STEP SIZE, DELTA, IS BELOW MINIMUM, RES2.
C      IF SO, ALLOW ONE RESTART THEN ABORT
C
       IF(ABS(DELTA).GE.RES2)GO TO 10
       IF(ONCE)GO TO 92
       GO TO 9
C      DETERMINE TANGENT FOR NEWLY FOUND POINT
    32 CALL VSUM(CP(1,L+1),CP(1,L),-1.,TAN)
       CALL UNIT(TAN,TAN,DIST)
       CALL PERP(TAN,DIR)
C ** IS THE CONTOUR CLOSED **
C     WHAT THIS NEXT CHECK DOES IS TO LOOK WITHIN A BOX
C     DELTA BY 2*SMAX, WHERE P AND THE PREVIOUS POINT ARE AT EACH
C     END OF THE LINE SEGMENT BETWEEN THE MIDPOINTS OF THE TWO
C     OPPOSITE SIDES 2*SMAX LONG, TO SEE IF THE STARTING POINT
C     IS WITHIN.
       CALL VSUM(CP,CP(1,L),-1.,CHORD)
       VAR1=DOTPRD(DIR,CHORD)
       VAR2=DOTPRD(TAN,CHORD)
       IF(ABS(VAR1).GE.RES*2.)GO TO 35
       IF((VAR2.LT.DIST).AND.(VAR2.GT.0.0))GO TO 90
C      BEGIN SEARCH FOR NEXT POINT
    35 L=L+1
       IF(CP(2,L).LT.RES2+YMIN)GO TO 91
       IF(L.GE.MAXDIM)GO TO 89
       DELTA=AMIN1(10.,DELTA*(SMAX+2.*RES)/((ABS(S)+RES)*2.),DELTA*2.)
       ONCE=.FALSE.
       TANTH=THETAN
       GO TO 10
    89 ENDCON=-1
       GO TO 99
    90 ENDCON=0
       L=L+1
       GO TO 99
    91 ENDCON=1
       GO TO 99
    92 ENDCON=2
   99  RETURN
       END
C
C ************************* SUBFILE 19 ***************************
C
       SUBROUTINE VSUM(A,B,S,C)
       IMPLICIT INTEGER*4 (I-N)
       DIMENSION A(2),B(2),C(2)
C *** C=A+S*B WHERE A,B,C ARE VECTORS AND S IS SCALAR
       DO 14 J=1,2
```

```
   14 C(J)=A(J)+S*B(J)
      RETURN
      END
C
C **********************  SUBFILE 20  **************************
C
      SUBROUTINE PERP(A,B)
      IMPLICIT INTEGER*4 (I-N)
      DIMENSION A(2),B(2)
C *** B IS ROTATED 90 DEGREES COUNTERCLOCKWISE FROM A
      B(1)=-A(2)
      B(2)=A(1)
      RETURN
      END
C
C *************************  SUBFILE 21  ***************************
C
      SUBROUTINE UNIT(A,B,XNORM)
      IMPLICIT INTEGER*4 (I-N)
      DIMENSION A(2),B(2)
C *** B IS THE NORM OF A, AND XNORM IS THE MAGNITUDE
      XNORM=SQRT(A(1)**2+A(2)**2)
      B(1)=A(1)/XNORM
      B(2)=A(2)/XNORM
      RETURN
      END
C
C **************************  SUBFILE 22  ***********************
C
      FUNCTION DOTPRD(A,B)
      IMPLICIT INTEGER*4 (I-N)
      DIMENSION A(2),B(2)
C
C     DOTPRD IS THE SCALER PRODUCT OF A AND B
C
      DOTPRD=A(1)*B(1)+A(2)*B(2)
      RETURN
      END
C
C *************************  SUBFILE 23  ************************
C
C        THIS SUBROUTINE TAKES AN ARRAY OF POINTS OF A CONTOUR
C     AND FINDS THE HIGHEST POINT, THE TWO POINTS ALONG A
C     LINE OF SIGHT (THESE TWO MAY NOT BE IN THE ARRAY AND WILL
C     BE FOUND BY INTERPOLATION FROM RATIOS) AND THEIR CENTROID,
C     THE CENTROID AND THE TWO POINTS WITH THE SAME Y VALUE, THE LENGTH
C     OF THE LINE OF SIGHT, AND THE SHEAR DISTANCE. NOTE- THE POINT ON
C     THE CENTROID LINE WITH THE GREATER X VALUE IS THE LEADING
```

```
C    EDGE POINT.
C
C    INPUT TERMS
C
C       SPCHT-A SPECIFIED HEIGHT AT WHICH THE WIDTH IS DESIRED
C       CP-THE ARRAY OF CONTOUR POINTS
C       L-THE NUMBER OF POINTS IN CP.
C
C
C    OUTPUT TERMS
C
C       ZHI- Z COORDINATE OF HIGHEST POINT ON CONTOUR.
C       XHI- X COORDINATE OF HIGHEST POINT ON CONTOUR.
C       XCL- X COORDINATE OF LEFTMOST POINT ON CONTOUR.
C       XCR- X COORDINATE OF RIGHTMOST POINT ON CONTOUR.
C       CNTRD-THE CENTROID
C       CENWTH- WIDTH AT THE CENTROID HEIGHT
C       XSPCL- X COORDINATE OF LEFTMOST POINT OF CONTOUR AT THE
C                     SPECIFIED HEIGHT, SPCHT
C       XSPCR- X COORDINATE OF RIGHTMOST POINT OF CONTOUR AT THE
C                SPECIFIED HEIGHT
C       SPCWTH- THE WIDTH OF THE CONTOUR AT THE SPECIFIED HEIGHT
C
C    SUBROUTINES CALLED
C
C       INTRP-PERFORMS A SIMPLE INTERPOLATION BY RATIO TO FIND
C             A POINT WITH A GIVEN Y COORDINATE, GIVEN A POINT
C             ON EITHER SIDE. THE CALL IS
C                 CALL INTRP(P1,P2,Y,P) WHERE P1,P2 ARE THE GIVEN
C             POINTS, Y THE GIVEN Y COORD., AND P THE POINT RETURNED.
C
C       IN INTRP... VSUM(A,B,S,C) WHERE A,B,S ARE GIVEN AND
C             C=A+S*B. A,B,C ARE 2-VECTORS AND S, A SCALAR.
C
C
      SUBROUTINE PTSPEC(L,CP,XHI,ZHI,XCL,CNTRD,XCR,CENWTH)
      IMPLICIT INTEGER*4 (I-N)
      DIMENSION CNTRD(2),CP(2,L)
      LOGICAL ERR,BTWN
      BTWN(A,B,C)=(A.LE.B.AND.B.LT.C).OR.(A.GE.B.AND.B.GT.C)
      ERR=.FALSE.
      NHI=1
      NLO=1
      NL=1
      NR=1
      DO 10 J=1,L
      IF(CP(2,J).GT.CP(2,NHI))NHI=J
      IF(CP(2,J).LT.CP(2,NLO))NLO=J
```

```
      IF(CP(1,J).LT.CP(1,NL))NL=J
      IF(CP(1,J).GT.CP(1,NR))NR=J
   10 CONTINUE
      XHI=CP(1,NHI)
      ZHI=CP(2,NHI)
      CNTRD(2)=AMAX1(CP(2,NL),CP(2,NR))
      NCR=NL
      NCL=NR
      DO 20 J=1,L
      IF(.NOT.BTWN(CP(2,J),CNTRD(2),CP(2,J+1)))GO TO 20
      IF(CP(1,J).LT.CP(1,NCL))NCL=J
      IF(CP(1,J).GT.CP(1,NCR))NCR=J
   20 CONTINUE
      CALL INTRP(CP(1,NCL),CP(1,NCL+1),CNTRD(2),XCL)
      CALL INTRP(CP(1,NCR),CP(1,NCR+1),CNTRD(2),XCR)
      CNTRD(1)=(XCL+XCR)/2.
      CENWTH=XCR-XCL
C       IF(.NOT.BTWN(ZLO,SPCHT,ZHI))GO TO 99
C       NCR=NL
C       NCL=NR
C       DO 30 J=1,L
C       IF(.NOT.BTWN(CP(2,J),SPCHT,CP(2,J+1)))GO TO 30
C       IF(CP(1,J).LT.CP(1,NCL))NCL=J
C       IF(CP(1,J).GT.CP(1,NCR))NCR=J
C    30 CONTINUE
C       CALL INTRP(CP(1,NCL),CP(1,NCL+1),SPCHT,XSPCL)
C       CALL INTRP(CP(1,NCR),CP(1,NCR+1),SPCHT,XSPCR)
C       SPCWTH=XSPCR-XSPCL
C       GO TO 999
C    99 ERR=.TRUE.
C       XSPCL=0.0
C       XSPCR=0.0
C       SPCWTH=0.0
  999 RETURN
      END
C
C ************************  SUBFILE 24  ***************************
C
      SUBROUTINE INTRP(P1,P2,Y,X)
      IMPLICIT INTEGER*4 (I-N)
C ******************************************************
C
C       GIVEN POINTS P1 AND P2 AND SECOND COORDINATE, Y,
C       INTRP DETERMINES THE FIRST COORDINATE, X, OF A POINT, (X,Y),
C       WHICH IS ON A LINE DRAWN THROUGH P1 AND P2
C
C ************************************************************
      DIMENSION P1(2),P2(2),P(2),DIF(2)
```

```
      IF(ABS(P2(2)-P1(2)).LT.1.E-30)GO TO 89
      RATIO=(Y-P1(2))/(P2(2)-P1(2))
      ONEM=-1.
      CALL VSUM(P2,P1,ONEM,DIF)
      CALL VSUM(P1,DIF,RATIO,P)
      X=P(1)
      GO TO 99
   89 X=P1(1)
   99 RETURN
      END
C  CALCULATION OF 0-ORDER AND INTERPOLATION OF HIGHER ORDER MOMENTS
C
C **********************  SUBFILE 25  *********************************
C
      SUBROUTINE MOMENT(VGRAV,ZIN,H,TIN,Q,XBAR,SIGW2,SIGP2)
      IMPLICIT INTEGER*4 (I-N)
      REAL M,N,NM
      DIMENSION AL(9),Z(9),T(9),XB(81,4,4),SW(81,4,4),SP(81,4,4),NM(9)
      DIMENSION VAL(16),XVAL(8),W(8),XI(4),IB(4),NTC(4),II(4),X(9,4)
      LOGICAL FIRST
      COMMON /WNDPRM/DXZO,DYXO,DZO,UO,M,N,ZINV
      COMMON /IOUNIT/ IOIN,IOOUT,ISPTPF,LOUNIT,NDIRTU,NBSCAT
      EQUIVALENCE (Z(1),X(1,1)),(T(1),X(1,2)),(AL(1),X(1,3))
      EQUIVALENCE (NM(1),X(1,4))
      DATA FIRST/.TRUE./,IB/4,4,1,3/,ITC/11/,HREF/1./
C *******************************************************************
C
C
C      PURPOSE
C
C        TO CONVERT PARAMETERS TO NONDIMENSIONAL FORM AND THEN COMPUTE
C      THE ZERO ORDER MOMENT AND INTERPOLATE FROM TABULATED VALUES OF
C      THE HIGHER ORDER MOMENTS
C
C      INPUT
C
C        VGRAV   THE GRAVITATIONAL SETTLING VELOCITIES OF THE PARTICLE
C                IN METERS / SEC
C        ZIN       THE HEIGHT (METERS) AT WHICH THE MOMENTS ARE DESIRED
C        H       THE HEIGHT OF RELEASE OF THE PARTICLES IN METERS
C        TIN     THE TIME IN SECONDS AFTER RELEASE
C
C      OUTPUT
C
C        Q       THE VERTICAL CONCENTRATION OF PARTICLES IN PARTS/METER
C                AT HEIGHT Z
C        XBAR    THE DISPLACEMENT (METERS) IN THE X (IE WIND) DIRECTION
C                OF THE CENTER OF MASS OF PARTICLES AT HEIGHT Z
```

```
C        SIGW2   THE SQUARE OF THE STANDARD DEVIATION OF THE WINDWARD
C                DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2
C        SIGP2   THE SQUARE OF THE STANDARD DEVIATION OF THE CROSS-WIND
C                DISPLACEMENT OF THE PARTICLES AT HEIGHT Z IN METERS**2
C
C
C     SUBROUTINES CALLED
C
C        DTERPS   PUTS THE NEEDED VALUES OF THE TABULATED MOMENTS
C                 INTO A ONE DIMENSIONAL ARRAY
C        DTERPI   A FUNCTION WHICH RETURNS THE INTERPOLATED VALUE
C                 FOR GIVEN ARGUMENTS AND ARRAYS
C        GREEN    CALCULATES THE GREENS FUNCTION WHICH IS THE
C                 0-ORDER MOMENT
C
C     CALLED BY CWIND
C
C ***************************************************************
      IF(.NOT.FIRST)GO TO 5
C
C     READ IN THE TABLE OF MOMENTS ON THE FIRST CALL OF MOMENT
C
C       Z       LOG OF NON-DIMENSIONAL HEIGHTS AT WHICH MOMENTS ARE TABULATED
C       T       LOG OF NON-DIMENSIONAL TIMES   AT WHICH MOMENTS ARE TABULATED
C       AL      NON-DIMENSIONAL SETTLING VELOCITIES AT WHICH MOMENTS ARE
C                     TABULATED
C       NM      DIFFUSIVITY POWER LAW EXPONENTS AT WHICH MOMENTS ARE
C                      TABULATED
C
C       XB      TABULATED VALUES OF LOGS OF FIRST ORDER MOMENTS (RELATED
C                   TO MEAN HORIZONTAL DISPLACEMENT)
C       SW      TABULATED VALUES OF LOGS OF WIND SHEAR COMPONENT OF SECOND
C                ORDER MOMENT (CONTRIBUTES TO VARIANCE IN WIND DIRECTION)
C       SP      TABULATED VALUES OF LOGS OF SECOND ORDER MOMENT COMMON TO
C                  WIND AND CROSS-WIND VARIANCES
C
      READ(NDIRTU,1) NZ,NT,NA,NN
    1 FORMAT(4I3)
      NTC(1)=NZ-1
      NTC(2)=NT-1
      NTC(3)=NA-1
      NTC(4)=NN-1
      READ(NDIRTU,2) (Z(I),I=1,NZ)
    2 FORMAT(6E13.5)
      READ(NDIRTU,2) (T(I),I=1,NT)
      READ(NDIRTU,2) (AL(I),I=1,NA)
      READ(NDIRTU,2) (NM(I),I=1,NN)
      NZT=NZ*NT
```

```
      DO 3 L=1,NN
      READ(NDIRTU,2) ((XB(IJ,K,L),IJ=1,NZT),K=1,NA)
      READ(NDIRTU,2) ((SW(IJ,K,L),IJ=1,NZT),K=1,NA)
      READ(NDIRTU,2) ((SP(IJ,K,L),IJ=1,NZT),K=1,NA)
    3 CONTINUE
      FIRST=.FALSE.
      REWIND NDIRTU
    5 CONTINUE
C
C     CONVERT INPUT PARAMETERS TO NONDIMENSIONAL FORM
C
      SCLU=DZO*H**(N-1.)
      XI(1)=ZIN/H
      XI(2)=SCLU*TIN/H
      XI(3)=VGRAV/SCLU
      XI(4)=N
      CALL GREEN(XI(1),HREF,XI(2),XI(3),Q,IER)
      Q=Q/H
      IF(Q .LE. 1.E-20) GO TO 999
C
C     TAKE LOGS FOR LOGARITHMIC INTERPOLATION
C
      XI(1)=ALOG(XI(1))
      XI(2)=ALOG(XI(2))
C
C     DETERMINE INDICES OF LOWEST CORNER POINT OF THE CUBE TO
C     BE USED IN INTERPOLATION MAKING SURE THAT ENOUGH CORNER POINTS
C     OF THE CUBE HAVE TABULATED VALUES
C
      DO 100 I=1,4
      II(I)=IB(I)
100   CONTINUE
      DO 101 III=1,4
      I=5-III
6     IA=II(I)
      IF(XI(I) .GE. X(IA,I) .AND. X(IA,I) .LE. X(IA+1,I)) GO TO 101
      IF(XI(I) .LT. X(IA,I) .AND. IA .EQ. 1) GO TO 101
      IF(XI(I) .GT. X(IA,I) .AND. IA .EQ. NTC(I)) GO TO 101
      ISAV=II(I)
      II(I)=IA + IFIX(SIGN(1.,XI(I)-X(IA,I)))
      IT=0
      DO 102 JI=1,2
      JIX=JI + II(1) - 1
      DO 102 IJ=1,2
      IJX=JIX + (IJ + II(2) - 2)*NZ
      DO 102 K=1,2
      KX=K-1 + II(3)
      DO 102 L=1,2
```

```
      LX=L-1 + II(4)
      IF(XB(IJX,KX,LX) .GT. -100.) IT=IT+1
102   CONTINUE
      IF(IT .GT. ITC) GO TO 6
      II(I)=ISAV
101   CONTINUE
C
C     PERFORM THE INTERPOLATION WITH DETERMINED CUBE OF POINTS
C
      DO 103 I=1,4
      I2=I*2
      I1=I2-1
      IA=II(I)
      XVAL(I1)=X(IA,I)
      XVAL(I2)=X(IA+1,I)
103   CONTINUE
      CALL DTERPS(II,XB,VAL,NZ)
      XBAR=DTERPI(4,XI,XVAL,VAL,-100.,W)
      CALL DTERPS(II,SW,VAL,NZ)
      SIGW2=DTERPI(-4,XI,XVAL,VAL,-100.,W)
      CALL DTERPS(II,SP,VAL,NZ)
      SIGP2=DTERPI(-4,XI,XVAL,VAL,-100.,W)
C
C     CONVERT THE LOG OF THE NONDIMENSIONAL VALUES INTERPOLATED
C     TO THE USUAL DIMENSIONAL FORM
C
      SCL=U0*H**(M+1.)/SCLU
      XBAR=SCL*EXP(XBAR)
      SIGW2=SCL*SCL*EXP(SIGW2)
      SIGP2=2.*DXZ0*H*H*EXP(SIGP2)
      SIGW2=SIGW2+SIGP2
      SIGP2=DYX0*SIGP2
999   RETURN
      END
C
C ************************  SUBFILE 26  **************************
C
      SUBROUTINE DTERPS(II,X,VAL,NZ)
      IMPLICIT INTEGER*4 (I-N)
      DIMENSION X(81,4,4),VAL(1),II(1)
C *****************************************************************
C
C
C     PURPOSE
C
C        TO SET UP A ONE DIMENSIONAL ARRAY OF THE VALUES CORRESPONDING
C        TO THE CORNERS OF THE CUBE WITHIN A TABULATED ARRAY WITH
C        LOWEST CORNER INDICES GIVEN
```

```
C
C     INPUT
C
C       II      SINGLY DIMENSIONED ARRAY CONTAINING THE INDICES OF THE
C               LOWEST CORNER OF THE CUBE
C       X       A TRIPLY DIMENSIONED ARRAY CONTAINING THE TABULATED
C               V^201ALUES TO BE SET UP.  THE FIRST INDEX IS THE COLLAPSED
C               INDEX FOR THE FIRST TWO INDICES OF A FOUR DIMENSIONAL
C               ARRAY
C       NZ      THE RANGE OF THE FIRST INDEX OF THE FOUR DIMENSIONAL
C               ARRAY
C
C
C     OUTPUT
C
C       VAL     SINGLY DIMENSIONED ARRAY CONTAINING THE VALUES OF X
C               FOR THE 16 CORNER POINTS OF THE CUBE
C
C     CALLED BY MOMENT
C
C *********************************************************************
      M=0
      DO 104 L=1,2
      LX=L + II(4) - 1
      DO 103 K=1,2
      KX=K + II(3) - 1
      DO 102 JI=1,2
      JIX=(JI + II(2) - 2)*NZ
      DO 101 IJ=1,2
      IJX=JIX + IJ + II(1) -1
      M=M+1
      VAL(M)=X(IJX,KX,LX)
101   CONTINUE
102   CONTINUE
103   CONTINUE
104   CONTINUE
      RETURN
      END
C
C **********************  SUBFILE 27  ********************************
C
      FUNCTION DTERPI(NDIM,XI,XVAL,VAL,VMIN,WORK)
      IMPLICIT INTEGER*4 (I-N)
C ***********************************************************
C
C
C     PURPOSE
C
```

```
C   PERFORMS AN N-DIMENSIONAL LINEAR INTERPOLATION
C
C
C      INPUT
C
C    NDIM - THE NUMBER OF DIMENSIONS.  (- DONT RECALCUALTE WEIGHTS)
C      XI - THE POINT IN THE HYPERSPACE AT WHICH THE INTERPOLATED
C           VALUE IS DESIRED.  XI MUST BE A VECTOR OF ATLEAST NDIM
C           IN LENGTH.
C    XVAL - THE COORDINATE VALUES AT THE CORNERS OF THE HYPERCUBE.
C           THE VECTOR MUST BE SET UP LIKE A TWO-DIMENSIONAL ARRAY
C           (2 X NDIM), WHERE THE FIRST SUBSCRIPT REFERS TO THE
C           HYPERCUBE COORDINATES IN THE SECOND SUBSCRIPTS
C           DIRECTION.
C     VAL - THE FUNCTIONAL VALUES AT THE CORNERS OF THE HYPERCUBE
C           SURROUNDING XI.  THIS VECTOR MUST BE FILLED EQUIVALENT
C           TO AN NDIM ARRAY WITH EACH DIMENSION AS 2.  THE SIZE
C           OF VAL SHOULD BE ATLEAST 2**NDIM.
C    VMIN - A MINIMUM VALUE OF VAL FOR WHICH THE INTERPOLATION
C           WILL USE A CORNER VALUE.
C    WORK - A WORK VECTOR OF ATLEAST NDIM*2.  USE TO STORE COOR-
C           DINATE WEIGHTS.
C
C
C      OUTPUT
C
C        RETURNS INTERPOLATED VALUE OF VAL AT XI
C
C      CALLED BY MOMENT
C
C *******************************************************************
      DIMENSION XI(1),XVAL(1),VAL(1),WORK(1)
C
C  SET UP THE COORDINATE WEIGHTS
C
      NDI=IABS(NDIM)
      IF(NDIM .LT. 0) GO TO 1
      DO 100 I=1,NDI
      I2=I*2
      I1=I2-1
      WORK(I2)=(XI(I)-XVAL(I1))/(XVAL(I2)-XVAL(I1))
      WORK(I1)=1. - WORK(I2)
100   CONTINUE
C
C  INTERPOLATE  -  USE BINARY COUNTER FOR COORDINATE LOCATION
C
1     DTERPI=0.
      SUM=0.
```

```
      ND=2**NDI
      DO 201 I=1,ND
      IF(VAL(I) .LT. VMIN) GO TO 201
      L=I-1
      WEIGHT=1.
      DO 200 J=1,NDI
      N=MOD(L,2) + J*2 - 1
      WEIGHT=WEIGHT*WORK(N)
      L=L/2
200   CONTINUE
      SUM=SUM + WEIGHT
      DTERPI=DTERPI + WEIGHT*VAL(I)
201   CONTINUE
      IF(SUM .EQ. 0.) GO TO 202
      DTERPI=DTERPI/SUM
      RETURN
C
202   STOP
      END
C
C **********************  SUBFILE 28  *****************************
C
      SUBROUTINE GREEN(Z,Z1,T,ALPHA,T0,IER)
      IMPLICIT INTEGER*4 (I-N)
C ************************************************************
C
C
C     PURPOSE
C
C       TO COMPUTE THE GENERALIZED GREENS FUNCTION
C
C      USES GREEN1
C
C      SEE GREEN1 FOR ARGUMENT LIST
C
C
C ************************************************************
      REAL N,M
      COMMON /WNDPRM/DXZ0,DYX0,DZ0,U0,M,N,ZINV
C
      IF(N .EQ. 1.) GO TO 2
      X2=2.-N
      AT=ALPHA*T
      T0=0.
      IF(AT .GE. Z1) RETURN
      CALL GREEN1((Z+AT)**X2,Z1**X2,X2*X2*T,(N-1.)/X2,T1,IER)
      T1=T1*X2*Z1**(1.-N)
      U=1.
      T2=0.
      IF(ABS(ALPHA) .LT. 1.E-4) GO TO 1
      ZMZ=Z-Z1+AT
      X2=N+1.
      AN1=ALPHA*X2
      ZMZN=Z1**X2 - (Z1-AT)**X2
      ARG=(-AN1*ZMZ*ZMZ)/(4.*ZMZN)
      IF(ARG .LT. -70.) GO TO 3
      T2=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
```

```
    3 IF(T1.LT.1.E-30 .AND. T2.LT.1.E-30) RETURN
C
C     CALCULATION OF MIXING RATIO, U, BY N=1 ANALOGY
C
      CALL GREEN1(Z+AT,Z1,T,0.,T1U,IER)
      X2=2.
      AN1=ALPHA*X2
      ZMZN=Z1**X2 - (Z1-AT)**X2
      T2U=0.
      ARG=(-AN1*ZMZ*ZMZ)/(4.*ZMZN)
      IF(ARG .LT. -70.) GO TO 4
      T2U=SQRT(AN1/(4.*3.1415926*ZMZN))*EXP(ARG)
    4 IF(T1U.LT.1.E-30 .AND. T2U.LT.1.E-30) GO TO 1
      CALL GREEN1(Z,Z1,T,ALPHA,G,IER)
      U=(G-T2U)/(T1U-T2U)
    1 IF(U .LT. 0.) U=0.
      IF(U .GT. 1.) U=1.
C
C     COMBINE LIMITING SOLUTIONS WITH DETERMINED MIXING RATIO
C
      T0=U*T1 + (1.-U)*T2
      RETURN
    2 CALL GREEN1(Z,Z1,T,ALPHA,T0,IER)
      RETURN
      END
C
C ********************  SUBFILE 29  *************************
C
C
C     ..............................................................
C
C        SUBROUTINE GREEN1
C
C        PURPOSE
C           COMPUTE THE I BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER
C                  AND MULTIPLY BY AN APPROPRIATE POWER OF THE ARGUMENT
C                  AND AN EXPONENTIAL IN ORDER TO CALCULATE THE GREENS
C                  FUNCTION FOR THE WIND DIFFUSION EQUATION
C
C        USAGE
C           CALL GREEN1(Z,Z1,T,NU,BI,IER)
C
C        DESCRIPTION OF PARAMETERS
C           Z,Z1,T   -THE ARGUMENTS OF THE FUNCTION DESIRED
C           NU  -THE ORDER OF THE I BESSEL FUNCTION
C           BI  -THE RESULTANT BESSEL FUNCTION
C           IER -RESULTANT ERROR CODE WHERE
C              IER=-1  EXPONENTIAL UNDERFLOW (NON-FATAL),  BI SET TO 0.0
C              IER=0 NO ERROR
C              IER=1 NU NEAR NEGATIVE INTEGER
C              IER=2 OVERFLOW IN GAMMA
C              IER=3 UNDERFLOW, BI .LT. 1.E-32, BI SET TO 0.0
C              IER=4 OVERFLOW, X .GT. 90 WHERE X .GT. N
C              IER=5 X IS NEGATIVE
C
C        REMARKS
C           NU IS A  REAL NUMBER
C           N AND X MUST BE .GE. ZERO
C           THIS SUBROUTINE IS A MODIFICATION OF BESI WHICH COMPUTES THE
C           I BESSEL FUNCTION FOR INTEGER ORDERS.  THE CHANGE REQUIRES THE
C           USE OF THE GAMMA FUNCTION FOR COMPUTING THE FIRST TERM OF THE
C           SERIES.  THE SUCCESSIVE TERMS ARE CALCULATED WITH THE SAME
C           RECURSION FORMULA AND THE ASYMPTOTIC APPROXIMATION IS ALSO
```

```
C           UNCHANGED.  BESI IS IN THE IBM SYSTEM/360 SCIENTIFIC SUBROUTINE
C           PACKAGE.  MODIFICATIONS MADE BY D. DVORE, AERODYNE RESEARCH,
C                    INC. JANUARY 15,1979.
C
C        SUBROUTINES AND FUNCTIONS REQUIRED
C           GAMMA  WHICH COMPUTES THE GAMMA FUNCTION
C
C        METHOD
C           COMPUTES I BESSEL FUNCTION USING SERIES OR ASYMPTOTIC
C           APPROXIMATION DEPENDING ON THE RANGE OF THE ARGUMENT.
C
C      CALLED BY MOMENT
C
C     ..........................................................
C
      SUBROUTINE GREEN1(Z,Z1,T,NU,BI,IER)
      IMPLICIT INTEGER*4 (I-N)
      REAL NU
      X=2.*SQRT(Z*Z1)/T
C
C     CHECK FOR ERRORS IN NU AND X AND EXIT IF ANY ARE PRESENT
C
      IER=0
      BI=1.0
      IF(NU)10,15,10
   10 IF(X)160,20,20
   15 IF(X)160,17,20
   17 ARG=-(Z+Z1)/T
      IF(ARG .LT. -80.) GO TO 170
      BI=EXP(ARG)/T
      RETURN
C
C     DEFINE TOLERANCE
C
   20 TOL=1.E-3
C
C     IF ARGUMENT GT 12 AND GT NU, USE ASYMPTOTIC FORM
C
      IF(X-12.)40,40,30
   30 IF(X-ABS(NU))40,40,110
C
C     COMPUTE FIRST TERM OF SERIES AND SET INITIAL VALUE OF THE SUM
C
   40 XX=X/2.
      N=INT(NU)
      FN=N
      R=NU-FN
      CALL GAMMA(1.+NU,GR,IER)
      IF(IER .EQ. 0) GO TO 60
   50 BI=0.0
      RETURN
   60 TERM=1./GR
   70 BI=TERM
      XX=XX*XX
C
C     COMPUTE TERMS, STOPPING WHEN ABS(TERM) LE ABS(SUM OF TERMS)*TOLERANCE
C
      DO 90 K=1,1000
      IF(ABS(TERM)-ABS(BI*TOL))95,95,80
   80 FK=K
      FK=FK*(NU+FK)
      TERM=TERM*(XX/FK)
   90 BI=BI+TERM
   95 ARG=-(Z+Z1)/T
```

```
      IF(ARG .LT. -80.) GO TO 170
      BI=BI*(Z1/T)**NU*EXP(ARG)/T
C
C     RETURN BI AS ANSWER
C
  100 RETURN
C
C     X GT 12 AND X GT NU, SO USE ASYMPTOTIC APPROXIMATION
C
  110 FN=4.*NU*NU
  115 XX=1./(8.*X)
      TERM=1.
      BI=1.
      DO 130 K=1,30
      IF(ABS(TERM)-ABS(BI*TOL)) 140,140,120
  120 FK=(2*K-1)**2
      TERM=TERM*XX*(FK-FN)/FLOAT(K)
  130 BI=BI+TERM
C
C     SIGNIFICANCE LOST AFTER 30 TERMS, TRY SERIES
C
      GO TO 40
  140 PI=3.141592653
      ARG=X-(Z+Z1)/T
      IF(ARG .LT. -80.) GO TO 170
      BI=BI*(Z1/Z)**(NU/2.)*EXP(ARG)/SQRT(2.*PI*X)/T
      GO TO 100
  160 IER=5
      GO TO 100
  170 BI=0.0
      GO TO 50
      END
C
C ******************  SUBFILE 30  ****************************
C
C
C    ..................................................................
C
C       SUBROUTINE GAMMA
C
C       PURPOSE
C          COMPUTES THE GAMMA FUNCTION FOR A GIVEN ARGUMENT
C
C       USAGE
C          CALL GAMMA(XX,GX,IER)
C
C       DESCRIPTION OF PARAMETERS
C          XX  -THE ARGUMENT FOR THE GAMMA FUNCTION
C          GX  -THE RESULTANT GAMMA FUNCTION
C          IER -THE RESULTANT ERROR CODE WHERE
C                IER=0  NO ERROR
C                IER=1  XX IS WITHIN .000001 OF BEING A NEGATIVE INTEGER
C                IER=2  XX GT 57, OVERFLOW, GX SET TO 1.E32
C
C       COMMENTS
C            NONE
C
C       SUBROUTINES AND FUNCTIONS
C            NONE
C
C       METHOD
C          THE RECURSION RLATION AND POLYNOMIAL APPROXIMATION
C          BY C. HASTINGS, JR., 'APPROXIMATIONS FOR DIGITAL COMPUTERS',
C          PRINCETON UNIVERSITY PRESS, 1955
```

```
C
C     .....................................................................
C
      SUBROUTINE GAMMA(XX,GX,IER)
      IMPLICIT INTEGER*4 (I-N)
      IF(XX-57.) 6,6,4
    4 IER=2
      GX=1.E32
      RETURN
    6 X=XX
      ERR=1.0E-6
      IER=0
       GX=1.0
      IF(X-2.0) 50,50,15
   10 IF(X-2.0) 110,110,15
   15 X=X-1.0
      GX=GX*X
      GO TO 10
   50 IF(X-1.0) 60,120,110
C
C     SEE IF X IS NEAR NEGATIVE INTEGER OR ZERO
C
   60 IF(X-ERR) 62,62,80
   62 Y=FLOAT(INT(X))-X
      IF(ABS(Y)-ERR) 130,130,70
C
C     X NOT NEAR A NEGATIVE INTEGER OR ZERO
C
   70 IF(X-1.0)80,80,110
   80 GX=GX/X
      X=X+1.0
      GO TO 70
  110 Y=X-1.0
      GY=1.0+Y*(-0.5771017+Y*(0.9858540+Y*(-0.8764218+Y*(0.8328212+
     1Y*(-0.5684729+Y*(0.2548205+Y*(-0.05149930)))))))
      GX=GX*GY
  120 RETURN
  130 IER=1
      RETURN
      END
      SUBROUTINE RKM(N,XL,XU,Y,HMIN,DEL,ACCURC,WK,ND)
      IMPLICIT INTEGER*4 (I-N)
C
C
C  NUMERICAL INTEGRATION ROUTINE FOR SYSTEMS OF ODE'S
C        USING THE RUNGE-KUTTA-MERSON TECHNIQUE
C
C
C  INPUT PARAMETERS
C
C      N  - NUMBER OF FIRST ORDER DIFFERENTIAL EQUATIONS
C     XL  - INITIAL ABCISSA OF THE INTERVAL
C     XU  - THE FINAL ABCISSA OF THE INTEGRATION INTERVAL
C      Y  - A SINGLY DIMENSIONED ARRAY OF LENGTH N.  WHEN
C             RKM IS CALLED IT MUST CONTAIN THE VALUES OF
C             THE DEPENDENT VARIABLES AT XL.  UPON RETURN
C             TO THE CALLING PROGRAM Y CONTAINS THE VALUES
C             OF THE DEPENDENT VARIABLES AT XU.
C   HMIN  - THE MINIMUM STEP SIZE THAT WILL BE USED FOR THE
C             INTEGRATION
C    DEL  - THE INITIAL ESTIMATE OF THE STEP SIZE AND UPON
C             RETURN TO THE CALLING PROGRAM DEL CONTAINS THE
C             FINAL STEP SIZE USED.  THIS VALUE SHOULD BE USED
C             IN THE NEXT CALL TO PRODUCE AN EFFICIENT INTEGRATION.
```

```
C             DEL IS RETURNED WITH THE VALUE ZERO IF IT HAS
C             BEEN HALVED BELOW HMIN.
C ACCURC - PREASSIGNED ACCURACY WHICH IS ALSO USED IN ADJUSTING
C             THE STEP SIZE.
C     WK - AT LEAST A BLOCK OF N BY 6 FLOATING POINT LOCATIONS
C             USED FOR A WORK ARRAY.
C     ND - THE DIMENSION OF ARRAYS Y AND WK.
C
C
C     IT IS REQUIRED THAT THE USER OF RKM WRITE A SUBROUTINE
C   DEFINING THE DIFFERENTIAL EQUATIONS.  THE SUBROUTINE
C   STATEMENT SHOULD LOOK LIKE - SUBROUTINE DIFEQ(N,X,Y,YP) .
C
C   WHERE
C         N - THE NUMBER OF EQUATIONS
C         X - THE INDEPENDENT VARIABLE
C         Y - SINGLY DIMENSIONED ARRAY OF DEPENDENT VARIABLES
C        YP - SINGLY DIMENSIONED ARRAY OF THE RATES OF Y AT X
C                    YP(I) = D Y(I)/DX
C
C
      DIMENSION Y(ND),WK(ND,6)
      LOGICAL FIRST,QUIT
C
C   SET UP NEEDED VARIABLES UPON ENTRY
C
      XN=XL
      H=DEL
      FIRST=.TRUE.
      QUIT=.FALSE.
C
C   CHECK IF XN IS CLOSE TO XU
C
   20 IF(XN+H .LT. XU) GO TO 30
      DEL=H
      H=XU-XN
      QUIT=.TRUE.
      IF(FIRST) DEL=H
C
C   MAKE FIRST CALL TO DIFEQ AT THE BEGINNING OF INTERVAL
C
   30 CALL DIFEQ(N,XN,Y,WK(1,1))
C
C   PERFORM THE RUNGE-KUTTA-MERSON ALGORITHM
C
   40 H3=H/3.
      DO 50 I=1,N
      WK(I,3)=H3*WK(I,1)
   50 WK(I,6)=Y(I)+WK(I,3)
      CALL DIFEQ(N,XN+H3,WK(1,6),WK(1,2))
      DO 60 I=1,N
   60 WK(I,6)=Y(I)+(WK(I,3)+H3*WK(I,2))/2.
      CALL DIFEQ(N,XN+H3,WK(1,6),WK(1,2))
      DO 70 I=1,N
      WK(I,4)=H3*WK(I,2)
   70 WK(I,6)=Y(I)+(3.*WK(I,3)+9.*WK(I,4))/8.
      CALL DIFEQ(N,XN+H/2.,WK(1,6),WK(1,2))
      DO 80 I=1,N
      WK(I,5)=H3*WK(I,2)
   80 WK(I,6)=Y(I)+(3.*WK(I,3)-9.*WK(I,4)+12.*WK(I,5))/2.
      CALL DIFEQ(N,XN+H,WK(1,6),WK(1,2))
C
C   FIND THE LARGEST RELATIVE ERROR
C
```

```
      TEST=0.
      DO 90 I=1,N
      YX=Y(I)
      IF(YX .EQ. 0.) YX=ACCURC
      E=((WK(I,3)-9.*WK(I,4)/2.+4.*WK(I,5)-H3*WK(I,2)/2.)/5.)/YX
   90 TEST=AMAX1(TEST,ABS(E))
      FIRST=.FALSE.
      IF(TEST .LT. ACCURC) GO TO 100
C
C  IF THE LARGEST ERROR IS GREATER THAN ACCURC HALF THE STEP
C    SIZE AND TRY AGAIN.
C
      H=H/2.
      IF(H .LT. HMIN) GO TO 10
      QUIT=.FALSE.
      GO TO 40
C
C  TRUNCATION ERROR LESS THAN ACCURC, RESET THE Y ARRAY TO
C    SET UP FOR THE NEXT INTERVAL
C
  100 XN=XN+H
      DO 110 I=1,N
  110 Y(I)=Y(I)+(WK(I,3)+4.*WK(I,5)+H3*WK(I,2))/2.
C
C  CHECK FOR STEP SIZE DOUBLING.  DOUBLE IF LARGEST RELATIVE
C     ERROR IS 32 TIMES LESS THAN ACCURC.
C
      IF(.NOT.(TEST .GE. ACCURC/32. .OR. QUIT)) H=H+H
      IF(.NOT. QUIT) GO TO 20
      RETURN
C
C  THE VALUE OF H (DEL) IS LESS THAN THE SPECIFIED MINIMUM.
C    REPORT THIS AND ERROR OUT.
C
   10 CONTINUE
      DEL=0.
      RETURN
      END
BOTTOM
```

U.S. GOVERNMENT PRINTING OFFICE 1980 – 677-119-54

DATE

FILMED

-8