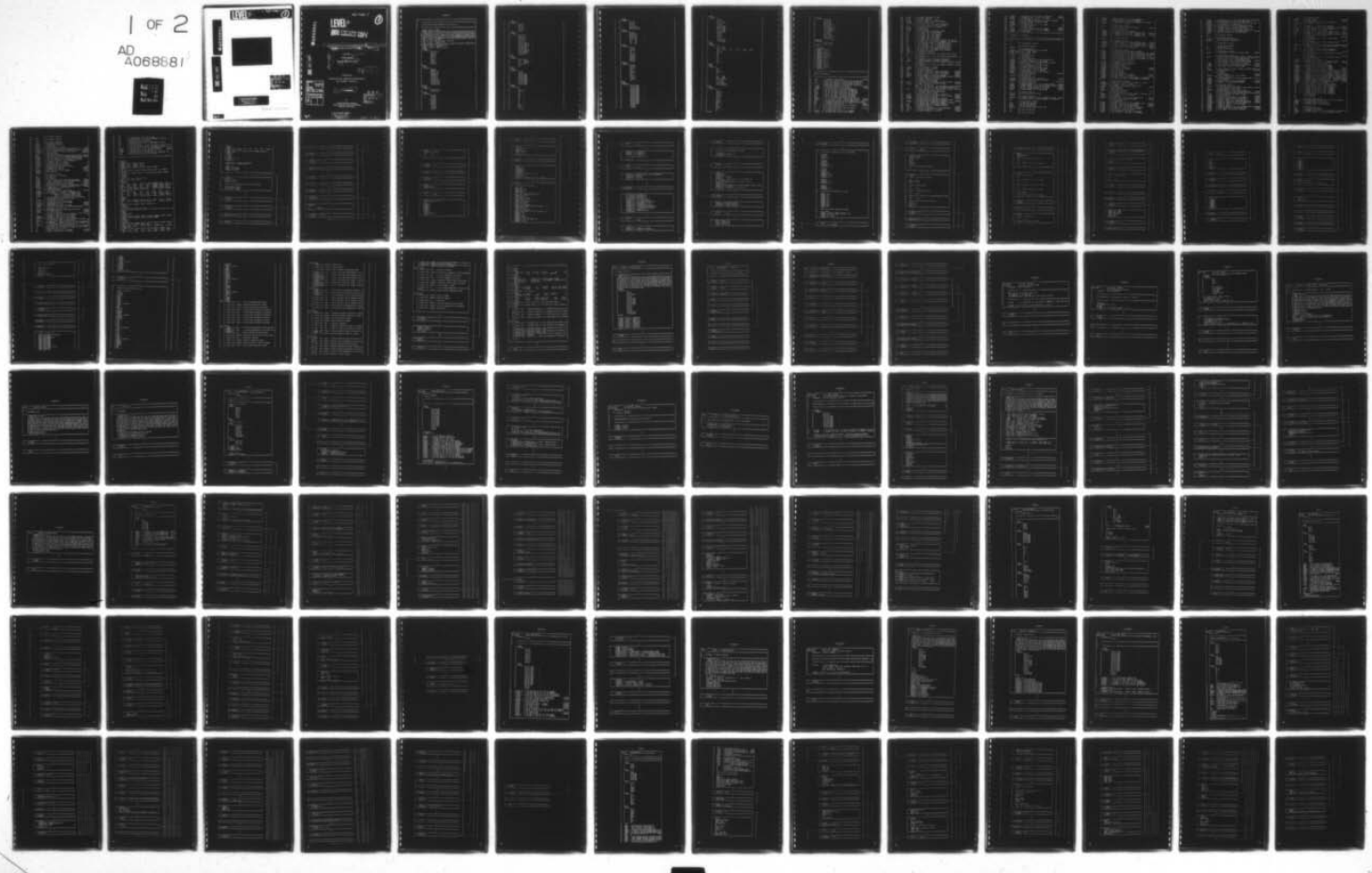
AD-A068 881 GENERAL ELECTRIC CO SYRACUSE N Y HEAVY MILITARY ELEC--ETC F/G 17/1
SONAR SIMULATION COMPUTER PROGRAMS. VOLUME 2. FLOW CHARTS.(U)
NOV 68 N00140-68-C-0372

UNCLASSIFIED NL

1 OF 2
AD A068681



MOST Project - 1

(1)

LEVEL II

OOVI LIBRARY COPY

AD A068881

DDC FILE COPY

(6) SONAR SIMULATION COMPUTER PROGRAMS Volume 2. Flow Charts

(9) PHASE I INTERIM TECHNICAL REPORT on Phase 1.

VOLUME 2

B021 097L

FLOW CHARTS

001951

Contract N00140-68-C-0372

(15)

(12) 137p.

Prepared for

United States Navy Underwater Sound Laboratory
Fort Trumbull, Connecticut

(11) 21 November 1968

By

General Electric Company
Heavy Military Electronics Systems
Syracuse, New York

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
Per ltr. on file
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

DDC
RECEIVED
MAY 23 1979
D

DISTRIBUTION STATEMENT
Approved for public release
Distribution Unlimited

Gp 6

149 510

```
                                    (ENTRANCE)
                                        I
                                        I
*****************************************************************************
*C                                                                           *
*C      ***************************************************************      *
*C      *                                                             *      *
*C      THIS IS MAIN CONTROL PROGRAM FOR SIMULATION                    *
*C      *                                                             *      *
*C      ***************************************************************      *
*C                                                                           *
*       COMMON /TWOB2/ B2S,B2E                                              *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSURS,AAAAAA,BBBBBB,MECO,VPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS    , FRWS   , FRLOSE,                                        *
*      A PTS(128), FXS(128), PNS(128), FVS(128),                          *
*      B FOE    , FRWE   , FRLOEV, F2E                                    *
*      7,F2S,PTE(128),FXE(128),PNE(128),FVE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                  *
*       COMMON / ANGRAD / ANGBER(2)                                       *
*       COMMON / INTSPD / VXEINT, VYEINT                                  *
*       COMMON                                                            *
*      C / ARRAYC /                                                       *
*      D            NARRAY(2)                                             *
*      C            ARRYH1,                                               *
*      C            ARRYH2,                                               *
*      C            ARRYW1,                                               *
*      C            ARRYW2,                                               *
*      C            DELF   ,                                              *
*      C            FRES1  ,                                              *
*      C            FRES2  ,                                              *
*      C            QTRAN1,                                               *
*      C            QTRAN2,                                               *
*      NDUM2                                                              *
*       COMMON                                                            *
*      C / ARRAYP /                                                       *
*      C            DGANGS,                                               *
*      C            DGANGE,                                               *
*      D            ARRAYD(3,2)      ,                                    *
*      D            COSPHI(2)        ,                                    *
*      D            COSRAD(2)        ,                                    *
*      C            MSHIPS,                                               *
*      D            SINPHI(2)        ,                                    *
*      D            SINRAD(2)        ,                                    *
*      D            TARRIV(3,2)      ,                                    *
*      D            TMATSE(3,3)      ,                                    *
*      D            TMATEV(3,3)      ,                                    *
*      D            TVECTR(3,2)      ,                                    *
*      DDUMMYS                                                            *
*       COMMON                                                            *
*      C / BEAMCR /                                                       *
*      D            EULANG(3,4)                                           *
*      D            BEMCOR(3,4)                                           *
*       COMMON                                                            *
*      C / CALPHA /                                                       *
*      C            ALPHAE,                                               *
*      C            ALPHAS,                                               *
*      C            FALPHE,                                               *
*      C            FALPHS,                                               *
*      C            ONMIAE,                                               *
*      C            ONMIAS,                                               *
*      C            STNPEV,                                               *
*      C            STNPSE,                                               *
*      C            TWOALE,                                               *
*      C            TWOALS,                                               *
*      C            NTIMEM                                                *
```

```
*      COMMON
*     C / CONSTN /                                                    *
*     C             DEPSER,                                           *
*     C             NCONSK,                                           *
*     C             SPAISE,                                           *
*     C             DEPEVA,                                           *
*     C             NCONSL,                                           *
*     C             SPAIEV                                            *
*      COMMON
*     C / FREQUN /                                                    *
*     D             COSAVE(2)       ,                                 *
*     C             DELRCZ,                                           *
*     D             DYHITH(2)       ,                                 *
*     D             ARAREA(2)       ,                                 *
*     C             CTWOPI,                                           *
*     D             SQAREA(2)       ,                                 *
*     D             FRQSIG(128,2)   ,                                 *
*     D             FRQLCN(128,2)   ,                                 *
*     D             FRQIDB(128,2)   ,                                 *
*     D             ANGDEP(2)       ,                                 *
*     D             FRQTRN(128,2)   ,                                 *
*     D             FRQASD(128,2)   ,                                 *
*     D             FRQNCN(128,2)   ,                                 *
*     D             FRQNOS(128,2)   ,                                 *
*     D             NUMFRQ(2)       ,                                 *
*     DDUM8
*      COMMON
*     C / INDEXS /                                                    *
*     D             ANDISO(16,2)    ,                                 *
*     D             DI1   (50,16)   ,                                 *
*     D             DI2   (50,16)   ,                                 *
*     D             DLDIAN(16,2)    ,                                 *
*     C             IO1   ,                                           *
*     C             IO2   ,                                           *
*     CDUMMYA
*      COMMON
*     C / LCONST /                                                    *
*     C             NULAPO,                                           *
*     C             DEPROT,                                           *
*     D             CONSG0(128)     ,                                 *
*     D             CONSG1(128)     ,                                 *
*     D             CONSG2(128)     ,                                 *
*     D             CONSV0(128)     ,                                 *
*     D             DELTAZ(128)     ,                                 *
*     D             DEPKYD(128)     ,                                 *
*     D             SLOPEJ(128)     ,                                 *
*     D             SPDKYD(128)                                       *
*      COMMON
*     C / PTHLNG /                                                    *
*     C             A1    ,                                           *
*     C             CI    ,                                           *
*     C             CDSQRD,                                           *
*     C             CV    ,                                           *
*     C             DI    ,                                           *
*     C             DZ1   ,                                           *
*     C             I     ,                                           *
*     C             BI    ,                                           *
*     C             SM    ,                                           *
*     C             H     ,                                           *
*     C             X     ,                                           *
*     C             Y1    ,                                           *
*     C             Y2    ,                                           *
*     C             DXDC  ,                                           *
*     C             PL    ,                                           *
*     C             TIMCON                                            *
```

```
      COMMON
     C / RANGES /
     C            NUANMO,
     C            ANGMAX,
     C            DELANG,
     C            DELRAD,
     D            ANGINT(200)
     D            RNGMOD(6,200)
      COMMON
     C / RAYPAR /
     C            RANGEH,
     D            BOTANG(6)      ,
     D            DRDXDC(6)      ,
     D            PATHLN(6)      ,
     D            RANGEC(6)      ,
     D            SPI   (6)      ,
     D            SPT   (6)      ,
     D            TIR   (6)      ,
     D            TTR   (6)
      COMMON
     C / RAYTRA /
     C            NCONCT,
     C            INITLK,
     C            ZSTART,
     C            Z2    ,
     C            SPVRSQ,
     C            ANGSTR,
     C            ANGARR,
     C            ANGBTM,
     C            ANGSUR,
     C            SPDVER,
     C            RANGET
      COMMON
     C / RCONST /
     C            ZONACZ,
     C            AMLSRD,
     C            ZONBCZ,
     C            HCI   ,
     C            HZSD  ,
     C            NCONSD,
     C            RCZ1  ,
     C            RCZ2  ,
     C            SDCON ,
     C            TCZAV1,
     C            TCZAV2,
     C            ZW    ,
     NDUM3
      COMMON
     C / SIGNAL /
     D            PRYSEV(128)    ,
     D            PRYSSE(128)    ,
     D            PRNOEV(128)    ,
     D            PRNOSE(128)    ,
     D            PROEVA(128)    ,
     D            PROSER(128)    ,
     D            VAREVA(128)    ,
     D            VARSER(128)    ,
     D            GMUEVA(128)    ,
     D            GMUSER(128)    ,
     D            DEVAEV(128)    ,
     D            DEVASE(128)    ,
     D            DEMUEV(128)    ,
     D            DEMUSE(128)    ,
     C            THREVA,
     C            THRSER,
     C            NTIMEN
```

```
      COMMON
     C / STATIC /
     C            NDSTAT,
     C            NESTAT,
     D            PFDEIN(128)
     C            PERANG,
     C            PPATHM,
     D            PRANGE(128)      ,
     D            PRANGS(128)      ,
     D            PRPTEV(128)      ,
     D            PRPTSE(128)      ,
     D            PSDEIN(128)      ,
     C            PSRANG,
     C            SMPEDT,
     C            SMPSDT,
     C            SMTONE,
     C            SMTONS,
     D            TCONEN(128)      ,
     D            TCONSN(128)      ,
     CDUMMYB
      COMMON
     C / STORAG /
     D            ADI1  (16)       ,
     D            ADI2  (16)       ,
     C            AR1X  , AR1Y  , AR1Z  , AR2X  , AR2Y  , AR2Z  ,
     C            ARRAY1,
     C            ARRAY2,
     C            E11   ,
     C            E12   ,
     C            E13   ,
     C            E21   ,
     C            E22   ,
     C            E23   ,
     C            F1E   ,
     C            F1S   ,
     CSDUMMY
      COMMON
     C / SURDUC /
     C            BLA1  ,
     C            BLA2  ,
     C            BLA3  ,
     C            DTRAD ,
     C            J     ,
     C            M     ,
     D            BF1   (128)      ,
     D            BF2   (128)      ,
     D            DELRAF(128,2)    ,
     D            FLN1(128)        ,
     D            FLN2(128)        ,
     D            CONLR2(40,50)    ,
     D            CONLR1(40,50)    ,
     DDUM9
      COMMON
     C / SURFAC /
     C            A6    ,
     C            CONST2,
     C            CONST4,
     C            CZANGL,
     C            CZANDL,
     C            CZRANG,
     C            G1SD  ,
     C            G2SD  ,
     C            NCZRAS,
     C            NZONE ,
     C            RSD   ,
     C            RSD1  ,
     C            SCSD  ,
     C            SORTZL,
     C            SS    ,
     C            ZL    ,
     CDUMMYZ
```

```
C
      DIMENSION                                                         *
     D            ANCZAV(2)          ,                                 *
     D            ARHITH(2)          ,                                 *
     D            ARQTRN(2)          ,                                 *
     D            ARWIDT(2)          ,                                 *
     D            BAFFUN(128,2)      ,                                 *
     D            CDYOVV(2)          ,                                 *
     D            DIRANG(16,2)       ,                                 *
     D            DIRSON(50,16,2)    ,                                 *
     D            FLONOS(128,2)      ,                                 *
     D            FRQLOW(2)          ,                                 *
     D            FRQRES(2)          ,                                 *
     D            NUDIAN(2)          ,                                 *
     D            SIGPLS(128,2)      ,                                 *
     D            SPDTER(2)          ,                                 *
     D            TMATRX(3,3,2)      ,                                 *
     DADUMMY(1)                                                         *
      DIMENSION                                                         *
     D            AINPUT(3,2)        ,                                 *
     D            EANGLE(3,2)        ,                                 *
     DDUMMYA(1)                                                         *
C                                                                       *
      EQUIVALENCE                                                       *
     Q ( ANCZAV, TCZAV1 ),                                              *
     Q ( DIRANG, ADI1 ),                                                *
     Q ( ANGDGA, DGANGS ),                                              *
     Q ( ARHITH, ARRYH1 ),                                              *
     Q ( ARQTRN, QTRAN1 ),                                              *
     Q ( AINPUT, AR1X ),                                                *
     Q ( ARWIDT, ARRYW1 ),                                              *
     Q ( BAFFUN, BF1 ),                                                 *
     Q ( DIRSON, DI1 ),                                                 *
     Q ( EANGLE, E11 ),                                                 *
     Q ( FLONOS, FLN1 ),                                                *
     Q ( FRQRES, FRES1 ),                                               *
     Q ( NUDIAN, IO1 ),                                                 *
     Q ( TMATRX, TMAISE ),                                              *
     Q ( NCONSR, NR )                                                   *
C                                                                       *
C     ****************************************************************  *
C     *                                                              *  *
C     INPUTS TO PROGRAM                                                 *
C     *                                                              *  *
C     ****************************************************************  *
C                                                                       *
C*0*   ACZ          = CONVERGENT ZONE CONSTANT                  (DIMENS*
C*0*   ADI1   ( )   = ANGLES OF DIFFERENT DIRECTIVITIES FOR SEARCHER   (*
C                    CHANGED FROM DEGREES TO RADIANS FOR INTERNAL USE   *
C*0*   ADI2   ( )   = ANGLES OF DIFFERENT DIRECTIVITIES FOR EVADER     (*
C                    CHANGED FROM DEGREES TO RADIANS FOR INTERNAL USE   *
C*0*   ALPHAE       = SMOOTHING PARAMETER FOR EVADER            (DIMENS*
C*0*   ALPHAS       = SMOOTHING PARAMETER FOR SEARCHER          (DIMENS*
C*0*   ANGMAX       = MAXIMUM ANGLE BEING CONSIDERED FOR RAYS       (*
C*0*   AR1X         = ONE OF THE ARRAY DIMENSIONS FOR SEARCHER        *
C*0*   AR1Y         = ONE OF THE ARRAY DIMENSIONS FOR SEARCHER        *
C*0*   AR1Z         = ONE OF THE ARRAY DIMENSIONS FOR SEARCHER        *
C*0*   AR2X         = ONE OF THE ARRAY DIMENSIONS FOR EVADER          *
C*0*   AR2Y         = ONE OF THE ARRAY DIMENSIONS FOR EVADER          *
C*0*   AR2Z         = ONE OF THE ARRAY DIMENSIONS FOR EVADER          *
C*0*   ARRAY1       = TYPE OF ARRAY CONTROL CONSTANT FOR SEARCHER (DIMENS*
C*0*   ARRAY2       = TYPE OF ARRAY CONTROL CONSTANT FOR EVADER  (DIMENS*
C*0*   B2E          = INTEGRATION BANDWIDTH OF POST-DETECTION OF EVADER *
C*0*   B2S          = INTEGRATION BANDWIDTH OF POST-DETECTION OF SEARCHER*
C*0*   BCZ          = CONVERGENT ZONE CONSTANT                  (DIMENS*
C*0*   BF1    ( )   = BAFFLING CORRECTION FACTOR FOR SEARCHER    (DIMENS*
C                    AS A FUNCTION OF ANGLE                             *
C*0*   BF2    ( )   = BAFFLING CORRECTION FACTOR FOR EVADER      (DIMENS*
C                    AS A FUNCTION OF ANGLE                             *
C*0*   CONLR1( , )= RADIATED POWER SPECTRUM FROM SEARCHER               *
C*0*   CONLR2( , )= RADIATED POWER SPECTRUM FROM EVADER                 *
```

```
*C*     DELANG      = DIFFERENCE IN ANGLES FOR RAYS                          (*
*C*     DELF        = FREQUENCY INCREMENT                                     *
*C*     DELHED      = CHANGE IN HEADING OF EVADER                            (*
*C*     DELRNG      = CHANGE IN RANGE OF CLOSEST APPROACH                     *
*C*     DEPKYD( )   = DEPTH TO TOP OF LAYER                                   *
*C*     DI1    ( , )= DIRECTIVITY INDEX FOR SONAR ON SEARCHER                 *
*C*     DI2    ( , )= DIRECTIVITY INDEX FOR SONAR ON EVADER                   *
*C*     DIFTI       = TIME BETWEEN POINTS ON A BRANCH                        (*
*C*     DPETEV      =  EVADER  DEPTH                                          *
*C*     DPETSE      = SEARCHER DEPTH                                          *
*C*     DTRAD       = ANGLE INCREMENT FOR RADIATED SIGNAL                    (*
*C*     E11         = EULER ANGLE INPUT FOR SEARCHER                         (*
*C*     E12         = EULER ANGLE INPUT FOR SEARCHER                          *
*C*     E13         = EULER ANGLE INPUT FOR SEARCHER                          *
*C*     E21         = EULER ANGLE INPUT FOR EVADER                            *
*C*     E22         = EULER ANGLE INPUT FOR EVADER                            *
*C*     E23         = EULER ANGLE INPUT FOR EVADER                            *
*C*     F0E         = PRE-DETECTION FILTER CENTER FREQUENCY FOR EVADER        *
*C*     F0S         = PRE-DETECTION FILTER CENTER FREQUENCY FOR SEARCHER      *
*C*     F1E         = LOWER FREQUENCY LIMIT OF EVADER EQUIPMENT               *
*C*     F1S         = LOWER FREQUENCY LIMIT OF SEARCHER EQUIPMENT             *
*C*     F2E         = UPPER FREQUENCY LIMIT OF EVADER EQUIPMENT               *
*C*     F2S         = UPPER FREQUENCY LIMIT OF SEARCHER EQUIPMENT             *
*C*     FBWE        = PRE-DETECTION FILTER BANDWIDTH OF EVADER                *
*C*     FBWS        = PRE-DETECTION FILTER BANDWIDTH OF SEARCHER              *
*C*     FLN1   ( )  = FLOW NOISE OF SEARCHER                                  *
*C*     FLN2   ( )  = FLOW NOISE OF EVADER                                    *
*C*     FNE    ( )  = ORDERED FREQUENCY POINTS FOR NOISE AROUND EVADER        *
*C*                   SAME AS FXE                                             *
*C*     FNS    ( )  = ORDERED FREQUENCY POINTS FOR NOISE AROUND SEARCHER      *
*C*                   SAME AS FXS                                             *
*C*     FRES1       = TRANSDUCER RESONANT FREQUENCY ON SEARCHER                *
*C*     FRES2       = TRANSDUCER RESONANT FREQUENCY ON EVADER                  *
*C*     FXE    ( )  = ORDERED FREQUENCY POINTS FOR SIGNAL RECIEVED BY EVADE*
*C*                   SAME AS FNE                                             *
*C*     FXS    ( )  = ORDERED FREQUENCY POINTS FOR SIGNAL RECIEVED BY SEARC*
*C*                   SAME AS FNS                                             *
*C*     HDINEV      = INITIAL HEADING OF EVADER                              (*
*C*     HEDMAX      = MAXIMUM HEADING OF EVADER                              (*
*C*     HS1         = HEADING OF SEARCHER                                    (*
*C*     IO1         = NUMBER OF DIRECTIVITY ANGLES FOR SEARCH       (DIMENS*
*C*     IO2         = NUMBER OF DIRECTIVITY ANGLES FOR EVADER       (DIMENS*
*C*     MAXLAY      = MAXIMUM LAYERS TO BE USED TO FIT PROFILE      (DIMENS*
*C*     MAXTIM      = MAXIMUM NUMBER OF POINTS ON A BRANCH          (DIMENS*
*C*     MECO        = CONTROL PARAMETER FOR EVASION COURSE OPTIONS  (DIMENS*
*C*                   1 = INVERSE BEARING RIDER OPTION                        *
*C*                   2 = NORMAL TO SEARCH EVASION                            *
*C*                   3 = CONTINUE INITIAL COURSE                             *
*C*     NEMAX       = MAXIMUM POINT ALONG BRANCH TO START EVASION   (DIMENS*
*C*     NEWLAY      = CONTROL CONSTANT FOR INITIALIZATION           (DIMENS*
*C*                   NEGATIVE = NEW PROFILE                                  *
*C*                   ZERO     = OLD PROFILE-- NEW SHIP DEPTHS OR DELTA ANGLE*
*C*                   POSITIVE = OLD PROFILE-- OLD SHIP DEPTHS AND ANGLES     *
*C*     NPCO        = CONTROL PARAMETER FOR PURSUIT COURSE OPTIONS  (DIMENS*
*C*                   1 = BEARING RIDER CLOSING TACTICS                       *
*C*                   2 = COLLISION COURSE CLOSING TACTICS                    *
*C*                   3 = CONTINUE INITIAL COURSE                             *
*C*     NPRINT      = CONTROL PARAMETER FOR AMOUNT OF PRINTOUT      (DIMENS*
*C*     NSMAX       = MAXIMUM POINT ALONG BRANCH TO START CLOSING   (DIMENS*
*C*     NULAPO      = NUMBER OF LAYERS PLUS ONE                     (DIMENS*
*C*     PDEMIN      = MINIMUM DETECTION PROBABILITY ALLOWED         (DIMENS*
*C*                   AFTER SHIPS START SEPARATING AS SEEN BY EVADER          *
*C*     PDSMIN      = MINIMUM DETECTION PROBABILITY ALLOWED         (DIMENS*
*C*                   AFTER SHIPS START SEPARATING AS SEEN BY SEARCHER         *
*C*     PHIE        = DEPRESSION ANGLE FOR EVADER STEERING                   (*
*C*     PHIS        = DEPRESSION ANGLE FOR SEARCHER STEERING                 (*
*C*     POR         = POROSITY OF BOTTOM                            (DIMENS*
*C*     PPAMIN      = MINIMUM PATH PROBABILITY TO BE CONSIDERED     (DIMENS*
*C*     PRE         = A-PRIORI PROBABILITY OF EVASION               (DIMENS*
*C*     PRK         = A-PRIORI PROBABILITY OF A KILL                (DIMENS*
```

```
C*    QTRAN1      = TRANSDUCER FIGURE OF MERIT ON SEARCHER          (DIMENS*
C*    QTRAN2      = TRANSDUCER FIGURE OF MERIT ON EVADER            (DIMENS*
C*    RGINEV      = INITIAL RANGE OF CLOSE                                  *
C*    RI          = INITIAL RANGE BETWEEN SHIPS AT START OF BRANCH          *
C*    RNGMAX      = MAXIMUM CLOSEST APPROACH DISTANCE                       *
C*    SPDEVA      = SPEED OF EVADER                                         *
C*    SPDKYD( )   = PROPAGATION SPEED AT TOP OF LAYER                  (K*
C*    SPDSER      = SPEED OF SEARCHER                                       *
C*    SS          = SEA STATE INPUT                                 (DIMENS*
C*    THREVA      = DETECTION THRESHOLD FOR THE EVADER              (DIMENS*
C*    THRSER      = DETECTION THRESHOLD FOR THE SEARCHER            (DIMENS*
C*    WRANGE      = WEAPON RANGE                                            *
C*****************************************************************************
C*                                                                         *
C*  CONSTANTS  IN  PROGRAM                                                 *
C*                                                                         *
C*****************************************************************************
C*****************************************************************************
C*                                                                         *
      A1          =                                                        *
                  = COMPUTATIONAL VARIABLE                                 *
      A6          =                                                        *
                  COMPUTATIONAL VARIABLE                                   *
      AAAAAA      =                                                        *
                  NOT USED OR REQUIRED                                     *
      AINPUT( , )= ARRAY PARAMETERS INPUT TO PROGRAM                       *
C     ALPXN = VALUE GEN. IN SUB. UPDATE                                    *
C     ALPYN = VALUE GEN. IN SUB. UPDATE                                    *
      ALSUBE      = -SEE ALPHAE-                                           *
                  NOT USED OR REQUIRED                                     *
      ALSUBS      = -SEE ALPHAS-                                           *
                  NOT USED OR REQUIRED                                     *
      AMLSRD      = CONSTANT REQQUIRED FOR AMOS CALCULATIONS               *
      ANCZAV( )   = AVERAGE OF CONVERGENT ZONE ANGLES                  (*
      ANDISO( , )= ANGLES OF DIFFERENT DIRECTIVITIES FOR SHIPS         (*
      ANGARR      = ARRIVAL ANGLE OF RAY AT Z2                         (*
      ANGBTM      = BOTTOM BOUNCE ANGLE OF RAY PORTION                 (*
      ANGDEP( )   = STEERING DEPRESSION ANGLES                              *
      ANGDGA( )   = AZIMUTHAL MAIN BEAM STEERING ANGLE                 (*
      ANGINT( )   = INITIAL ANGLE OF RAY AT SEARCHER                   (*
      ANGSTR      = STARTING ANGLE OF RAY AT ZSTART                    (*
      ANGSUR      = SURFACE BOUNCE ANGLE OF RAY PORTION                (*
      ARAREA( )   = ARRAY AREA CONSTANT                          ((YD-SEC/K*
                  NOT USED OR REQUIRED                                     *
      ARHITH( )   = ARRAY HEIGTH                                           *
                  NOT USED OR REQUIRED                                     *
      ARQTRN( )   = Q-FACTOR FOR ARRAY TRANSDUCERS                 (DIMENS*
      ARRAYD( , )= MODIFIED ARRAY DIMENSIONS FOR SHIPS                     *
      ARWIDT( )   = ARRAY WIDTH CONSTANT                         ((YD.SEC/K*
                  NOT USED OR REQUIRED                                     *
      AVBRKL      = AVERAGE PROBABILITY OF KILL ALONG A BRANCH     (DIMENS*
      AVBREV      = AVERAGE PROBABILITY OF EVASION ALONG A BRANCH  (DIMENS*
      AVEEVA      = OVERALL AVERAGE PROBABILITY OF EVADE           (DIMENS*
      AVEKIL      = OVERALL AVERAGE PROBABILITY OF KILL            (DIMENS*
      AVEPTH      = AVERAGE PROBABILITY OF CONTINUING              (DIMENS*
      B2          =                                                        *
                  NOT USED OR REQUIRED                                     *
      BAFFUN( , )= BAFFLING VALUES FOR SHIPS                               *
      BBBBBB      =                                                        *
                  NOT USED OR REQUIRED                                     *
      BEMCOR( , )= FREQUENCY INDEPENDENT BEAM-CORRECTION FACTOR  (DIMENS*
      BEP         = AZIMUTHAL ANGLE OF EVADER MEASURED FROM SEARCHER    (*
      B1          =                                                        *
                  = COMPUTATIONAL VARIABLE                                 *
      BLA1        = BOTTOM LOSS CONSTANT                                   *
      BLA2        = BOTTOM LOSS CONSTANT                                   *
      BLA3        = BOTTOM LOSS CONSTANT                                   *
      BOTANG( )   = BOTTOM BOUNCE ANGLE FOR RAY TYPE                   (*
      BPE         =                                                        *
                  NOT USED OR REQUIRED                                     *
      BPS         =                                                        *
                  NOT USED OR REQUIRED                                     *
```

```
*      BRNEVA      = BRANCH SUMMATION OF EVASION PROBABILITY                    *
*      BRNKIL      = BRANCH SUMMATION OF KILL PROBABILITY                       *
*      BSP         = AZIMUTHAL ANGLE OF SEARCHER MEASURED FROM EVADER     (*
*      CDSQRD      = COMPUTATIONAL VARIABLE                                     *
*      CI          =                                                            *
*                  = COMPUTATIONAL VARIABLE                                     *
*      CLPH        = CONTROL CONSTANT FOR CLOSE PHASE                 (DIMENS*
*                  0 = NOT IN CLOSE PHASE                                       *
*                  1 = IN CLOSE PHASE                                           *
*      CONNOR      = CONSTANT USED FOR NORMALIZING                              *
*      CONSG0( )   = G0 CONSTANT AS COMPUTED BY LAYERS ROUTINE     (SEC**2/*
*      CONSG1( )   = G1 CONSTANT AS COMPUTED BY LAYERS ROUTINE     (SEC**2/*
*      CONSG2( )   = G2 CONSTANT AS COMPUTED BY LAYERS ROUTINE           (K*
*      CONST2      =                                                            *
*                  COMPUTATIONAL VARIABLE                                       *
*      CONST4      =                                                            *
*                  COMPUTATIONAL VARIABLE                                       *
*      CONSV0( )   = V0 CONSTANT AS COMPUTED BY LAYERS ROUTINE      ((SEC/K*
*      COSAVE( )   = COSINE OF AVERAGE OF CONVERGENT ZONE ANGLES    (DIMENS*
*      COSPHI( )   = COSINE OF STEERING DEPRESSION ANGLE FOR SHIPS  (DIMENS*
*      COSRAD( )   = COSINE OF AZIMUTHAL SIGNAL ARRIVAL ANGLE       (DIMENS*
*      CTWOPI      = CONSTANT EQUAL TO TWO TIMES PI                 (DIMENS*
*      CV          =                                                            *
*                  = COMPUTATIONAL VARIABLE                                     *
*      CZANDL      = DELTA ANGLE FOR CONVERGENT ZONE SEARCH               (*
*      CZANEO      = SMALLER CONVERGENT ZONE ANGLE (IN RADIANS) FOR EVADER*
*      CZANET      = LARGER  CONVERGENT ZONE ANGLE (IN RADIANS) FOR EVADER*
*      CZANGL      = ANGLE BEING USED FOR CONVERGENT SONE RANGE SEARCH    (*
*      CZANSO      = SMALLER CONVERGENT ZONE ANGLE (IN RADIANS) FOR SEARCH*
*      CZANST      = LARGER  CONVERGENT ZONE ANGLE (IN RADIANS) FOR SEARCH*
*      CZRANG      = RANGE FOR CONVERGENT ZONE RAY                              *
*      D           = DEPTH (IN K-YD) IN LAYER                                   *
*      DELBAF( , )= DIFFERENCE IN BAFFLING VALUES                               *
*      DELRAD      = DIFFERENCE IN ANGLES FOR RAYS                        (*
*      DELRCZ      = WIDTH OF CONVERGENT ZONE                                   *
*      DELTAZ( )   = THICKNESS OF LAYER                                         *
*      DEMUEV( )   = MODIFIED MU FOR EVADER                         (DIMENS*
*      DEMUSE( )   = MODIFIED MU FOR SEARCHER                       (DIMENS*
*      DEPBOT      = DEPTH OF BOTTOM                                            *
*      DEPEVA      = DEPTH OF EVADER                                            *
*      DEPSER      = DEPTH OF SEARCHER                                          *
*      DEVAEV( )   = MODIFIED VARIANCE FOR EVADER                   (DIMENS*
*      DEVASE( )   = MODIFIED VARIANCE FOR SEARCHER                 (DIMENS*
*      DGANGS      = AZIMUTHAL MAIN-BEAM STEERING ANGLE FOR SEARCHER      (*
*      DGANGE      = AZIMUTHAL MAIN-BEAM STEERING ANGLE FOR SEARCHER      (*
*      DI          =                                                            *
*                  = COMPUTATIONAL VARIABLE                                     *
*      DIRANG( , )= ANGLES USED WITH DIFFERENT DIRECTIVITY CURVES        (*
*      DIRSON(,,) = DIRECTIVITY INDEX FOR SONAR                                 *
*      DLDIAN( , )= DIFFERENCES IN ANGLES OF DIRECTIVITIES               (*
*      DRDMDC( )   = DERIVATIVE (DXDC) FOR RAY TYPE                 ((SEC/K*
*      DUMMY1      = DUMMY VARIABLE                                             *
*      DXDC        = RANGE DERIVATIVE                                           *
*      DYHITH( )   = ARRAY HEIGTH CONSTANT                             (YD-S*
*                  NOT USED OR REQUIRED                                         *
*      DZ1         = DEPTH (IN K-YD) FROM TOP OF LAYER TO STARTING POINT  (*
*      EANGLE( , )= EULER ANGLES INPUT TO PROGRAM                               *
*      EDEPTH      = SEE -DEPEVA-                                               *
*      EULANG( , )= EULERIAN ANGLE AND THEIR SINE AND COSINE        (DIMENS*
*      EVPH        = CONTROL CONSTANT FOR EVADE PHASE               (DIMENS*
*                  0 = NOT IN EVADE PHASE                                       *
*                  1 = IN EVADE PHASE                                           *
*      FACTOR      = MODIFICATION FACTOR FOR PATH PROBABILITIES     (DIMENS*
*      FALPHE      = FRACTION  (ONMIAE**2/(1-ONMIAE**2))            (DIMENS*
*      FALPHS      = FRACTION  (ONMIAS**2/(1-ONMIAS**2))            (DIMENS*
*      FILSER      = FUNCTION FOR FILTER RESPONSE OF SEARCHER                   *
*      FILEVA      = FUNCTION FOR FILTER RESPONSE OF EVADER                     *
*      FLONOS( , )= FLOW AND SELF NOISE FOR SHIPS                               *
*      FRLOEV      = LOWER INTEGRATION FREQUENCY FOR  EVADER                    *
*      FRLOSE      = LOWER INTEGRATION FREQUENCY FOR SEARCHER                   *
```

```
C     FRQASD( , )= INTER-RANGE CONSTANT TIME SQRT( FREQUENCY/LAYER DEPTH
C     FRQIDB( , )= TEN*LOG( FREQUENCY ) AS A FUNCTION OF SHIP
C     FRQLCN( , )= LAYER CONSTANT AS A FUNCTION OF FREQUENCY AND SHIP
C     FRQLOW( )  = LOWER FREQUENCY OF INTEGRATION
C     FRQNCN( , )= NOISE SPECTRA UNALTERED BY SONAR DIRECTIVITY
C     FRQNOS( , )= FREQUENCY POINTS OF NOISE  AS FUNCTION OF SHIP
C     FRQRES( )  = TRANSDUCER RESONANT FREQUENCY FOR SHIPS
C     FRQSIG( , )= FREQUENCY POINTS USED FOR SIGNAL COMPUTATIONS
C     FRQTRN( , )= FREQUENCY RESPONSE OF TRANSDUCER ELEMENTS      (DIMENS
C     FTTOKY     = FUNCTION TO CHANGE FEET TO KILO-YARDS
C     G1SD       =
C                COMPUTATIONAL VARIABLE
C     G2SD       =
C                COMPUTATIONAL VARIABLE
C     GMUSER( )  = MEAN OF SMOOTHED SEARCHER S/N
C     GMUEVA( )  = MEAN OF SMOOTHED EVADER S/N
C     H          =
C                = COMPUTATIONAL VARIABLE
C     HCI        =
C                = COMPUTATIONAL VARIABLE
C     HE1        = HEADING OF EVADER                                (
C     HZSD       = SURFACE DUCT CONSTANT
C     I          =
C                = COMPUTATIONAL VARIABLE
C     INITLK     = LAYER NUMBER IN WHICH RAY PORTION STARTS       (DIMENS
C     J          =
C                = COMPUTATIONAL VARIABLE
C     JUMPIF     = CONTROL PARAMETER FOR RECALCULATING A LAYER CONSTANT
C     K          = INITIALLY THE LAYER IN WHICH THE STARTING POINT IS FO
C                LAYER IN WHICH CALCULATIONS ARE BEING MADE
C     K          = INDEX VARIABLE USED IN INTEGRATION ROUTINE     (DIMENS
C     L          = LAYER IN WHICH FINAL POINT (Z2) IS LOCATED
C     LAYERL     = LAYER IN WHICH A LOWER VERTEX IS FOUND         (DIMENS
C     LAYERM     = LAYER IN WHICH VELOCITY PROFILE HAS A ZERO     (DIMENS
C     LAYERS     = LAYER WHICH CONTAINS SHALLOWER SHIP            (DIMENS
C     M          =
C     MAXLAY     = MAXIMUM NUMBER OF LAYERS ALLOWED INCLUDING ADDED POIN
C     MSHIPS     = CONTROL NUMBER OF SHIP                         (DIMENS
C                1 = SEARCHER
C                2 = EVADER
C     N          = POINT ON BRANCH                                (DIMENS
C     NARRAY( )  = CONTROL PARAMETER FOR TYPE OF ARRAY            (DIMENS
C                NOT USED OR REQUIRED
C     NCONCI     = CONTROL PARAMETER FOR DIFFERENT SUBROUTINES TO BE CAL
C     NCONSD     = AMOS CONSTANT SD
C     NCONSK     = NUMBER OF LAYER CONTAINING SEARCHER            (DIMENS
C     NCONSL     = NUMBER OF LAYER CONTAINING EVADER              (DIMENS
C     NCZRAS     =
C                COMPUTATIONAL VARIABLE
C     NDSTAT     = INTEGER VALUE OF THE D-STATE
C     NEI        = POINT ALONG BRANCH WHEN EVADER STARTS EVADING  (DIMENS
C     NESTAT     = INTEGER VALUE OF THE E-STATE
C     NNFLAG     = DEBUG PARAMETER TO SHOW PROGRAM FLOW
C     NR         = CONTROL CONSTANT FOR POSITION IN BRANCHES      (DIMENS
C     NSI        = POINT ALONG BRANCH WHEN SEARCHER START CLOSING (DIMENS
C     NSTART     = CONTROL PARAMETER FOR START OF LOOP
C     NTIMEM     = NTIMEN MINUS ONE                               (DIMENS
C     NTIMEN     = POSITION ALONG PROBABILITY TREE BRANCH         (DIMENS
C     NUANMO     = NUMBER OF ANGLES FOR RAYS MINUS ONE            (DIMENS
C     NUDIAN( )  = NUMBER OF DIRECTIVITY ANGLES FOR SHIPS         (DIMENS
C     NUMANG     = NUMBER OF ANGLE BEING CONSIDERED FOR RAYS      (DIMENS
C     NUMFRQ( )  = NUMBER OF FREQUENCY POINTS FOR INTEGRATION     (DIMENS
C     NUMHED     = NUMBER OF HEADINGS TO BE CONSIDERED
C     NUMLAY     = NUMBER OF LAYERS
C     NUMRNG     = NUMBER OF RANGES FOR CLOSEST APPROACH BEING CONSIDERE
C     NVTXLO     = LAYER NUMBER BELOW LOWER VERTEX LAYER          (DIMENS
C     NVTXUP     = LAYER NUMBER IN WHICH RAY VERTEXES             (DIMENS
C     NZONE      =
```

```
C                COMPUTATIONAL VARIABLE                                  *
C     ONMIAE     = ONE MINUS ALPHAE                                      (DIMENS*
C     ONMIAS     = ONE MINUS ALPHAS                                      (DIMENS*
C     P          = DUMMY VARIABLE FOR COMPUTING PATHLENGTHS               *
C     PATHLN( )  = PATH LENGTH OF RAY TYPE                                *
C     PDE    ( ) =                                                        *
C     PDS    ( ) = PROBABILITY OF DETECTION BY SEARCHER                   (DIMENS*
C     PE     ( ) = PROBABILITY OF EVASION AS A FUNCTION OF STATE          (DIMENS*
C     PEDETN( )  = INITIAL PROBABILITY OF  EVADER DETECTION               (DIMENS*
C     PERANG     = RANGE OF DETECTION BY EVADER                           *
C     PEVADE( )  = PROBABILITY OF EVASION AT EACH BRANCH POINT            (DIMENS*
C     PGE    ( ) = PROBABILITY OF SEARCHER EVENT OTHER THAN KILL          (DIMENS*
C     PGS    ( ) = PROBABILITY OF EVADER EVENT OTHER THAN EVADE           (DIMENS*
C     PHI        = ANGLE RAY MAKES WITH SHIP AXIS                         (*
C     PIE    ( ) = SEE -PRYSEV-                                          (DIMENS*
C                A-PRIORI PROBABILITY OF DETECTION BY EVADER              *
C                NOT USED OR REQUIRED                                     *
C     PIS    ( ) = SEE -PRYSSE-                                          (DIMENS*
C                A-PRIORI PROBABILITY OF DETECTION BY SEARCHER            *
C                NOT USED OR REQUIRED                                     *
C     PKDS   ( ) = PROBABILITY OF KILL GIVEN DETECTION BY SEARCHE        (DIMENS*
C     PKILL  ( ) = PROBABILITY OF KILL AT EACH POINT ON BRANCH            (DIMENS*
C     PL         = PATH LENGTH OF RAY                                     *
C     PNE    ( ) = NOISE POWER SPECTRUM AS SEEN BY EVADER                 *
C     PNS    ( ) = NOISE POWER SPECTRUM AS SEEN BY SEARCHER               *
C     POE    ( ) = SEE -PRNOEV-                                          (DIMENS*
C                A-PRIORI PROBABILITY THAT NO DETECTION BY EVADER         *
C                NOT USED OR REQUIRED                                     *
C     POS    ( ) = SEE -PRNOSE-                                          (DIMENS*
C                A-PRIORI PROBABILITY OF NO DETECTION BY SEARCHER         *
C                NOT USED OR REQUIRED                                     *
C     POWRCD     = COMPUTATIONAL VARIABLE                                 *
C     PPATH  ( ) = PROBABILITY THAT POINT ON BRANCH IS PASSED             (DIMENS*
C     PPATHM     = PROBABILITY OF REACHING THE PRESENT POINT              (DIMENS*
C                I. E. P-OF-PATH FOR N-MINUS-ONE                          *
C     PRANGE( )  = RANGE OF FIRST DETECTION BY EVADER                     *
C     PRANGS( )  = RANGE OF FIRST DETECTION BY SEARCHER                   *
C     PROEVA( )  =  PROBABILITY OF DETECTION BY EVADER                    (DIMENS*
C     PROSER( )  =  PROBABILITY OF DETECTION BY SEARCHER                  (DIMENS*
C     PRNOEV( )  =  EVADER  DECISION PROBABILITY OF NO-DETECTION          (DIMENS*
C     PRNOSE( )  = SEARCHER DECISION PROBABILITY OF NO-DETECTION          (DIMENS*
C     PRNTEV     = PROBABILITY OF NOT BEING DETECTED BY  EVADER           (DIMENS*
C     PRNTSE     = PROBABILITY OF NOT BEING DETECTED BY SEARCHER          (DIMENS*
C     PRPTEV( )  = PROBABILITY OF PATH  OF DETECTIONS FOR  EVADER        (DIMENS*
C     PRPTSE( )  = PROBABILITY OF PATH  OF DETECTIONS FOR SEARCHE        (DIMENS*
C     PRYSEV( )  =  EVADER  DECISION PROBABILITY OF    DETECTION         (DIMENS*
C     PRYSSE( )  = SEARCHER DECISION PROBABILITY OF    DETECTION         (DIMENS*
C     PSDETN( )  = INITIAL PROBABILITY OF SEARCHER DETECTION              (DIMENS*
C     PSRANG     = RANGE OF DETECTION BY SEARCHER                         *
C     PTE    ( ) = SIGNAL POWER SPECTRUM AS SEEN BY EVADER                *
C     PTS    ( ) = SIGNAL POWER SPECTRUM AS SEEN BY SEARCHER              *
C     PXE    ( ) =  EVADER  X-POSITION AT POINT ON BRANCH                 *
C     PXS    ( ) = SEARCHER X-POSITION AT POINT ON BRANCH                 *
C     PYE    ( ) =  EVADER  Y-POSITION AT POINT ON BRANCH                 *
C     PYS    ( ) = SEARCHER Y-POSITION AT POINT ON BRANCH                 *
C     PZE    ( ) =  EVADER  Z-POSITION AT POINT ON BRANCH                 *
C     PZS    ( ) = SEARCHER Z-POSITION AT POINT ON BRANCH                 *
C     Q1         = DUMMY VARIABLE                                         *
C     Q2         = DUMMY VARIABLE                                         *
C     RAD        =                                                        *
C                = COMPUTATIONAL VARIABLE                                 *
C     RANGE  ( ) = HORIZONTAL RANGE BETWEEN SHIPS AT POINT ON BRANCH      *
C     RANGEC( )  = COMPUTED RANGE FOR RAY TYPE                            *
C     RANGEH     = HORIZONTAL RANGE BETWEEN SHIPS                         *
C     RANGET     = HORIZONTAL TRAVEL OF RAY PORTION                       *
C     RC         =                                                        *
C     RCJ        = RANGE OF CLOSE                                         *
C     RCZ1       = CONVERGENT ZONE RANGE                                  *
C     RCZ2       = CONVERGENT ZONE RANGE                                  *
C     RNGMOD( , )= HORIZONTAL RANGE OF RAY FOR EACH MODE OF PROPAGATION   *
C     RNGSTR     = RANGE OF CLOSING TO INITIALIZE RNGCLS PROPERLY         *
C     RSD        =                                                        *
```

```
C                COMPUTATIONAL VARIABLE                                  *
C      RSD1      =                                                       *
C                COMPUTATIONAL VARIABLE                                  *
C      SCSD      =                                                       *
C                COMPUTATIONAL VARIABLE                                  *
C      SDCON     = WAVE HEIGTH PARAMETER                                 *
C      SDCON     = SURFACE DUCT CONSTANT                                 *
C      SDEPTH    = SEE -DEPSER-                                          *
C      SE1       = CONSTANT SPEED OF EVADER                              (K*
C      SIGPLS( , )= SIGNAL TO NOISE RATIO PLUS ONE FOR SHIPS             (DIMENS*
C      SINPHI( ) = SINE OF STEERING DEPRESSION ANGLES FOR SHIPS          (DIMENS*
C      SINRAD( ) = SINE OF AZIMUTHAL SIGNAL ARRIVAL ANGLE                (DIMENS*
C      SLOPEJ( ) = SLOPE IN LAYER                                        (SEC**2/*
C      SM        = SUMMATION VARIABLE                                    *
C      SMPEDT    = SUMMATION FOR COMPUTING AVERAGE EVADED DETECTI        (DIMENS*
C      SMPSDT    = SUMMATION FOR COMPUTING AVERAGE SEARCHER DETEC        (DIMENS*
C      SMTCNE    = TOTAL TIME EVADER IS IN CONTACT WITH SEARCHER          *
C      SMTCNS    = TOTAL TIME SEARCHER IS IN CONTACT WITH EVADER          *
C      SPAFUN    = FUNCTION TO COMPUTE  THE SPA REQUIRED                 *
C      SPATEV    = PROPAGATION SPEED AT EVADER                           (K*
C      SPATSE    = PROPAGATION SPEED AT SEARCHER                         *
C      SPDTER( ) = SPEED OF PROPAGATION AT SHIPS                         (K*
C      SPDVER    = PROPAGATION SPEED AT VERTEX OF RAY                    (K*
C      SPI( )    = SPREADING LOSS CONSTANT                               (DIMENS*
C      SPT( )    = SPREADING LOSS CONSTANT                               (DIMENS*
C      SPVRSQ    = VERTEX VELOCITY OF RAY SQUARED                        ((K-YD2*
C      SQAREA( ) = ARRAY CONSTANT                                        (YD-S*
C      SQRTZL    =                                                       *
C                COMPUTATIONAL VARIABLE                                  *
C      SS1       = CONSTANT SPEED OF SEARCHER                            (K*
C      STATD     = SEE -NDSTAT-                                          *
C      STATE     = SEE -NESTAT-                                          *
C      STNPEV    = SIGNAL TO NOISE RATIO PLUS ONE FOR EVADER             (DIMENS*
C      STNPSE    = SIGNAL TO NOISE RATIO PLUS ONE FOR SEARCHER           (DIMENS*
C      SUMEVA    = ACCUMLATIVE PROBABILITY OF EVASION                    (DIMENS*
C      SUMKIL    = ACCUMLATIVE PROBABILITY OF EVASION                    (DIMENS*
C      SUMPTH    = SUMMATION OF PATH PROBABILITIES AT BRANCH ENDS        (DIMENS*
C      SUMTOT    = A SUMMATION OF ALL POSSIBILE PROBABILITIES            (DIMENS*
C      SURANG( ) = SURFACE ANGLE FOR THE RAY TYPE                        (*
C                NOT USED OR REQUIRED                                    *
C      TARRIV( , )= TRANSFORMED ARRIVAL ANGLE VECTOR                     (DIMENS*
C      TCONEN( ) = TIME  EVADER  IS IN CONTACT AT BRANCH POINT           *
C      TCONSN( ) = TIME SEARCHER IS IN CONTACT AT BRANCH POINT           *
C      TCZAV1    =           AVERAGE OF CONVERGENT ZONE ANGLES           (*
C      TCZAV2    =           AVERAGE OF CONVERGENT ZONE ANGLES           (*
C      TI        = STARTING ANGLE (IN DEGREES) OF RAY FROM SEARCHER      *
C      TI        = TIME INCREMENT FOR RAY TO GO X K-YD                   *
C      TIMCON    = PROPAGATION TIME OF RAY                               *
C      TIR   ( ) = INITIAL ANGLE OF RAY WITH RANGEH                      (*
C      TMATEV( , )= TRANSFORMATION MATRIX FOR EVADER                     (DIMENS*
C      TMATRX(,,) = TRANSFORMATION MATRIX                                (DIMENS*
C      TMATSE( , )= TRANSFORMATION MATRIX FOR EVADER                     (DIMENS*
C      TSTOKY    = FUNCTION TO CHANGE KNOTS TO KILO-YARDS/SECOND         *
C      TTR   ( ) = ARRIVAL ANGLE OF RAY TYPE                             (*
C      TVECTR( , )= TRANSFORMED BEAM-STEERING-DIRECTION VECTOR           (DIMENS*
C      TWOALE    = TWO TIMES ALPHAE**2                                   (DIMENS*
C      TWOALS    = TWO TIMES ALPHAS**2                                   (DIMENS*
C      UC        = VELOCITY OF PROPAGATION (IN K-YD/SEC) AT POINT IN LAY*
C      V         = PROPAGATION SPEED AT END POINT OF RAY                 (K*
C      VAREVA( ) = VARIANCE OF SMOOTHED S/N FOR EVADER                   *
C      VARSER( ) = VARIANCE OF SMOOTHED S/N FOR SEARCHER                 *
C      VXE   ( ) = X-COMPONENT OF EVADER VELOCITY AT POINT ON BRANCH     (K*
C      VXEINT    = INITIAL EVADER SPEED IN X-DIRECTION                   (K*
C      VXS   ( ) = X-COMPONENT OF SEARCHER VELOCITY AT POINT ON BRANCH   (K*
C      VYE   ( ) = Y-COMPONENT OF EVADER VELOCITY AT POINT ON BRANCH     (K*
C      VYEINT    = INITIAL EVADER SPEED IN Y-DIRECTION                   (K*
C      VYS   ( ) = Y-COMPONENT OF SEARCHER VELOCITY AT POINT ON BRANCH   (K*
C      X         = RANGE (IN K-YD) OF HORIZONTAL TRAVEL OF RAY WITHIN LA*
C      X         = FRACTIONAL PART OF LAYER                              *
C      XE        = SIGNAL TO NOISE RATIO AT EVADER                       (DIMENS*
C      XS        = SIGNAL TO NOISE RATIO AT SEARCHER                     (DIMENS*
```

```
C      Y1          =                                                        *
C      Y2          =                                                        *
C      Z1          = STARTING DEPTH (IN K-YD) OF RAY                        *
C      Z1B         = STARTING DEPTH (IN K-YD) OF CONVERGENT ZONE RAY        *
C      Z1B         = DEPTH (IN K-YD) OF EVADER                              *
C      Z2          = ENDING DEPTH (IN K-YD) OF RAY                          *
C      Z2B         = ENDING DEPTH (IN K-YD) OF CONVERGENT ZONE RAY          *
C      Z2B         = DEPTH OF EVADER                                        *
C      ZB          = DEPTH (IN K-YD) OF POINT WITHIN LAYER TO WHICH RAY IS  *
C      ZL          = DEPTH OF MAX OR MIN POINT IN VELOCITY PROFILE          *
C      ZONACZ      = MODIFIED VALUE OF ACZ FOR SIMPLIFICATION      (DIMENS  *
C      ZONBCZ      = MODIFIED VALUE OF BCZ FOR SIMPLIFICATION      (DIMENS  *
C      ZSTART      = STARTING DEPTH OF RAY PORTION                          *
C      ZVLO        = DEPTH (IN K-YD) OF VERTEX POINT OF CONVERGENT RAY      *
C      ZVUP        = DEPTH OF UPPER VERTEX POINT FOR RAY                    *
C                                                                           *
C***************************************************************************
C                                                                           *
C                                                                           *
      NAMELIST                                                              *
     N / AVERAG /                                                           *
     N SUMTOT, SUMKIL, SUMEVA, SUMDTH,                                      *
     N TRENOR, TREKIL, TREEVA, TREDTH,                                      *
     N CONNOR, AVEKIL, AVEEVA, AVEDTH,                                      *
     N SMPEDT, SMPSDT, SMTCNS, SMTCNE, PERANG, PSRANG                       *
     N / BOTTOM /                                                           *
     N TIR, TTR, RANGEC, PATHLN, DRDXDC, BOTANG, SPI, SPT, RANGEH           *
     N / BRANCH /                                                           *
     N NEI, NSI, NR, SUMKIL, BRNKIL, SUMEVA, BRNEVA, AVBREV, AVBRKL         *
     N / CHECKS /                                                           *
     N RCJ, HE1                                                             *
     N / CZONEN /                                                           *
     N AMLSRD, HCI, HZSD, SDCON, NCONSD,                                    *
     N RCZ1, RCZ2, TCZAV1, TCZAV2                                           *
      NAMELIST                                                              *
     N / DATAIN /                                                           *
     N ACZ   , ADI1  , ADI2  , ALPHAE, ALPHAS, ANGMAX, AR1X  , AR1Y  ,      *
     N AR1Z  , AR2X  , AR2Y  , AR2Z  , ARRAY1, ARRAY2, ARRYH1, ARRYH2,      *
     N ARRYW1, ARRYW2, B2E   , B2S   , BCZ   , BETAE , BETAS , BF1   ,      *
     N BF2   , CONLR1, CONLR2, DELANG, DELF  , DELHED, DELRNG, DELTAE,      *
     N DELTAS, DEPEVA, DEPKYD, DEPSER, DI1   , DI2   , DIFTI , DTRAD ,      *
     N DPFTEV, DPFTSE,                                                      *
     N E11   , E12   , E13   , E21   , E22   , E23   , FOE   , FOS   ,      *
     N F1E   , F1S   , F2E   , F2S   , FBWE  , FBWS  , FLN1  , FLN2  ,      *
     N FNE   , FNS   , FRES1 , FRES2 , FXE   , FXS   , HDINEV, HEDMAX,      *
     N HS1   , IO1   , IO2   , MAXLAY, MAXTIM, MECO  , NEMAX , NPCO  ,      *
     N NEWLAY,                                                              *
     N NPRINT,                                                              *
     N NSMAX , NULAPD, PDEMIN, PDSMIN, PHIE  , PHIS  , POR   , PPAMIN,      *
     N PRE   , PRK   , QTRAN1, QTRAN2, RGINEV, RI    , RNGMAX, SPDEVA,      *
     N SS    ,                                                              *
     N SPDKYD, SPDSER, THREVA, THRSER, WRANGE, ZW                           *
      NAMELIST                                                              *
     N / DATAOU /                                                           *
     N ADI1, ADI2, DI1, DI2, IO1, IO2,                                      *
     N BF1, BF2, CONLR1, CONLR2, DEPKYD, FLN1, FLN2, FNE, FNS, FXE, FXS,    *
     N ARRAYD,                                                              *
     N FULANG,                                                              *
     N SPDKYD, DUMMY1                                                       *
     N / FVALUE /                                                           *
     N COSAVE, DELRCZ, DYHITH, ARAREA, CTWOPI, SQAREA, ANGDEP, ARWIDT,      *
     N FRQLCN, FRQIDB, FRQTRN, FRQASD, FRQNCN, NUMFRQ                       *
      NAMELIST                                                              *
     N / INPUTS /                                                           *
     N ACZ, BCZ,                                                            *
     N ALPHAE, ALPHAS, ANGMAX, ARRAY1, ARRAY2, ARRYH1, ARRYH2, ARRYW1,      *
     N AR1X, AR1Y, AR1Z, AR2X, AR2Y, AR2Z,                                  *
     N ARRYW2, B2E   , B2S   , BETAE , BETAS , DELANG, DELF  , DELHED,      *
     N DPFTEV, DPFTSE,                                                      *
     N DELRNG, DELTAE, DELTAS, DEPEVA, DEPSER, DIFTI , DTRAD , FOE   ,      *
     N FOS   , F1E   , F1S   , F2E   , F2S   , FBWE  , FBWS  , FRES1 ,      *
     N FRES2 , HDINEV, HS1   , MAXLAY, MAXTIM, MECO  , NEMAX , NPCO  ,      *
```

```
*     N NPRINT,
*     N NEWLAY,
*     N NSMAX , NULAPO, PHIE  , PHIS  , PDR   , PRE   , PRK   , QTRAN1,
*     N PPAMIN, PDSMIN, PDEMIN,
*     N QTRAN2, RGINEV, RI    , RNGMAX, SPDEVA, SPDSER, WRANGE, ZW
*     N,THREVA, THRSER
*      NAMELIST
*     N / NOISES /
*     N XE, XS
*     N / POWERS /
*     N PTE, PTS, PNE, PNS
*     N / TABLES /
*     N PDS, PGS, PGE
*C
*      TSTOKY( DUMMY1 ) = DUMMY1*5.6266567E-4
*      FTTOKY( DUMMY1 ) = DUMMY1/3000.0
*C
*      NEWLAY = -1
*      DGPRRD = 57.2957795
*      CONSPI = 3.14159265
*      CTWOPI = 2.0*CONSPI
*C
```

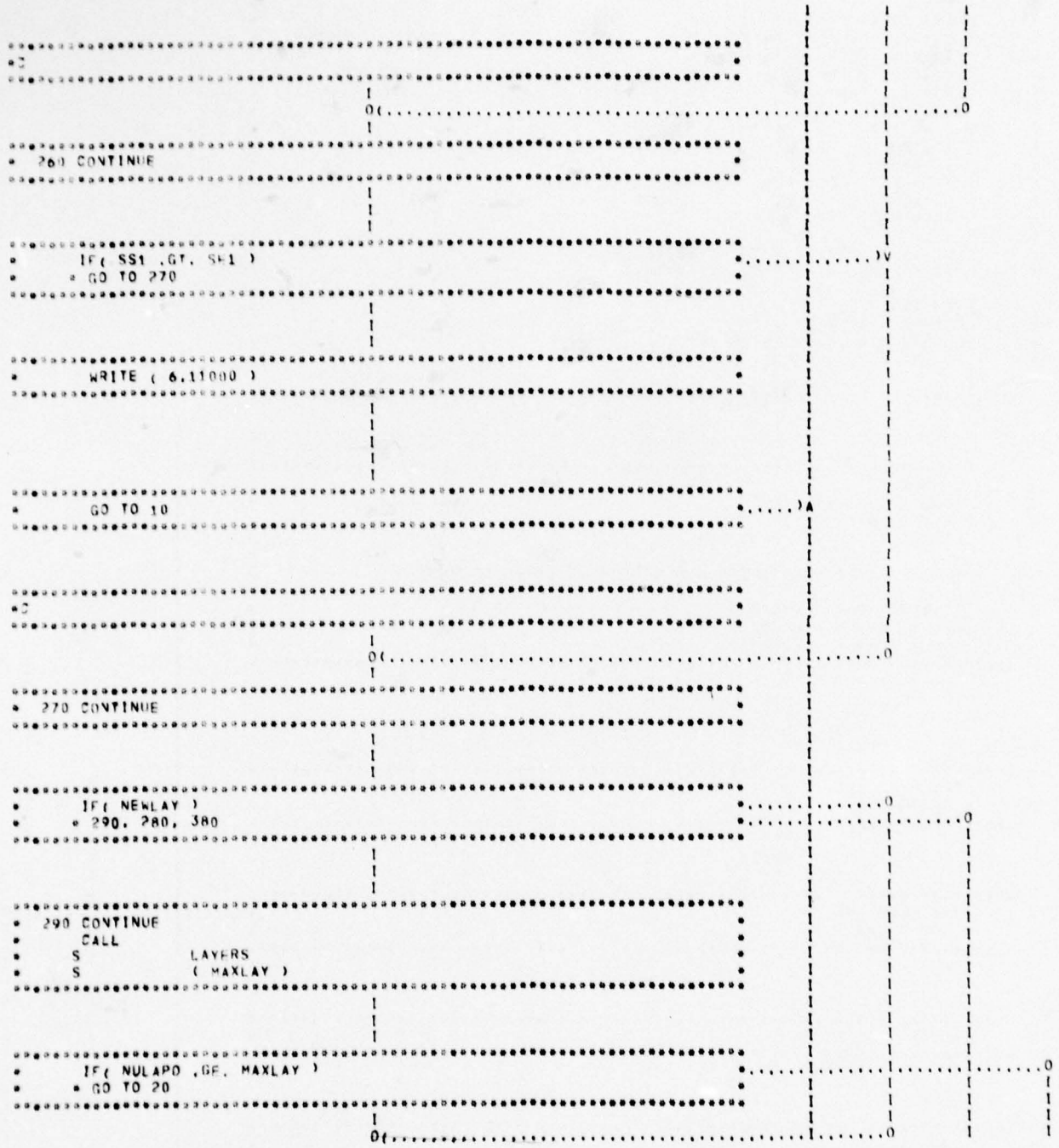
```
*   10 CONTINUE
*      READ( 5, DATAIN )
*      WRITE ( 6, 1000 )
*      WRITE( 6, INPUTS )
*C
*C     SET UP LAYER CONSTANSTS AND DIFFERENT TYPES OF RAYS
*C
*      DELRAD = DELANG/DGPRRD
*      SF1 = TSTOKY( SPDEVA )
*      SS1 = TSTOKY( SPDSER )
```

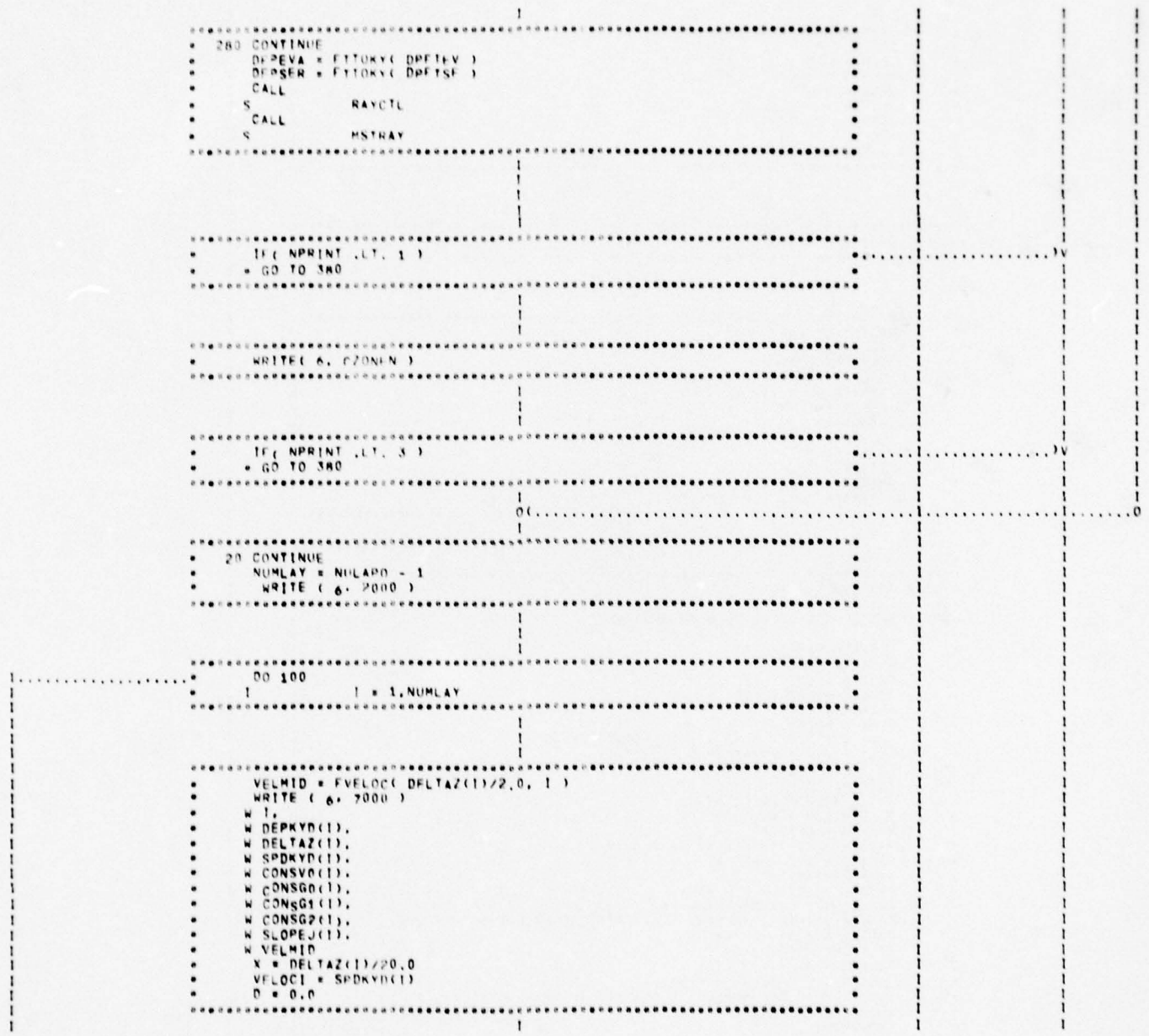
```
*      IF( NPCO .NE. 2 )
*     * GO TO 270
```

```
*      IF( MECO .NE. 2 )
*     * GO TO 260
```

```
*      WRITE ( 6, 10000 )
```

```
*       GO TO 10
```


C
260 CONTINUE
IF(SS1 .GT. SH1)
GO TO 270
WRITE (6,11000)
GO TO 10
C
270 CONTINUE
IF(NEWLAY)
290, 280, 380
290 CONTINUE
CALL
S LAYERS
S (MAXLAY)
IF(NULAPO .GE. MAXLAY)
GO TO 20


280 CONTINUE
DEPEVA = FTTOKY(DPETEV)
DEPSER = FTTOKY(DPETSE)
CALL
S RAYCTL
CALL
S MSTRAY
IF(NPRINT .LT. 1)
GO TO 380
WRITE(6, FZONEN)
IF(NPRINT .LT. 3)
GO TO 380
20 CONTINUE
NUMLAY = NULAPO - 1
WRITE (6, 2000)
DO 100
I I = 1,NUMLAY
VELMID = FVELOC(DELTAZ(I)/2.0, I)
WRITE (6, 7000)
W I,
W DEPKYD(I),
W DELTAZ(I),
W SPDKYD(I),
W CONSV0(I),
W CONSG0(I),
W CONSG1(I),
W CONSG2(I),
W SLOPEJ(I),
W VELMID
X = DELTAZ(I)/20.0
VELOCI = SPDKYD(I)
D = 0.0

```
      DO 100
     !        J = 1,20
```

```
      D = D + X
      UC = FVELOC( D, I )
      DELTAV = VELOCI - UC
      VELOCI = UC
      SLOPEN = DELTAV/X
      WRITE ( 6, 3000 )
     W UC
     W , DELTAV, SLOPEN
```

```
  100 CONTINUE
```

```
      WRITE( 6, 7000 )
     W NULAPO,
     W DEPKYD(NULAPO),
     W DELTAZ(NULAPO),
     W SPDKYD(NULAPO),
     W CONSVD(NULAPO),
     W CONSGD(NULAPO)
      IF( NULAPO .GE. MAXLAY )  CALL  DUMP
      N1 = NUANMO + 1
      WRITE( 6, 4000 )
      WRITE ( 6, 5000 )
     W ( ANGINT(J), ( RNGMOD(I,J), I = 1,6 ), J = 1,N1 )
```

```
  380 CONTINUE
C
C     SET UP CONSTANTS USED IN PROGRAM
C
      NFMAX = MAXTIM
      NSMAX = MAXTIM
      ZONACZ = 60.0 - ACZ
      ZONBCZ = BCZ - ACZ
      AMAXNN = MAXTIM
      BLA1 = POR/0.24
      BLA2 = ( 1.0 - BLA1 )/3.125/CONSP1/CONSP1
      BLA3 = 6.0 + 22.0*( POR - 0.27 )
      DELRCZ = RCZ2 - RCZ1
      EDEPTH = DEPEVA
      HCI = 30.0 + HCI + HZSD
      RNGSTR = RGINEV - DELRNG
      SDEPTH = DEPSER
      WAVECN = 1.035*ALOG10( ZW ) - 44.0
      ONMIAE = 1.0 - ALPHAE
      A1 = ONMIAE*ONMIAE
      FALPHE = A1/( 1.0 - A1 )
      TWOALE = 2.0*ALPHAE*ALPHAE
      THREVA = AMIN1( THREVA, 1.4142*TWOALE*FALPHE/A1 + 1.0 )
      ONMIAS = 1.0 - ALPHAS
      A1 = ONMIAS*ONMIAS
      FALPHS = A1/( 1.0 - A1 )
      TWOALS = 2.0*ALPHAS*ALPHAS
      THRSER = AMIN1( THRSER, 1.4142*TWOALS*FALPHS/A1 + 1.0 )
      ANGDEP(1) = PHIS
      ANGDEP(2) = PHIE
      FRQLOW(1) = F1S
      FRQLOW(2) = F1E
      NARRAY(1) = ARRAY1
      NARRAY(2) = ARRAY2
      NUMFRQ(1) = ( F2S - F1S )/DELF + 1.0
      NUMFRQ(2) = ( F2E - F1E )/DELF + 1.0
      SPDTER(1) = SPATSE
      SPDTER(2) = SPATEV
```

```
      DO 400
     I          M = 1,2
C
      COSAVE(M) = COS( ANCZAV(M) )
      COSPHI(M) = COS( ANGDEP(M) )
      SINPHI(M) = SIN( ANGDEP(M) )
C
C     SET UP TRANSFORMATION MATRICES
C
      MS = M + 2
      DO 200
     I          I = 1,3
      ARRAYD(I,M) = CONSPI*( AINPUT(I,M)/3.0 )/SPDTER(M)
      A1 = EANGLE(I,M)/DGPRRD
      EULANG(I,MS) = SIN( A1 )
      EULANG(I,M)  = COS( A1 )
  200 CONTINUE
      A1 = EULANG(3,M)*EULANG(1,M)
      A2 = EULANG(3,MS)*EULANG(1,MS)
      A3 = EULANG(3,M)*EULANG(1,MS)
      A4 = EULANG(3,MS)*EULANG(1,M)
      TMATRX(1,1,M) = A1 - A2*EULANG(2,M)
      TMATRX(1,2,M) = A3 + A4*EULANG(2,M)
      TMATRX(1,3,M) = EULANG(2,MS)*EULANG(3,MS)
      TMATRX(2,1,M) =-A3*EULANG(2,M) - A4
      TMATRX(2,2,M) = A1*EULANG(2,M) - A2
      TMATRX(2,3,M) = EULANG(2,MS)*EULANG(3,M)
      TMATRX(3,1,M) =  EULANG(2,MS)*EULANG(1,MS)
      TMATRX(3,2,M) = -EULANG(2,MS)*EULANG(1,M)
      TMATRX(3,3,M) = EULANG(2,M)
      ANDISO(1,M) = DIRANG(1,M)/DGPRRD
      N2 = NUDIAN(M)
      DO 300
     I          I1 = 2,N2
      I = I1 - 1
      ANDISO(I1,M) = DIRANG(I1,M)/DGPRRD
      DLDIAN(I,M) = ANDISO(I1,M) - ANDISO(I,M)
```

```
  300 CONTINUE

C
C     SET UP THE DIFFERENT FREQUENCY CONSTANTS
C
      F = FRQLOW(M)
      FRQLOW(M) = FRQLOW(M) - DELF
      N1 = NUMFRQ(M) + 1

      DO 400
     I          J = 1,N1

      FRQSIG(J,M) = F
      FRQNOS(J,M) = F
      A2 = SQRT(F)
      FRQASP(J,M) = SPCON*A2
      A1 = F*F
      FRQLCN(J,M) = ( 40.0/( 4100.0 + A1 ) + 2.75E-4 )*A1
      A1 = F/FRQRES(M)
      A2 = 1.0/(1.0 +(ARQTRN(M)*( A1 - 1.0/A1 ) ) **2 )
      FRQTRN(J,M) = A2
      A3 = ALOG10( F )
      FRQNCN(J,M)=(ALOG10(WAVECN-1.667*A3)/2.0+ALOG10(FLONOS(J,M)))*A2
      FRQDB(J,M) = A3 + BLA3
      DELBAF(J,M) = BAFFUN(J+1,M) - BAFFUN(J,M)
      F = F + DELF

  400 CONTINUE

      ARRAYD(1,1) = ARRAYD(1,1)/ARRAY1
      ARRAYD(1,2) = ARRAYD(1,2)/ARRAY2
      N1 = MAX0( NUMFRQ(1), NUMFRQ(2) ) + 2

      DO 500
     I          J = 1,N1

      FNE(J) = FRQNOS(J,2)
      FNS(J) = FRQNOS(J,1)
      FXE(J) = FRQSIG(J,2)
      FXS(J) = FRQSIG(J,1)
```

```
  500 CONTINUE

      SET UP CONSTANTS USED IN D- AND E- STATE TABLES

      PE(1)=PRE
      PGE(1)=1.
      PGS(1)=1.
      PKDS(1)=PRK

      PDE(2)=1.
      PE(2)=PRE
      PGS(2)=1.
      PKDS(2)=0.
      PRNDEV(2) = 0.0
      PRYSEV(2) = 1.0

      PDE(3)=0.
      PDS(3)=1.
      PE(3)=0.
      PKDS(3)=PRK
      PRNDEV(3) = 1.0
      PRNDSE(3) = 0.0
      PRYSEV(3) = 0.0
      PRYSSE(3) = 1.0

      PDS(4)=1.
      PKDS(4)=0.
      PRNDSE(4) = 0.0
      PRYSSE(4) = 1.0

      PDS(5)=0.
      PKDS(5)=0.
      PRNDSE(5) = 1.0
      PRYSSE(5) = 0.0

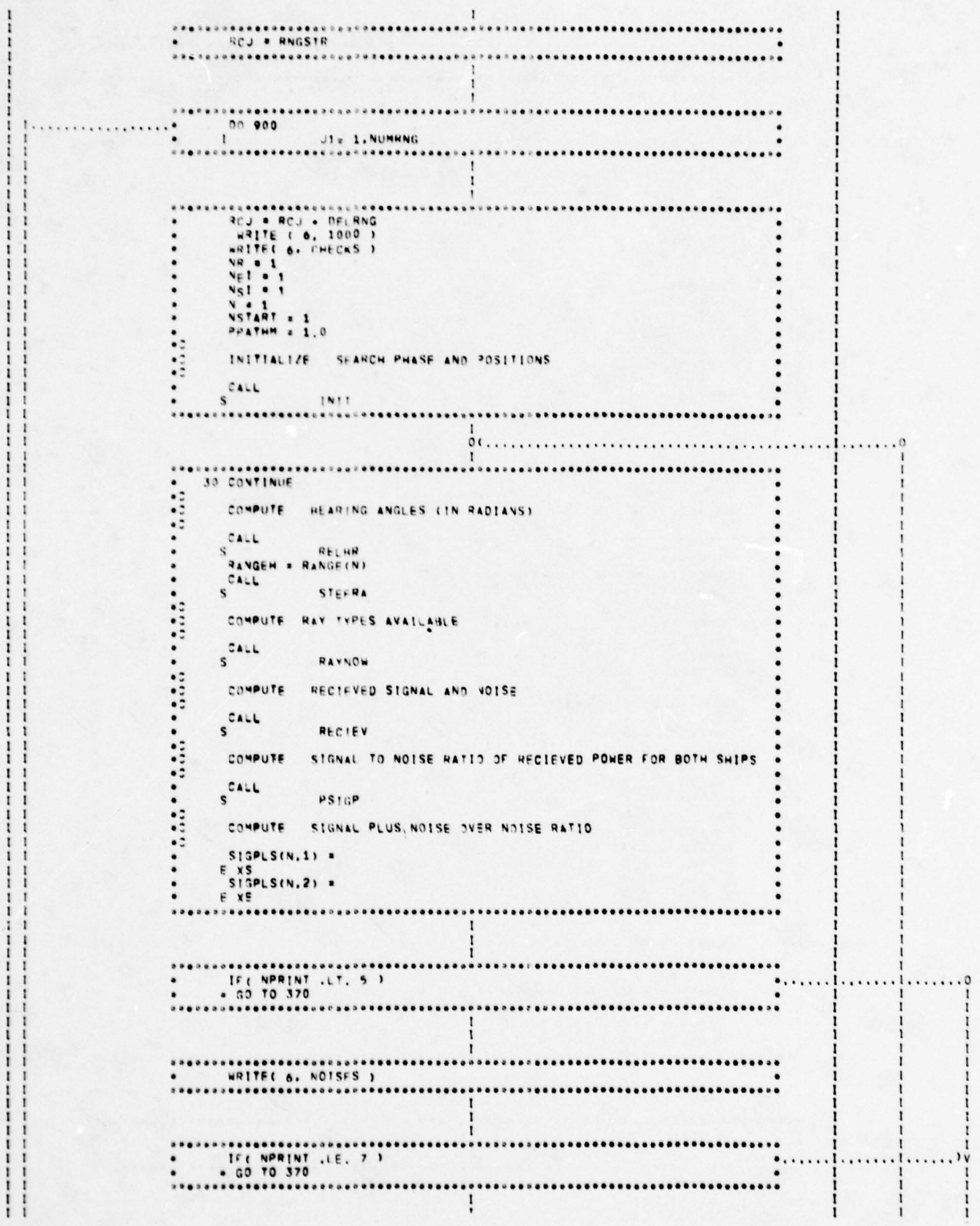
      INITIALIZE SUMMATION CONSTANTS TO ZERO

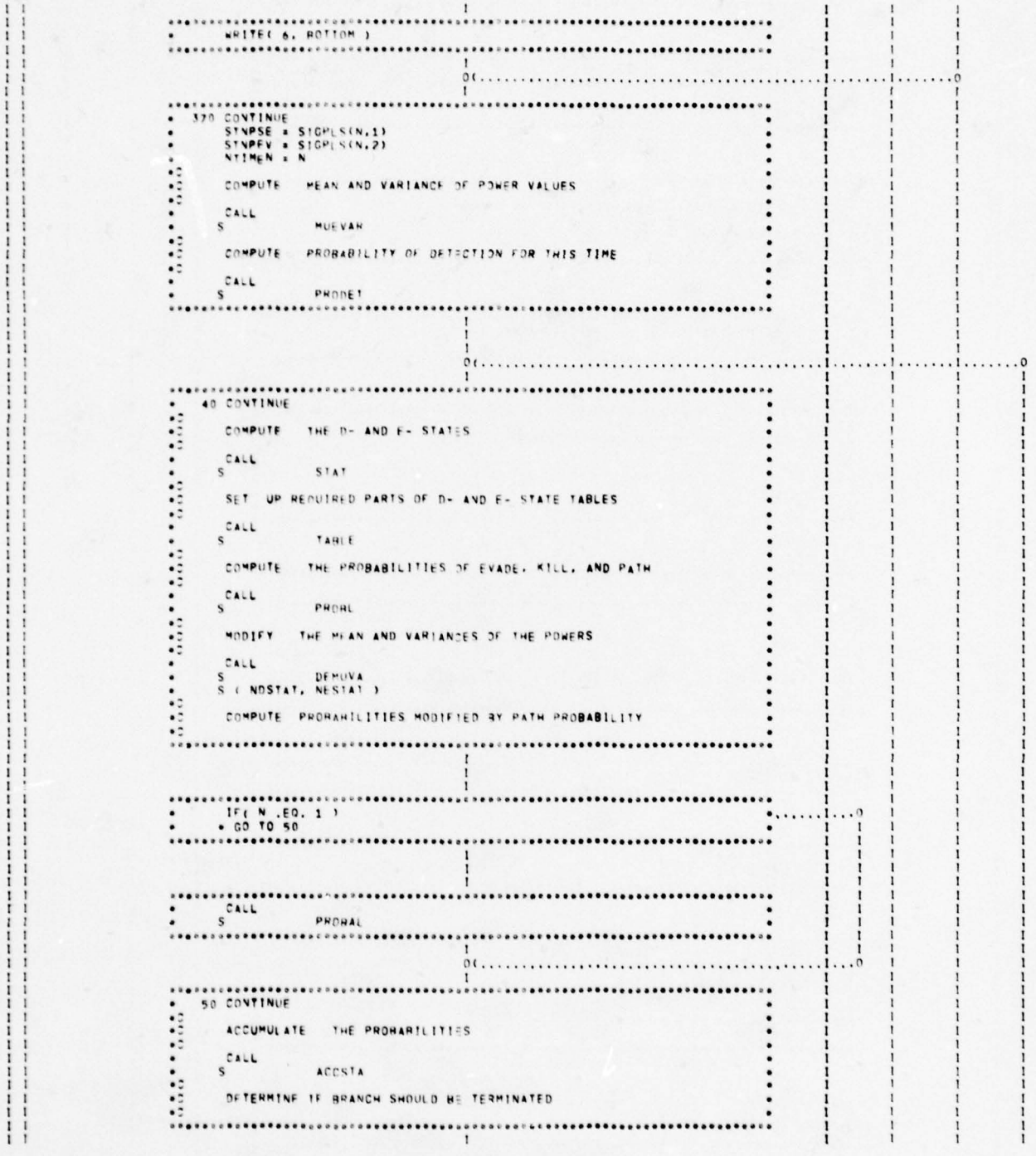
      BRNEVA = 0.0
      BRNKIL = 0.0
      PERANG = 0.0
      PSRANG = 0.0
      SMPEDT = 0.0
      SMPSDT = 0.0
      SMRANG = 0.0
      SMTCNE = 0.0
      SMTCNS = 0.0
      SUMEVA = 0.0
      SUMPTH = 0.0
      SUMKIL = 0.0

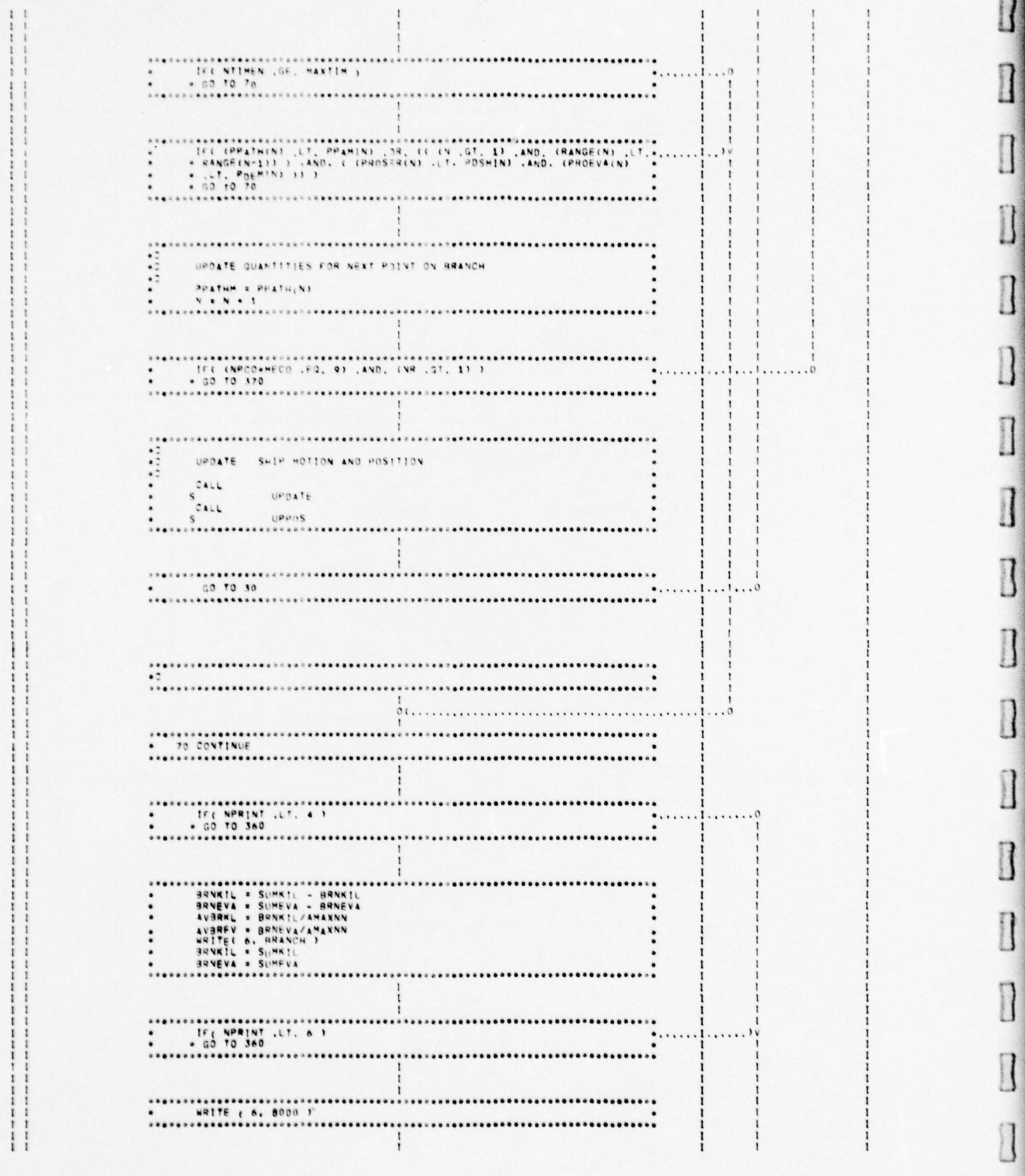
      START LOOPS THROUGH HEADINGS AND RANGES

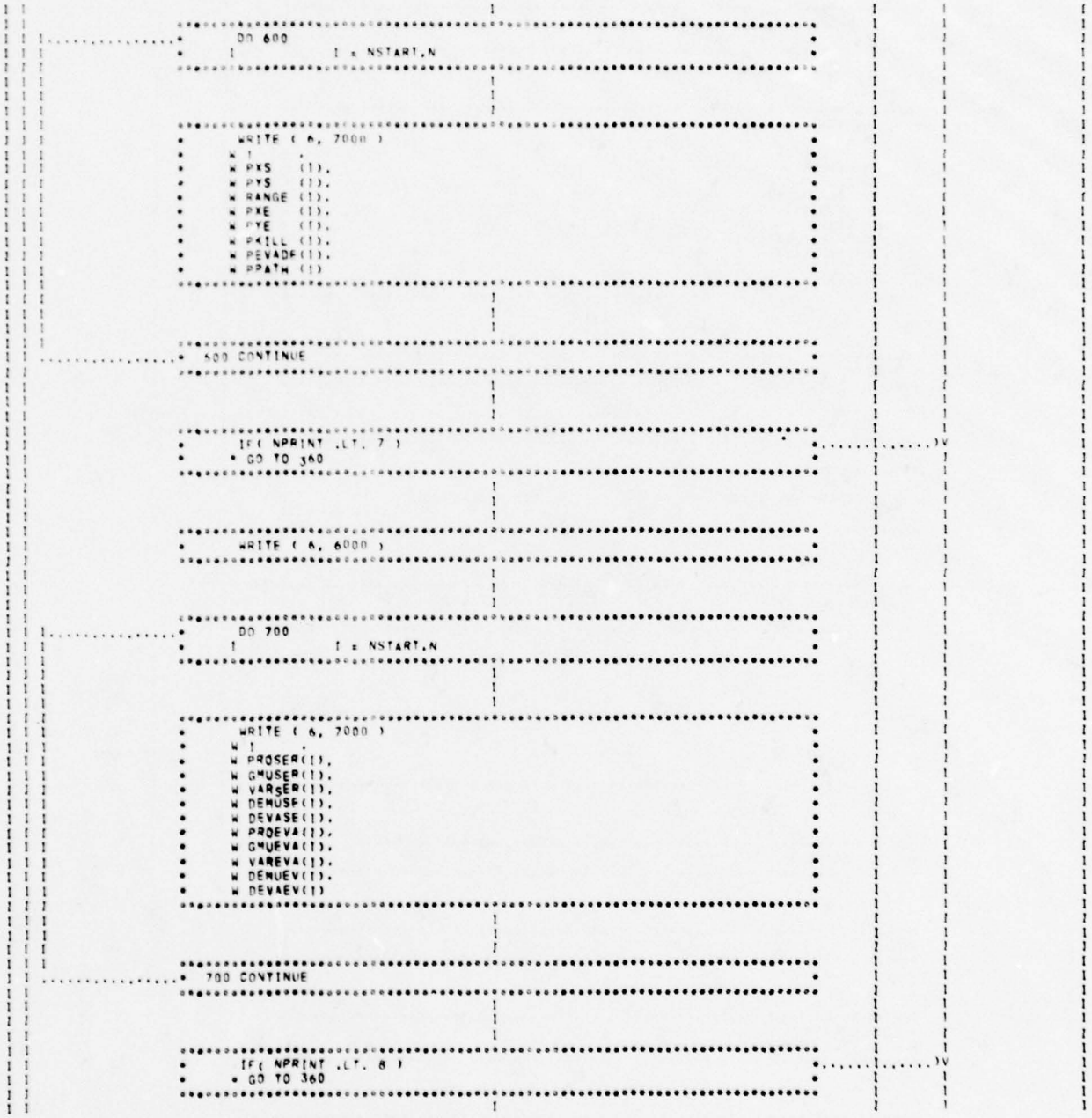
      HE1 = HDINEV
      NUMHED = ( HEDMAX - HDINEV )/DELHED + 1.0
      NUMRNG = ( RNGMAX - RGINEV )/DELRNG + 1.0
      FRLOEV = F1E - DELF
      FRLOSE = F1S - DELF

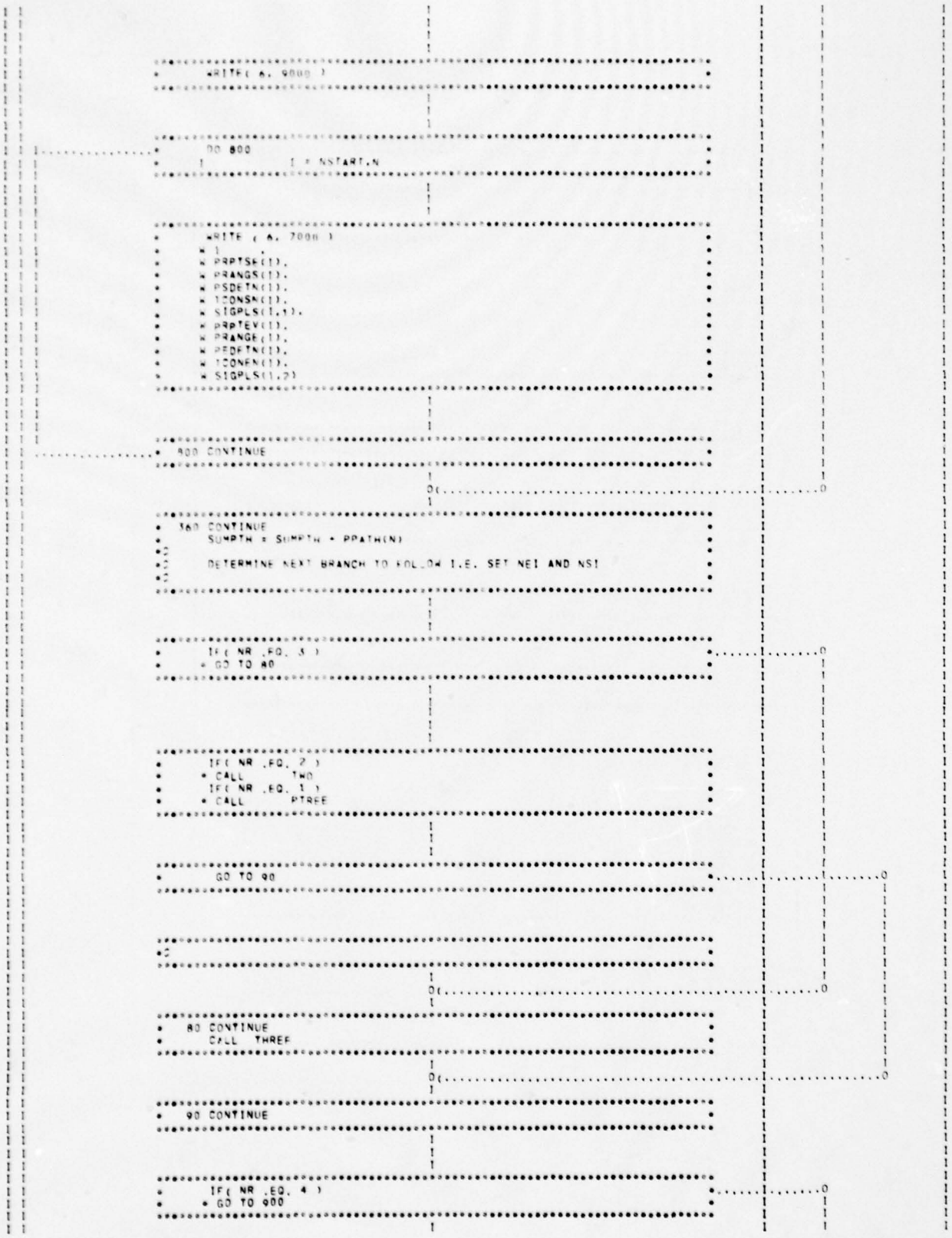
      DO 110
     I          K1= 1,NUMHED
```

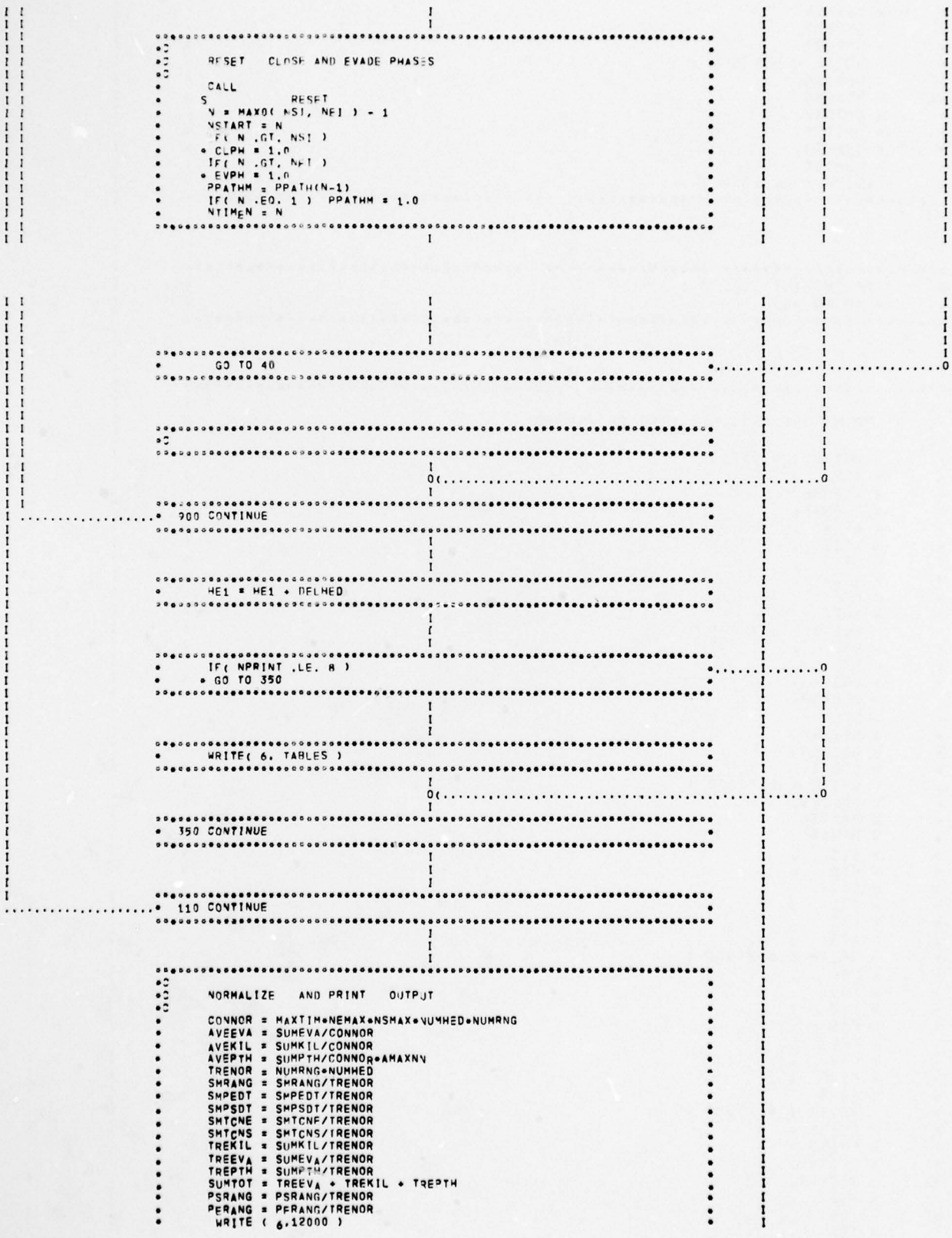

RCJ = RNGSTR
DO 900
J1= 1,NUMRNG
RCJ = RCJ + DELRNG
WRITE (6, 1000)
WRITE(6, CHECKS)
NR = 1
NEI = 1
NSI = 1
N = 1
NSTART = 1
PRATHM = 1.0
INITIALIZE SEARCH PHASE AND POSITIONS
CALL
INIT
30 CONTINUE
COMPUTE BEARING ANGLES (IN RADIANS)
CALL
RELHR
RANGEM = RANGE(N)
CALL
STERA
COMPUTE RAY TYPES AVAILABLE
CALL
RAYNOW
COMPUTE RECIEVED SIGNAL AND NOISE
CALL
RECIEV
COMPUTE SIGNAL TO NOISE RATIO OF RECIEVED POWER FOR BOTH SHIPS
CALL
PSIGP
COMPUTE SIGNAL PLUS NOISE OVER NOISE RATIO
SIGPLS(N,1) =
E XS
SIGPLS(N,2) =
E XE
IF(NPRINT .LT. 5)
GO TO 370
WRITE(6, NOISES)
IF(NPRINT .LE. 7)
GO TO 370


WRITE(6, BOTTOM)
320 CONTINUE
STNPSE = SIGPLS(N,1)
STNPFV = SIGPLS(N,2)
NTIMEN = N
COMPUTE MEAN AND VARIANCE OF POWER VALUES
CALL
S MUEVAR
COMPUTE PROBABILITY OF DETECTION FOR THIS TIME
CALL
S PRODET
40 CONTINUE
COMPUTE THE D- AND E- STATES
CALL
S STAT
SET UP REQUIRED PARTS OF D- AND E- STATE TABLES
CALL
S TABLE
COMPUTE THE PROBABILITIES OF EVADE, KILL, AND PATH
CALL
S PRORL
MODIFY THE MEAN AND VARIANCES OF THE POWERS
CALL
S DEMUVA
S (NDSTAT, NESTAT)
COMPUTE PROBABILITIES MODIFIED BY PATH PROBABILITY
IF(N .EQ. 1)
GO TO 50
CALL
S PRORAL
50 CONTINUE
ACCUMULATE THE PROBABILITIES
CALL
S ACCSTA
DETERMINE IF BRANCH SHOULD BE TERMINATED


IF(NTIMEN .GE. MAXTIM)
GO TO 70
IF((PPATH(N) .LT. PPAMIN) .OR. ((N .GT. 1) .AND. (RANGE(N) .LT. RANGE(N-1))) .AND. ((PROSER(N) .LT. PDSMIN) .AND. (PROEVA(N) .LT. PDEMIN))))
GO TO 70
C
C UPDATE QUANTITIES FOR NEXT POINT ON BRANCH
C
PPATHM = PPATH(N)
N = N + 1
IF((NPCO+MECO .EQ. 9) .AND. (NR .GT. 1))
GO TO 370
C
C UPDATE SHIP MOTION AND POSITION
C
CALL UPDATE
CALL UPDS
GO TO 30
C
70 CONTINUE
IF(NPRINT .LT. 4)
GO TO 360
BRNKIL = SUMKIL - BRNKIL
BRNEVA = SUMEVA - BRNEVA
AVBRKL = BRNKIL/AMAXNN
AVBREV = BRNEVA/AMAXNN
WRITE(6, BRANCH)
BRNKIL = SUMKIL
BRNEVA = SUMEVA
IF(NPRINT .LT. 6)
GO TO 360
WRITE (6, 8000)


DO 600
I = NSTART,N
WRITE (6, 7000)
W I
W PXS (I),
W PYS (I),
W RANGE (I),
W PXE (I),
W PYE (I),
W PKILL (I),
W PEVADE(I),
W PPATH (I)
600 CONTINUE
IF(NPRINT .LT. 7)
GO TO 360
WRITE (6, 6000)
DO 700
I = NSTART,N
WRITE (6, 7000)
W I
W PROSER(I),
W GMUSER(I),
W VARSER(I),
W DEMUSR(I),
W DEVASE(I),
W PROEVA(I),
W GMUEVA(I),
W VAREVA(I),
W DEMUEV(I),
W DEVAEV(I)
700 CONTINUE
IF(NPRINT .LT. 8)
GO TO 360


WRITE(6, 9000)
DO 800
I = NSTART,N
WRITE (6, 7000)
N I ,
N PRPTSE(I),
N PRANGS(I),
N PSDETN(I),
N ICONSN(I),
N SIGPLS(I,1),
N PRPTEV(I),
N PRANGE(I),
N PEDETN(I),
N ICONEN(I),
N SIGPLS(I,2)
800 CONTINUE
360 CONTINUE
SUMPTH = SUMPTH + PPATH(N)
C
C DETERMINE NEXT BRANCH TO FOLLOW I.E. SET NEI AND NSI
C
IF(NR .EQ. 3)
GO TO 80
IF(NR .EQ. 2)
CALL TWO
IF(NR .EQ. 1)
CALL PTREE
GO TO 90
C
80 CONTINUE
CALL THREE
90 CONTINUE
IF(NR .EQ. 4)
GO TO 900


C
C RESET CLOSE AND EVADE PHASES
C
CALL
S RESET
N = MAX0(NSI, NEI) - 1
NSTART = N
IF(N .GT. NSI)
• CLPH = 1.0
IF(N .GT. NEI)
• EVPH = 1.0
PPATHM = PPATH(N-1)
IF(N .EQ. 1) PPATHM = 1.0
NTIMEN = N
GO TO 40
C
900 CONTINUE
HE1 = HE1 + DELHED
IF(NPRINT .LE. 8)
• GO TO 350
WRITE(6, TABLES)
350 CONTINUE
110 CONTINUE
C
C NORMALIZE AND PRINT OUTPUT
C
CONNOR = MAXTIM•NEMAX•NSMAX•NUMHED•NUMRNG
AVEEVA = SUMEVA/CONNOR
AVEKIL = SUMKIL/CONNOR
AVEPTH = SUMPTH/CONNOR•AMAXNN
TRENOR = NUMRNG•NUMHED
SMRANG = SMRANG/TRENOR
SMPEDT = SMPEDT/TRENOR
SMPSDT = SMPSDT/TRENOR
SMTCNE = SMTCNE/TRENOR
SMTCNS = SMTCNS/TRENOR
TREKIL = SUMKIL/TRENOR
TREEVA = SUMEVA/TRENOR
TREPTH = SUMPTH/TRENOR
SUMTOT = TREEVA + TREKIL + TREPTH
PSRANG = PSRANG/TRENOR
PERANG = PERANG/TRENOR
WRITE (6,12000)

```
      W TREKIL,
      W TREFVA,
      W SMPEDT,
      W SMPSDT
        WRITE ( 6,13000 )
      W PERANG,
      W PSRANG,
      W SMTCNE,
      W SMTCNS,
      W TRENOR,
      W SUMTOT
       WRITE ( 6, 1000 )
       IF( NPRINT .LE. 0 )
      * GO TO 40
C
C     PRINT OUT DETAILED LIST OF INPUTS
C
        WRITE ( 6,21000 )
      W ACZ   ,
      W ALPHAE,
      W ALPHAS,
      W AR1X  ,
      W AR1Y  ,
      W AR1Z  ,
      W AR2X  ,
      W AR2Y  ,
      W AR2Z
        WRITE ( 6,22000 )
      W B2E   ,
      W B2S   ,
      W BCZ   ,
      W DELANG,
      W DELF  ,
      W DELHED,
      W DELRNG,
      W DIFTI
        WRITE ( 6,23000 )
      W DPFTEV,
      W DPFTSE,
      W DTRAD ,
      W F11   ,
      W F12   ,
      W F13   ,
      W F21   ,
      W F22   ,
      W F23
        WRITE ( 6,24000 )
      W F0E   ,
      W F0S   ,
      W F1E   ,
      W F1S   ,
      W F2E   ,
      W F2S   ,
      W FBWE  ,
      W FBWS
        WRITE ( 6,25000 )
      W FRES1 ,
      W FRES2 ,
      W HDINEV,
      W HEDMAX,
      W HS1   ,
      W I01   ,
      W I02   ,
```

```
*      W MAXLAY,                                                          *
*      W MAXTIM                                                           *
*        WRITE ( 6,26000 )                                               *
*      W MECO   ,                                                         *
*      W NEMAX  ,                                                         *
*      W NEWLAY,                                                          *
*      W NPCO   ,                                                         *
*      W NPRINT,                                                          *
*      W NSMAX  ,                                                         *
*      W NULAPO,                                                          *
*      W PDEMIN,                                                          *
*      W PDSMIN                                                           *
*        WRITE ( 6,27000 )                                               *
*      W POR    ,                                                         *
*      W PPAMIN,                                                          *
*      W PRE    ,                                                         *
*      W PRK    ,                                                         *
*      W QTRAN1,                                                          *
*      W QTRAN2,                                                          *
*      W RGINEV,                                                          *
*      W RI     ,                                                         *
*      W RNGMAX                                                           *
*        WRITE ( 6,28000 )                                               *
*      W SPDEVA,                                                          *
*      W SPDSER,                                                          *
*      W SS     ,                                                         *
*      W THREVA,                                                          *
*      W THRSER,                                                          *
*      W WRANGE                                                           *
*21000 FORMAT(                                                           *
*     F / F15.6, 99H = ACZ    = FIRST CONVERGENCE ZONE CONSTANT           *
*     F                                                                   *
*     F / F15.6, 99H = ALPHAE = SMOOTHING PARAMETER FOR EVADER            *
*     F                                                                   *
*     F / F15.6, 99H = ALPHAS = SMOOTHING PARAMETER FOR SEARCHER          *
*     F                                                                   *
*     F / F15.6, 99H = AR1X   = X-PRIME ARRAY DIMENSION FOR SEARCHER      *
*     F                                                                   *
*     F / F15.6, 99H = AR1Y   = Y-PRIME ARRAY DIMENSION FOR SEARCHER      *
*     F                                                                   *
*     F / F15.6, 99H = AR1Z   = Z-PRIME ARRAY DIMENSION FOR SEARCHER      *
*     F                                                                   *
*     F / F15.6, 99H = AR2X   = X-PRIME ARRAY DIMENSION FOR EVADER        *
*     F                                                                   *
*     F / F15.6, 99H = AR2Y   = Y-PRIME ARRAY DIMENSION FOR EVADER        *
*     F                                                                   *
*     F / F15.6, 99H = AR2Z   = Z-PRIME ARRAY DIMENSION FOR EVADER        *
*     F                                                                   *
*     F )                                                                 *
*22000 FORMAT(                                                           *
*     F / F15.6, 99H = B2E    = INTEGRATION BANDWIDTH OF POST-DETECTION O*
*     FF EVADER                                                          *
*     F / F15.6, 99H = B2S    = INTEGRATION BANDWIDTH OF POST-DETECTION O*
*     FF SEARCHER                                                        *
*     F / F15.6, 99H = BCZ    = SECOND CONVERGENCE ZONE CONSTANT          *
*     F                                                                   *
*     F / F15.6, 99H = DELANG = ANGLE STEP SIZE FOR RAY PRE-TRACE TABLE   *
*     F                                                                   *
*     F / F15.6, 99H = DELF   = FREQUENCY INCREMENT                       *
*     F                                                                   *
*     F / F15.6, 99H = DELHED = CHANGE IN HEADING OF EVADER               *
*     F                                                                   *
*     F / F15.6, 99H = DELRNG = CHANGE IN RANGE OF CLOSEST APPROACH       *
*     F                                                                   *
*     F / F15.6, 99H = DIFTI  = TIME BETWEEN POINTS ON A BRANCH           *
*     F                                                                   *
*     F )                                                                 *
```

```
*23000 FORMAT(
*     F / F15.6, 99H = DPFTEV = EVADER DEPTH
*     F
*     F / F15.6, 99H = DPFTSE = SEARCHER DEPTH
*     F
*     F / F15.6, 99H = DTRAD  = ANGLE INCREMENT FOR RADIATED SIGNAL
*     F
*     F / F15.6, 99H = E11    = FIRST  EULER ANGLE FOR SEARCHER (Z AXIS R
*     FOTATION)
*     F / F15.6, 99H = E12    = SECOND EULER ANGLE FOR SEARCHER (LINE OF
*     FNODES ROTATION)
*     F / F15.6, 99H = E13    = THIRD  EULER ANGLE FOR SEARCHER (Z-PRIME
*     FAXIS ROTATION)
*     F / F15.6, 99H = E21    = FIRST  EULER ANGLE FOR  EVADER  (Z AXIS R
*     FOTATION)
*     F / F15.6, 99H = E22    = SECOND EULER ANGLE FOR  EVADER  (LINE OF
*     FNODES ROTATION)
*     F / F15.6, 99H = E23    = THIRD  EULER ANGLE FOR  EVADER  (Z-PRIME
*     FAXIS ROTATION)
*     F )
*24000 FORMAT(
*     F / F15.6, 99H = F0E    = PRE-DETECTION FILTER CENTER FREQUENCY FOR
*     F EVADER
*     F / F15.6, 99H = F0S    = PRE-DETECTION FILTER CENTER FREQUENCY FOR
*     F SEARCHER
*     F / F15.6, 99H = F1E    = LOWER FREQUENCY LIMIT OF EVADER EQUIPMENT
*     F
*     F / F15.6, 99H = F1S    = LOWER FREQUENCY LIMIT OF SEARCHER EQUIPME
*     FNT
*     F / F15.6, 99H = F2E    = UPPER FREQUENCY LIMIT OF EVADER EQUIPMENT
*     F
*     F / F15.6, 99H = F2S    = UPPER FREQUENCY LIMIT OF SEARCHER EQUIPME
*     FNT
*     F / F15.6, 99H = FBWE   = PRE-DETECTION FILTER BANDWIDTH OF EVADER
*     F
*     F / F15.6, 99H = FBWS   = PRE-DETECTION FILTER BANDWIDTH OF SEARCHE
*     FR
*     F )
*25000 FORMAT(
*     F / F15.6, 99H = FRES1  = TRANSDUCER RESONANT FREQUENCY ON SEARCHER
*     F
*     F / F15.6, 99H = FRES2  = TRANSDUCER RESONANT FREQUENCY ON EVADER
*     F
*     F / F15.6, 99H = HDINEV = INITIAL HEADING OF EVADER
*     F
*     F / F15.6, 99H = HEDMAX = MAXIMUM HEADING OF EVADER
*     F
*     F / F15.6, 99H = HS1    = HEADING OF SEARCHER
*     F
*     F / I15,   99H = I01    = NUMBER OF DIRECTIVITY ANGLES FOR SEARCHER
*     F
*     F / I15,   99H = I02    = NUMBER OF DIRECTIVITY ANGLES FOR EVADER
*     F
*     F / I15,   99H = MAXLAY = MAXIMUM NUMBER OF LAYERS TO BE USED TO FI
*     FT PROFILE
*     F / I15,   99H = MAXTIM = MAXIMUM NUMBER OF POINTS ON A BRANCH
*     F
*     F )
*26000 FORMAT(
*     F / I15,   99H = MECO   = CONTROL PARAMETER FOR EVASION COURSE OPTI
*     FONS
*     F / I15,   99H = NEMAX  = MAXIMUM POINT ALONG BRANCH TO START EVASI
*     FON
*     F / I15,   99H = NEWLAY = CONTROL CONSTANT FOR INITIALIZATION
*     F
*     F / I15,   99H = NPCO   = CONTROL PARAMETER FOR PURSUIT COURSE OPTI
*     FON
*     F / I15,   99H = NPRINT = CONTROL PARAMETER FOR AMOUNT OF PRINTOUT
*     F
*     F / I15,   99H = NSMAX  = MAXIMUM POINT ALONG BRANCH TO START CLOSI
*     FNG
*     F / I15,   99H = NULAPO = NUMBER OF LAYERS PLUS ONE
*     F
```

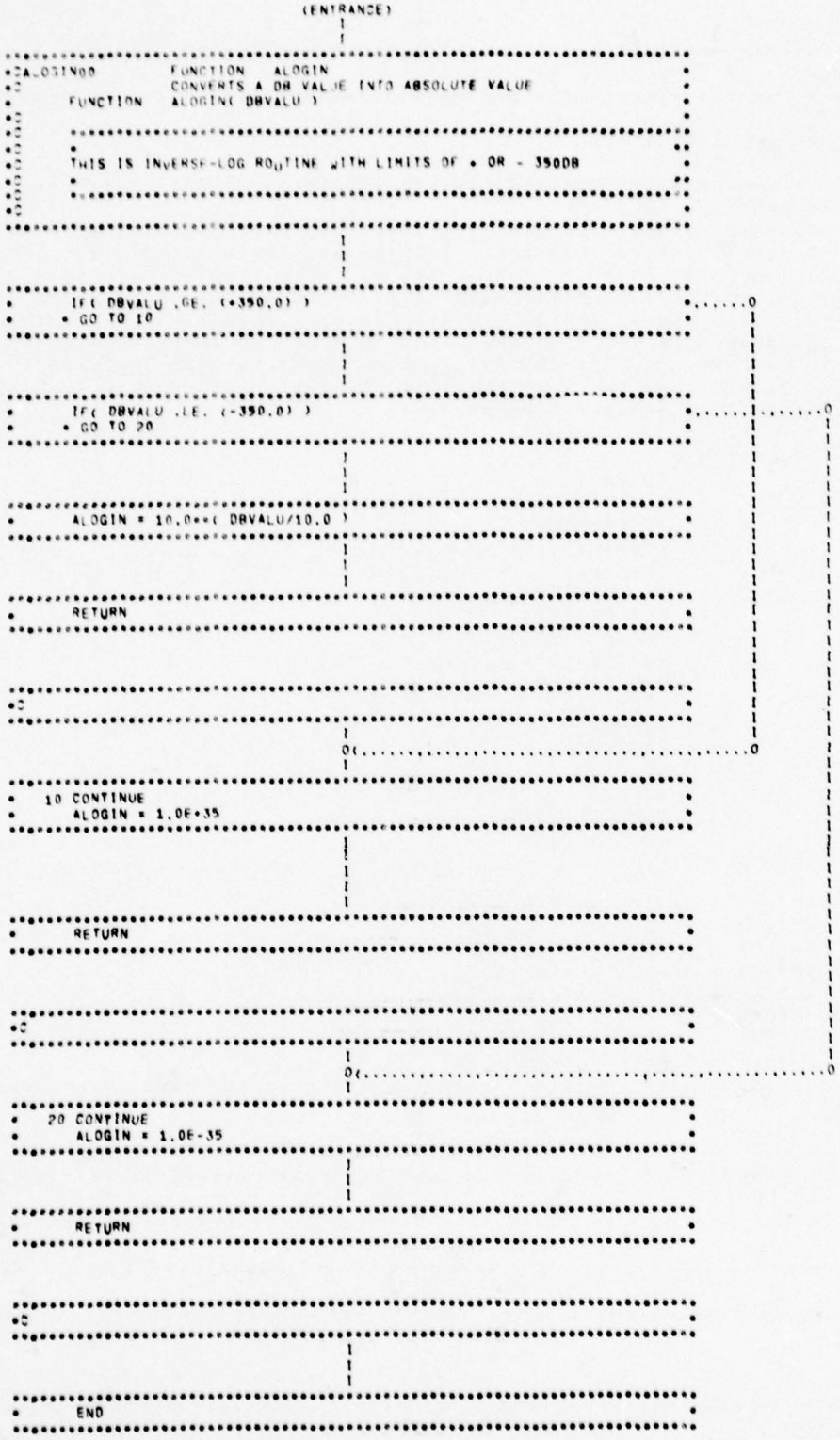
```
*      F / F15.6, 99H = PDEMIN = MINIMUM DETECTION PROBABILITY ALLOWED AF
*      FER SHIPS START SEPARATING AS SEEN BY EVADER
*      F / F15.6, 99H = PDSMIN = MINIMUM DETECTION PROBABILITY ALLOWED AF
*      FER SHIPS START SEPARATING AS SEEN BY SEARCHER
*      F )
*27000 FORMAT(
*      F / F15.6, 99H = POR    = POROSITY OF BOTTOM
*      F
*      F / F15.6, 99H = PPAMIN = MINIMUM PATH PROBABILITY TO BE CONSIDERED
*      F
*      F / F15.6, 99H = PRE    = A-PRIORI PROBABILITY OF EVASION
*      F
*      F / F15.6, 99H = PRK    = A-PRIORI PROBABILITY OF A KILL
*      F
*      F / F15.6, 99H = QTRAN1 = TRANSDUCER FIGURE OF MERIT ON SEARCHER
*      F
*      F / F15.6, 99H = QTRAN2 = TRANSDUCER FIGURE OF MERIT ON EVADER
*      F
*      F / F15.6, 99H = RGINEV = INITIAL RANGE OF CLOSE
*      F
*      F / F15.6, 99H = RI     = INITIAL RANGE BETEEN SHIPS AT START OF BR
*      FANCH
*      F / F15.6, 99H = RNGMAX = MAXIMUM CLOSEST APPROACH DISTANCE
*      F
*      W )
*28000 FORMAT(
*      F / F15.6, 99H = SPDEVA = SPEED OF EVADER
*      F
*      F / F15.6, 99H = SPDSER = SPEED OF SEARCHER
*      F
*      F / F15.6, 99H = SS     = SEA STATE
*      F
*      F / F15.6, 99H = THREVA = DETECTION THRESHOLD FOR THE EVADER
*      F
*      F / F15.6, 99H = THRSER = DETECTION THRESHOLD FOR THE SEARCHER
*      F
*      F / F15.6, 99H = WRANGE = WEAPON RANGE
*      F
*      W )
*
*      IF( NPRINT .LE. 9 )
*      * GO TO 340
*
*      WRITE( 6, POWERS )
*      WRITE( 6, EVALUE )
*      WRITE( 6, DATAOU )
*      CALL  PDUMP
*
*  340 CONTINUE
*      NEWLAY = 1
*
*       GO TO 10
```

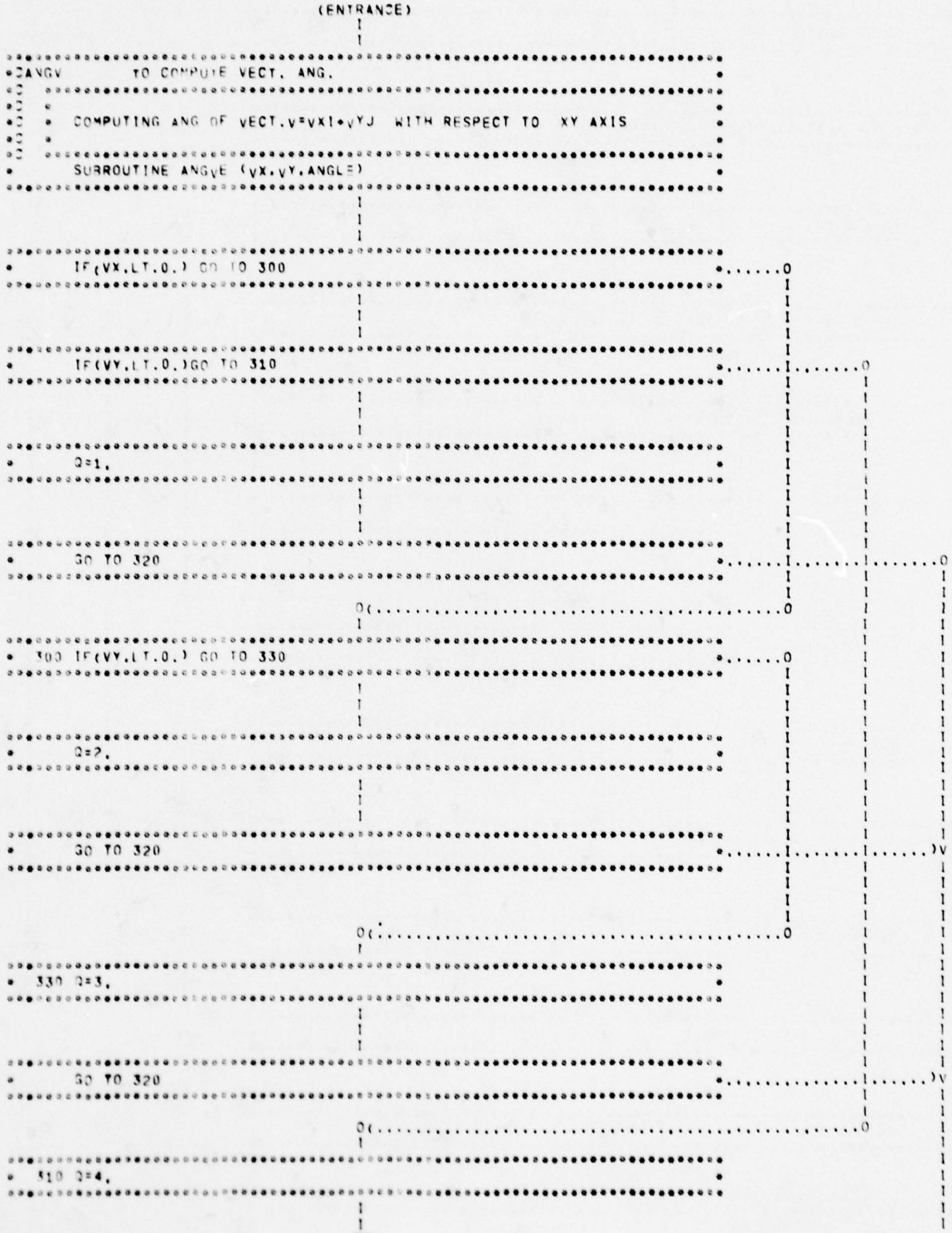
```
*C                                                                      *
* 1000 FORMAT(                                                          *
*     F 1H1 )                                                           *
* 2000 FORMAT(                                                          *
*     F    58H1LAYER  -DEPTH    -DELTA-Z  -VELOCITY        -V0        -G0 *
*     F   59H            -G1        -G2     -SLOPE  -MIDSPEED          )*
* 3000 FORMAT(                                                          *
*     F 6E20.10 )                                                       *
* 4000 FORMAT(                                                          *
*     F 52H1RADIAN ANGLE      RANGES FOR DIFFERENT BOUNCE MODES  /      *
*     F 24X, 90H DIRECT      ONE-SURFACE      ONE-BOTTOM   SURFACE-BOTTOM *
*     F  BOTTOM-SURFACE      TWO-SURFACE )                              *
* 5000 FORMAT(                                                          *
*     F 7E17.7 )                                                        *
* 6000 FORMAT(                                                          *
*     F  //59H N  PR-SEARCHER      MU-S    SIGMA-S     DEL-MU  DEL-SIGMA *
*     F   59H      PR-EVADER       MU-E    SIGMA-E     DEL-MU  DEL-SIGMA)*
* 7000 FORMAT(                                                          *
*     F 13, 1P5E11.3, 5X, 5E11.3 )                                      *
* 8000 FORMAT(                                                          *
*     F  / 58H N   SEARCHER-X        -Y       RANGE   EVADER-X         -Y *
*     F   59H PROBABIL.-KILL      -EVADE      -PATH                      )*
* 9000 FORMAT(                                                          *
*     F /  58H N       S-PATH     -RANGE -DETECTION      -TIME     -S/N+1 *
*     F   59H          E-PATH     -RANGE -DETECTION      -TIME     -S/N+1)*
*10000 FORMAT(                                                          *
*     F 51H COLLISION PURSUIT AND NORMAL ESCAPE NOT COMPATIBLE  )       *
*11000 FORMAT(                                                          *
*     F 60H COLLISION COURSE PURSUIT WILL NOT WORK WITH SLOWER SEARCHER )*
*12000 FORMAT(                                                          *
*     F 1H1                                                             *
*     F / F15.6, 94H   = AVERAGE PROBABILITY OF SEARCHER NEUTRALIZING EVA*
*     FDER                                                              *
*     F / F15.6, 94H   = AVERAGE PROBABILITY OF EVADER ESCAPING FROM SEAR*
*     FCHER                                                             *
*     F / F15.6, 94H   = AVERAGE PROBABILITY OF FIRST DETECTION FOR EVADE*
*     FR                                                                *
*     F / F15.6, 94H   = AVERAGE PROBABILITY OF FIRST DETECTION FOR SEARC*
*     FHER                                                              *
*     F )                                                               *
*13000 FORMAT(                                                          *
*     F / F15.6, 94H   = AVERAGE RANGE OF FIRST DETECTION FOR EVADER (KYD*
*     F)                                                                *
*     F / F15.6, 94H   = AVERAGE RANGE OF FIRST DETECTION FOR SEARCHER (K*
*     FYD)                                                              *
*     F / F15.6, 94H   = AVERAGE LENGTH OF TIME DURING WHICH EVADER IS DE*
*     FTECTING SEARCHER PER TIME INTERVAL (SEC)                         *
*     F / F15.6, 94H   = AVERAGE LENGTH OF TIME DURING WHICH SEARCHER IS *
*     FDETECTING EVADER PER TIME INTERVAL (SEC)                         *
*     F / F15.6, 94H   = NUMBER OF COMBINATIONS OF INITIAL HEADINGS AND C*
*     FLOSEST APPROACH DISTANCES                                        *
*     F / F15.6, 94H   = SUM OF KILL, EVADE AND RESIDUE PATH PROBABILITIE*
*     FS                                                                *
*     F 1H1 )                                                           *
*C                                                                      *
**********************************************************************
                                    I
                                    I
                                    I
**********************************************************************
*      END                                                              *
**********************************************************************
```

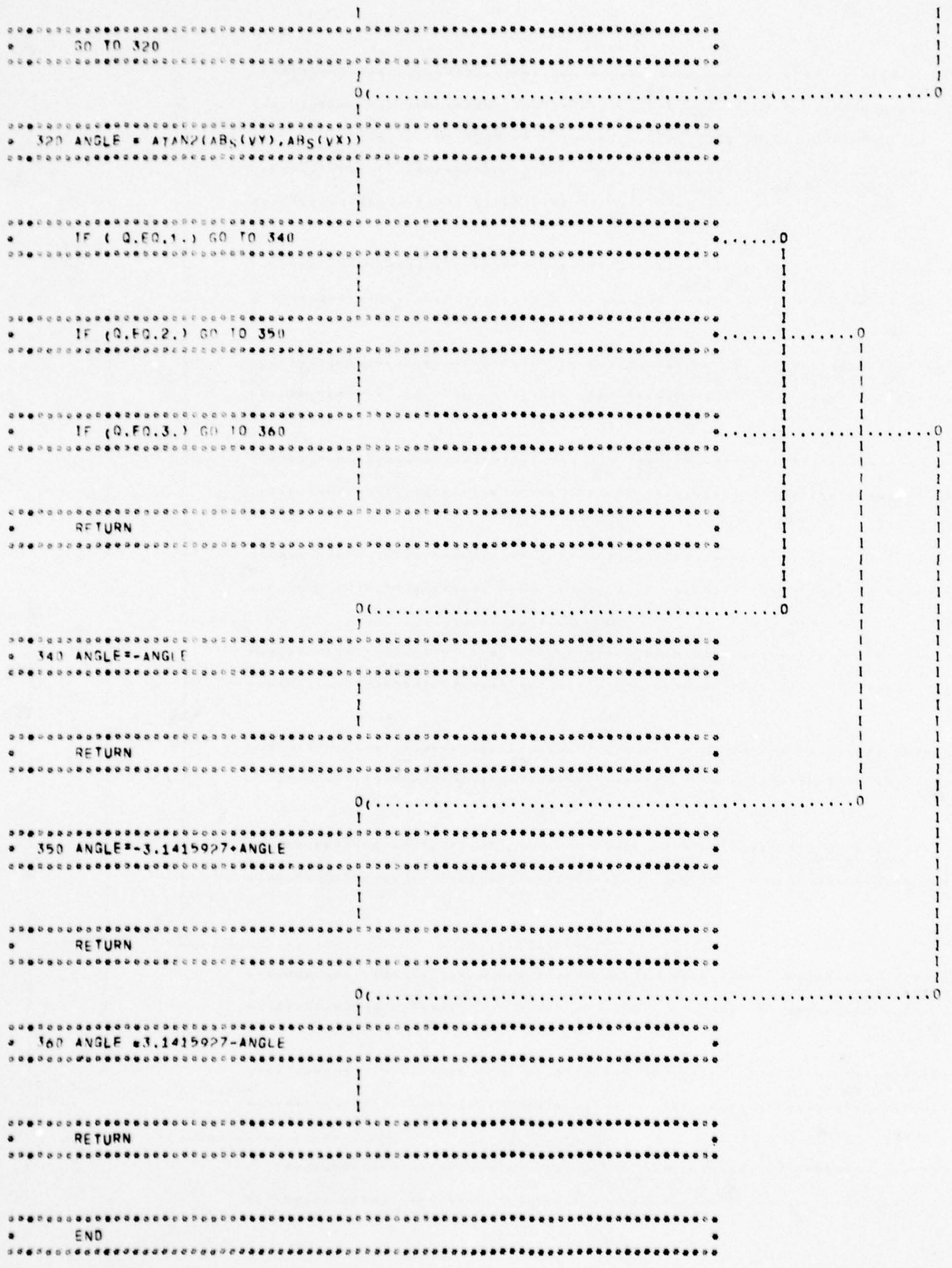
(ENTRANCE)

```
*CACCV        ACCUMULATIVE STATISTICS
*C  ***********************************************************************
*C  *                                                                     *
*C  *  ACCUM. STATISTICS                                                  *
*C  *                                                                     *
*C  ***********************************************************************
*       SUBROUTINE ACCSTA                                                 *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NS1,*
*      2NE1,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,F0S,FRWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,F8SE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE;P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX
*       COMMON
*      C / STATIC /
*      C            NOSTAT,
*      C            NESTAT,
*      D            PEDETN(128)      ,
*      C            PERANG,
*      C            PPATHM,
*      D            PRANGE(128)      ,
*      D            PRANGS(128)      ,
*      D            PRPTEV(128)      ,
*      D            PRPTSE(128)      ,
*      D            PSDETN(128)      ,
*      C            PSRANG,
*      C            SMPEDT,
*      C            SMPSDT,
*      C            SMTCNE,
*      C            SMTCNS,
*      D            TCONEN(128)
*      D            TCONSN(128)
*      CDUMMYB
*C
*       PERANG = PERANG + PRANGE(N)                                       *
*       PSRANG = PSRANG + PRANGS(N)                                       *
*       SMPSDT = SMPSDT + PSDETN(N)                                       *
*       SMPEDT = SMPEDT + PEDETN(N)                                       *
*       SMTCNE = SMTCNE + TCONEN(N)                                       *
*       SMTCNS = SMTCNS + TCONSN(N)                                       *
*       SUMEVA = SUMEVA + PEVADE(N)                                       *
*       SUMKIL = SUMKIL + PKILL(N)                                        *
*************************************************************************
                                    |
                                    |
                                    |
*************************************************************************
*       RETURN                                                            *
*************************************************************************

*************************************************************************
*C                                                                        *
*************************************************************************
                                    |
                                    |
                                    |
*************************************************************************
*       END                                                               *
*************************************************************************
```


(ENTRANCE)
C ALOGIN00 FUNCTION ALOGIN
C CONVERTS A DB VALUE INTO ABSOLUTE VALUE
FUNCTION ALOGIN(DBVALU)
C
C
C THIS IS INVERSE-LOG ROUTINE WITH LIMITS OF + OR - 350DB
C
C
IF(DBVALU .GE. (+350.0))
GO TO 10
IF(DBVALU .LE. (-350.0))
GO TO 20
ALOGIN = 10.0**(DBVALU/10.0)
RETURN
C
10 CONTINUE
ALOGIN = 1.0E+35
RETURN
C
20 CONTINUE
ALOGIN = 1.0E-35
RETURN
C
END

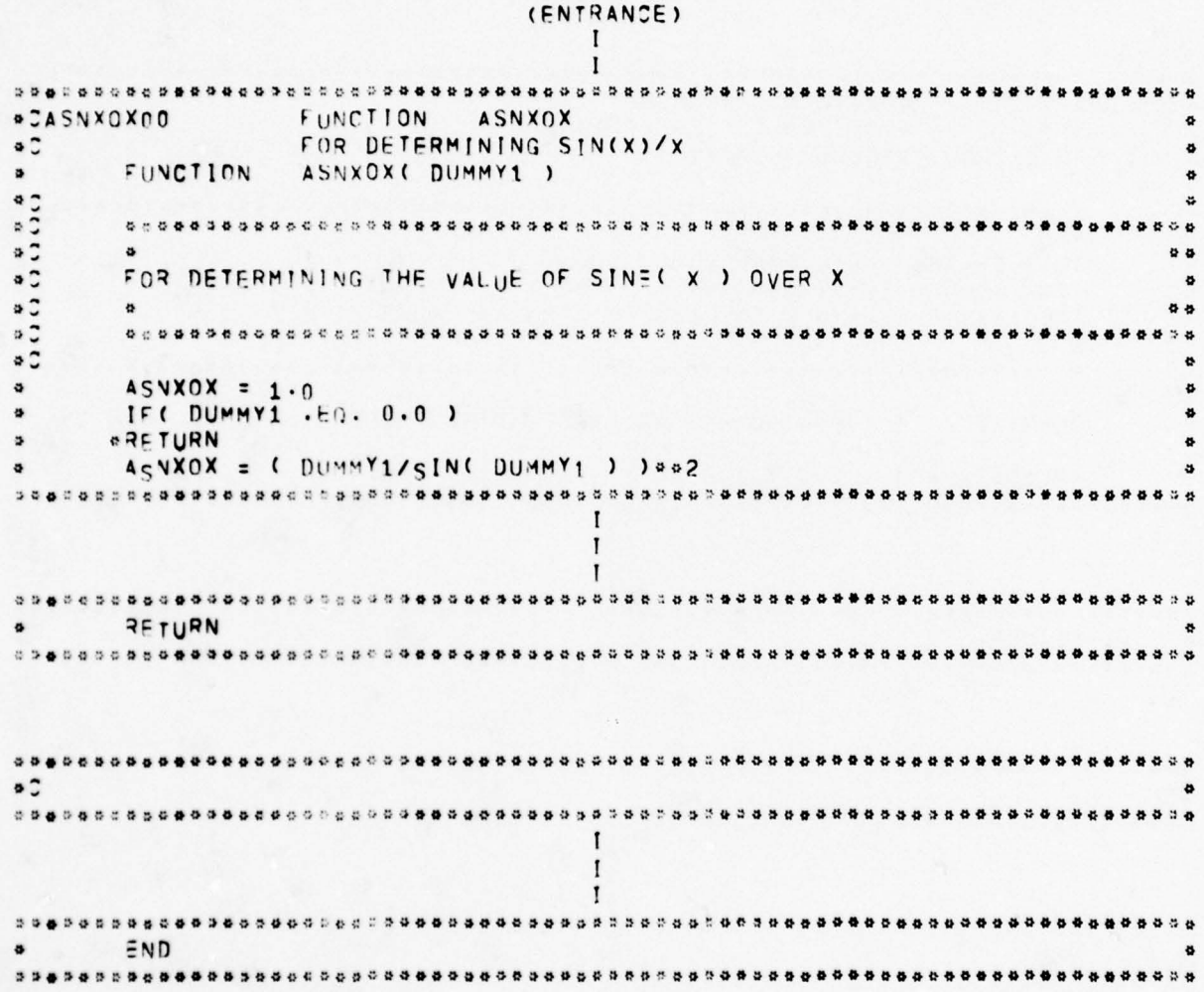

(ENTRANCE)
CANGV TO COMPUTE VECT. ANG.
COMPUTING ANG OF VECT.V=VXI+VYJ WITH RESPECT TO XY AXIS
SUBROUTINE ANGVE (VX,VY,ANGLE)
IF(VX.LT.0.) GO TO 300
IF(VY.LT.0.)GO TO 310
Q=1.
GO TO 320
300 IF(VY.LT.0.) GO TO 330
Q=2.
GO TO 320
330 Q=3.
GO TO 320
310 Q=4.


GO TO 320
320 ANGLE = ATAN2(ABS(VY),ABS(VX))
IF (Q.EQ.1.) GO TO 340
IF (Q.EQ.2.) GO TO 350
IF (Q.EQ.3.) GO TO 360
RETURN
340 ANGLE=-ANGLE
RETURN
350 ANGLE=-3.1415927+ANGLE
RETURN
360 ANGLE =3.1415927-ANGLE
RETURN
END

```
                                    (ENTRANCE)
                                        I
                                        I
*****************************************************************************
*CARCCOS00          FUNCTION   ARCCOS                                         *
*CFARCC000          FUNCTION FOR ARC-COSINE                                   *
*       FUNCTION    ARCCOS( DUMMY1 )                                          *
*C                                                                            *
*C      ***************************************************************     **
*C      *                                                                     **
*C      THIS IS THE ARC COSINE FUNCTION                                       *
*C       THE ANSWRR IS IN RADIANS                                             *
*C      THE ANSWER ALWAYS LIES BETWEEN ZERO AND PLUS PI/2                     *
*C      *                                                                     **
*C      ***************************************************************     **
*C                                                                            *
*C      DUMMY1      = THE ARGUMENT OF THE FUNCTION HOPEFULLY LESS THAN ONE    *
*C                                                                            *
*       ARCCOS = ATAN( SQRT( 1.0/DUMMY1/DUMMY1 - 1.0 ) )                     *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*       RETURN                                                               *
*****************************************************************************



*****************************************************************************
*C                                                                           *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*       END                                                                  *
*****************************************************************************
```


(ENTRANCE)
C ASNXOX00 FUNCTION ASNXOX
C FOR DETERMINING SIN(X)/X
FUNCTION ASNXOX(DUMMY1)
FOR DETERMINING THE VALUE OF SINE(X) OVER X
ASNXOX = 1.0
IF(DUMMY1 .EQ. 0.0)
RETURN
ASNXOX = (DUMMY1/SIN(DUMMY1))**2
RETURN
END

```
                                    (ENTRANCE)
                                        I
                                        I
*****************************************************************************
*CBAFFLE00        FUNCTION   BAFFLE                                          *
*C                INTERPOLATES THE BAFFLE AND RADIATED CURVES                *
*       FUNCTION   BAFFLE( N, DUMMY1, X1 )                                   *
*       COMMON                                                               *
*      C / SURDUC /                                                          *
*      C           BLA1   ,                                                  *
*      C           BLA2   ,                                                  *
*      C           BLA3   ,                                                  *
*      C           DTRAD  ,                                                  *
*      C           J      ,                                                  *
*      C           M      ,                                                  *
*      D           BAFFUN(128,2)     ,                                       *
*      D           DELBAF(128,2),                                            *
*      D           FLN1(128)         ,                                       *
*      D           FLN2(128)         ,                                       *
*      D           RADSPC(40,50,2)   ,                                       *
*      C           NTIMEN                                                    *
*C                                                                           *
*       X1 = AMOD( DUMMY1, DTRAD )/DTRAD                                     *
*       N = DUMMY1/DTRAD + 1.0                                               *
*       ENTRY BAFCON( N, X1 )                                                *
*       BAFFLE = BAFFUN(N,M) + X1*DELBAF(N,M)                                *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*       RETURN                                                               *
*****************************************************************************


*****************************************************************************
*C                                                                           *
*       ENTRY SPETRM( N, DUMMY1, X1 )                                        *
*       X1 = AMOD( DUMMY1, DTRAD )/DTRAD                                     *
*       N = DUMMY1/DTRAD + 1.0                                               *
*       ENTRY OSPECT( N, X1 )                                                *
*       BAFFLE = RADSPC(N,J,M) + X1*( RADSPC(N+1,J,M) - RADSPC(N,J,M) )      *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*       RETURN                                                               *
*****************************************************************************


*****************************************************************************
*C                                                                           *
*****************************************************************************
                                        I
                                        I
                                        I
                                        I
                                        I
*****************************************************************************
*       END                                                                  *
*****************************************************************************
```

```
                                    (ENTRANCE)
                                        I
                                        I
*****************************************************************************
*CBEAR       RELATIVE BEARING ANGLES AND RANGE                                *
*C   *************************************************************************
*C   *                                                                       *
*C   *  COMPUTING REL. BEARING ANGLES   AND   RANGE                          *
*C   *                                                                       *
*C   *************************************************************************
*       SUBROUTINE RELBR                                                     *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NFI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,F0S,FBWS,F1S,PTS(128),PXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*      7,F2S,PTE(128),PXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                    *
*       PX=PXE(N)-PXS(N)                                                    *
*       PY=PYE(N)-PYS(N)                                                    *
*       PROD2=PX**2+PY**2                                                   *
*       RANGE(N) = SQRT(PROD2)                                              *
*       CBPE = (-VXE(N)*PX-VYE(N)*PY)/(SE1*RANGE(N))                        *
*       CBPS = (VXS(N)*PX+VYS(N)*PY)/(SS1*RANGE(N))                         *
*       BSP=ACOS(CBPS)                                                      *
*       BEP=ACOS(CBPE)                                                      *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*       RETURN                                                              *
*****************************************************************************


*****************************************************************************
*       END                                                                 *
*****************************************************************************
```

```
                                    (ENTRANCE)
                                        I
                                        I
*****************************************************************************
*CBRID       BEARING RIDER                                                    *
*C  ***********************************************************************  *
*C  *                                                                         *
*C    *BEARING RIDER                                                         *
*C  *                                                                         *
*C  ***********************************************************************  *
*      SUBROUTINE BRIDER                                                      *
*      COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*     1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NS1,*
*     2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*     3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*     4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*     5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*     6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*     7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PREP*
*     8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                      *
*      VXS(N-1)=SS1*ALPXN                                                     *
*      VYS(N-1)=SS1*ALPYN                                                     *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*      RETURN                                                                 *
*****************************************************************************

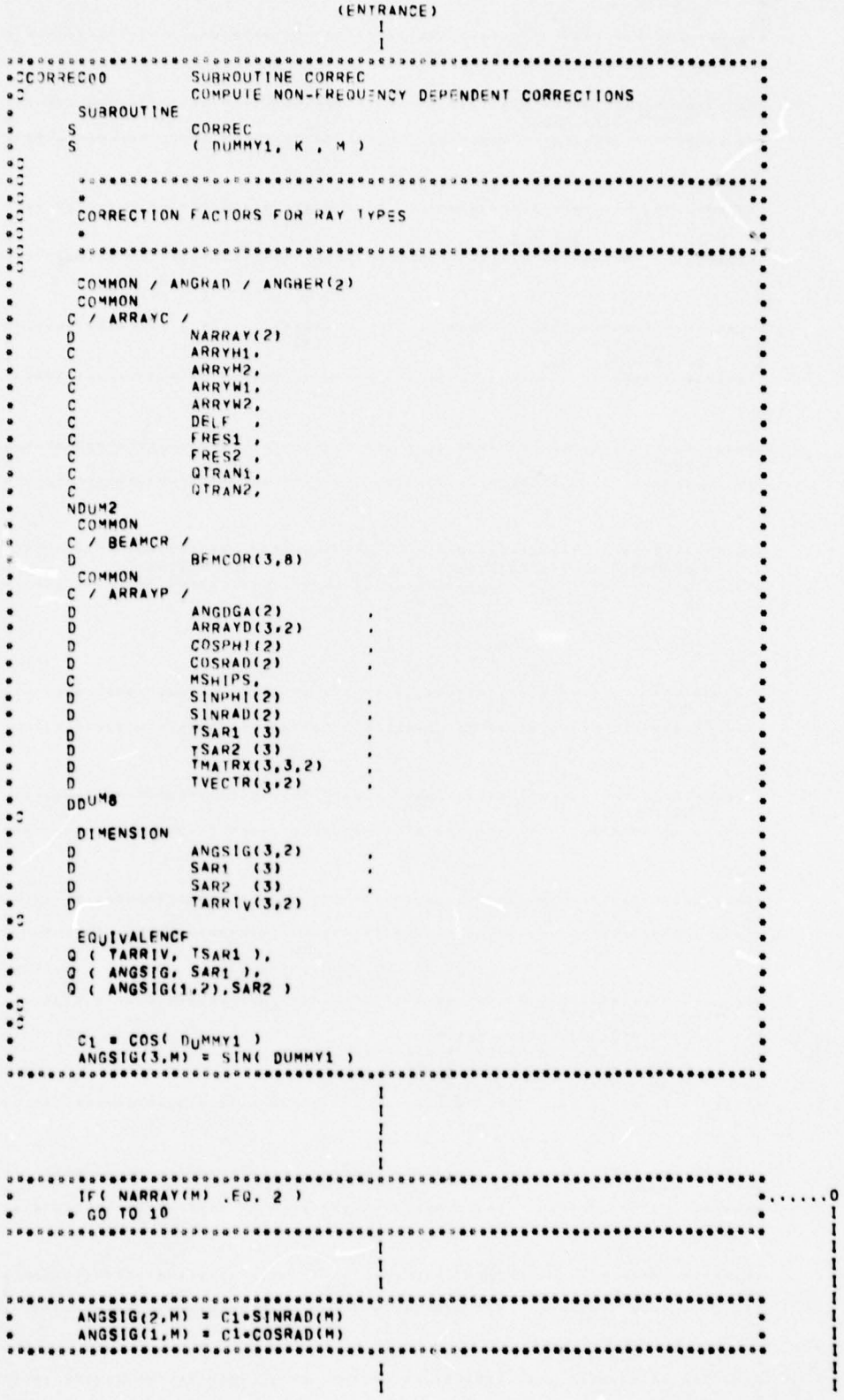


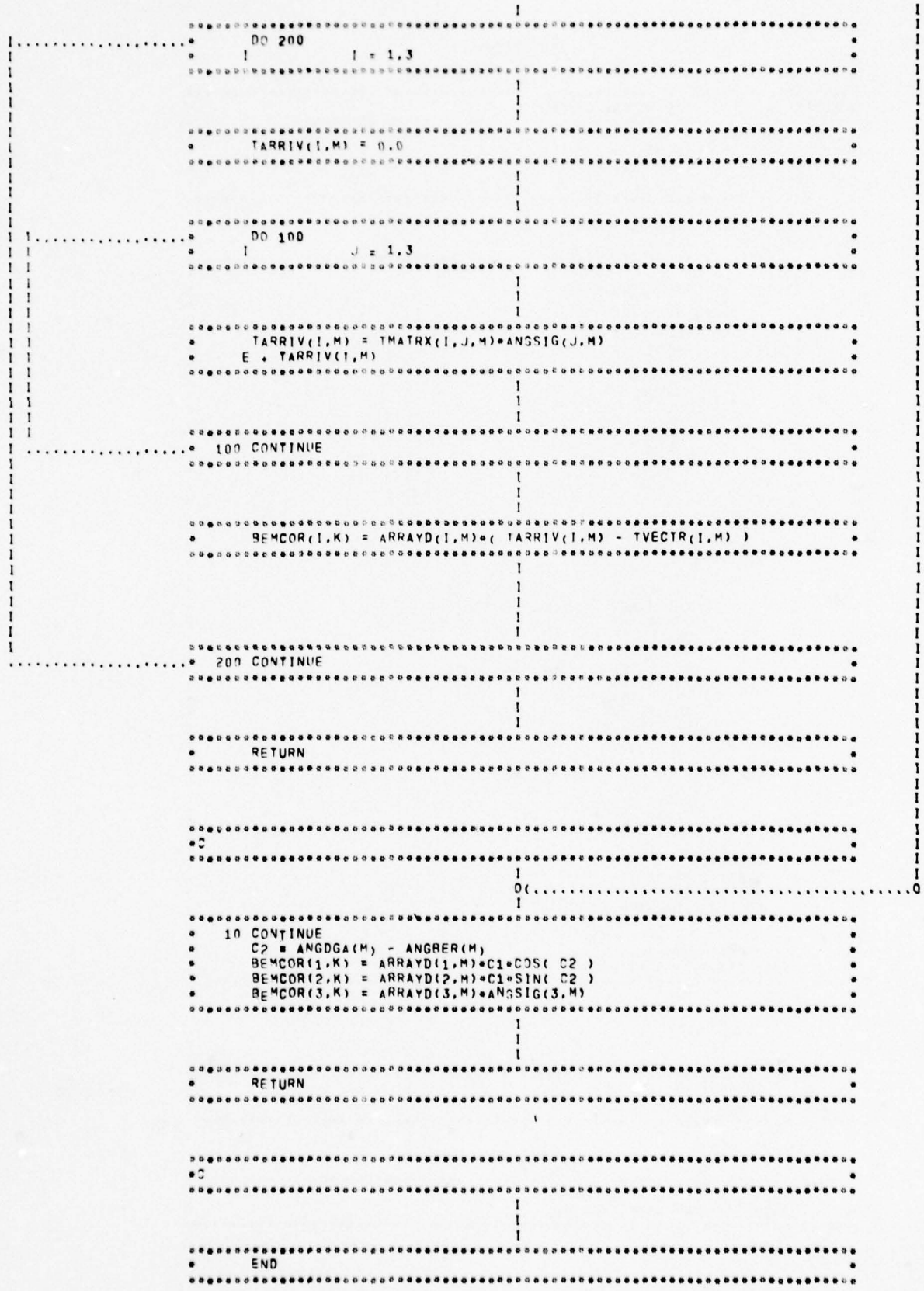
*****************************************************************************
*      END                                                                    *
*****************************************************************************
```

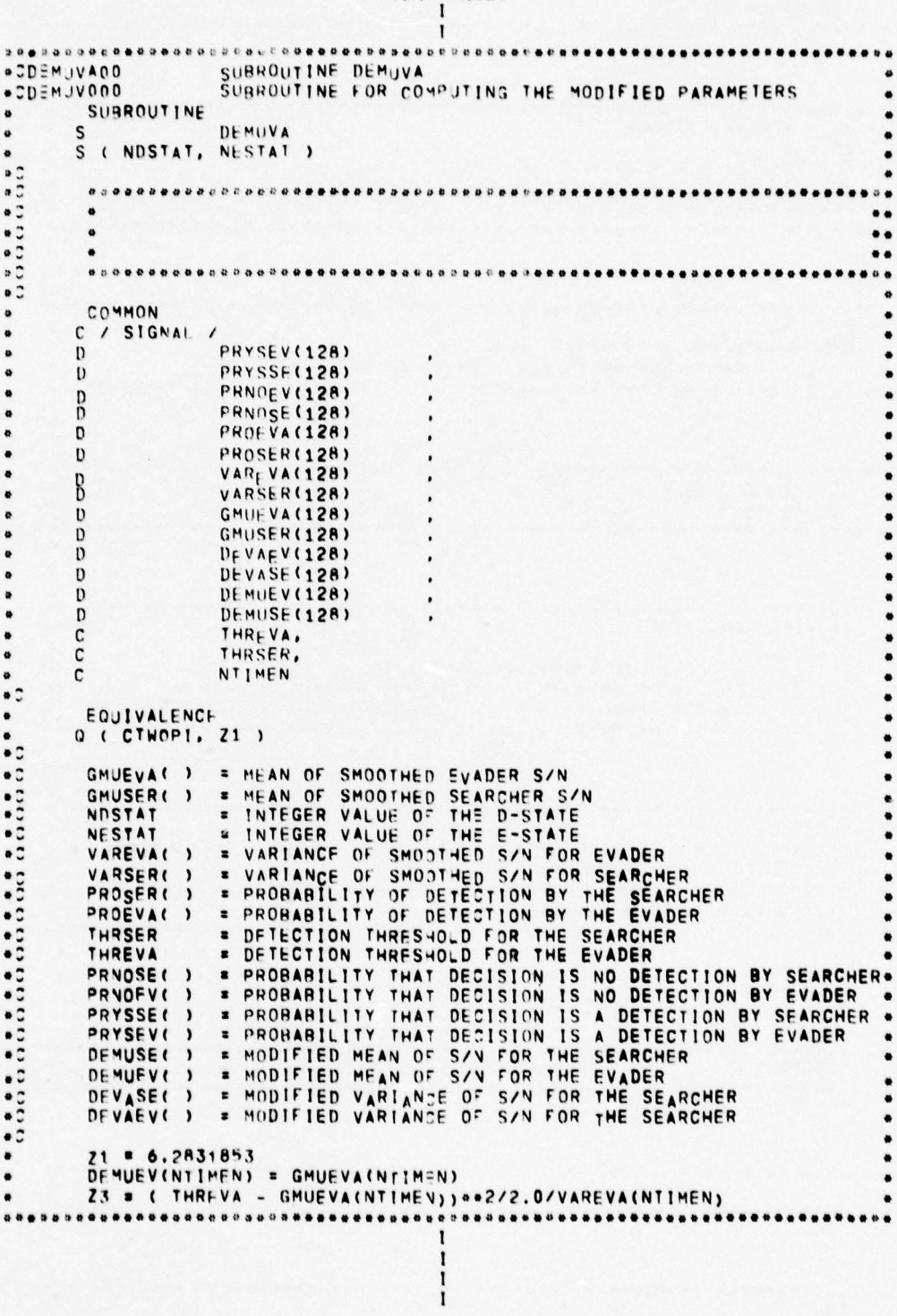
(ENTRANCE)

```
*CCOL_I      COLLISION
*C   ************************************************************
*C   *                                                          *
*C     *COLLISION                                              *
*C   *                                                          *
*C   ************************************************************
*       SUBROUTINE COLLIS                                      *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NFI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                       *
*       NSUB1=N-1                                              *
*       AL1= ALPXN*VXE(NSUB1)+ALPYN*VYE(NSUB1)                 *
*       AL=SQRT(AL1**2+SS1**2-SE1**2)-AL1                      *
*       VXS(NSUB1)=VXE(NSUB1)+AL*ALPXN                         *
*       VYS(NSUB1)=VYE(NSUB1)+AL*ALPYN                         *
************************************************************
                                    I
                                    I
                                    I
************************************************************
*       RETURN                                                 *
************************************************************


************************************************************
*       END                                                    *
************************************************************
```


(ENTRANCE)
CCORREC00 SUBROUTINE CORREC
C COMPUTE NON-FREQUENCY DEPENDENT CORRECTIONS
SUBROUTINE
S CORREC
S (DUMMY1, K , M)
CORRECTION FACTORS FOR RAY TYPES
COMMON / ANGRAD / ANGRER(2)
COMMON
C / ARRAYC /
D NARRAY(2)
C ARRYH1,
C ARRYH2,
C ARRYW1,
C ARRYW2,
C DELF ,
C FRES1 ,
C FRES2 ,
C QTRAN1,
C QTRAN2,
NDUM2
COMMON
C / BEAMCR /
D BEMCOR(3,8)
COMMON
C / ARRAYP /
D ANGDGA(2) ,
D ARRAYD(3,2) ,
D COSPHI(2) ,
D COSRAD(2) ,
C MSHIPS,
D SINPHI(2) ,
D SINRAD(2) ,
D TSAR1 (3) ,
D TSAR2 (3) ,
D TMATRX(3,3,2) ,
D TVECTR(3,2)
DDUM8
C
DIMENSION
D ANGSIG(3,2) ,
D SAR1 (3) ,
D SAR2 (3) ,
D TARRIV(3,2)
C
EQUIVALENCE
Q (TARRIV, TSAR1),
Q (ANGSIG, SAR1),
Q (ANGSIG(1,2),SAR2)
C
C
C1 = COS(DUMMY1)
ANGSIG(3,M) = SIN(DUMMY1)
IF(NARRAY(M) .EQ. 2)
GO TO 10
ANGSIG(2,M) = C1*SINRAD(M)
ANGSIG(1,M) = C1*COSRAD(M)


DO 200
I = 1,3
TARRIV(I,M) = 0.0
DO 100
J = 1,3
TARRIV(I,M) = TMATRX(I,J,M)*ANGSIG(J,M)
100 CONTINUE
BEMCOR(I,K) = ARRAYD(I,M)*(TARRIV(I,M) - TVECTR(I,M))
200 CONTINUE
RETURN
C
C
10 CONTINUE
C2 = ANGDGA(M) - ANGRER(M)
BEMCOR(1,K) = ARRAYD(1,M)*C1*COS(C2)
BEMCOR(2,K) = ARRAYD(2,M)*C1*SIN(C2)
BEMCOR(3,K) = ARRAYD(3,M)*ANGSIG(3,M)
RETURN
C
C
END


(ENTRANCE)
*CDEMUVA00 SUBROUTINE DEMUVA
*CDEMUV000 SUBROUTINE FOR COMPUTING THE MODIFIED PARAMETERS
SUBROUTINE
S DEMUVA
S (NDSTAT, NESTAT)
COMMON
C / SIGNAL /
D PRYSEV(128) ,
D PRYSSE(128) ,
D PRNOEV(128) ,
D PRNOSE(128) ,
D PROEVA(128) ,
D PROSER(128) ,
D VAREVA(128) ,
D VARSER(128) ,
D GMUEVA(128) ,
D GMUSER(128) ,
D DEVAEV(128) ,
D DEVASE(128) ,
D DEMUEV(128) ,
D DEMUSE(128) ,
C THREVA,
C THRSER,
C NTIMEN
EQUIVALENCE
Q (CTWOPI, Z1)
GMUEVA( ) = MEAN OF SMOOTHED EVADER S/N
GMUSER( ) = MEAN OF SMOOTHED SEARCHER S/N
NDSTAT = INTEGER VALUE OF THE D-STATE
NESTAT = INTEGER VALUE OF THE E-STATE
VAREVA( ) = VARIANCE OF SMOOTHED S/N FOR EVADER
VARSER( ) = VARIANCE OF SMOOTHED S/N FOR SEARCHER
PROSER( ) = PROBABILITY OF DETECTION BY THE SEARCHER
PROEVA( ) = PROBABILITY OF DETECTION BY THE EVADER
THRSER = DETECTION THRESHOLD FOR THE SEARCHER
THREVA = DETECTION THRESHOLD FOR THE EVADER
PRNOSE( ) = PROBABILITY THAT DECISION IS NO DETECTION BY SEARCHER
PRNOEV( ) = PROBABILITY THAT DECISION IS NO DETECTION BY EVADER
PRYSSE( ) = PROBABILITY THAT DECISION IS A DETECTION BY SEARCHER
PRYSEV( ) = PROBABILITY THAT DECISION IS A DETECTION BY EVADER
DEMUSE( ) = MODIFIED MEAN OF S/N FOR THE SEARCHER
DEMUEV( ) = MODIFIED MEAN OF S/N FOR THE EVADER
DEVASE( ) = MODIFIED VARIANCE OF S/N FOR THE SEARCHER
DEVAEV( ) = MODIFIED VARIANCE OF S/N FOR THE SEARCHER
Z1 = 6.2831853
DEMUEV(NTIMEN) = GMUEVA(NTIMEN)
Z3 = (THREVA - GMUEVA(NTIMEN))**2/2.0/VAREVA(NTIMEN)

```
      IF( Z3 .GT. 88.0 )                                                    .....0
     * GO TO 10                                                                 I
                                                                               I
                                                                               I
                                                                               I
      Z4 = PROEVA(NTIMEN)                                                      I
      Z5 = 0.0                                                                 I
      IF( Z4 .NE. 0.0 ) Z5 = PRYSEV(NESTAT)/Z4                                 I
      IF( Z4 .NE. 1.0 )  Z5 = Z5 - PRNDEV(NESTAT)/( 1.0 - Z4 )                 I
      DEMUEV(NTIMEN)=DEMUEV(NTIMEN)+SQRT(VAREVA(NTIMEN)/Z1)*Z5/EXP(Z3)         I
                                                                               I
                                                                               I
                                                                               
```


(ENTRANCE)
*CEXCHNG00 SUBROUTINE EXCHNG
*CSEXCH000 SUBROUTINE FOR EXCHANGING TWO VALUE
SUBROUTINE
S EXCHNG
S (FIRSTO, SECOND)
THIS EXCHANGES THE VALUES OF THE INPUTS
DUMMY1 = FIRSTO
FIRSTO = SECOND
SECOND = DUMMY1
RETURN
END

```
                                    (ENTRANCE)
                                        I
                                        I
*****************************************************************************
*CFILT       PRE-DETECT. FILTER RESPONSE                                     *
*C   ***************************************************************       *
*C   *                                                                 *     *
*C   *  SUB. FOR COMPUTING PRE-DETECT. FILTER RESPONSE                 *     *
*C   *                                                                 *     *
*C   ***************************************************************       *
*       SUBROUTINE FILTER (FO,FBW,F,Y)                                      *
*       Y=1./(1.+((F-FO)/FBW)**2)                                           *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*       RETURN                                                              *
*****************************************************************************



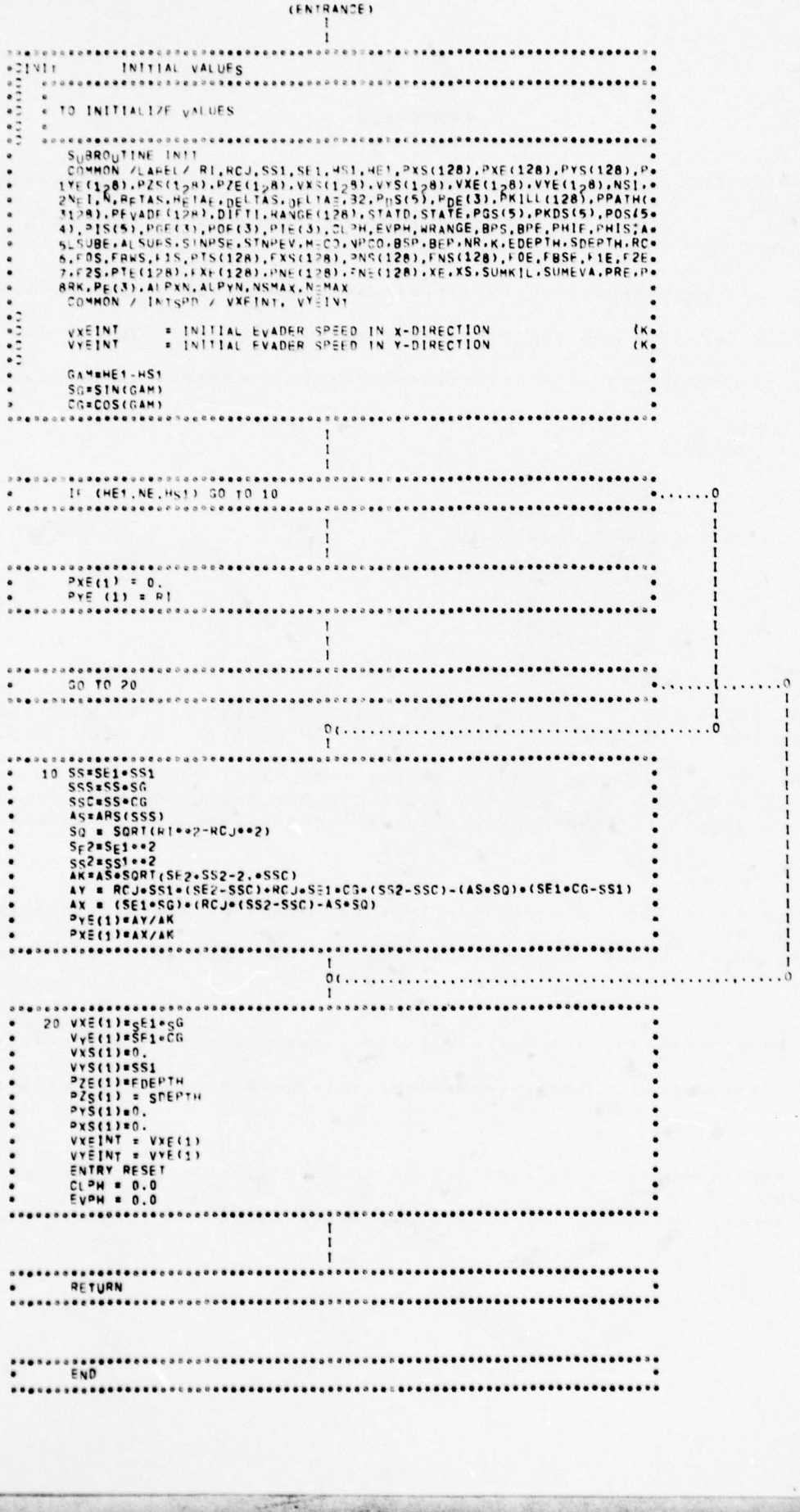
*****************************************************************************
*       END                                                                 *
*****************************************************************************
```

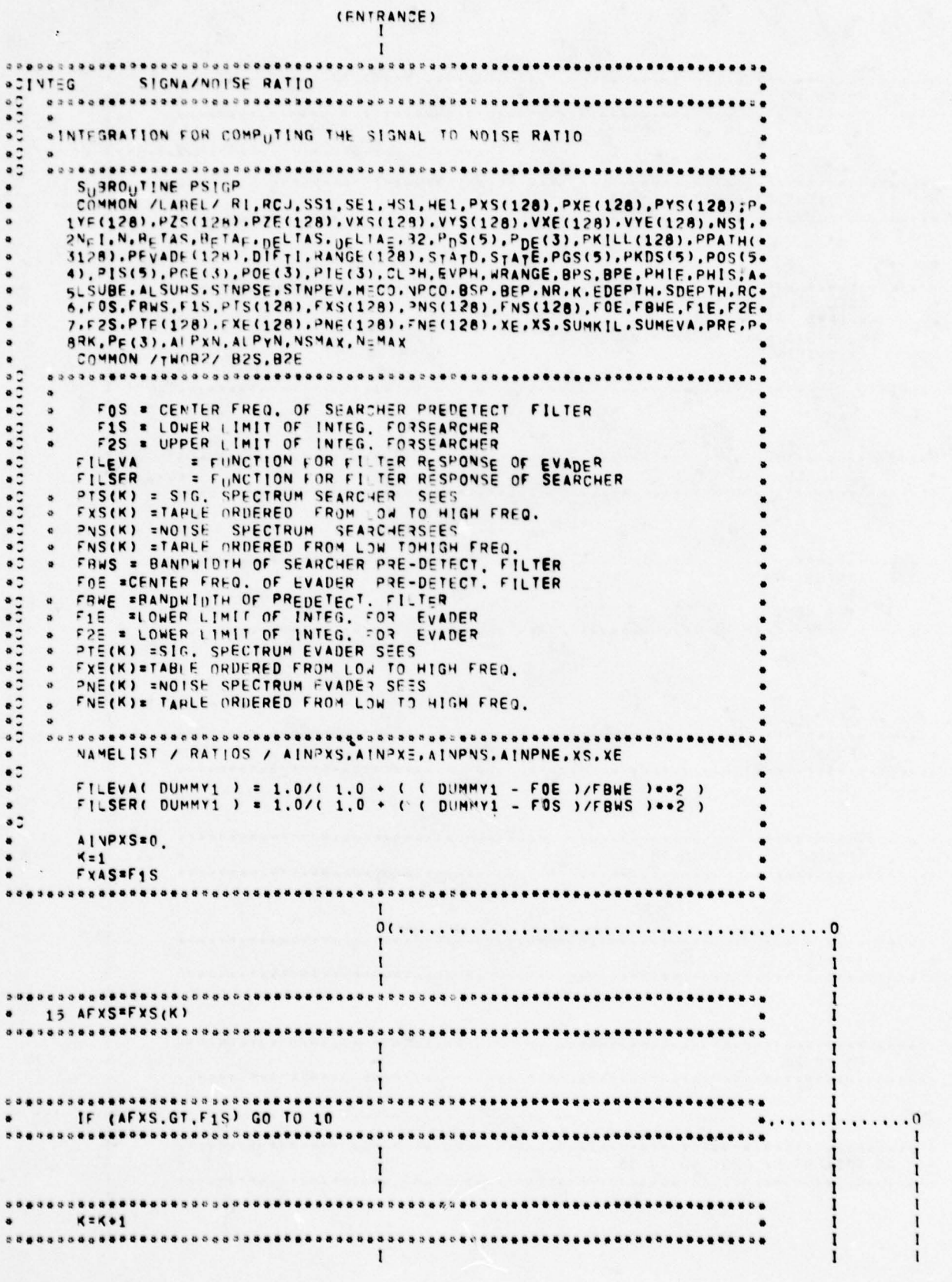
(ENTRANCE)

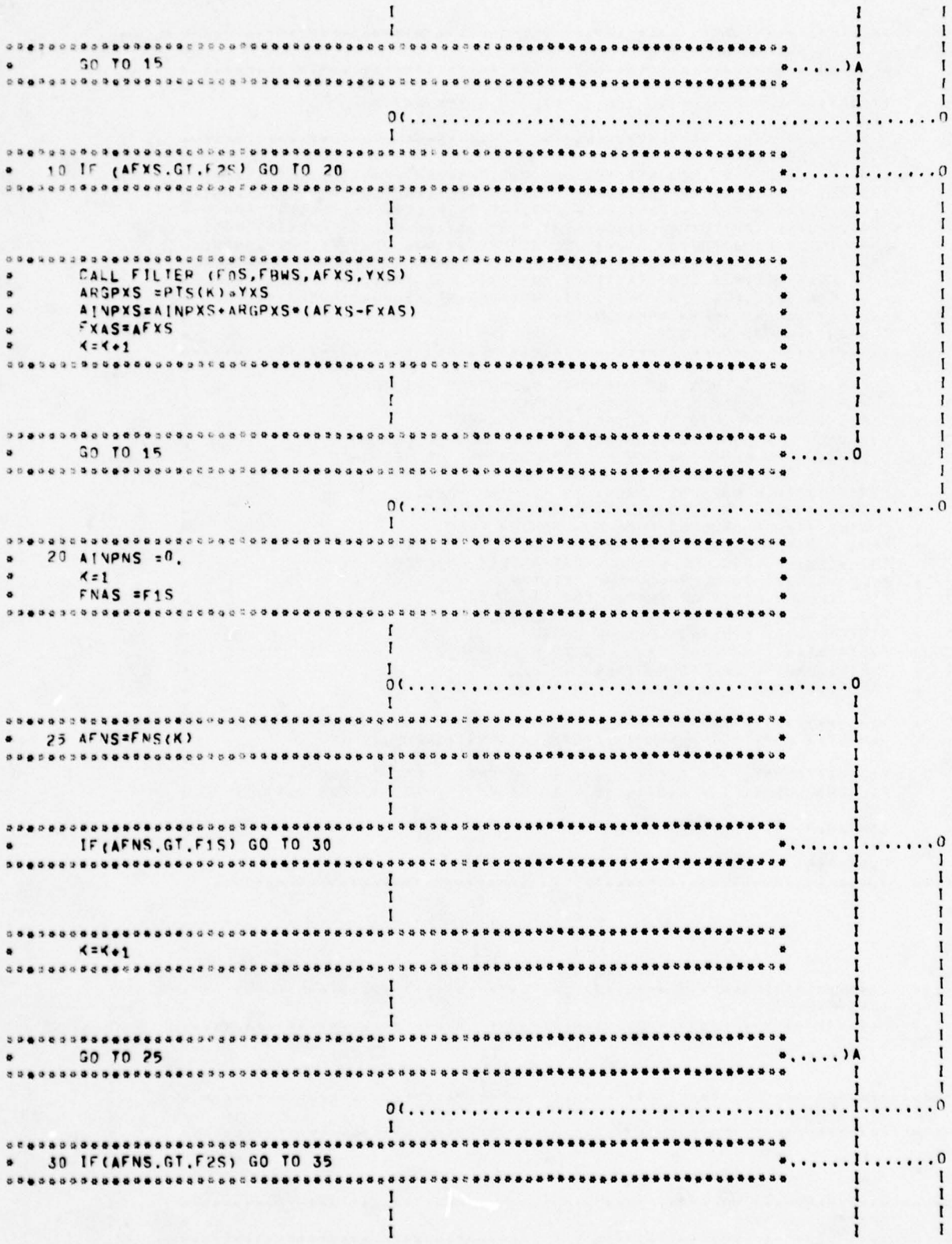
```
*CFVE_0000        FUNCTION   FVELOC                                          *
*CFVE_C000        FUNCTION FOR COMPUTING THE VELOCITY AT ANY DEPTH          *
*       FUNCTION   FVELOC( DELTAD, LAYERN )                                 *
*C                                                                           *
*C      ***************************************************************      *
*C      *                                                             *      *
*C      THIS COMPUTES THE VELOCITY (IN K_YD/SEC) AT ANY POINT IN THE LAYER*
*C      *                                                             *      *
*C      ***************************************************************      *
*C                                                                           *
*       COMMON                                                              *
*      C / LCONST /                                                         *
*      C            NULAPO,                                                *
*      C            DEPROT,                                                *
*      D            CONSG0(128)      ,                                     *
*      D            CONSG1(128)      ,                                     *
*      D            CONSG2(128)      ,                                     *
*      D            CONSV0(128)      ,                                     *
*      D            DELTAZ(128)      ,                                     *
*      D            DEPKYD(128)      ,                                     *
*      D            SLOPEJ(128)      ,                                     *
*      D            SPDKYD(128)                                            *
*C                                                                          *
*C     DELTAD      = DEPTH FROM TOP OF LAYER TO POINT OF INTEREST (IN K-YD*
*C     LAYERN      = NUMBER OF LAYER  IN WHICH VELOCITY IS SOUGHT (DIMENSI*
*C                                                                          *
*       FVELOC = 1.0/SQRT( CONSV0(LAYERN) + DELTAD*( CONSG0(LAYERN) +      *
*     E DELTAD*CONSG1(LAYERN) )/( 1.0 + CONSG2(LAYERN)*DELTAD )**2 )       *
*****************************************************************************
                                    I
                                    I
                                    I
*****************************************************************************
*       RETURN                                                              *
*****************************************************************************



*****************************************************************************
*C                                                                          *
*****************************************************************************
                                    I
                                    I
                                    I
*****************************************************************************
*       END                                                                 *
*****************************************************************************
```


(ENTRANCE)
INIT INITIAL VALUES
TO INITIALIZE VALUES
SUBROUTINE INIT
COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P
1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,
2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(
3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5
4),PIS(5),PGE(3),PDE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A
5LSUBE,ALSUBS,SINPSE,STNPEV,MECO,VPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC
6,FOS,FRWS,FIS,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBSE,FIE,F2E
7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P
8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX
COMMON / INTSPD / VXEINT, VYEINT
VXEINT = INITIAL EVADER SPEED IN X-DIRECTION (K
VYEINT = INITIAL EVADER SPEED IN Y-DIRECTION (K
GAM=HE1-HS1
SG=SIN(GAM)
CG=COS(GAM)
IF (HE1.NE.HS1) GO TO 10
PXE(1) = 0.
PYE (1) = RI
GO TO 20
10 SS=SE1*SS1
SSS=SS*SG
SSC=SS*CG
AS=ABS(SSS)
SQ = SQRT(RI**2-RCJ**2)
SE2=SE1**2
SS2=SS1**2
AK=AS*SQRT(SE2*SS2-2.*SSC)
AY = RCJ*SS1*(SE2-SSC)+RCJ*SE1*CG*(SS2-SSC)-(AS*SQ)*(SE1*CG-SS1)
AX = (SE1*SG)*(RCJ*(SS2-SSC)-AS*SQ)
PYE(1)=AY/AK
PXE(1)=AX/AK
20 VXE(1)=SE1*SG
VYE(1)=SE1*CG
VXS(1)=0.
VYS(1)=SS1
PZE(1)=EDEPTH
PZS(1) = SDEPTH
PYS(1)=0.
PXS(1)=0.
VXEINT = VXE(1)
VYEINT = VYE(1)
ENTRY RESET
CLPH = 0.0
EVPH = 0.0
RETURN
END


(ENTRANCE)
C INTEG SIGNA/NOISE RATIO
C
C INTEGRATION FOR COMPUTING THE SIGNAL TO NOISE RATIO
C
SUBROUTINE PSIGP
COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),PYE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,ALSUBE,ALSUBS,STNPSE,STNPEV,MECO,VPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC,FOS,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBWE,F1E,F2E,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,PBRK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX
COMMON /TWOB2/ B2S,B2E
F0S = CENTER FREQ. OF SEARCHER PREDETECT FILTER
F1S = LOWER LIMIT OF INTEG. FORSEARCHER
F2S = UPPER LIMIT OF INTEG. FORSEARCHER
FILEVA = FUNCTION FOR FILTER RESPONSE OF EVADER
FILSER = FUNCTION FOR FILTER RESPONSE OF SEARCHER
PTS(K) = SIG. SPECTRUM SEARCHER SEES
FXS(K) =TABLE ORDERED FROM LOW TO HIGH FREQ.
PNS(K) =NOISE SPECTRUM SEARCHERSEES
FNS(K) =TABLE ORDERED FROM LOW TOHIGH FREQ.
FBWS = BANDWIDTH OF SEARCHER PRE-DETECT. FILTER
F0E =CENTER FREQ. OF EVADER PRE-DETECT. FILTER
FBWE =BANDWIDTH OF PREDETECT. FILTER
F1E =LOWER LIMIT OF INTEG. FOR EVADER
F2E = LOWER LIMIT OF INTEG. FOR EVADER
PTE(K) =SIG. SPECTRUM EVADER SEES
FXE(K)=TABLE ORDERED FROM LOW TO HIGH FREQ.
PNE(K) =NOISE SPECTRUM EVADER SEES
FNE(K)= TABLE ORDERED FROM LOW TO HIGH FREQ.
NAMELIST / RATIOS / AINPXS,AINPXE,AINPNS,AINPNE,XS,XE
FILEVA(DUMMY1) = 1.0/(1.0 + ((DUMMY1 - F0E)/FBWE)**2)
FILSER(DUMMY1) = 1.0/(1.0 + ((DUMMY1 - F0S)/FBWS)**2)
AINPXS=0.
K=1
FXAS=F1S
15 AFXS=FXS(K)
IF (AFXS.GT.F1S) GO TO 10
K=K+1


GO TO 15
10 IF (AFXS.GT.F2S) GO TO 20
CALL FILTER (F0S,FBWS,AFXS,YXS)
ARGPXS =PTS(K)*YXS
AINPXS=AINPXS+ARGPXS*(AFXS-FXAS)
FXAS=AFXS
K=K+1
GO TO 15
20 AINPNS =0.
K=1
FNAS =F1S
25 AFNS=FNS(K)
IF(AFNS.GT.F1S) GO TO 30
K=K+1
GO TO 25
30 IF(AFNS.GT.F2S) GO TO 35

```
      CALL FILTER(FOS,FBWS,AFNS,YNS)
      ARGPNS =(PNS(K)*YNS)**2
      AINPNS = AINPNS+ARGPNS*(AFNS-FNAS)
      FNAS=AFNS
      K=K+1

      GO TO 25

   35 CONTINUE
      XS = 1.0 + AINPXS/SQRT( 0.002*AINPNS*B2S )
   37 AINPXE=0.
      K=1
      FXAE=F1E

   40 AFXE=FXE(K)

      IF(AFXE.GT.F1E) GO TO 45

      K=K+1

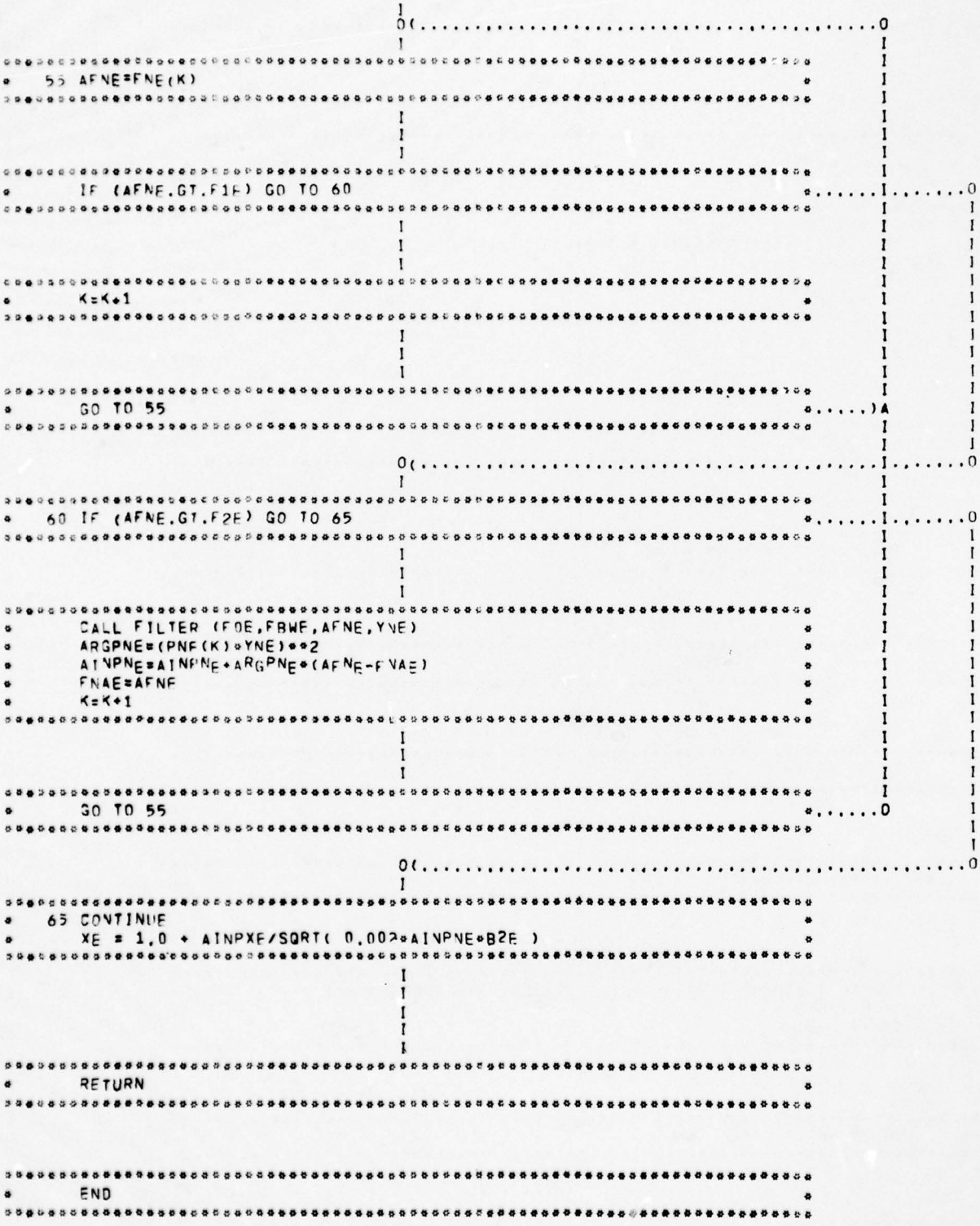
      GO TO 40

   45 IF(AFXE.GT.F2E) GO TO 50

      AINPXE = AINPXE + PTE(K)*FILEVA( AFXE )*( AFXE - FXAE )
      FXAE =AFXE
      K=K+1

      GO TO 40

   50 AINPNE = 0.
      K=1
      FNAE=F1E
```

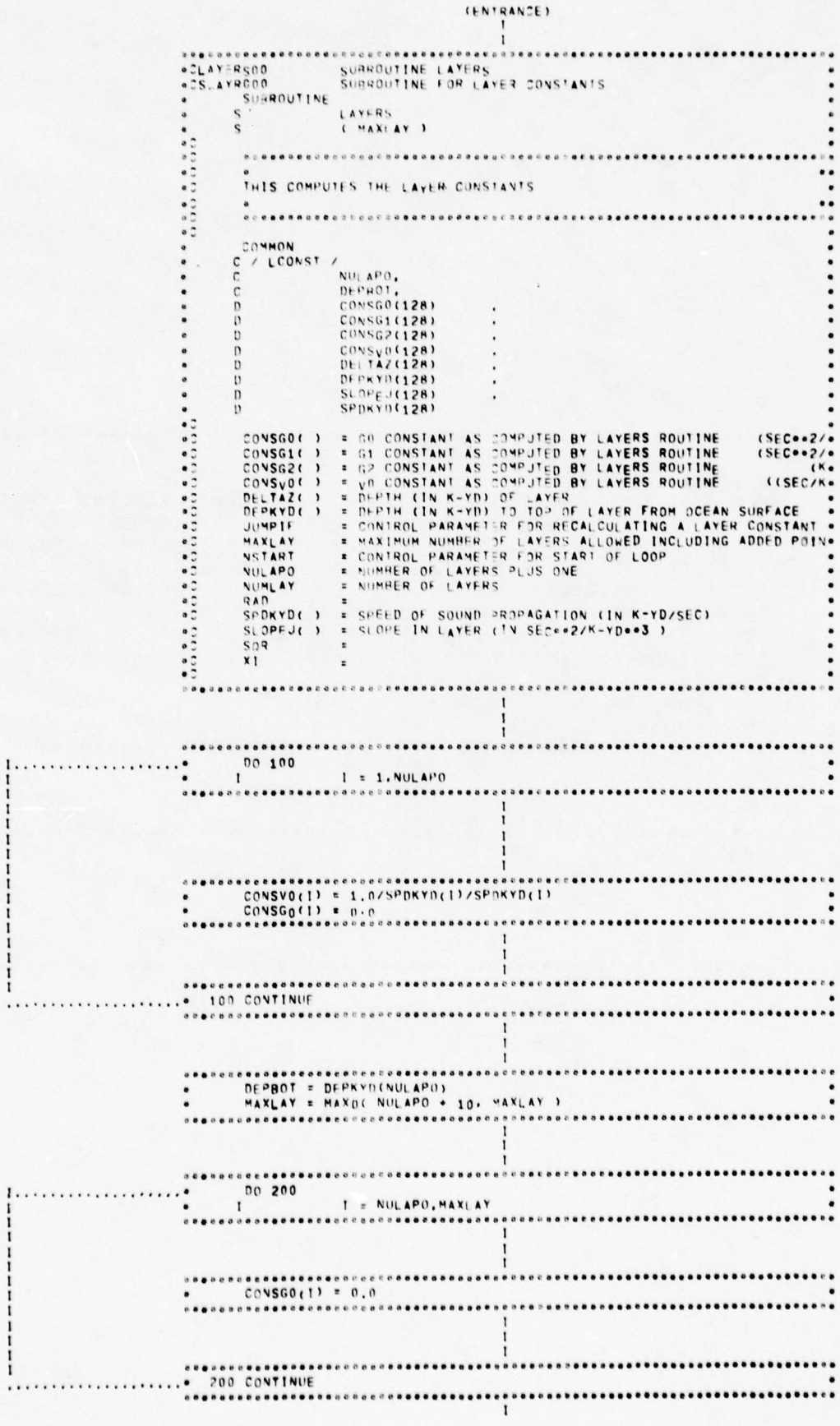

55 AFNE=FNE(K)
IF (AFNE.GT.F1E) GO TO 60
K=K+1
GO TO 55
60 IF (AFNE.GT.F2E) GO TO 65
CALL FILTER (F0E,FBWE,AFNE,YNE)
ARGPNE=(PNE(K)*YNE)**2
AINPNE=AINPNE+ARGPNE*(AFNE-FNAE)
FNAE=AFNE
K=K+1
GO TO 55
65 CONTINUE
XE = 1.0 + AINPXE/SQRT(0.002*AINPNE*B2E)
RETURN
END

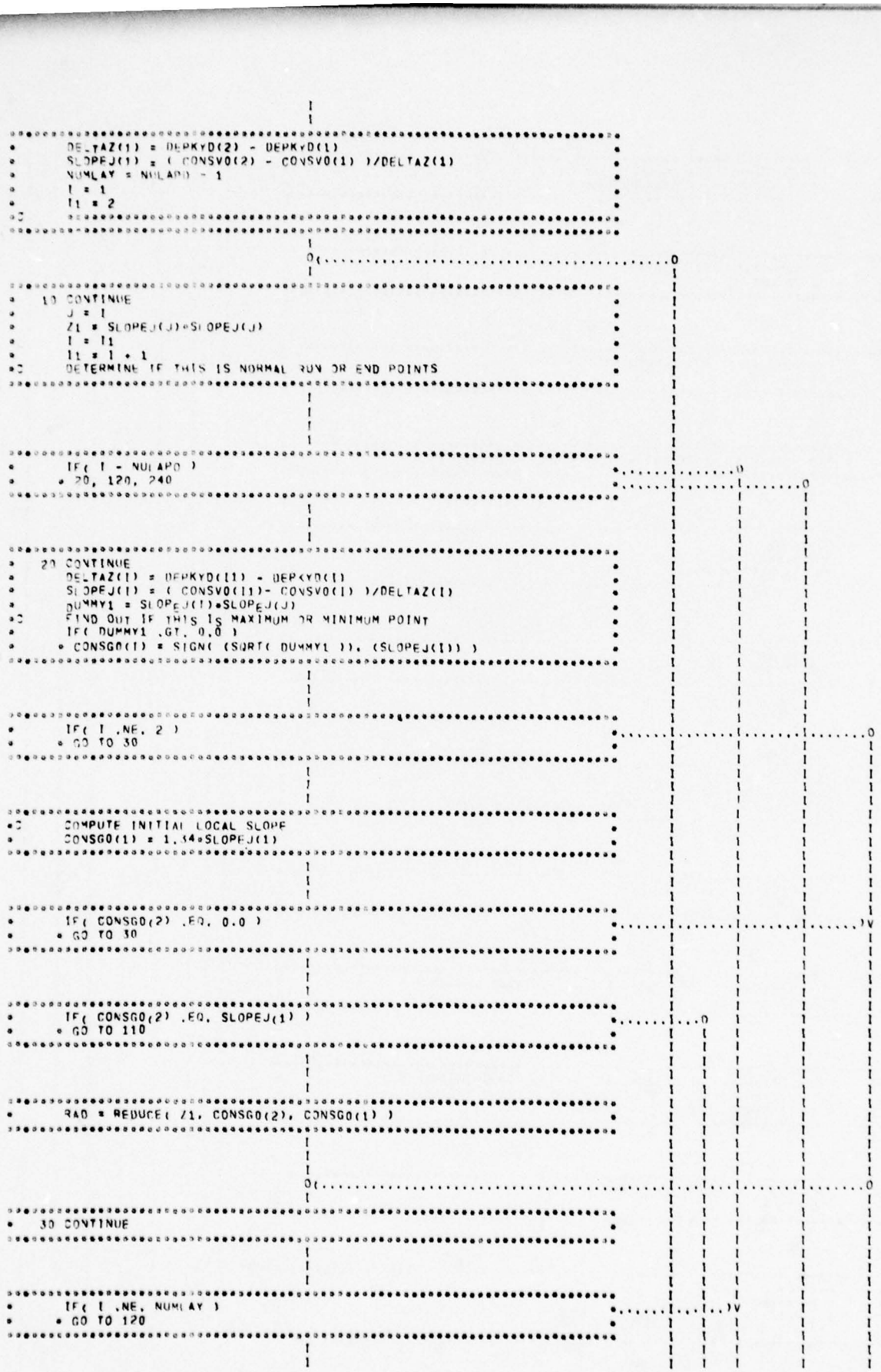
(ENTRANCE)

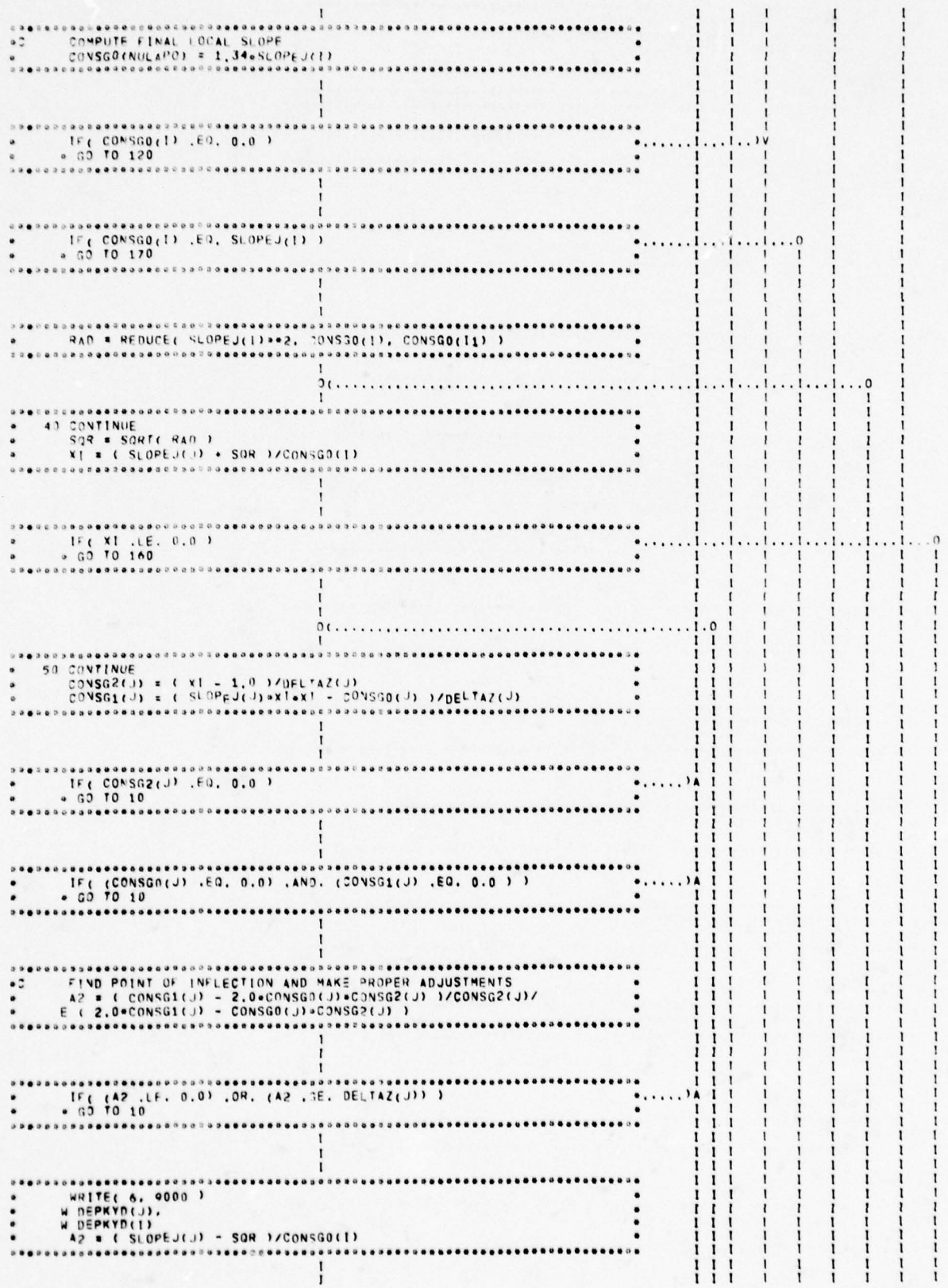
```
*CINV3        INVERSE BEARING RIDER                                          *
*       SUBROUTINE INVBRI                                                    *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSURS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PREP*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                    *
*       VXE(N-1)=SF1*ALPXN                                                  *
*       VYE(N-1)=SF1*ALPYN                                                  *
```

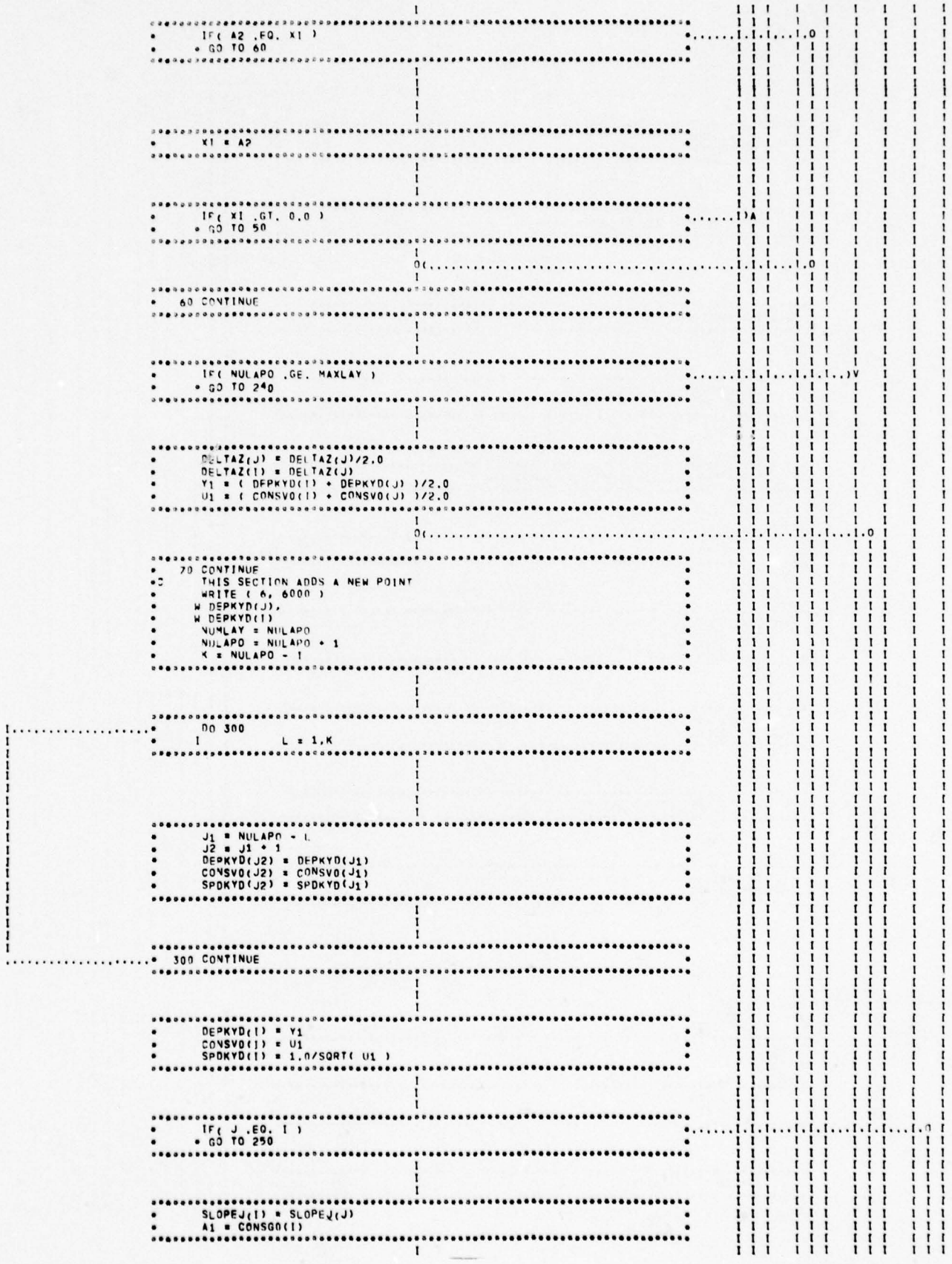
```
*       RETURN                                                               *
```

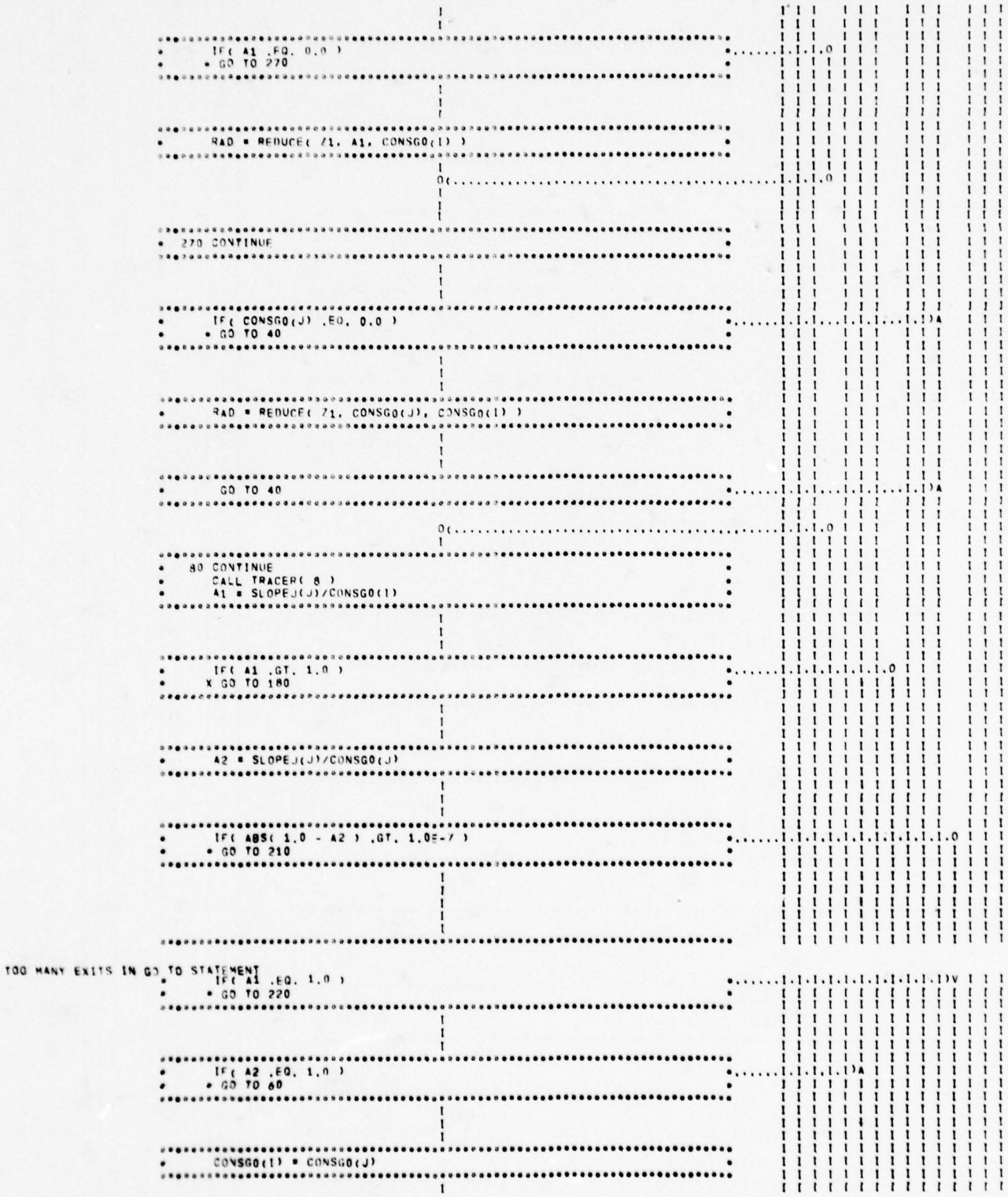
```
*       END                                                                  *
```


(ENTRANCE)
CLAYERS00 SUBROUTINE LAYERS
CSLAYR000 SUBROUTINE FOR LAYER CONSTANTS
SUBROUTINE
S LAYERS
S (MAXLAY)
THIS COMPUTES THE LAYER CONSTANTS
COMMON
C / LCONST /
C NULAPO,
C DEPBOT,
D CONSG0(128) ,
D CONSG1(128) ,
D CONSG2(128) ,
D CONSV0(128) ,
D DELTAZ(128) ,
D DEPKYD(128) ,
D SLOPEJ(128) ,
D SPDKYD(128)
CONSG0( ) = G0 CONSTANT AS COMPUTED BY LAYERS ROUTINE (SEC**2/*
CONSG1( ) = G1 CONSTANT AS COMPUTED BY LAYERS ROUTINE (SEC**2/*
CONSG2( ) = G2 CONSTANT AS COMPUTED BY LAYERS ROUTINE (K*
CONSV0( ) = V0 CONSTANT AS COMPUTED BY LAYERS ROUTINE ((SEC/K*
DELTAZ( ) = DEPTH (IN K-YD) OF LAYER
DEPKYD( ) = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE
JUMPIF = CONTROL PARAMETER FOR RECALCULATING A LAYER CONSTANT
MAXLAY = MAXIMUM NUMBER OF LAYERS ALLOWED INCLUDING ADDED POIN*
NSTART = CONTROL PARAMETER FOR START OF LOOP
NULAPO = NUMBER OF LAYERS PLUS ONE
NUMLAY = NUMBER OF LAYERS
RAD =
SPDKYD( ) = SPEED OF SOUND PROPAGATION (IN K-YD/SEC)
SLOPEJ( ) = SLOPE IN LAYER (IN SEC**2/K-YD**3)
SQR =
X1 =
DO 100
I I = 1,NULAPO
CONSV0(I) = 1.0/SPDKYD(I)/SPDKYD(I)
CONSG0(I) = 0.0
100 CONTINUE
DEPBOT = DEPKYD(NULAPO)
MAXLAY = MAX0(NULAPO + 10, MAXLAY)
DO 200
I I = NULAPO,MAXLAY
CONSG0(I) = 0.0
200 CONTINUE


DELTAZ(1) = DEPKYD(2) - DEPKYD(1)
SLOPEJ(1) = (CONSV0(2) - CONSV0(1))/DELTAZ(1)
NUMLAY = NULAPO - 1
I = 1
I1 = 2
C
10 CONTINUE
J = I
Z1 = SLOPEJ(J)*SLOPEJ(J)
I = I1
I1 = I + 1
C DETERMINE IF THIS IS NORMAL RUN OR END POINTS
IF(I - NULAPO)
* 20, 120, 240
20 CONTINUE
DELTAZ(I) = DEPKYD(I1) - DEPKYD(I)
SLOPEJ(I) = (CONSV0(I1)- CONSV0(I))/DELTAZ(I)
DUMMY1 = SLOPEJ(I)*SLOPEJ(J)
C FIND OUT IF THIS IS MAXIMUM OR MINIMUM POINT
IF(DUMMY1 .GT. 0.0)
* CONSGO(I) = SIGN((SQRT(DUMMY1)), (SLOPEJ(I)))
IF(I .NE. 2)
* GO TO 30
C COMPUTE INITIAL LOCAL SLOPE
CONSGO(1) = 1.34*SLOPEJ(1)
IF(CONSGO(2) .EQ. 0.0)
* GO TO 30
IF(CONSGO(2) .EQ. SLOPEJ(1))
* GO TO 110
RAD = REDUCE(Z1, CONSGO(2), CONSGO(1))
30 CONTINUE
IF(I .NE. NUMLAY)
* GO TO 120

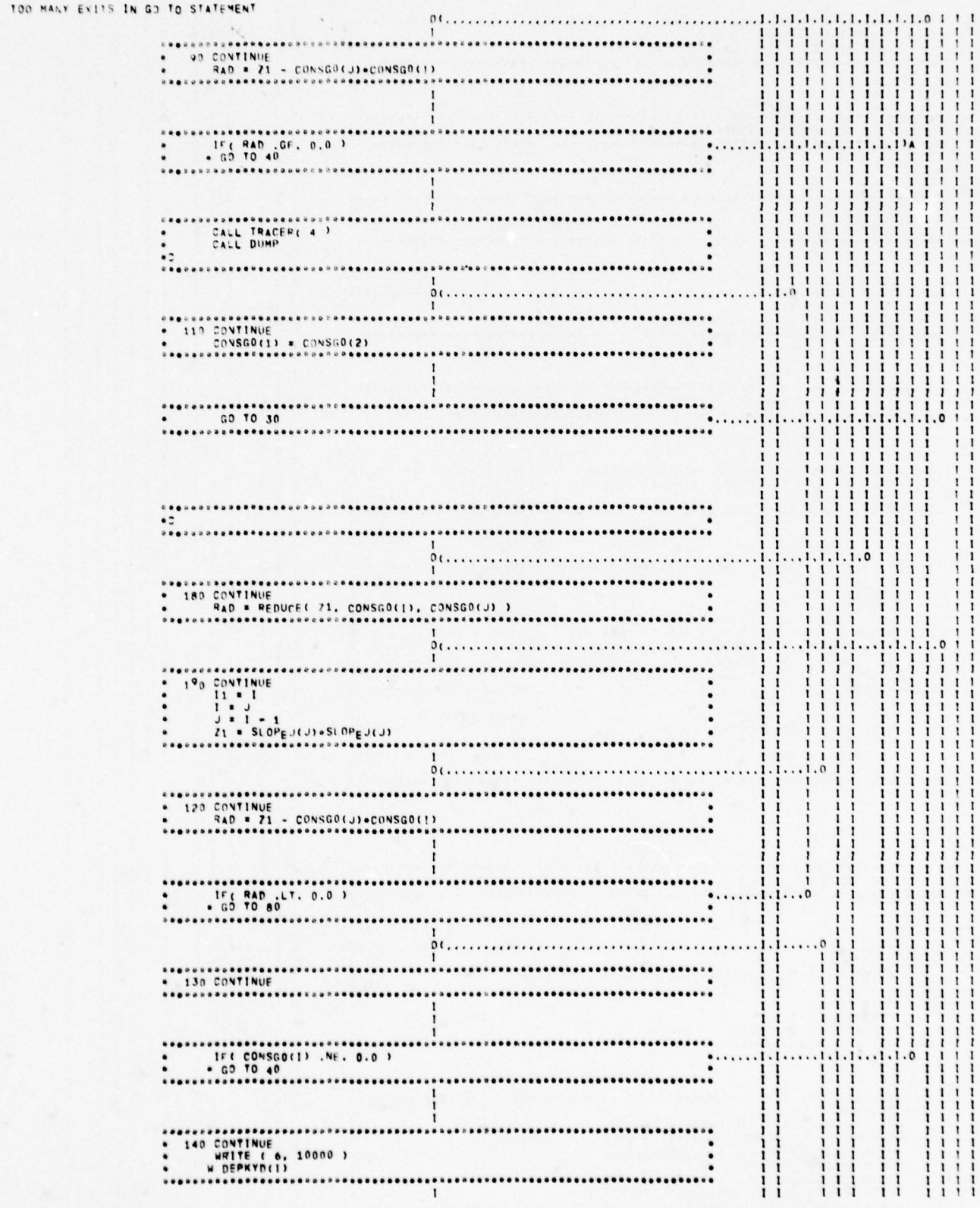

C COMPUTE FINAL LOCAL SLOPE
CONSG0(NULAPO) = 1.34*SLOPEJ(I)
IF(CONSG0(I) .EQ. 0.0)
GO TO 120
IF(CONSG0(I) .EQ. SLOPEJ(I))
GO TO 170
RAD = REDUCE(SLOPEJ(I)**2, CONSG0(I), CONSG0(I))
40 CONTINUE
SQR = SQRT(RAD)
XI = (SLOPEJ(J) + SQR)/CONSG0(I)
IF(XI .LE. 0.0)
GO TO 160
50 CONTINUE
CONSG2(J) = (XI - 1.0)/DELTAZ(J)
CONSG1(J) = (SLOPEJ(J)*XI*XI - CONSG0(J))/DELTAZ(J)
IF(CONSG2(J) .EQ. 0.0)
GO TO 10
IF((CONSG0(J) .EQ. 0.0) .AND. (CONSG1(J) .EQ. 0.0))
GO TO 10
C FIND POINT OF INFLECTION AND MAKE PROPER ADJUSTMENTS
A2 = (CONSG1(J) - 2.0*CONSG0(J)*CONSG2(J))/CONSG2(J)/
E (2.0*CONSG1(J) - CONSG0(J)*CONSG2(J))
IF((A2 .LE. 0.0) .OR. (A2 .GE. DELTAZ(J)))
GO TO 10
WRITE(6, 9000)
W DEPKYD(J),
W DEPKYD(I)
A2 = (SLOPEJ(J) - SQR)/CONSG0(I)

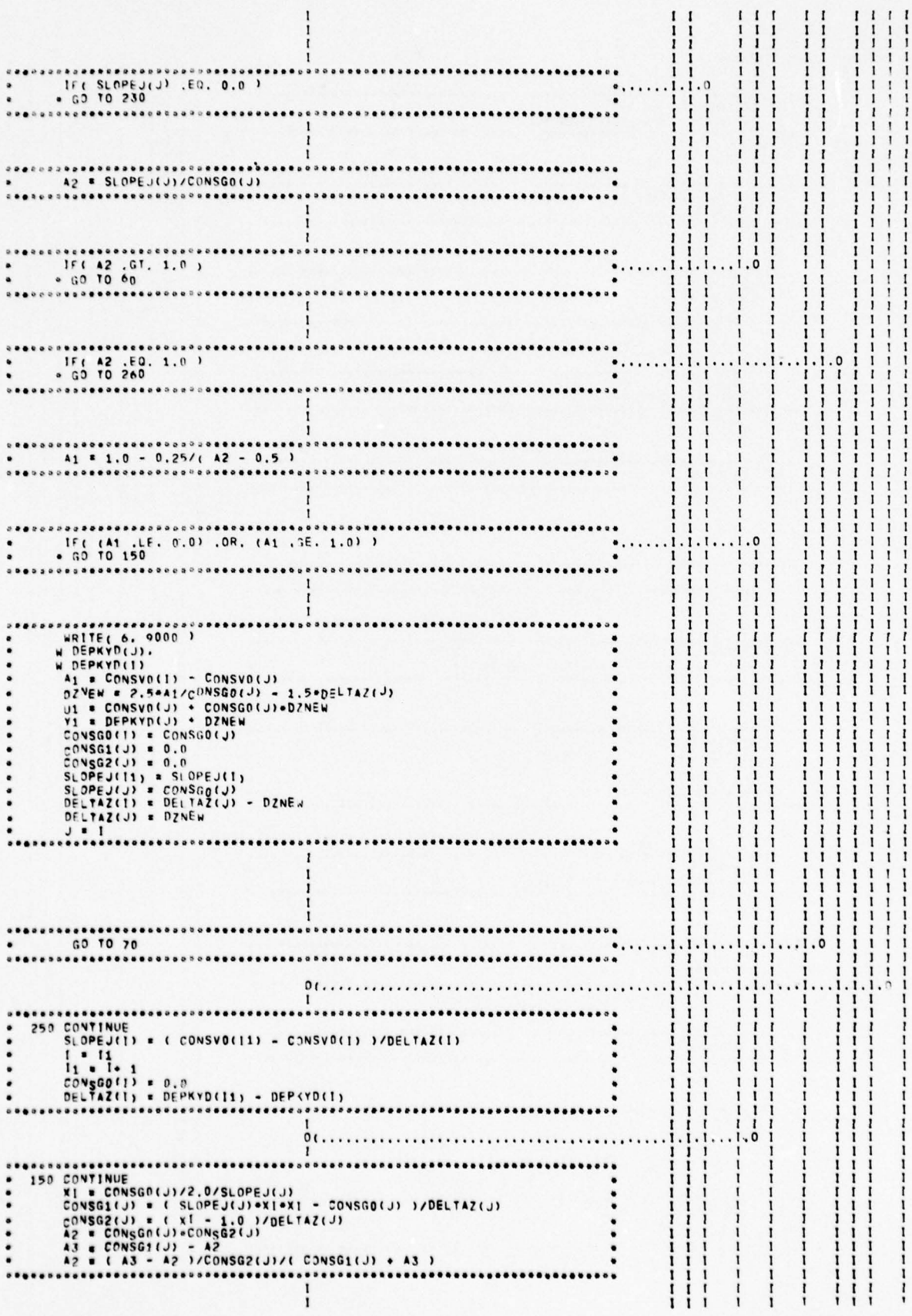

IF(A2 .EQ. X1)
GO TO 60
X1 = A2
IF(X1 .GT. 0.0)
GO TO 50
60 CONTINUE
IF(NULAPO .GE. MAXLAY)
GO TO 240
DELTAZ(J) = DELTAZ(J)/2.0
DELTAZ(I) = DELTAZ(J)
Y1 = (DEPKYD(I) + DEPKYD(J))/2.0
U1 = (CONSVO(I) + CONSVO(J))/2.0
70 CONTINUE
C THIS SECTION ADDS A NEW POINT
WRITE (6, 6000)
W DEPKYD(J),
W DEPKYD(I)
NUMLAY = NULAPO
NULAPO = NULAPO + 1
K = NULAPO - 1
DO 300
I L = 1,K
J1 = NULAPO - L
J2 = J1 + 1
DEPKYD(J2) = DEPKYD(J1)
CONSVO(J2) = CONSVO(J1)
SPDKYD(J2) = SPDKYD(J1)
300 CONTINUE
DEPKYD(I) = Y1
CONSVO(I) = U1
SPDKYD(I) = 1.0/SQRT(U1)
IF(J .EQ. I)
GO TO 250
SLOPEJ(I) = SLOPEJ(J)
A1 = CONSGO(I)

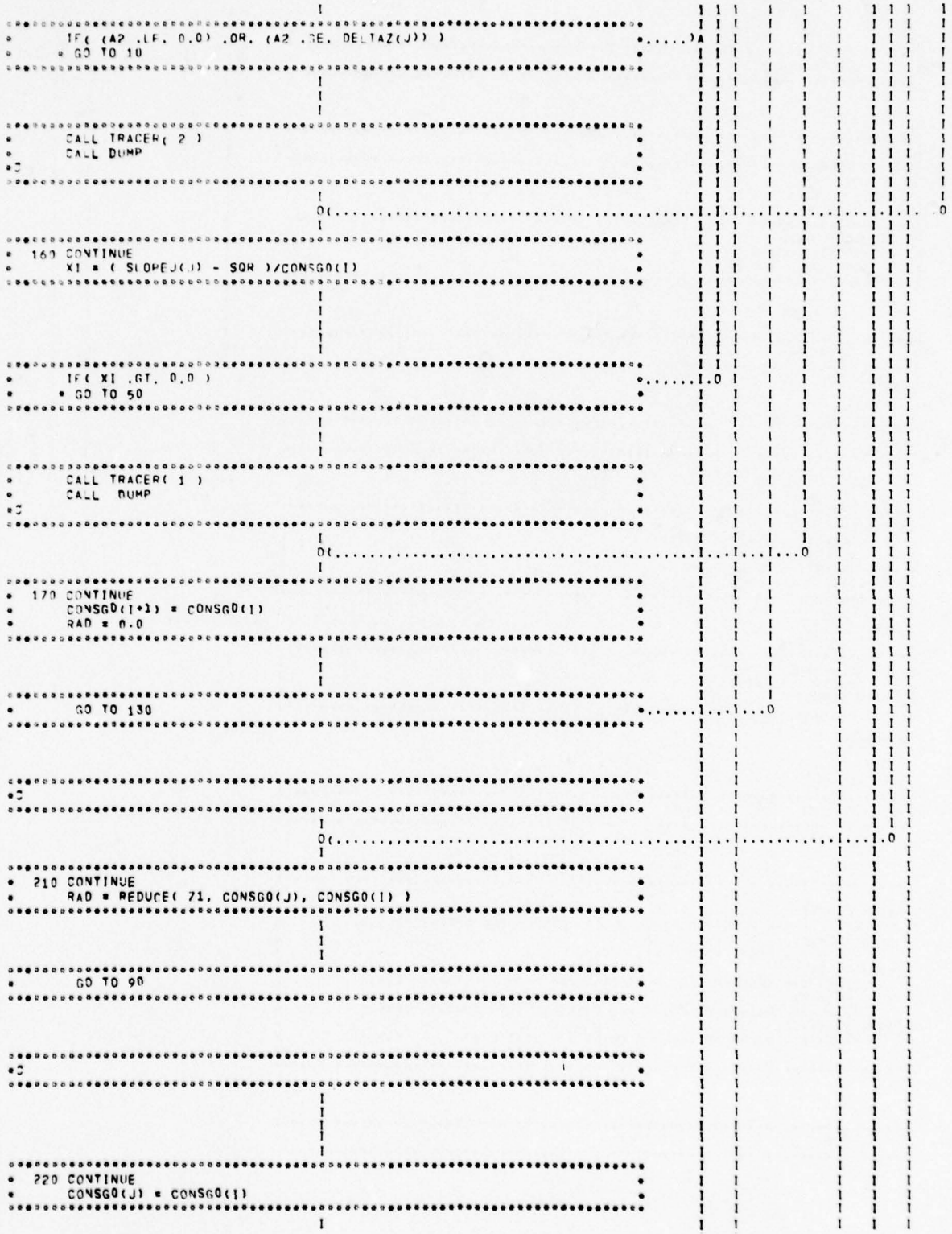


TOO MANY EXITS IN GO TO STATEMENT

TOO MANY EXITS IN GO TO STATEMENT




IF(SLOPEJ(J) .EQ. 0.0)
GO TO 230
A2 = SLOPEJ(J)/CONSG0(J)
IF(A2 .GT. 1.0)
GO TO 60
IF(A2 .EQ. 1.0)
GO TO 260
A1 = 1.0 - 0.25/(A2 - 0.5)
IF((A1 .LE. 0.0) .OR. (A1 .GE. 1.0))
GO TO 150
WRITE(6, 9000)
W DEPKYD(J),
W DEPKYD(I)
A1 = CONSV0(I) - CONSV0(J)
DZNEW = 2.5*A1/CONSG0(J) - 1.5*DELTAZ(J)
U1 = CONSV0(J) + CONSG0(J)*DZNEW
Y1 = DEPKYD(J) + DZNEW
CONSG0(I) = CONSG0(J)
CONSG1(J) = 0.0
CONSG2(J) = 0.0
SLOPEJ(I1) = SLOPEJ(I)
SLOPEJ(J) = CONSG0(J)
DELTAZ(I) = DELTAZ(J) - DZNEW
DELTAZ(J) = DZNEW
J = I
GO TO 70
250 CONTINUE
SLOPEJ(I) = (CONSV0(I1) - CONSV0(I))/DELTAZ(I)
I = I1
I1 = I+ 1
CONSG0(I) = 0.0
DELTAZ(I) = DEPKYD(I1) - DEPKYD(I)
150 CONTINUE
X1 = CONSG0(J)/2.0/SLOPEJ(J)
CONSG1(J) = (SLOPEJ(J)*X1*X1 - CONSG0(J))/DELTAZ(J)
CONSG2(J) = (X1 - 1.0)/DELTAZ(J)
A2 = CONSG0(J)*CONSG2(J)
A3 = CONSG1(J) - A2
A2 = (A3 - A2)/CONSG2(J)/(CONSG1(J) + A3)


IF((A2 .LE. 0.0) .OR. (A2 .GE. DELTAZ(J)))
GO TO 10
CALL TRACER(2)
CALL DUMP
160 CONTINUE
XI = (SLOPEJ(J) - SQR)/CONSG0(I)
IF(XI .GT. 0.0)
GO TO 50
CALL TRACER(1)
CALL DUMP
170 CONTINUE
CONSG0(I+1) = CONSG0(I)
RAD = 0.0
GO TO 130
210 CONTINUE
RAD = REDUCE(71, CONSG0(J), CONSG0(I))
GO TO 90
220 CONTINUE
CONSG0(J) = CONSG0(I)

```
      GO TO 190

C

  230 CONTINUE
      CONSG1(J) = 0.0
      CONSG2(J) = 0.0

      GO TO 10

C

  260 CONTINUE
      I1 = I - NULAPO + MAXLAY - 11
      NULAPO = MAXLAY
      WRITE ( 6, 7000 )
     W I1

  240 CONTINUE
      WRITE ( 6, 8000 )
     W DEPBOT

      RETURN

C
 6000 FORMAT(
     F/46H A NEW POINT IS BEING ADDED BETWEEN THE DEPTHS 2E15.6)
 7000 FORMAT(
     F 22H NEW POINT NEEDED AT    I4 )
 8000 FORMAT(
     F/46H THE CONSTANT MAXIMUM OCEAN DEPTH IS AT (K-YD)  E15.6)
 9000 FORMAT(
     F/46H AN INFLECTION POINT WAS FOUND BETWEEN POINTS  2E15.6)
10000 FORMAT(
     F/46H A MAXIMUM OR MINIMUM POINT IS LOCATED AT       E15.6)
C

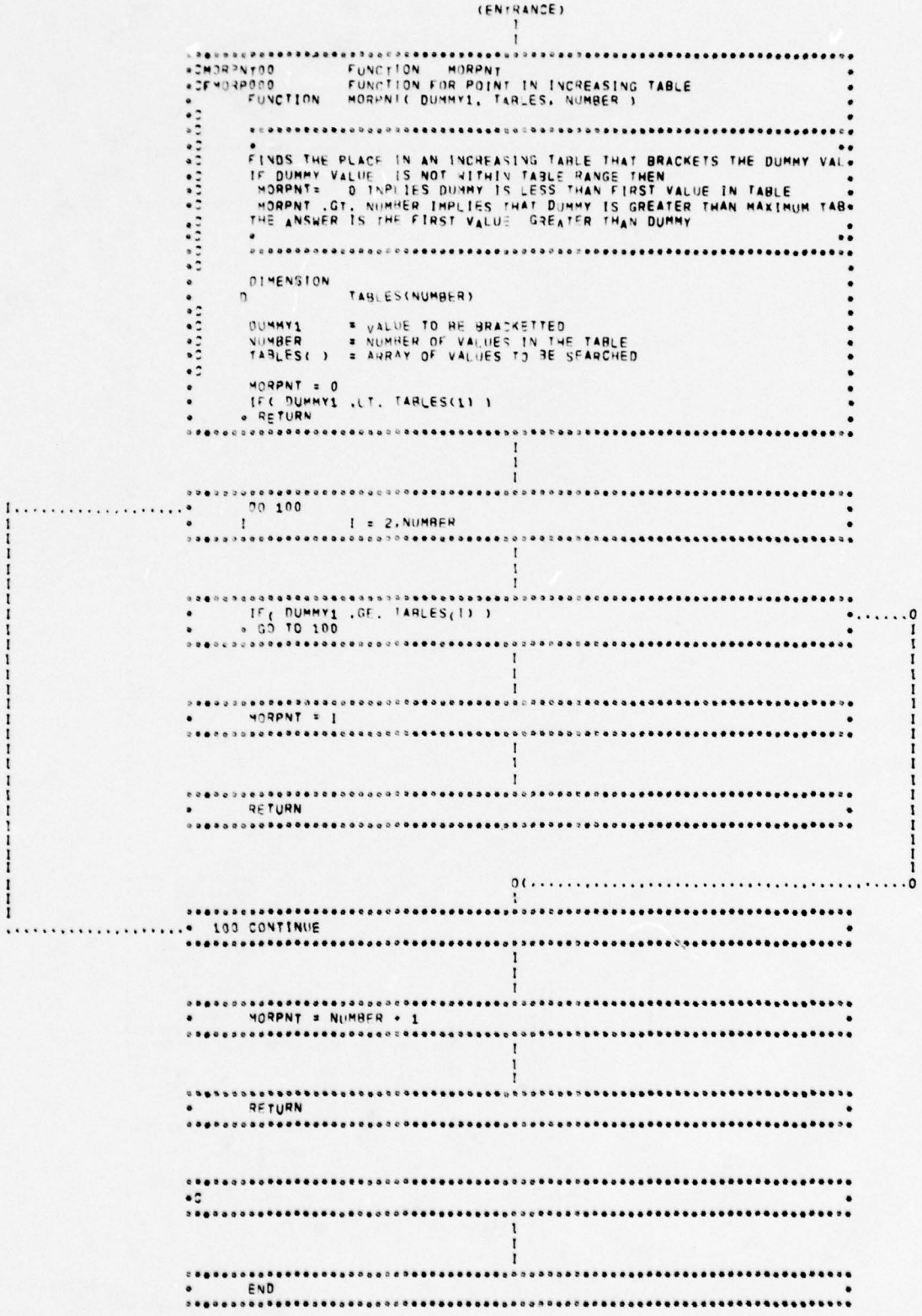
      END
```

(ENTRANCE)

```
C*LENGTH00        SUBROUTINE LENGTH                                          *
C                 FOR COMPUTING PATHLENGTHS, TIME AND DERIVATIVES            *
      SUBROUTINE                                                             *
     S            LENGTH                                                     *
     S ( N, M, DUMMY1, DUMMY2 )                                             *
C                                                                            *
C     ***************************************************************      *
C     *                                                             *      *
C     *                                                             *      *
C     ***************************************************************      *
C                                                                            *
      COMMON                                                                 *
     C / CONSTN /                                                            *
     C            DEPSER,                                                    *
     C            NCONSK,                                                    *
     C            SPATSK,                                                    *
     C            DEPEVA,                                                    *
     C            NCONSL,                                                    *
     C            SPATEV                                                     *
      COMMON                                                                 *
     C / LCONST /                                                            *
     C            NULAPO,                                                    *
     C            DEPROT,                                                    *
     D            CONSG0(128)       ,                                        *
     D            CONSG1(128)       ,                                        *
     D            CONSG2(128)       ,                                        *
     D            CONSV0(128)       ,                                        *
     D            DELTAZ(128)       ,                                        *
     D            DEPKY0(128)       ,                                        *
     D            SLOPE0(128)       ,                                        *
     D            SPDKY0(128)                                                *
      COMMON                                                                 *
     C / PTHLNG /                                                            *
     C            A1     ,                                                   *
     C            B1     ,                                                   *
     C            CDSORD,                                                    *
     C            C1     ,                                                   *
     C            D1     ,                                                   *
     C            DXDC   ,                                                   *
     C            DZ1    ,                                                   *
     C            DZM    ,                                                   *
     C            K      ,                                                   *
     C            PL     ,                                                   *
     C            SM     ,                                                   *
     C            H      ,                                                   *
     C            X      ,                                                   *
     C            Y1     ,                                                   *
     C            Y2     ,                                                   *
     C            TIMCON                                                     *
      COMMON                                                                 *
     C / RANGES /                                                            *
     C            NUMANG,                                                    *
     C            ANGMAX,                                                    *
     C            DELANG,                                                    *
     C            DELRAD,                                                    *
     D            ANGINT(200)                                                *
     D            RNGMOD(6,200)                                              *
      COMMON                                                                 *
     C / RAYPAR /                                                            *
     C            RANGEH,                                                    *
     D            BOTLOS(6)         ,                                        *
     D            DRDXDC(6)         ,                                        *
     D            PATHLN(6)         ,                                        *
     D            RANGEC(6)         ,                                        *
     D            SPI(6)            ,                                        *
     D            SPT(6)            ,                                        *
     D            TIR(6)            ,                                        *
     D            TTR   (6)                                                  *
      COMMON                                                                 *
     C / RAYTRA /                                                            *
     C            NCONCI,                                                    *
     C            INITLK,                                                    *
     C            Z1     ,                                                   *
     C            Z2     ,                                                   *
     C            SPVRSO,                                                    *
     C            ANGSTR,                                                    *
     C            ANGARR,                                                    *
     C            ANGRTM,                                                    *
     C            ANGSUR,                                                    *
     C            SPDVER,                                                    *
     C            RANGET                                                     *
```

```
*      COMMON                                                          *
*     C / SURDUC /                                                     *
*     C            BLA1  ,                                             *
*     C            BLA2  ,                                             *
*     C            BLA3  ,                                             *
*     C            DTRAD ,                                             *
*     C            J     ,                                             *
*     C            MSHIP ,                                             *
*     D            BF1   (128)      ,                                  *
*     D            BF2   (128)      ,                                  *
*     D            DELRAF(128,2),                                      *
*     D            FLN1(128)        ,                                  *
*     D            FLN2(128)        ,                                  *
*     D            CONLR2(40,50)    ,                                  *
*     D            CONLR1(40,50)    ,                                  *
*     NDUM1                                                            *
*C                                                                     *
*C     SPI( )      = SPREADING LOSS CONSTANT                  (DIMENS*
*C     SPT( )      = SPREADING LOSS CONSTANT                  (DIMENS*
*C                                                                     *
*      SPAFUN( DUMMY2 ) = ALOG10( COS( DUMMY2 )**2/A1 )                *
*     E - 6.0                                                          *
*C                                                                     *
*      INITLK = M                                                      *
*      Z1 = DUMMY1                                                     *
*      Z2 = DUMMY2                                                     *
*      CALL                                                            *
*     S            RATRAC                                             *
*      RANGEC(N) = RANGEC(N) + RANGET                                  *
*      ANGSTR = -ANGARR                                                *
*****************************************************************     *
                                    I                                  I
                                    I                                  I
                                    I                                  I
                                    I                                  I
                                    I                                  I
*****************************************************************     *
*      IF( Z2 .NE. DEPBOT )                                       *.....0
*     * GO TO 10                                                  *     I
*****************************************************************     I
                                    I                                  I
                                    I                                  I
                                    I                                  I
*****************************************************************     I
*      BOTLOS(N) = TANH( BLA1*ANGARR ) + BLA2*ANGARR*ANGARR       *     I
*****************************************************************     I
                                    I                                  I
                                    0(.....................................0
                                    I
*****************************************************************
*   10 CONTINUE                                                       *
*      IF( Z2 .NE. DEPEVA )                                           *
*     *RETURN                                                         *
*C                                                                     *
*      DRDXDC(N) = DXDC                                               *
*      PATHLN(N) = PL                                                 *
*      A1 = ABS( RANGEH*SIN( TIR(N) )*SIN( ANGARR )*SPDVER*DRDXDC(N) )*
*      SPI(N) = 10.0*SPAFUN( TIR(N) )                                 *
*      SPT(N) = 10.0*SPAFUN( ANGARR )                                 *
*      TTR(N) = ANGARR                                                *
*****************************************************************     *
                                    I                                  *
                                    I                                  *
                                    I                                  *
*****************************************************************     *
*      RETURN                                                         *
*****************************************************************     *

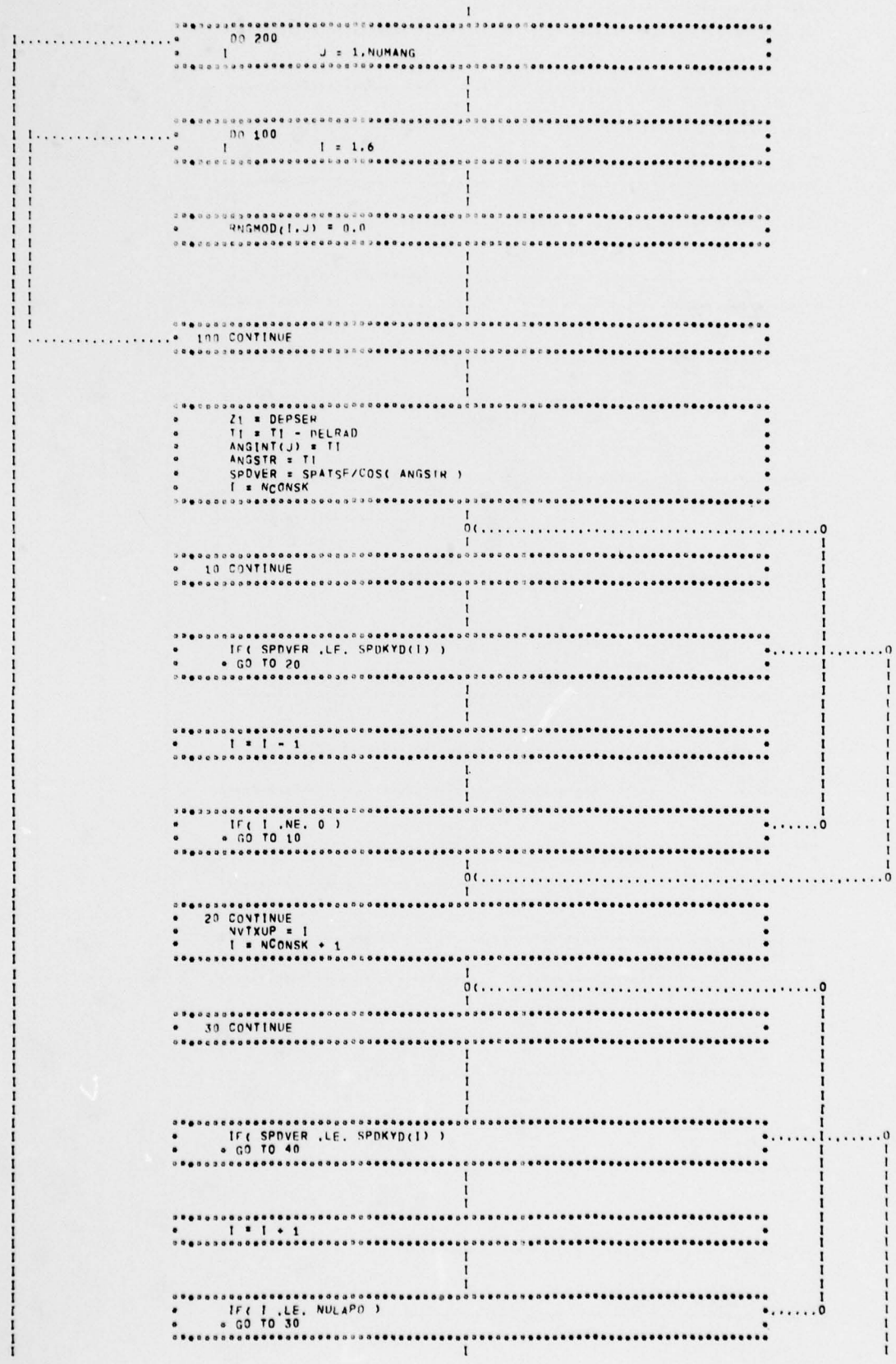
*****************************************************************     *
*C                                                                    *
*****************************************************************     *
                                    I                                  *
                                    I                                  *
                                    I                                  *
*****************************************************************     *
*      END                                                            *
*****************************************************************     *
```

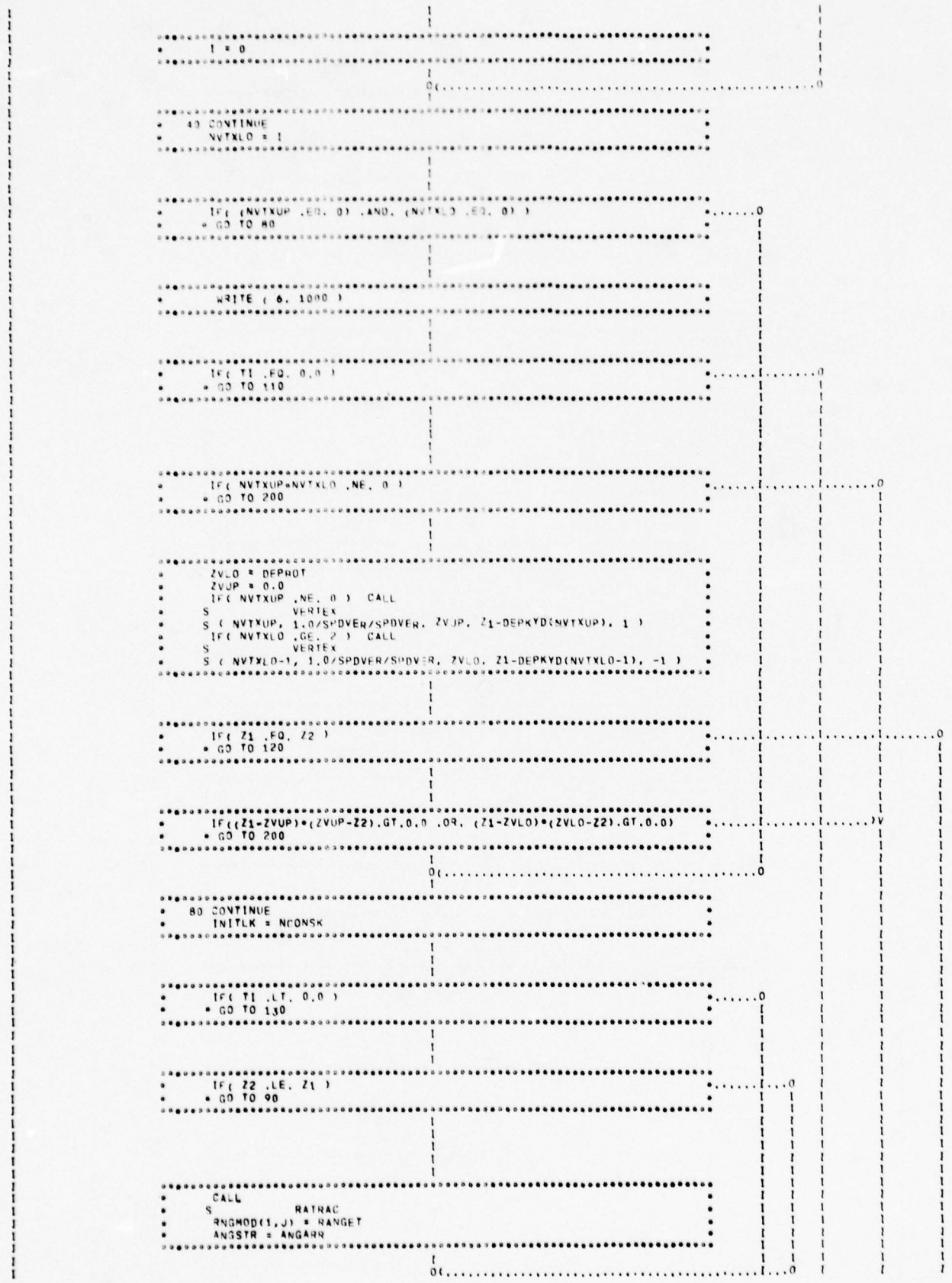

(ENTRANCE)
C MORPNT00 FUNCTION MORPNT
C FMORP000 FUNCTION FOR POINT IN INCREASING TABLE
FUNCTION MORPNT(DUMMY1, TABLES, NUMBER)
C
FINDS THE PLACE IN AN INCREASING TABLE THAT BRACKETS THE DUMMY VAL
IF DUMMY VALUE IS NOT WITHIN TABLE RANGE THEN
MORPNT= 0 IMPLIES DUMMY IS LESS THAN FIRST VALUE IN TABLE
MORPNT .GT. NUMBER IMPLIES THAT DUMMY IS GREATER THAN MAXIMUM TAB
THE ANSWER IS THE FIRST VALUE GREATER THAN DUMMY
DIMENSION
TABLES(NUMBER)
DUMMY1 = VALUE TO BE BRACKETTED
NUMBER = NUMBER OF VALUES IN THE TABLE
TABLES( ) = ARRAY OF VALUES TO BE SEARCHED
MORPNT = 0
IF(DUMMY1 .LT. TABLES(1))
RETURN
DO 100
I = 2,NUMBER
IF(DUMMY1 .GE. TABLES(I))
GO TO 100
MORPNT = I
RETURN
100 CONTINUE
MORPNT = NUMBER + 1
RETURN
C
END

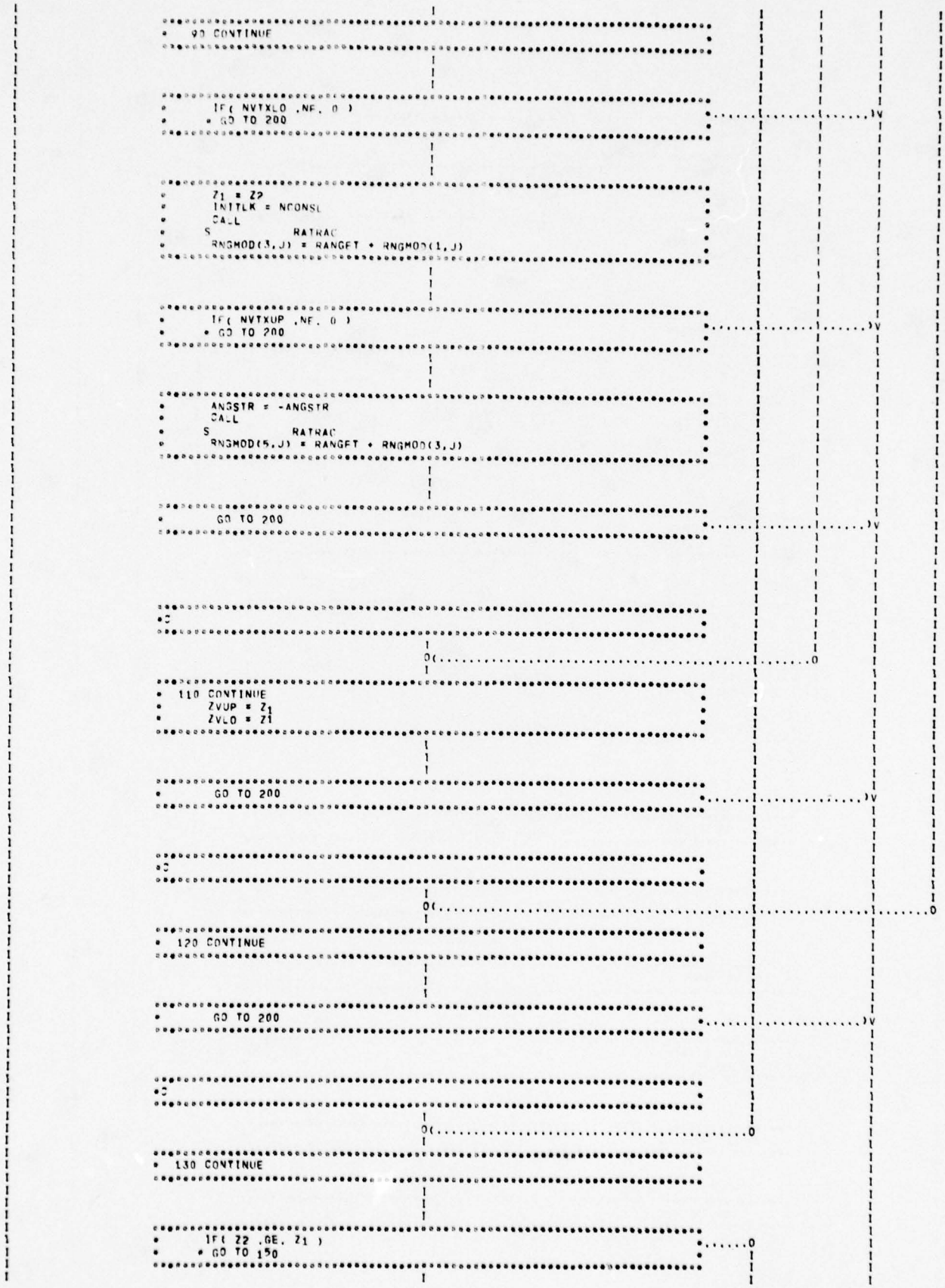
(ENTRANCE)

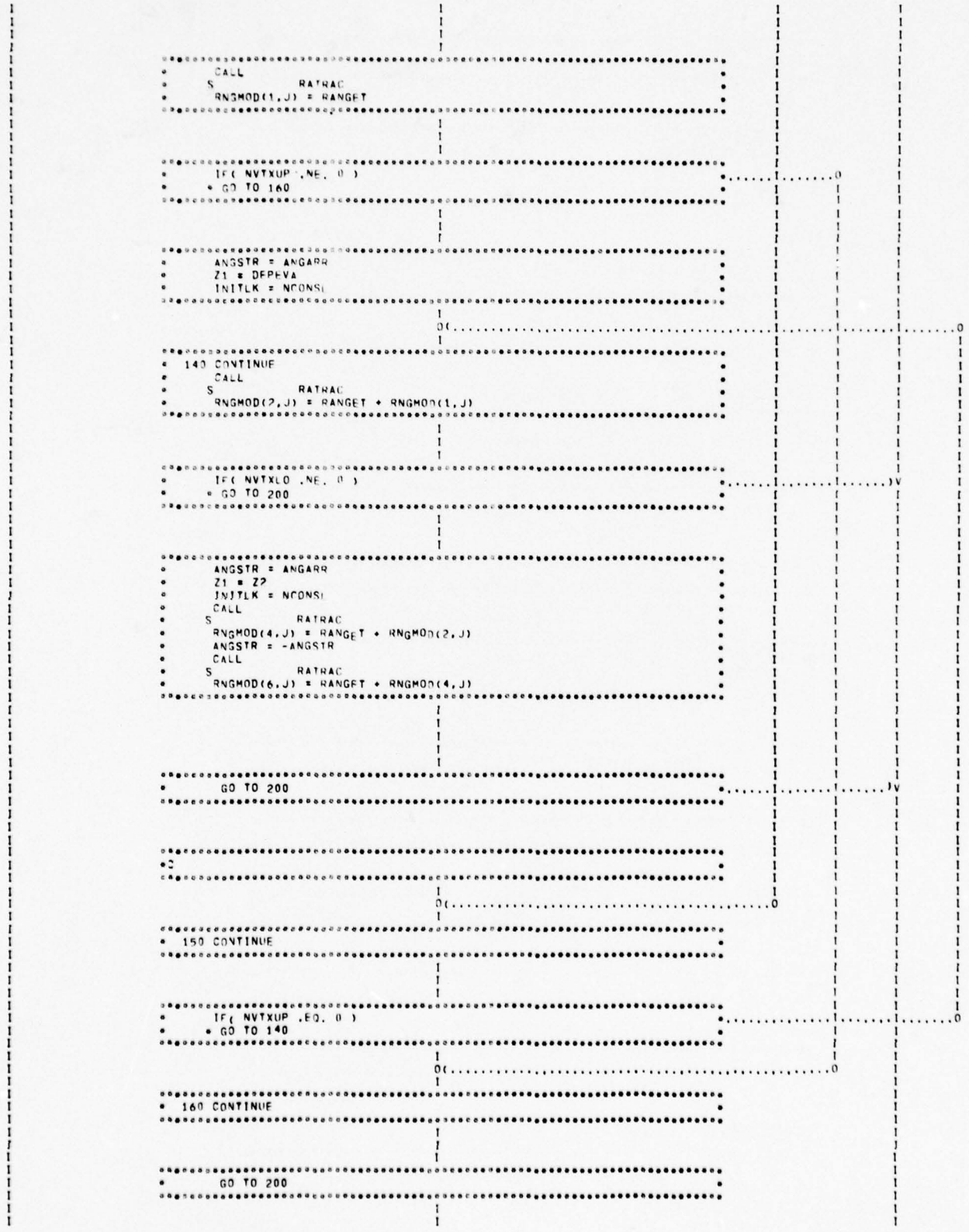
```
*CMSTRAY00        SUBROUTINE MSTRAY                                          *
*CSMSTA000        MASTER RAY-TRACING PROGRAM                                 *
*CSMSTR000        SUBROUTINE FOR SETTING UP RANGE TABLE FOR RAYS             *
      SUBROUTINE
     S            MSTRAY                                                    *
C                                                                           *
C     ***********************************************************           *
C     *                                                         *           *
C     THIS IS MASTER RAY TRACING PROGRAM                        *           *
C     *                                                         *           *
C     ***********************************************************           *
C                                                                           *
      COMMON
     C / CONSTN /
     C            DEPSER,                                                   *
     C            NCONSK,                                                   *
     C            SPATSE,                                                   *
     C            DEPEVA,                                                   *
     C            NCONSL,                                                   *
     C            SPATEV                                                    *
      COMMON
     C / LCONST /
     C            NULAPO,                                                   *
     C            DEPROT,                                                   *
     D            CONSG0(128)      ,                                        *
     D            CONSG1(128)      ,                                        *
     D            CONSG2(128)      ,                                        *
     D            CONSV0(128)      ,                                        *
     D            DELTAZ(128)      ,                                        *
     D            DEPKYD(128)      ,                                        *
     D            SLOPEJ(128)      ,                                        *
     D            SPDKYD(128)                                               *
      COMMON
     C / RANGES /
     C            NUANMO,                                                   *
     C            ANGMAX,                                                   *
     C            DELANG,                                                   *
     C            DELRAD,                                                   *
     D            ANGINT(200)                                               *
     D            RNGMOD(6,200)                                             *
      COMMON
     C / RAYTRA /
     C            NCONCJ,                                                   *
     C            INITLK,                                                   *
     C            Z1     ,                                                  *
     C            Z2     ,                                                  *
     C            SPVRSD,                                                   *
     C            ANGSTR,                                                   *
     C            ANGARR,                                                   *
     C            ANGBTM,                                                   *
     C            ANGSUR,                                                   *
     C            SPDVER,                                                   *
     C            RANGET                                                    *
C                                                                           *
      ANGARR      = ANGLE (IN RADIANS) AT ARRIVAL POINT (Z2)                *
      ANGBTM      = ANGLE (IN RADIANS) OF BOTTOM BOUNCE OF RAY              *
      ANGMAX      = MAXIMUM ANGLE BEING CONSIDERED FOR RAYS                 (*
      ANGSTR      = ANGLE (IN RADIANS) OF START OF RAY (Z1)                 *
      ANGSUR      = ANGLE (IN RADIANS) OF SURFACE BOUNCE OF RAY             *
      CONSG0( )   = G0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
      CONSG1( )   = G1 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
      CONSG2( )   = G2 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
      CONSV0( )   = V0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
      DELSER      = DEPTH (IN K-YD) OF SEARCHER IN ITS LAYER                *
      DEPKYD( )   = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE      *
      DGPRRD      = DEGREES PER RADIAN                                      (DEGREES*
      DZVX        =                                                         *
      INITLK      = INITIAL VALUE OF K I.E. THE INITIAL STARTING LAYER      *
      NCONSK      = LAYER NUMBER OF LAYER IN WHICH SEARCHER IS FOUND        *
      NCONSL      = LAYER NUMBER OF LAYER IN WHICH ECHADER IS FOUND         *
      NUANMO      = NUMBER OF ANGLES FOR RAYS MINUS ONE                     (DIMENS*
      NUMANG      = NUMBER OF DIFFERENT ANGLES BEING CONSIDERED FOR RAYS    *
      NULAPO      = NUMBER OF LAYERS PLUS ONE                               *
      NVTXLO      = LAYER NUMBER BELOW LOWER VERTEX LAYER                   (DIMENS*
      NVTXUP      = LAYER NUMBER IN WHICH RAY VERTEXES                      (DIMENS*
      R           = SUMMATION OF THE DIFFERENT LAYER RANGES                 *
      RANGET      = SUMMATION OF THE DIFFERENT LAYER RANGES                 *
      RNGMOD( , )= RANGE (IN K-YD) FOR THE DIFFERENT MODES OF PROPAGATIO*
                   FIRST SUBSCRIPT IS MODE TYPE                             *
                   SECOND SUBSCRIPT IS RELATED TO DEPARTURE ANGLE           *
      SPDKYD( )   = SPEED OF SOUND PROPAGATION (IN K-YD/SEC)                *
      SPLASE      = PROPAGATION VELOCITY (IN K-YD/SEC) IN LAYER FOR SEARC*
      TI          = STARTING ANGLE (IN DEGREES) OF RAY FROM SEARCHER        *
      ZVLO        = DEPTH (IN K-YD) OF VERTEXING POINT                      *
      ZVUP        = DEPTH OF UPPER VERTEX POINT FOR RAY                     *
C                                                                           *
      DGPRRD = 57.2957795                                                   *
      Z2 = DEPEVA                                                           *
      TI = ANGMAX/DGPRRD + DELRAD                                           *
      NUMANG = 2.0*ANGMAX/DELANG + 1.0                                      *
```

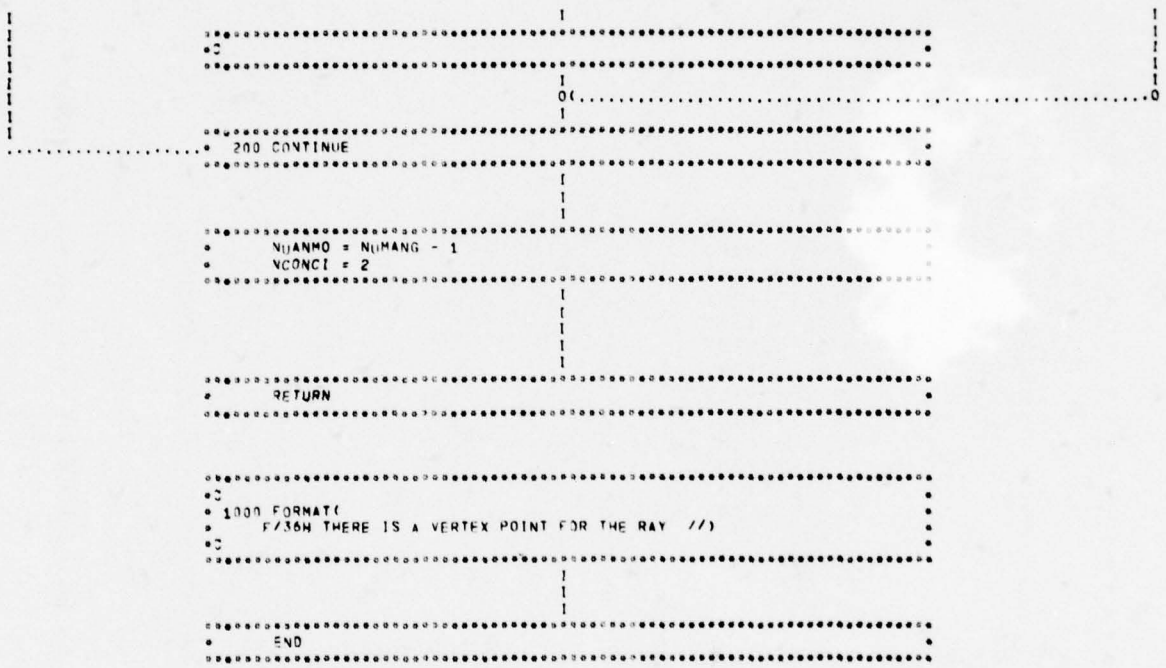
(EXIT)


DO 200
J = 1,NUMANG
DO 100
I = 1,6
RNGMOD(I,J) = 0.0
100 CONTINUE
Z1 = DEPSEA
TI = TI - DELRAD
ANGINT(J) = TI
ANGSTR = TI
SPDVER = SPATSF/COS(ANGSTR)
I = NCONSK
10 CONTINUE
IF(SPDVER .LE. SPDKYD(I))
GO TO 20
I = I - 1
IF(I .NE. 0)
GO TO 10
20 CONTINUE
NVTXUP = I
I = NCONSK + 1
30 CONTINUE
IF(SPDVER .LE. SPDKYD(I))
GO TO 40
I = I + 1
IF(I .LE. NULAPO)
GO TO 30


I = 0
40 CONTINUE
NVTXLO = 1
IF((NVTXUP .EQ. 0) .AND. (NVTXLO .EQ. 0))
GO TO 80
WRITE (6, 1000)
IF(TI .EQ. 0.0)
GO TO 110
IF(NVTXUP*NVTXLO .NE. 0)
GO TO 200
ZVLO = DEPROT
ZVUP = 0.0
IF(NVTXUP .NE. 0) CALL
S VERTEX
S (NVTXUP, 1.0/SPDVER/SPDVER, ZVUP, Z1-DEPKYD(NVTXUP), 1)
IF(NVTXLO .GE. 2) CALL
S VERTEX
S (NVTXLO-1, 1.0/SPDVER/SPDVER, ZVLO, Z1-DEPKYD(NVTXLO-1), -1)
IF(Z1 .EQ. Z2)
GO TO 120
IF((Z1-ZVUP)*(ZVUP-Z2).GT.0.0 .OR. (Z1-ZVLO)*(ZVLO-Z2).GT.0.0)
GO TO 200
80 CONTINUE
INITLK = NCONSK
IF(TI .LT. 0.0)
GO TO 130
IF(Z2 .LE. Z1)
GO TO 90
CALL
S RATRAC
RNGMOD(1,J) = RANGET
ANGSTR = ANGARR


90 CONTINUE
IF(NVTXLO .NE. 0)
GO TO 200
Z1 = Z2
INITLK = NCONSL
CALL
S RATRAC
RNGMOD(3,J) = RANGET + RNGMOD(1,J)
IF(NVTXUP .NE. 0)
GO TO 200
ANGSTR = -ANGSTR
CALL
S RATRAC
RNGMOD(5,J) = RANGET + RNGMOD(3,J)
GO TO 200
C
110 CONTINUE
ZVUP = Z1
ZVLO = Z1
GO TO 200
C
120 CONTINUE
GO TO 200
C
130 CONTINUE
IF(Z2 .GE. Z1)
GO TO 150


CALL
S RATRAC
RNGMOD(1,J) = RANGET
IF(NVTXUP .NE. 0)
GO TO 160
ANGSTR = ANGARR
Z1 = DEPEVA
INITLK = NCONSI
140 CONTINUE
CALL
S RATRAC
RNGMOD(2,J) = RANGET + RNGMOD(1,J)
IF(NVTXLO .NE. 0)
GO TO 200
ANGSTR = ANGARR
Z1 = Z2
INITLK = NCONSI
CALL
S RATRAC
RNGMOD(4,J) = RANGET + RNGMOD(2,J)
ANGSTR = -ANGSTR
CALL
S RATRAC
RNGMOD(6,J) = RANGET + RNGMOD(4,J)
GO TO 200
C
150 CONTINUE
IF(NVTXUP .EQ. 0)
GO TO 140
160 CONTINUE
GO TO 200


200 CONTINUE
NUANMO = NUMANG - 1
NCONCI = 2
RETURN
1000 FORMAT(
F/36H THERE IS A VERTEX POINT FOR THE RAY ///)
END

(ENTRANCE)

```
*CMUEVAR00          SUBROUTINE MUEVAR                                          *
*CMUEVA000          SUBROUTINE FOR COMPUTING MU- AND VARIANCE FOR SHIPS         *
*       SUBROUTINE                                                            *
*      S            MUEVAR                                                    *
*
*       ************************************************************          **
*       *                                                                     *
*       ************************************************************
*
*       COMMON                                                                *
*      C / CALPHA /                                                           *
*      C            ALPHAE,                                                   *
*      C            ALPHAS,                                                   *
*      C            FALPHE,                                                   *
*      C            FALPHS,                                                   *
*      C            ONMIAE,                                                   *
*      C            ONMIAS,                                                   *
*      C            STNPEV,                                                   *
*      C            STNPSE,                                                   *
*      C            TWOALE,                                                   *
*      C            TWOALS,                                                   *
*      C            NTIMEM                                                    *
*       COMMON                                                                *
*      C / SIGNAL /                                                           *
*      D            PRYSEV(128)      ,                                        *
*      D            PRYSSE(128)      ,                                        *
*      D            PRNOEV(128)      ,                                        *
*      D            PRNOSE(128)      ,                                        *
*      D            PROEVA(128)      ,                                        *
*      D            PROSER(128)      ,                                        *
*      D            VAREVA(128)      ,                                        *
*      D            VARSER(128)      ,                                        *
*      D            GMUEVA(128)      ,                                        *
*      D            GMUSER(128)      ,                                        *
*      D            DEVAEV(128)      ,                                        *
*      D            DEVASE(128)      ,                                        *
*      D            DEMUEV(128)      ,                                        *
*      D            DEMUSE(128)      ,                                        *
*      C            THREVA,                                                   *
*      C            THRSER,                                                   *
*      C            NTIMEN                                                    *
*
*       DEMUEV( )  = MODIFIED MEAN OF S/N FOR THE EVADER                      *
*       DEMUSE( )  = MODIFIED MEAN OF S/N FOR THE SEARCHER                    *
*       DEVAEV( )  = MODIFIED VARIANCE OF S/N FOR THE SEARCHER                *
*       DEVASE( )  = MODIFIED VARIANCE OF S/N FOR THE SEARCHER                *
*       FALPHE     = FRACTION  (ONMIAE**2/(1-ONMIAE**2))          (DIMENS*
*       FALPHS     = FRACTION  (ONMIAS**2/(1-ONMIAS**2))          (DIMENS*
*       GMUEVA( )  = MEAN OF SMOOTHED EVADER S/N                              *
*       GMUSER( )  = MEAN OF SMOOTHED SEARCHER S/N                            *
*       NTIMEM     = NTIMEN MINUS ONE                             (DIMENS*
*       OMASQE     = ONE MINUS ALPHA-E   SQUARED                  (DIMENS*
*       OMASQS     = ONE MINUS ALPHA-S   SQUARED                  (DIMENS*
*       ONMIAE     = ONE MINUS ALPHAE                             (DIMENS*
*       ONMIAS     = ONE MINUS ALPHAS                             (DIMENS*
*       STNPSE     = SIGNAL TO NOISE RATIO PLUS ONE FOR THE SEARCHER      *
*       STNPEV     = SIGNAL TO NOISE RATIO PLUS ONE FOR THE EVADER        *
*       TWOALE     = TWO TIMES ALPHAE                             (DIMENS*
*       TWOALS     = TWO TIMES ALPHAS                             (DIMENS*
*       VAREVA( )  = VARIANCE OF SMOOTHED S/N FOR EVADER                      *
*       VARSER( )  = VARIANCE OF SMOOTHED S/N FOR SEARCHER                    *
*
*************************************************************************
```


IF(NTIMEN .EQ. 1)
GO TO 10
NTIMEM = NTIMEN - 1
OMASQS = ONMIAS*ONMIAS
OMASQE = ONMIAE*ONMIAE
GMUSER(NTIMEN) = ALPHAS*STNPSE + ONMIAS*DEMUSE(NTIMEM)
GMUEVA(NTIMEN) = ALPHAE*STNPEV + ONMIAE*DEMUEV(NTIMEM)
VARSER(NTIMEN) = TWOALS*STNPSE*STNPSE + OMASQS*DEVASE(NTIMEM)
VAREVA(NTIMEN) = TWOALE*STNPEV*STNPEV + OMASQE*DEVAEV(NTIMEM)
RETURN
10 CONTINUE
GMUSER(1) = ALPHAS*STNPSE + ONMIAS
GMUEVA(1) = ALPHAE*STNPEV + ONMIAE
VARSER(1) = TWOALS*(STNPSE*STNPSE + FALPHS)
VAREVA(1) = TWOALE*(STNPEV*STNPEV + FALPHE)
RETURN
END

```
                                    (ENTRANCE)
                                        I
                                        I
*********************************************************************************
*CNTSE        NORMAL TO SEARCH EVASION                                            *
*C  ***************************************************************************
*C  *                                                                            *
*C     *NORMAL TO SEARCH EVASION                                                 *
*C  *                                                                            *
*C  ***************************************************************************
*       SUBROUTINE NTSE                                                         *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,VPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                       *
*       N1 = N - 1                                                             *
*       A1 = VXS(N1)/VYS(N1)                                                   *
*       VECTXN = ( PXE(N1) - A1*PYE(N1)) /( 1.0 + A1*A1 )                      *
*       VECTYN = -A1*VECTXN                                                    *
*       SQSV = SQRT (VECTXN**2+VECTYN**2)                                      *
*       DXN=VECTXN/SQSV                                                        *
*       DYN=VECTYN/SQSV                                                        *
*       VXE(N-1)=SE1*DXN                                                       *
*       VYE(N-1)=SE1*DYN                                                       *
*********************************************************************************
                                        I
                                        I
                                        I
*********************************************************************************
*       RETURN                                                                  *
*********************************************************************************



*********************************************************************************
*       END                                                                     *
*********************************************************************************
```

```
                                    (ENTRANCE)
                                        I
                                        I
*********************************************************************************
*CPAPERF00          FUNCTION   PAPERF                                             *
*CPAPER000          PAPOULIS ERROR FUNCTION ROUTINE                               *
*       FUNCTION    PAPERF( DUMMY1 )                                              *
*C                                                                                *
*C      ***************************************************************************
*C      *                                                                         **
*C      THIS IS THE PAPOULIS ERROR FUNCTION WHICH HAS RESULTS BETWEEN  0 A*
*C      *                                                                         **
*C      ***************************************************************************
*C                                                                                *
*C      DUMMY1      = X OF EQUATION                                               *
*C                    1/SQRT(2*PI) TIMES THE INTEGRAL FROM ZERO TO X  OF          *
*C                    EXP( -Y**2/2 )  W.R.T. Y                                    *
*C                    SEE PAGE 64 OF -PAPOULIS-                                   *
*C                                                                                *
*       PAPERF = 0.5*( 1.0 + ERF( 0.707107*DUMMY1 ) )                            *
*********************************************************************************
                                        I
                                        I
                                        I
*********************************************************************************
*       RETURN                                                                    *
*********************************************************************************



*********************************************************************************
*C                                                                                *
*********************************************************************************
                                        I
                                        I
                                        I
*********************************************************************************
*       END                                                                       *
*********************************************************************************
```

```
                                    (ENTRANCE)
                                        I
                                        I
*****************************************************************************
*CPROB       PROBABILITY DEP. ON STATE                                        *
*C   ***************************************************************       *
*C   *                                                             *       *
*C     *GENERATION OF PROB DEPENDING ON SHIP STATE                 *       *
*C   *                                                             *       *
*C   ***************************************************************       *
*       SUBROUTINE PROBL                                                    *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2VEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                   *
*       COMMON                                                             *
*      C / STATIC /                                                        *
*      C            NDSTAT,                                               *
*      C            NESTAT,                                               *
*      D            PEDETN(128)      ,                                    *
*      C            PERANG,                                               *
*      C            PPATHM,                                               *
*      D            PRANGE(128)      ,                                    *
*      D            PRANGS(128)      ,                                    *
*      D            PRPTEV(128)      ,                                    *
*      D            PRPTSE(128)      ,                                    *
*      D            PSDETN(128)      ,                                    *
*      C            PSRANG,                                               *
*      C            SMPEDT,                                               *
*      C            SMPSDT,                                               *
*      C            SMTCNE,                                               *
*      C            SMTCNS,                                               *
*      D            TCONEN(128)      ,                                    *
*      D            TCONSN(128)      ,                                    *
*      COUMMYB                                                            *
*C                                                                         *
*       REAL MULT2                                                         *
*C                                                                         *
*       I=STATD                                                            *
*       J=STATE                                                            *
*       PES1=1.-PE(J)                                                      *
*       PGMULT=PGS(I)*PGE(J)                                               *
*       DIV=1.-PDS(I)*PKDS(I)*PE(J)                                        *
*       MULT2=1.-PDS(I)*PKDS(I)                                            *
*       PKILL(N)=(PGMULT*PDS(I)*PKDS(I)*PES1)/DIV                          *
*       PEVADE(N)=(PGMULT*PE(J)*MULT2)/DIV                                 *
*       PPATH(N)=(PGMULT*MULT2*PES1)/DIV                                   *
*       NDSTAT = STATD                                                     *
*       NESTAT = STATE                                                     *
*       FACTOR = PGS(NDSTAT)*PGE(NESTAT)                                   *
*       PEDETN(N) = FACTOR*PDE(NESTAT)                                     *
*       PRANGE(N) = FACTOR*RANGE(N)*PDE(NESTAT)                            *
*       PRANGS(N) = FACTOR*RANGE(N)*PDS(NDSTAT)                            *
*       PRNTEV = 1.0 - PDE(NESTAT)                                         *
*       PRPTEV(N) = FACTOR*PRNTEV                                          *
*       PRNTSE = 1.0 - PDS(NDSTAT)                                         *
*       PRPTSE(N) = FACTOR*PRNTSE                                          *
*       PSDETN(N) = FACTOR*PDS(NDSTAT)                                     *
*       TCONEN(N) = PEDETN(N)*DIFTI                                        *
*       TCONSN(N) = PSDETN(N)*DIFTI                                        *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*       RETURN                                                              *
*****************************************************************************


*****************************************************************************
*C                                                                          *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*       END                                                                 *
*****************************************************************************
```


(ENTRANCE)
CPRBJ UPDATING PROBABILITY
UPDATING PROBABILITIES
SUBROUTINE PROBAL
COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),PYE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(54),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC6,FOS,FBWS,F1S,PTS(128),PXS(128),PNS(128),FNS(128),FOE,FBSE,F1E,F2E7,F2S,PTE(128),PXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX
COMMON
C / STATIC /
C NDSTAT,
C NESTAT,
D PEDETN(128) ,
C PERANG,
C PPATHM,
D PRANGE(128) ,
D PRANGS(128) ,
D PRPTEV(128) ,
D PRPTSE(128) ,
D PSDETN(128) ,
C PSRANG,
C SMPEDT,
C SMPSDT,
C SMTCNE,
C SMTCNS,
D TCONEN(128) ,
D TCONSN(128) ,
CDUMMYB
NTIMEM = N - 1
PKILL(N) = PKILL(N)*PPATHM
PPATH(N) = PPATH(N)*PPATHM
PEVADE(N) = PEVADE(N)*PPATHM
PEDETN(N) = PEDETN(N)*PRPTEV(NTIMEM)
PRANGE(N) = PRANGE(N)*PRPTEV(NTIMEM)
PRPTEV(N) = PRPTEV(N)*PRPTEV(NTIMEM)
TCONEN(N) = TCONEN(N)*PRPTEV(NTIMEM)
PRANGS(N) = PRANGS(N)*PRPTSE(NTIMEM)
PRPTSE(N) = PRPTSE(N)*PRPTSE(NTIMEM)
PSDETN(N) = PSDETN(N)*PRPTSE(NTIMEM)
TCONSN(N) = TCONSN(N)*PRPTSE(NTIMEM)
RETURN
END

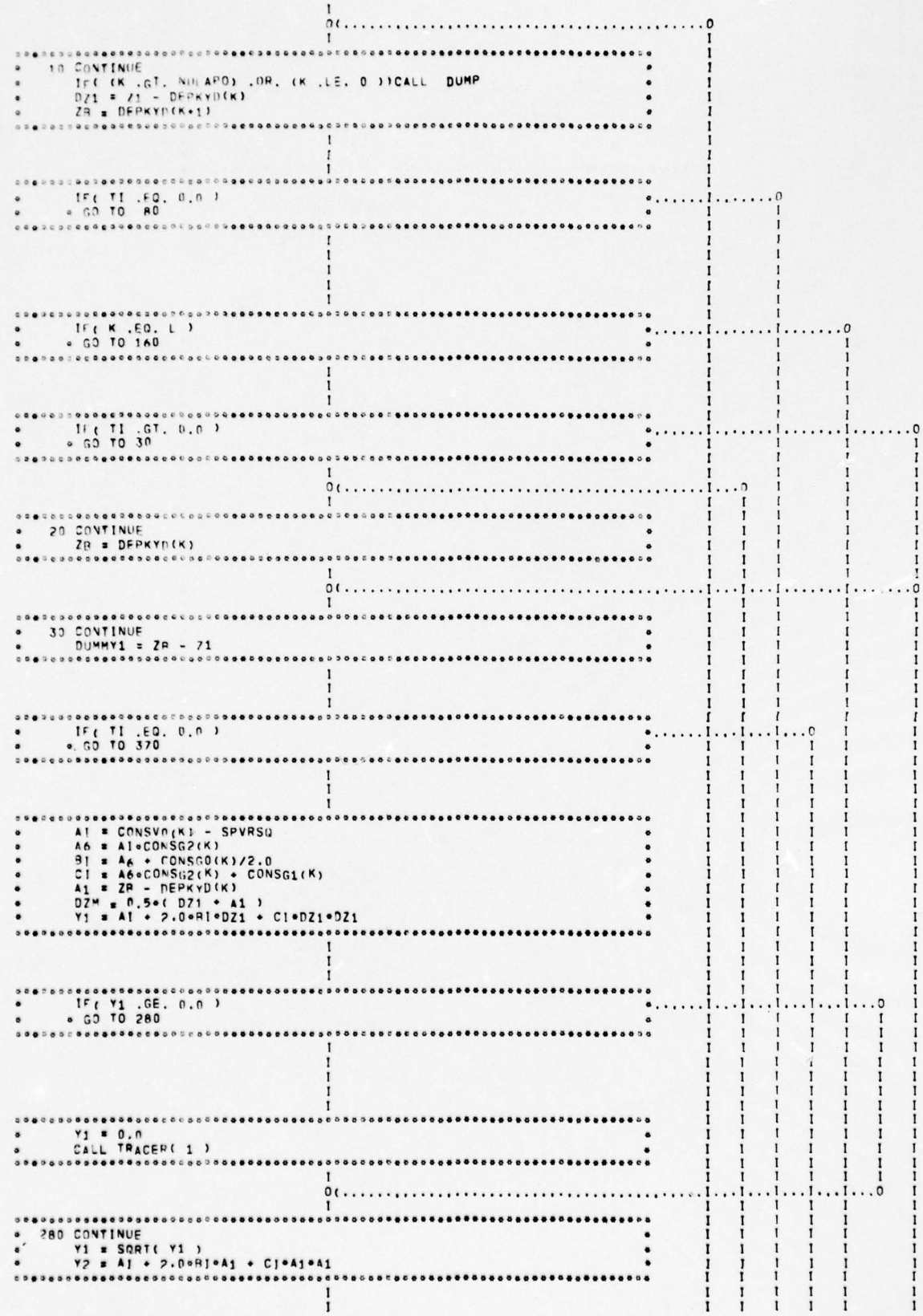
(ENTRANCE)

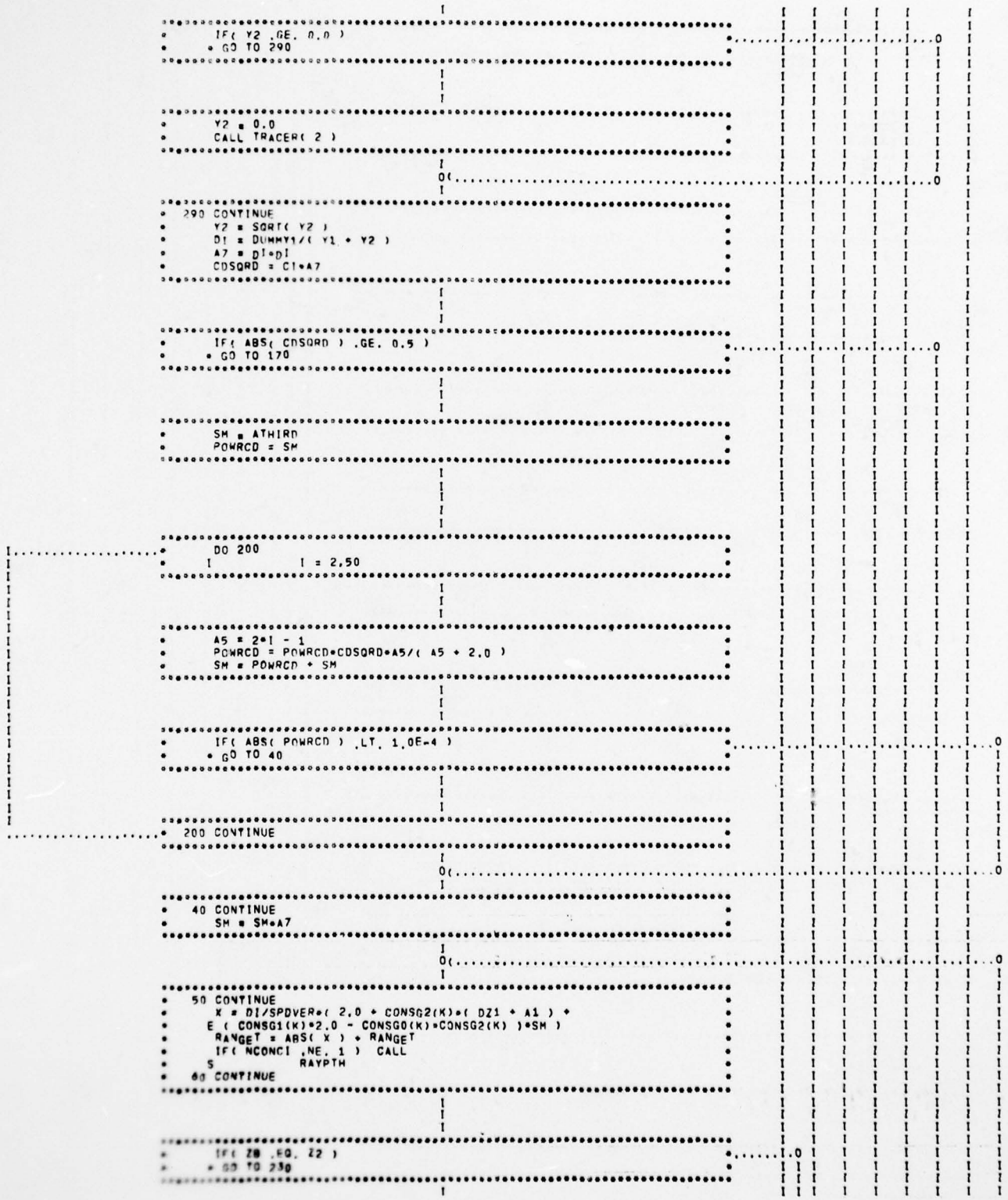
```
*CPRODET00          SUBROUTINE PRODET                                          *
*CPRODE000          SUBROUTINE FOR COMPUTING THE DETECTION PROBABILITIES       *
*       SUBROUTINE                                                            *
*      S            PRODET                                                    *
*C                                                                             *
*C     *****************************************************************     *
*C     *                                                               *     *
*C     *                                                               *     *
*C     *****************************************************************     *
*                                                                             *
*      COMMON                                                                 *
*     C / SIGNAL /                                                            *
*     D             PRYSEV(128)      ,                                        *
*     D             PRYSSE(128)      ,                                        *
*     D             PRNOEV(128)      ,                                        *
*     D             PRNOSE(128)      ,                                        *
*     D             PROEVA(128)      ,                                        *
*     D             PROSER(128)      ,                                        *
*     D             VAREVA(128)      ,                                        *
*     D             VARSER(128)      ,                                        *
*     D             GMUEVA(128)      ,                                        *
*     D             GMUSER(128)      ,                                        *
*     D             DEVAEV(128)      ,                                        *
*     D             DEVASE(128)      ,                                        *
*     D             DEMUEV(128)      ,                                        *
*     D             DEMUSE(128)      ,                                        *
*     C             THREVA,                                                  *
*     C             THRSER,                                                  *
*     C             NTIMEN                                                   *
*C                                                                            *
*C     GMUEVA( )  = MEAN OF SMOOTHED EVADER S/N                               *
*C     GMUSER( )  = MEAN OF SMOOTHED SEARCHER S/N                             *
*C     VAREVA( )  = VARIANCE OF SMOOTHED S/N FOR EVADER                       *
*C     VARSER( )  = VARIANCE OF SMOOTHED S/N FOR SEARCHER                     *
*C     PROSER( )  = PROBABILITY OF DETECTION BY THE SEARCHER                  *
*C     PROEVA( )  = PROBABILITY OF DETECTION BY THE EVADER                    *
*C                                                                            *
*      PROSER(NTIMEN) =                                                       *
*     E PAPERF( ( GMUSER(NTIMEN) - THRSER )/SQRT( VARSER(NTIMEN) ) )          *
*      PROEVA(NTIMEN) =                                                       *
*     E PAPERF( ( GMUEVA(NTIMEN) - THREVA )/SQRT( VAREVA(NTIMEN) ) )          *
*****************************************************************************
                                    I
                                    I
                                    I
*****************************************************************************
*       RETURN                                                                *
*****************************************************************************

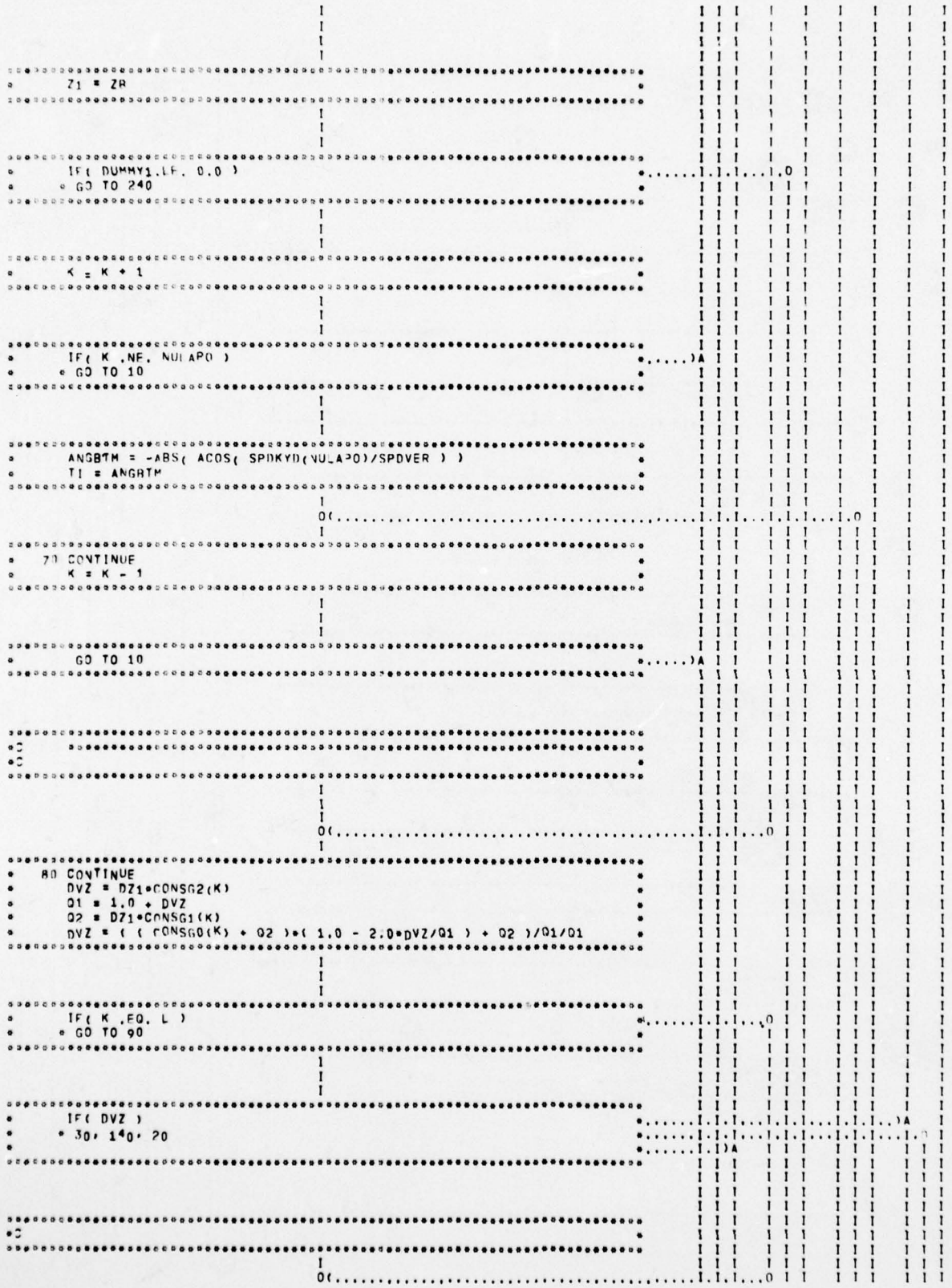
*****************************************************************************
*C                                                                            *
*****************************************************************************
                                    I
                                    I
                                    I
*****************************************************************************
*       END                                                                   *
*****************************************************************************
```

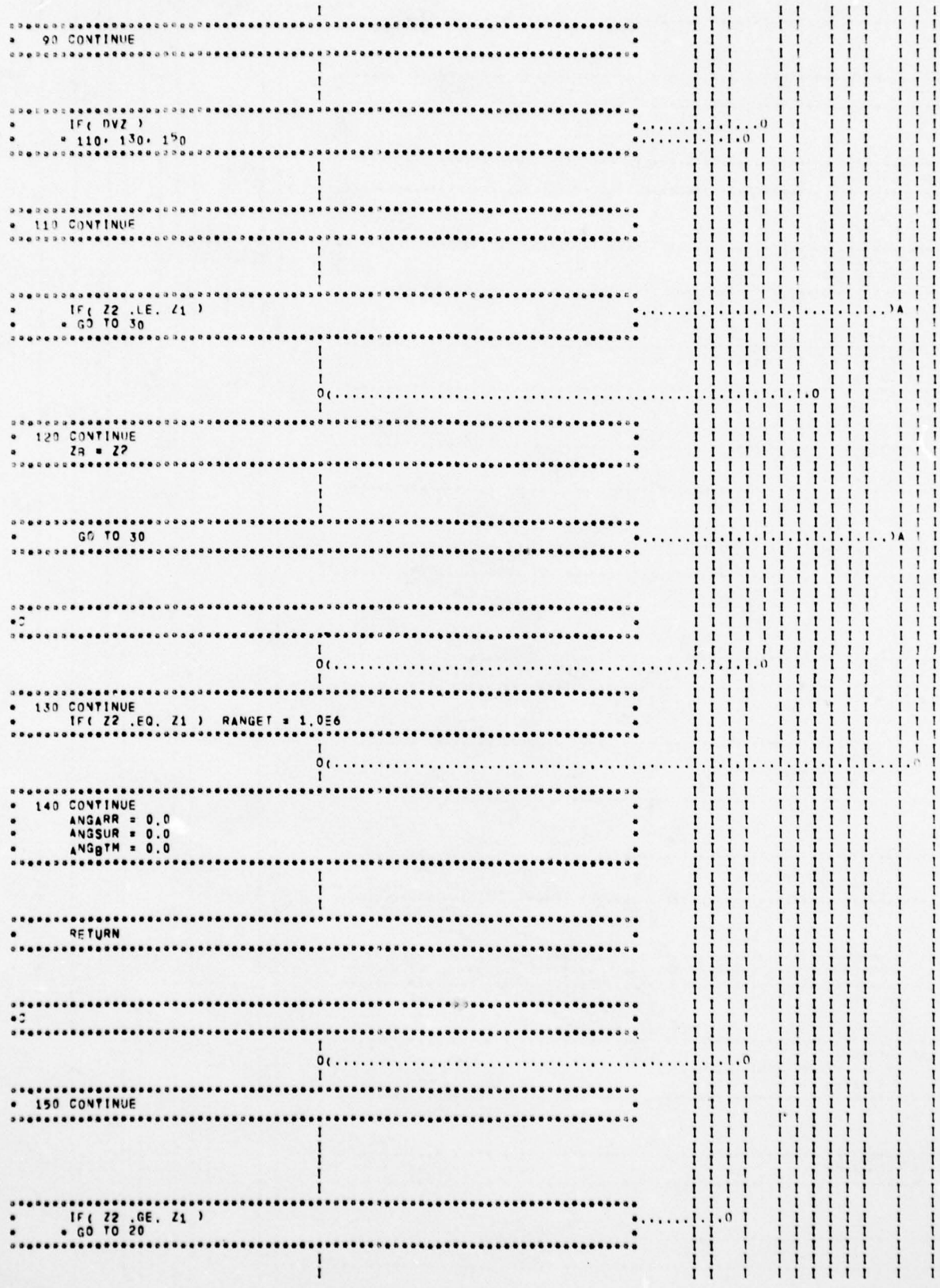
(ENTRANCE)

```
*CRATRAC00        SUBROUTINE RATRAC                                          *
*CSRATR000        SUBROUTINE FOR RAY-TRACING                                 *
      SUBROUTINE
     S            RATRAC                                                     *
C                                                                            *
C     ***********************************************************************
C     *                                                                     **
C     THIS IS THE RAY-TRACING PROGRAM                                       *
C      ASSUMING THE RAY DOES NOT VERTEX IN THIS DEPTH RANGE                 *
C     *                                                                     **
C     ***********************************************************************
C                                                                            *
      COMMON
     C / RAYTRA /
     C            NCONCI,
     C            INITLK,
     C            ZSTART,
     C            Z2     ,
     C            SPVRSQ,
     C            ANGSTR,
     C            ANGARR,
     C            ANGBTM,
     C            ANGSUR,
     C            SPDVER,
     C            RANGET
      COMMON
     C / LCONST /
     C            NULAPO,
     C            DEPROT,
     D            CONSG0(128)      ,
     D            CONSG1(128)      ,
     D            CONSG2(128)      ,
     D            CONSV0(128)      ,
     D            DELTAZ(128)      ,
     D            DEPKYD(128)      ,
     D            SLOPEJ(128)      ,
     D            SPDKYD(128)
      COMMON
     C / PTHLNG /
     C            A1     ,
     C            BI     ,
     C            CDSQRD,
     C            CI     ,
     C            DI     ,
     C            DXDC   ,
     C            DZ1    ,
     C            DZM    ,
     C            K      ,
     C            PL     ,
     C            SM     ,
     C            V      ,
     C            X      ,
     C            Y1     ,
     C            Y2     ,
     C            TIMCON
C                                                                            *
C     A1           = DEPTH FROM  END  OF RAY PORTION TO TOP OF LAYER       *
C                    ALWAYS POSITIVE OR ZERO                                 *
C     ANGARR       = ANGLE (IN RADIANS) AT ARRIVAL POINT (Z2)              *
C     ANGBTM       = ANGLE (IN RADIANS) OF BOTTOM BOUNCE OF RAY            *
C     ANGSTR       = ANGLE (IN RADIANS) OF START OF RAY (Z1)               *
C     ANGSUR       = ANGLE (IN RADIANS) OF SURFACE BOUNCE OF RAY           *
C     AR     ( )   = ARRIVAL ANGLE (IN RADIANS)                            *
C     BB     ( )   = BOTTOM BOUNCE ANGLE (IN RADIANS)                      *
C     CONSG0( )    = G0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT *
C     CONSG1( )    = G1 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT *
C     CONSG2( )    = G2 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT *
C     CONSV0( )    = V0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT *
C     CV           = SEE -SPDVER-                                          *
C     DEPKYD( )    = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE    *
C     DZ1          = DEPTH FROM START OF RAY PORTION TO TOP OF LAYER       *
C                    ALWAYS POSITIVE OR ZERO                                 *
C     INITLK       = INITIAL VALUE OF K I.E. THE INITIAL STARTING LAYER    *
C     J            = CONTROL SUBSCRIPT                                     *
C     NCONCI       = CONTROL PARAMETER FOR DIFFERENT SUBPROGRAMS TO BE CAL *
C     NULAPO       = NUMBER OF LAYERS PLUS ONE                             *
C     RANGET( )    = SUMMATION OF THE TOTAL RANGE THROUGH THE DIFFERENT LA *
C     SF     ( )   = SURFACE ARRIVAL ANGLE (IN RADIANS)                    *
C     SPDKYD( )    = SPEED OF SOUND PROPAGATION (IN K-YD/SEC)              *
C     SPDVER       = VERTEX VELOCITY I.E. SPEED AT VERTEX                  *
C     T      ( )   = ANGLE (IN RADIANS)                                    *
C     Z1           = DEPTH OF START POINT OF RAY PORTION                   *
C     Z2           = DEPTH (IN K-YD) OF ENDING DEPTH                       *
C     Z8           = DEPTH OF  END  POINT OF RAY PORTION                   *
C     ZSTART       = STARTING DEPTH (IN K-YD) OF RAY                       *
C                                                                            *
      ATHIRD = 1.0/3.0
      Z1 = ZSTART
      T1 = ANGSTR
      K = INITLK
      RANGET = 0.0
      L = NORPNT( Z2, DEPKYD, NULAPO ) - 1
      IF( (Z1 .EQ. DEPKYD(K)) .AND. (T1 .LT. 0.0) )  K = K - 1
      CV = SPDVER
      SPVRSQ = 1.0/SPDVER/SPDVER
```

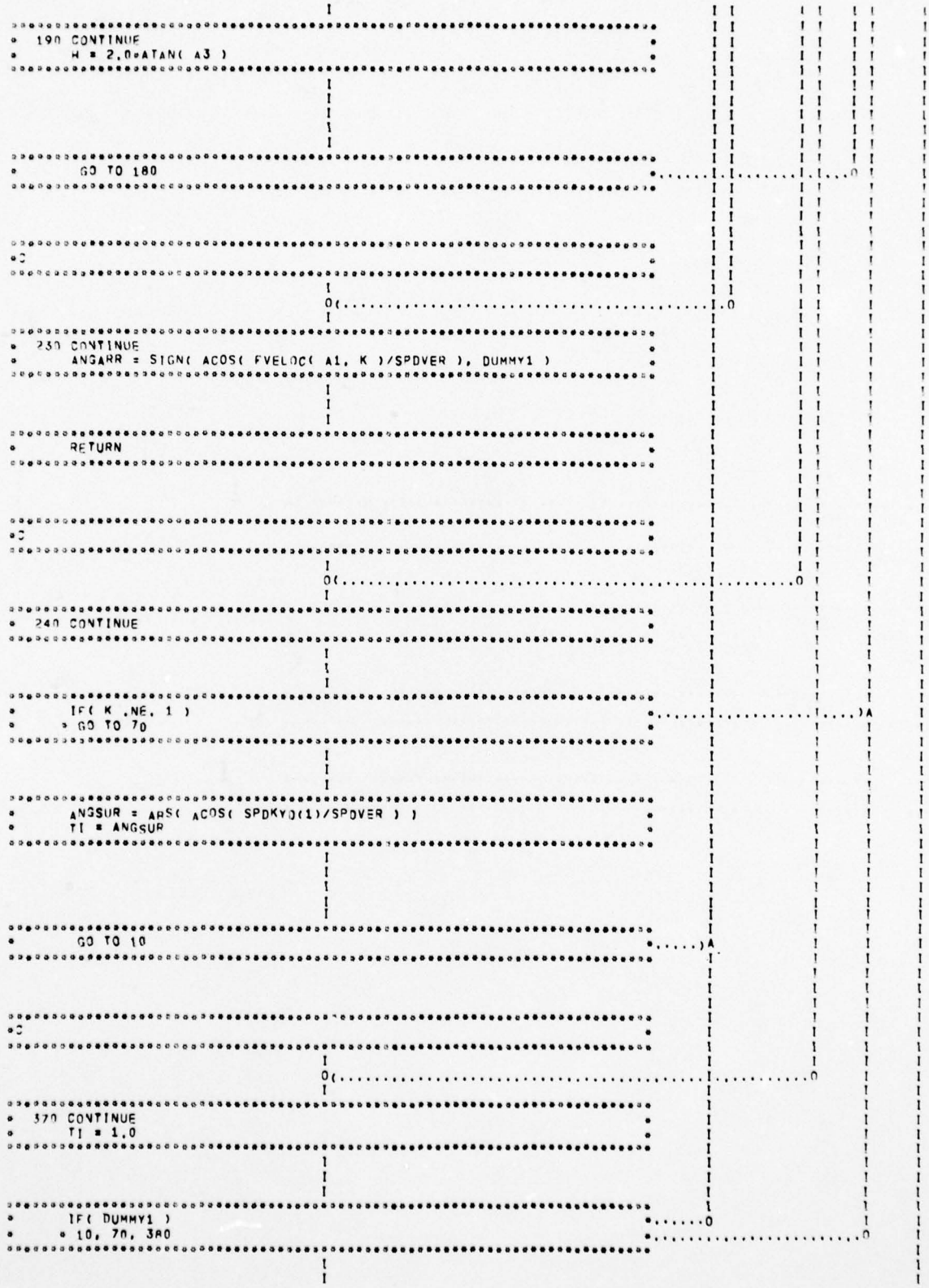

10 CONTINUE
IF((K .GT. NOLAPO) .OR. (K .LE. 0))CALL DUMP
DZ1 = Z1 - DEPKYD(K)
ZR = DEPKYD(K+1)
IF(TI .EQ. 0.0)
GO TO 80
IF(K .EQ. L)
GO TO 160
IF(TI .GT. 0.0)
GO TO 30
20 CONTINUE
ZR = DEPKYD(K)
30 CONTINUE
DUMMY1 = ZR - Z1
IF(TI .EQ. 0.0)
GO TO 370
A1 = CONSV0(K) - SPVRSQ
A6 = A1*CONSG2(K)
B1 = A6 + CONSG0(K)/2.0
C1 = A6*CONSG2(K) + CONSG1(K)
A1 = ZR - DEPKYD(K)
DZM = 0.5*(DZ1 + A1)
Y1 = A1 + 2.0*B1*DZ1 + C1*DZ1*DZ1
IF(Y1 .GE. 0.0)
GO TO 280
Y1 = 0.0
CALL TRACER(1)
280 CONTINUE
Y1 = SQRT(Y1)
Y2 = A1 + 2.0*B1*A1 + C1*A1*A1

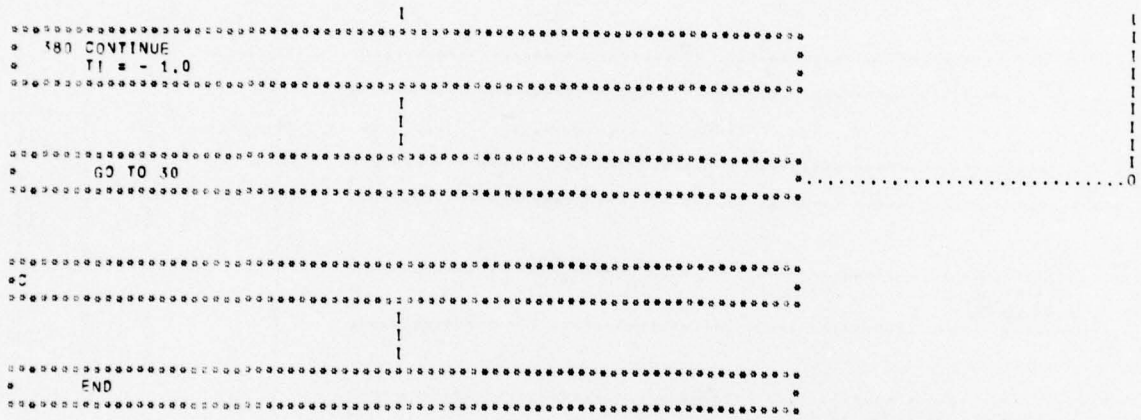

IF(Y2 .GE. 0.0)
GO TO 290
Y2 = 0.0
CALL TRACER(2)
290 CONTINUE
Y2 = SQRT(Y2)
D1 = DUMMY1/(Y1 + Y2)
A7 = D1*D1
CDSQRD = C1*A7
IF(ABS(CDSQRD) .GE. 0.5)
GO TO 170
SM = ATHIRD
POWRCD = SM
DO 200
I = 2,50
A5 = 2*I - 1
POWRCD = POWRCD*CDSQRD*A5/(A5 + 2.0)
SM = POWRCD + SM
IF(ABS(POWRCD) .LT. 1.0E-4)
GO TO 40
200 CONTINUE
40 CONTINUE
SM = SM*A7
50 CONTINUE
X = D1/SPOVER*(2.0 + CONSG2(K)*(DZ1 + A1) +
E (CONSG1(K)*2.0 - CONSG0(K)*CONSG2(K))*SM)
RANGET = ABS(X) + RANGET
IF(NCONCI .NE. 1) CALL
S RAYPTH
60 CONTINUE
IF(ZB .EQ. Z2)
GO TO 230


Z1 = ZR
IF(DUMMY1.LE. 0.0)
GO TO 240
K = K + 1
IF(K .NE. NULAPO)
GO TO 10
ANGBTM = -ABS(ACOS(SPDKYD(NULAPO)/SPDVER))
TI = ANGBTM
70 CONTINUE
K = K - 1
GO TO 10
80 CONTINUE
DVZ = DZ1*CONSG2(K)
Q1 = 1.0 + DVZ
Q2 = DZ1*CONSG1(K)
DVZ = ((CONSG0(K) + Q2)*(1.0 - 2.0*DVZ/Q1) + Q2)/Q1/Q1
IF(K .EQ. L)
GO TO 90
IF(DVZ)
30, 140, 20


90 CONTINUE
IF(DVZ)
• 110, 130, 150
110 CONTINUE
IF(Z2 .LE. Z1)
• GO TO 30
120 CONTINUE
ZB = Z2
GO TO 30
C
130 CONTINUE
IF(Z2 .EQ. Z1) RANGET = 1.0E6
140 CONTINUE
ANGARR = 0.0
ANGSUR = 0.0
ANGBTM = 0.0
RETURN
C
150 CONTINUE
IF(Z2 .GE. Z1)
• GO TO 20

```
      GO TO 120
C
  160 CONTINUE
      IF( RANGET .GT. 0.0 )
     * GO TO 120
      IF( T1 .LE. 0.0 )
     * GO TO 150
      IF( Z2 .GT. Z1 )
     * GO TO 120
      GO TO 30
C
  170 CONTINUE
      A2 = SQRT( ABS( CI ) )
      A3 = A2*DI
      IF( CI .LE. 0.0 )
     * GO TO 190
      H = ALOG( ABS( ( 1.0 + A3 )/( 1.0 - A3 ) ) )
  180 CONTINUE
      SM = ( H/A2*0.5/DI - 1.0 )/CI
      V = H/A2
      GO TO 50
C
```

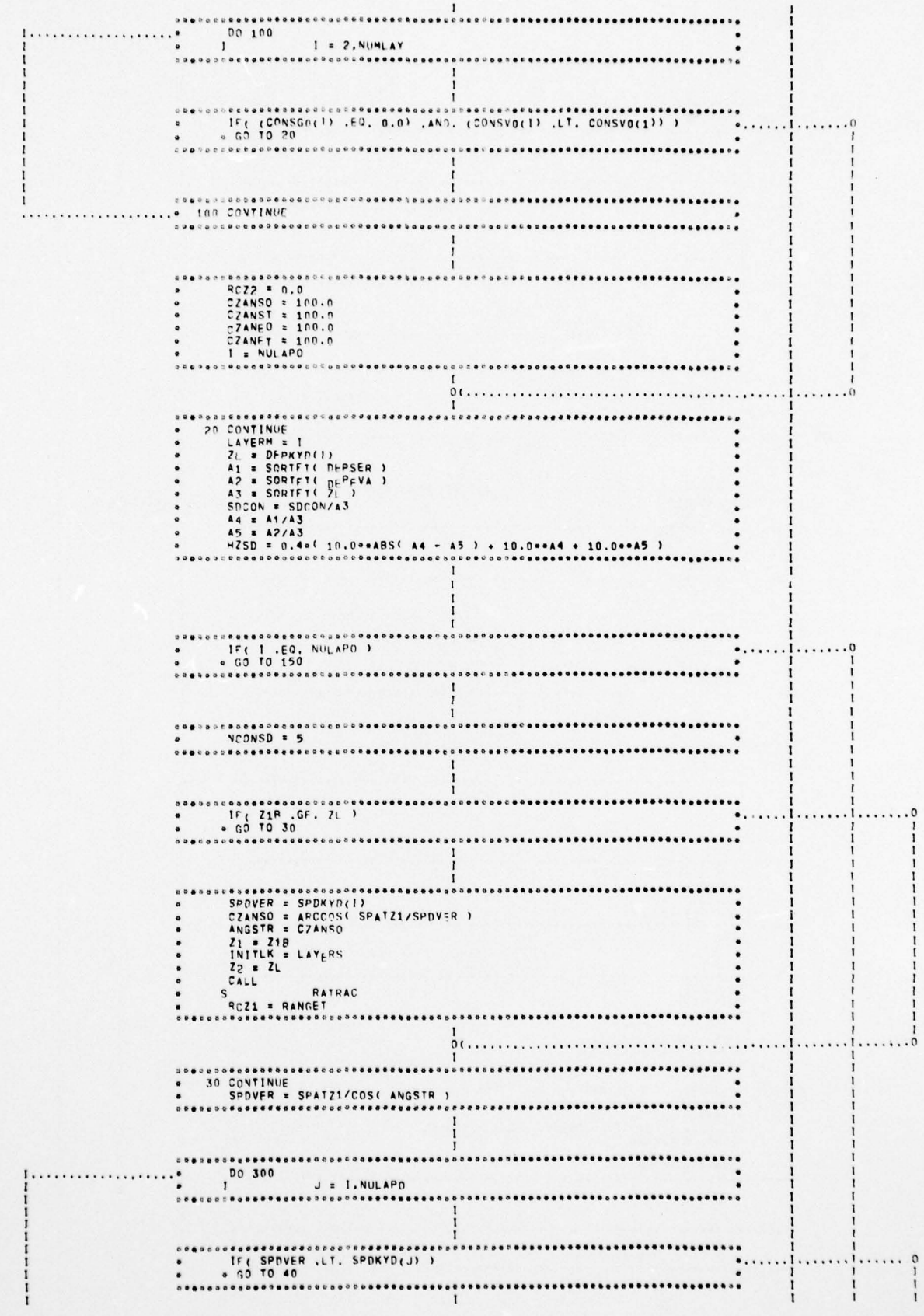

190 CONTINUE
H = 2.0*ATAN(A3)
GO TO 180
C
230 CONTINUE
ANGARR = SIGN(ACOS(FVELOC(A1, K)/SPDVER), DUMMY1)
RETURN
C
240 CONTINUE
IF(K .NE. 1)
GO TO 70
ANGSUR = ABS(ACOS(SPDKY()(1)/SPDVER))
TI = ANGSUR
GO TO 10
C
370 CONTINUE
TI = 1.0
IF(DUMMY1)
10, 70, 380

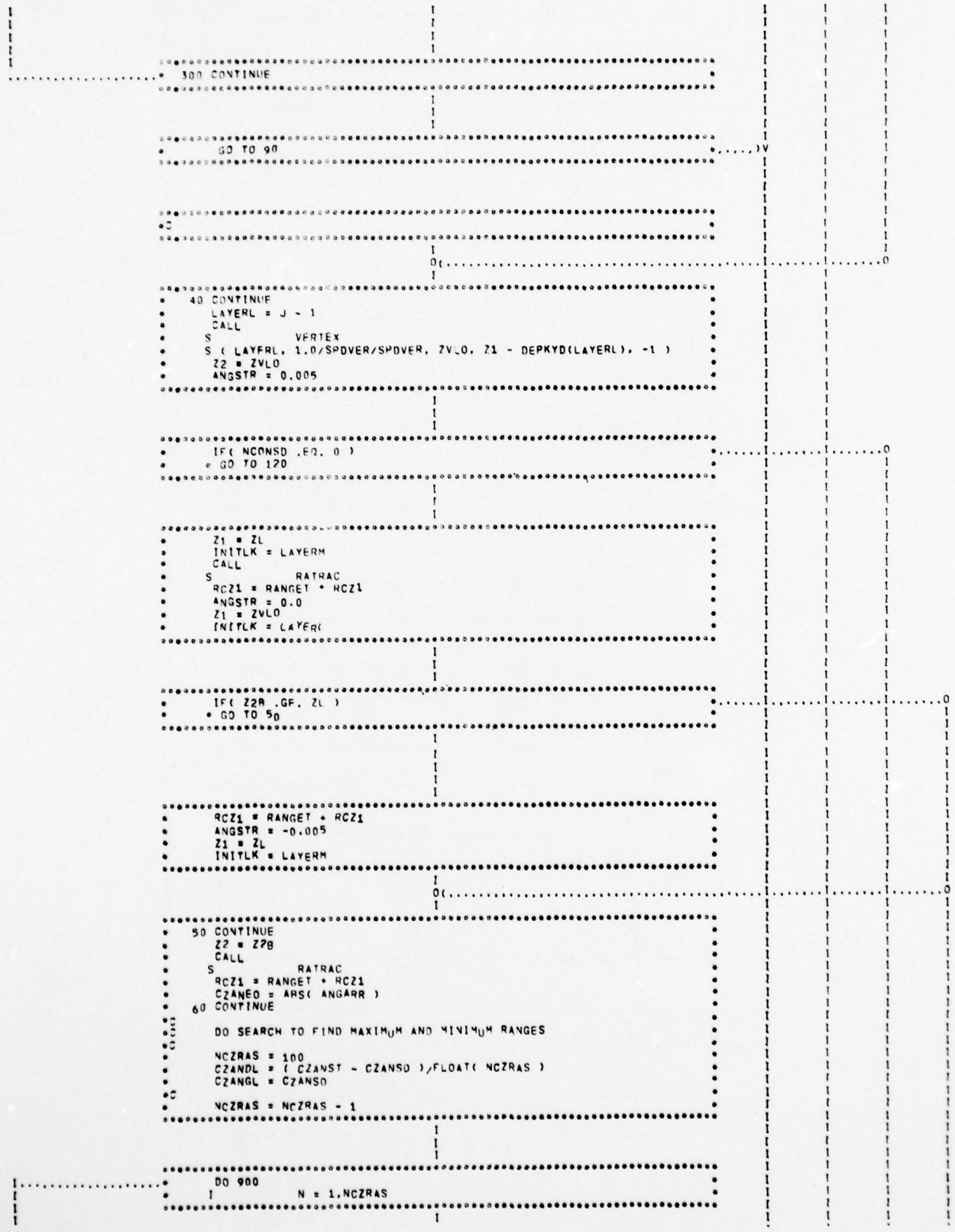

580 CONTINUE
TI = - 1.0
GO TO 30
C
END

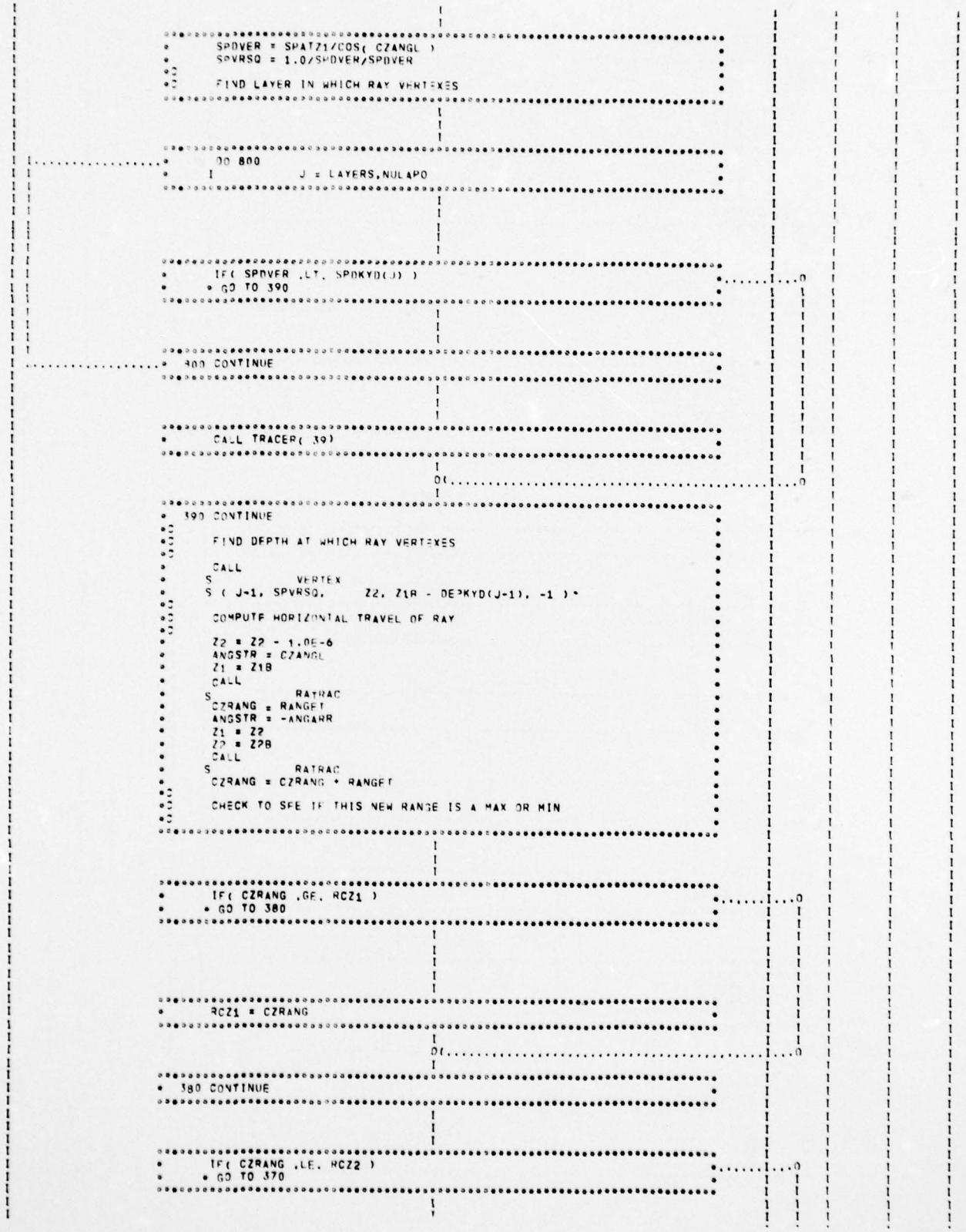
(ENTRANCE)

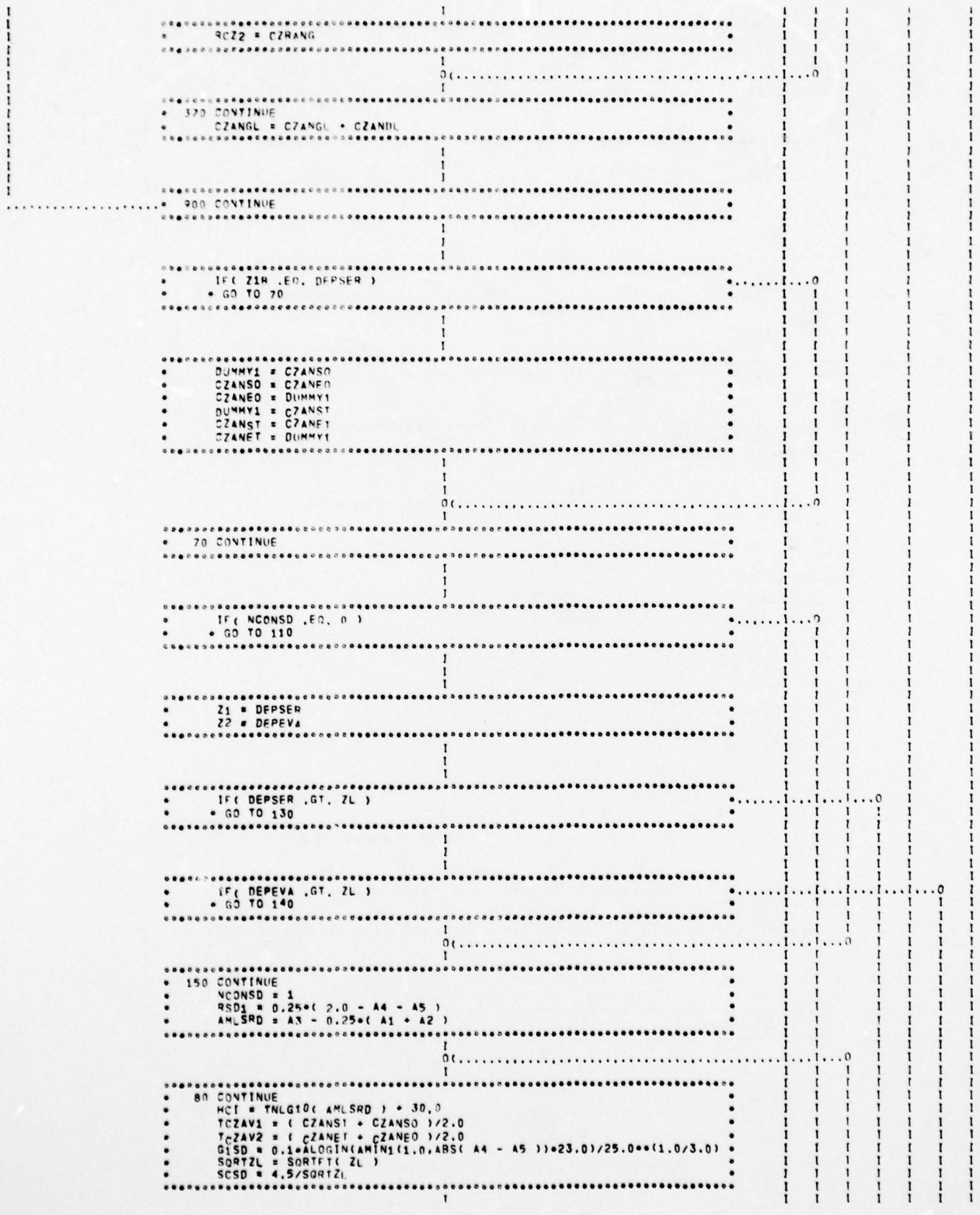
```
*CRAYCTL00        SUBROUTINE RAYCTL                                          *
*CSRAYC           SUBROUTINE FOR SETTING UP PROPER RAY TABLES                 *
*     SUBROUTINE                                                            *
*     S           RAYCTL                                                    *
*                                                                           *
*     ************************************************************          *
*     *                                                          *          *
*     THIS SETS UP THE REQUIRED RAY TRACE TABLES PLUS CONVERGENT ZONE VA*    *
*     *                                                          *          *
*     ************************************************************          *
*                                                                           *
*      COMMON                                                               *
*     C / CONSTN /                                                          *
*     C           DEPSER,                                                   *
*     C           NCONSK,                                                   *
*     C           SPATSE,                                                   *
*     C           DEPEVA,                                                   *
*     C           NCONSL,                                                   *
*     C           SPATEV                                                    *
*      COMMON                                                               *
*     C / LCONST /                                                          *
*     C           NULAPO,                                                   *
*     C           DEPBOT,                                                   *
*     D           CONSG0(128)      ,                                        *
*     D           CONSG1(128)      ,                                        *
*     D           CONSG2(128)      ,                                        *
*     D           CONSV0(128)      ,                                        *
*     D           DELTAZ(128)      ,                                        *
*     D           DEPKYD(128)      ,                                        *
*     D           SLOPEJ(128)      ,                                        *
*     D           SPDKYD(128)                                               *
*      COMMON                                                               *
*     C / RAYTRA /                                                          *
*     C           NCONCI,                                                   *
*     C           INITLK,                                                   *
*     C           Z1    ,                                                  *
*     C           Z2    ,                                                  *
*     C           SPVRSQ,                                                   *
*     C           ANGSTR,                                                   *
*     C           ANGARR,                                                   *
*     C           ANGBTM,                                                   *
*     C           ANGSUR,                                                   *
*     C           SPDVER,                                                   *
*     C           RANGET                                                    *
*      COMMON                                                               *
*     C / RCONST /                                                          *
*     C           ACZ   ,                                                   *
*     C           AMLSRD,                                                   *
*     C           BCZ   ,                                                   *
*     C           HCI   ,                                                   *
*     C           HZSD  ,                                                   *
*     C           NCONSD,                                                   *
*     C           RCZ1  ,                                                   *
*     C           RCZ2  ,                                                   *
*     C           SDCON ,                                                   *
*     C           TCZAV1,                                                   *
*     C           TCZAV2,                                                   *
*     C           ZW    ,                                                  *
*     NDUM6                                                                 *
*      COMMON                                                               *
*     C / SURFAC /                                                          *
*     C           A6    ,                                                  *
*     C           CONST2,                                                   *
*     C           CONST4,                                                   *
*     C           CZANGL,                                                   *
*     C           CZANDL,                                                   *
*     C           CZRANG,                                                   *
*     C           G1SD  ,                                                   *
*     C           G2SD  ,                                                   *
*     C           NCZRAS,                                                   *
*     C           NZONE ,                                                   *
*     C           RSD   ,                                                   *
*     C           RSD1  ,                                                   *
*     C           SCSD  ,                                                   *
*     C           SQRTZL,                                                   *
*     C           SS    ,                                                  *
*     C           ZL    ,                                                  *
*     CDUMMYZ                                                               *
*                                                                           *
*      AMLSRD     = CONSTANT REQQUIRED FOR AMOS CALCULATIONS                 *
*      ANGARR     = ANGLE (IN RADIANS) OF ARRIVAL AT Z2                      *
*      ANGBTM     = ANGLE (IN RADIANS) OF BOTTOM BOUNCE OF RAY               *
*      ANGSTR     = ANGLE (IN RADIANS) OF START OF RAY (Z1)                  *
*      ANGSUR     = ANGLE (IN RADIANS) OF SURFACE BOUNCE OF RAY              *
*      CONSG0( )  = G0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*   *
*      CONSG1( )  = G1 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*   *
*      CONSG2( )  = G2 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*   *
*      CONSV0( )  = V0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*   *
*      COSZ1B = COSINE OF STARTING ANGLE OF RAY                  (DIMENS*   *
*      CUB        =                                                          *
*      CUBCAL     =                                                          *
*      CZANEO     = SMALLER CONVERGENT ZONE ANGLE (IN RADIANS) FOR EVADER*   *
*      CZANET     = LARGER  CONVERGENT ZONE ANGLE (IN RADIANS) FOR EVADER*   *
*      CZANSO     = SMALLER CONVERGENT ZONE ANGLE (IN RADIANS) FOR SEARCH*   *
*      CZANST     = LARGER  CONVERGENT ZONE ANGLE (IN RADIANS) FOR SEARCH*   *
*      DEPKYD( )  = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE       *
*      DPTOZ1     = DEPTH (IN K-YD) FROM TOP OF LAYER TO Z1                  *
*      DZVX       = DEPTH BELOW LAYER (IN K-YD) OF POINT OF VERTEX           *
```

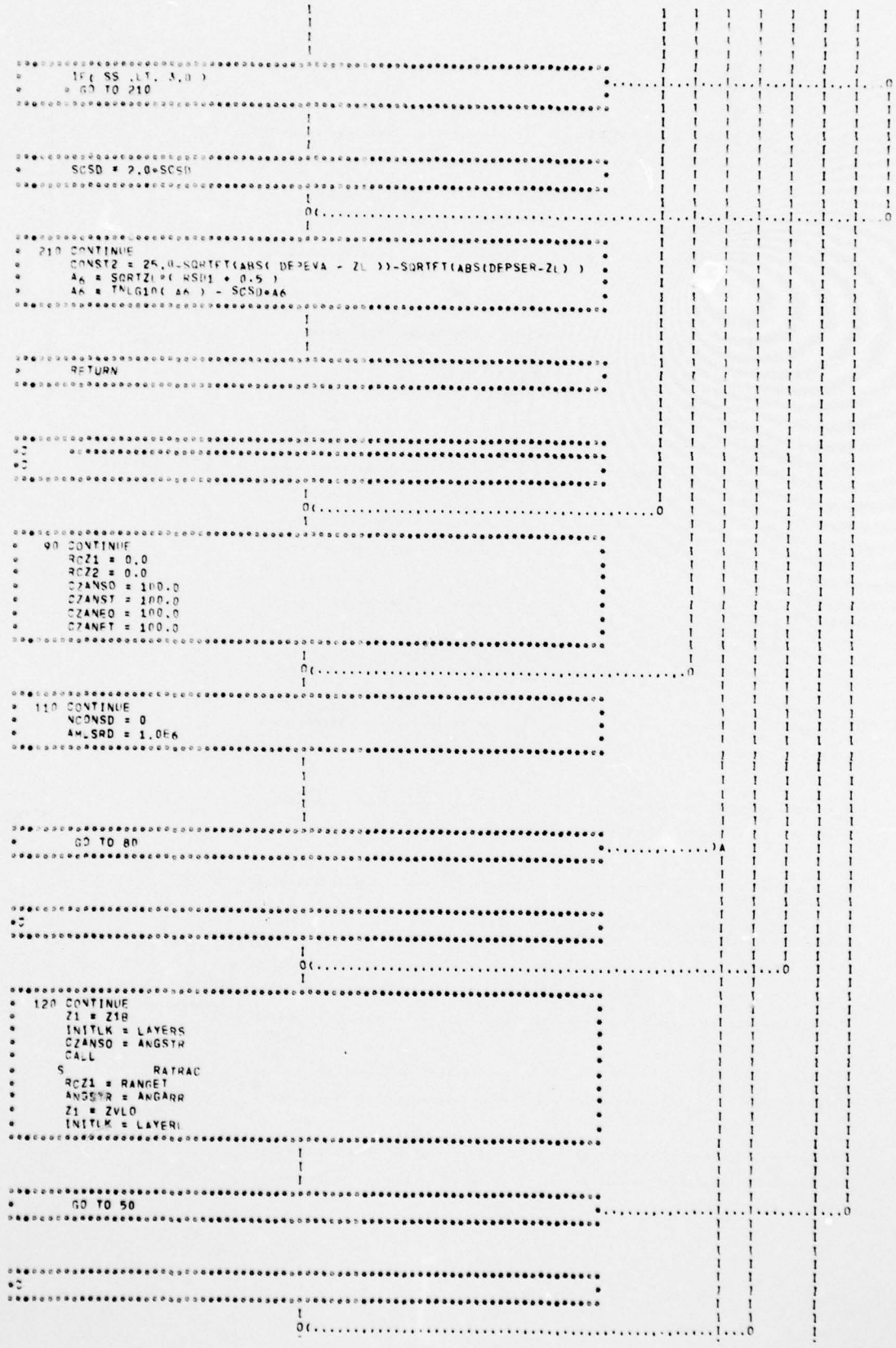
```
C      IVMAX    =                                                        C
C      KR       = LAYER WHICH CONTAINS Z1B                               C
C      LAYERL   = LAYER IN WHICH A LOWER VERTEX IS FOUND          (DIMENS C
C      LAYERM   = LAYER IN WHICH VELOCITY PROFILE HAS A ZERO      (DIMENS C
C      LAYERS   = LAYER NUMBER OF HIGHER DEPTH OF SHIP            (DIMENS C
C      NCONCI   = CONTROL PARAMETER FOR DIFFERENT SUBROUTINES TO BE CAL C
C      NCONSD   = AMOS CONSTANT SD                                       C
C      NCONZ1   = LAYER NUMBER WHICH CONTAINS Z1                         C
C      NULAPO   = NUMBER OF LAYERS PLUS ONE                              C
C      RANGET   = TOTAL RANGE (IN K-YD) AS COMPUTED BY RAY-TRACING SUBR C
C      RCZ1     =                                                        C
C      SDCON    = WAVE HEIGTH PARAMETER                                  C
C      SPATEV   = PROPAGATION SPEED AT EVADER                      (K C
C      SPATSE   = PROPAGATION SPEED AT SEARCHER                          C
C      SPATZ1   = PROPAGATION SPEED (IN K-YD/SEC) AT POINT STARTING POI C
C      SPDKYD( )= SPEED OF SOUND PROPAGATION (IN K-YD/SEC)               C
C      SQRTFT   = FUNCTION TO COMPUTE THE SQUARE ROOT OF K-YD IN FEET    C
C      TCZAV1   =           AVERAGE OF CONVERGENT ZONE ANGLES         (C
C      TCZAV2   =           AVERAGE OF CONVERGENT ZONE ANGLES         (C
C      TNLG10   = FUNCTION TO CONVERT DIMENSIONLESS QUANTITY TO DB       C
C      XSD      = SEE -AMLSRD-                                     (SQR C
C      Z1       = STARTING DEPTH (IN K-YD) OF RAY                        C
C      Z2       = ENDING DEPTH (IN K-YD) OF RAY                          C
C      Z2B      = ENDING DEPTH (IN K-YD) OF CONVERGENT ZONE RAY          C
C      ZL       = DEPTH OF MAX OR MIN POINT IN VELOCITY PROFILE          C
C      ZVLO     = DEPTH (IN K-YD) OF VERTEX POINT OF CONVERGENT RAY      C
C                                                                       C
      NAMELIST / SEARCH / CZANGL, CZRANG, Z1                            C
C                                                                       C
      SQRTFT( DUMMY1 ) = SQRT( 3000.0*DUMMY1 )                          C
C                                                                       C
      RCZ1 = 0.0                                                        C
      NCONCI = 1                                                        C
      SDCON = 4.5                                                       C
      IF( ZW .GT. 4.0 )  SDCON = 2.0*SDCON                              C
      NCONSK = MORPNT( DEPSER, DEPKYD, NULAPO ) - 1                     C
      SPATSE = FVELOC( DEPSER - DEPKYD(NCONSK), NCONSK )                C
      NCONSL = MORPNT( DEPEVA, DEPKYD, NULAPO ) - 1                     C
      SPATEV = FVELOC( DEPEVA - DEPKYD(NCONSL), NCONSL )                C
      Z2 = AMAX1( DEPSER, DEPEVA )                                      C
      Z1 = AMIN1( DEPSER, DEPEVA )                                      C
      SPATZ1 = SPATSE                                                   C
      LAYERS = NCONSK                                                   C
C                                                                       C
      IF( DEPEVA .GE. DEPSER )                                          C
     * GO TO  10                                                        C
C                                                                       C
      SPATZ1 = SPATEV                                                   C
      LAYERS = NCONSL                                                   C
C                                                                       C
   10 CONTINUE                                                          C
      COSZ1B = SPATZ1/SPDKYD(NULAPO)                                    C
C                                                                       C
      IF( COSZ1B .GE. 1.0 )                                             C
     * GO TO 90                                                        C
C                                                                       C
      Z1B = Z1                                                          C
      Z2B = Z2                                                          C
      SPDVER = SPDKYD(NULAPO)                                           C
      CZANST = ACOS( COSZ1B )                                           C
      ANGSTR = CZANST                                                   C
      INITLK = LAYERS                                                   C
      Z2 = DEPKYD(NULAPO)                                               C
      CALL                                                              C
     S             RATRAC                                             C
      RCZ2 = RANGET                                                     C
      ANGSTR = -ANGARR                                                  C
      Z1 = Z2                                                           C
      INITLK = NULAPO                                                   C
      Z2 = Z2B                                                          C
      CALL                                                              C
     S             RATRAC                                             C
      RCZ2 = RANGET + RCZ2                                              C
      CZANET = ABS( ANGARR )                                            C
      NUMLAY = NULAPO - 1                                               C
C                                                                       C
```


DO 100
I = 2,NUMLAY
IF((CONSGO(I) .EQ. 0.0) .AND. (CONSVO(I) .LT. CONSVO(1)))
GO TO 20
100 CONTINUE
RCZ2 = 0.0
CZANSO = 100.0
CZANST = 100.0
CZANEO = 100.0
CZANET = 100.0
I = NULAPO
20 CONTINUE
LAYERM = I
ZL = DEPKYD(I)
A1 = SQRTFT(DEPSER)
A2 = SQRTFT(DEPEVA)
A3 = SQRTFT(ZL)
SDCON = SDCON/A3
A4 = A1/A3
A5 = A2/A3
HZSD = 0.4*(10.0**ABS(A4 - A5) + 10.0**A4 + 10.0**A5)
IF(I .EQ. NULAPO)
GO TO 150
NCONSD = 5
IF(Z1R .GE. ZL)
GO TO 30
SPDVER = SPDKYD(I)
CZANSO = ARCCOS(SPATZ1/SPDVER)
ANGSTR = CZANSO
Z1 = Z1R
INITLK = LAYERS
Z2 = ZL
CALL
S RATRAC
RCZ1 = RANGET
30 CONTINUE
SPDVER = SPATZ1/COS(ANGSTR)
DO 300
J = I,NULAPO
IF(SPDVER .LT. SPDKYD(J))
GO TO 40


300 CONTINUE
GO TO 90
C
40 CONTINUE
LAYERL = J - 1
CALL
S VERTEX
S (LAYERL, 1.0/SPDVER/SPDVER, ZVLO, Z1 - DEPKYD(LAYERL), -1)
Z2 = ZVLO
ANGSTR = 0.005
IF(NCONSD .EQ. 0)
GO TO 120
Z1 = ZL
INITLK = LAYERM
CALL
S RATRAC
RCZ1 = RANGET + RCZ1
ANGSTR = 0.0
Z1 = ZVLO
INITLK = LAYERL
IF(Z2R .GE. ZL)
GO TO 50
RCZ1 = RANGET + RCZ1
ANGSTR = -0.005
Z1 = ZL
INITLK = LAYERM
50 CONTINUE
Z2 = Z2R
CALL
S RATRAC
RCZ1 = RANGET + RCZ1
CZANEO = ARS(ANGARR)
60 CONTINUE
C
C DO SEARCH TO FIND MAXIMUM AND MINIMUM RANGES
C
NCZRAS = 100
CZANDL = (CZANST - CZANSO)/FLOAT(NCZRAS)
CZANGL = CZANSO
C
NCZRAS = NCZRAS - 1
DO 900
N = 1,NCZRAS


SPDVER = SPATZ1/COS(CZANGL)
SPVRSQ = 1.0/SPDVER/SPDVER
FIND LAYER IN WHICH RAY VERTEXES
DO 800
J = LAYERS,NULAPO
IF(SPDVER .LT. SPDKYD(J))
GO TO 390
800 CONTINUE
CALL TRACER(39)
390 CONTINUE
FIND DEPTH AT WHICH RAY VERTEXES
CALL
VERTEX
(J-1, SPVRSQ, Z2, Z1B - DEPKYD(J-1), -1)
COMPUTE HORIZONTAL TRAVEL OF RAY
Z2 = Z2 - 1.0E-6
ANGSTR = CZANGL
Z1 = Z1B
CALL
RATRAC
CZRANG = RANGET
ANGSTR = -ANGARR
Z1 = Z2
Z2 = Z2B
CALL
RATRAC
CZRANG = CZRANG + RANGET
CHECK TO SEE IF THIS NEW RANGE IS A MAX OR MIN
IF(CZRANG .GE. RCZ1)
GO TO 380
RCZ1 = CZRANG
380 CONTINUE
IF(CZRANG .LE. RCZ2)
GO TO 370


RCZ2 = CZRANG
370 CONTINUE
CZANGL = CZANGL + CZANDL
900 CONTINUE
IF(Z1R .EQ. DEPSER)
GO TO 70
DUMMY1 = CZANSO
CZANSO = CZANEO
CZANEO = DUMMY1
DUMMY1 = CZANST
CZANST = CZANET
CZANET = DUMMY1
70 CONTINUE
IF(NCONSD .EQ. 0)
GO TO 110
Z1 = DEPSER
Z2 = DEPEVA
IF(DEPSER .GT. ZL)
GO TO 130
IF(DEPEVA .GT. ZL)
GO TO 140
150 CONTINUE
NCONSD = 1
RSD1 = 0.25*(2.0 - A4 - A5)
AMLSRD = A3 - 0.25*(A1 + A2)
80 CONTINUE
HCI = TNLG10(AMLSRD) + 30.0
TCZAV1 = (CZANST + CZANSO)/2.0
TCZAV2 = (CZANET + CZANEO)/2.0
GISD = 0.1*ALOGIN(AMIN1(1.0,ABS(A4 - A5))*23.0)/25.0**(1.0/3.0)
SQRTZL = SQRTFT(ZL)
SCSD = 4.5/SQRTZL


IF(SS .LT. 3.0)
GO TO 210
SCSD = 2.0*SCSD
210 CONTINUE
CONST2 = 25.0-SQRTFT(ABS(DEPEVA - ZL))-SQRTFT(ABS(DEPSER-ZL))
A6 = SQRTZ(R(RSD1 + 0.5)
A6 = TNLG10(A6) - SCSD*A6
RETURN
90 CONTINUE
RCZ1 = 0.0
RCZ2 = 0.0
CZANSO = 100.0
CZANST = 100.0
CZANEO = 100.0
CZANET = 100.0
110 CONTINUE
NCONSD = 0
AM_SRD = 1.0E6
GO TO 80
120 CONTINUE
Z1 = Z1B
INITLK = LAYERS
CZANSO = ANGSTR
CALL
S RATRAC
RCZ1 = RANGET
ANGSTR = ANGARR
Z1 = ZVLO
INITLK = LAYERL
GO TO 50


130 CONTINUE
IF(DEPEVA .GT. ZL)
GO TO 180
NCONSD = 3
RSD1 = 0.25*(1.0 - A5) + 0.2*SQRT(A4*A4 - 1.0)
AM_SRD = 0.75*A3 - 0.5*A2 + 0.2*SQRTFT(DEPSER - ZL)
GO TO 80
C
140 CONTINUE
NCONSD = 2
AM_SRD = 0.75*A3 - 0.5*A1 + 0.2*SQRTFT(DEPEVA - ZL)
RSD1 = 0.25*(1.0 - A4) + 0.2*SQRT(A5*A5 - 1.0)
GO TO 80
C
180 CONTINUE
NCONSD = 4
RSD1 = 0.2*(SQRT(A4*A4 - 1.0) + SQRT(A5*A5 - 1.0))
GO TO 80
C
END

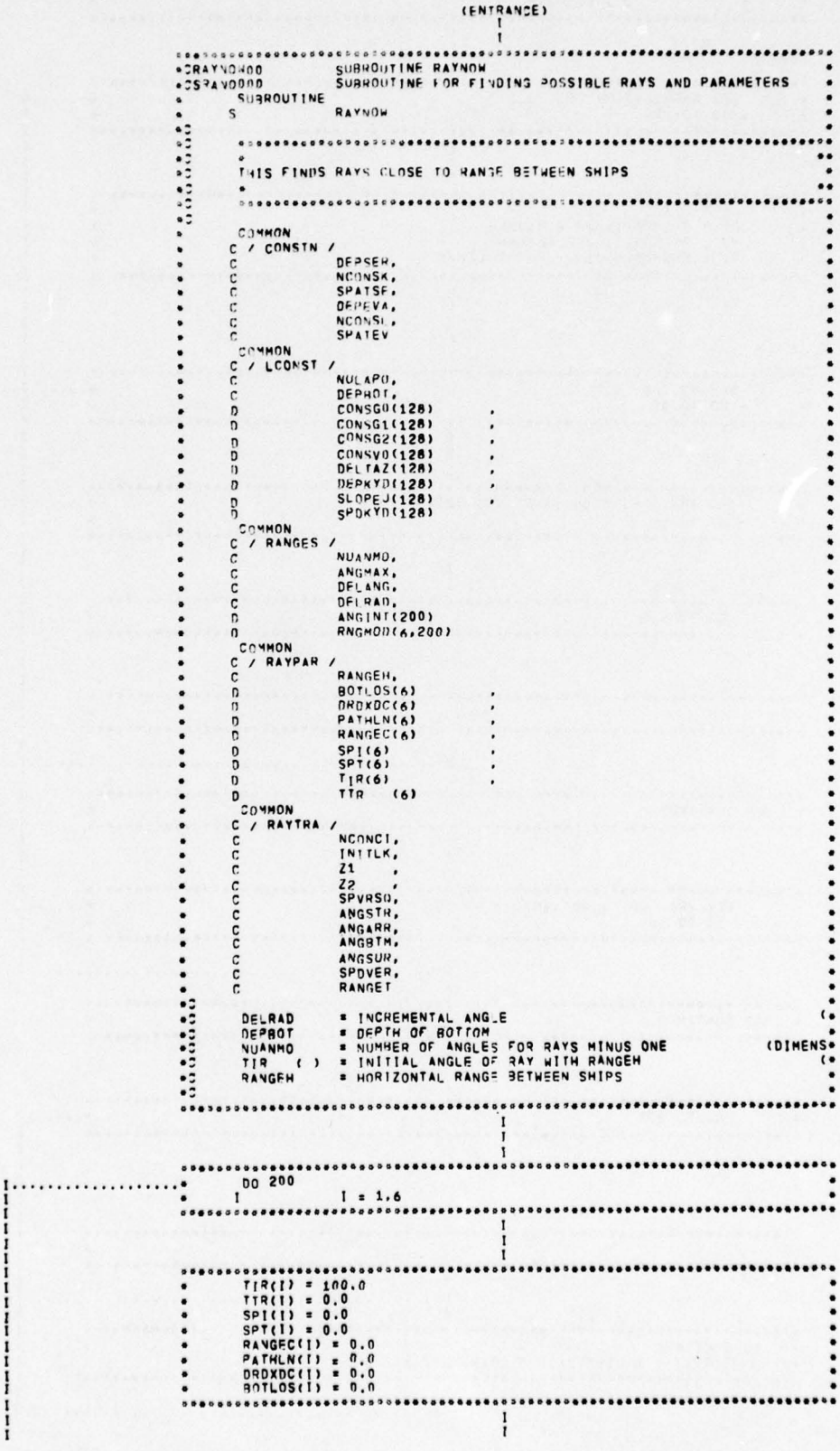
AD-A068 881 GENERAL ELECTRIC CO SYRACUSE N Y HEAVY MILITARY ELEC--ETC F/G 17/1
SONAR SIMULATION COMPUTER PROGRAMS. VOLUME 2. FLOW CHARTS.(U)
NOV 68 N00140-68-C-0372

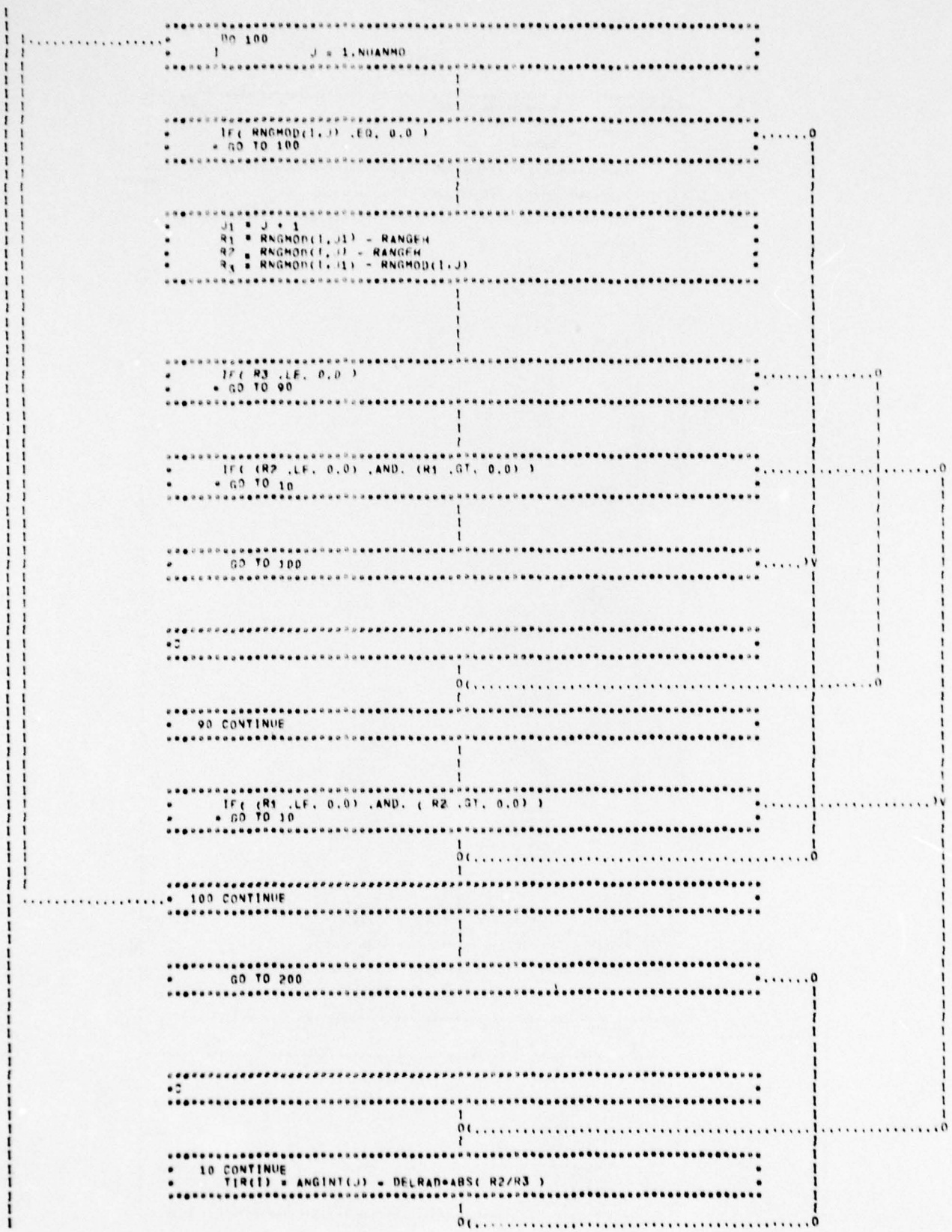
UNCLASSIFIED NL

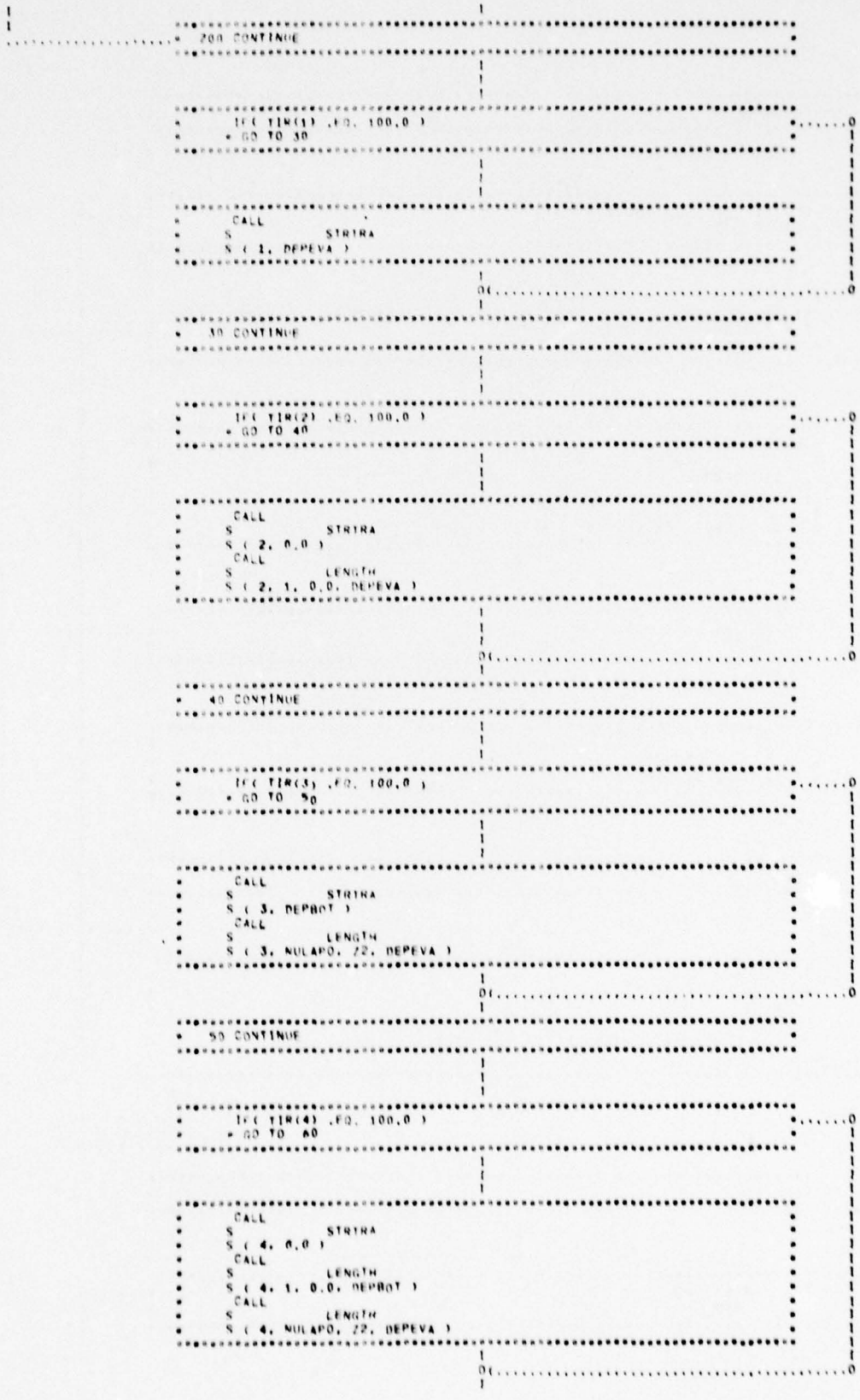
2 OF 2
AD
A068881

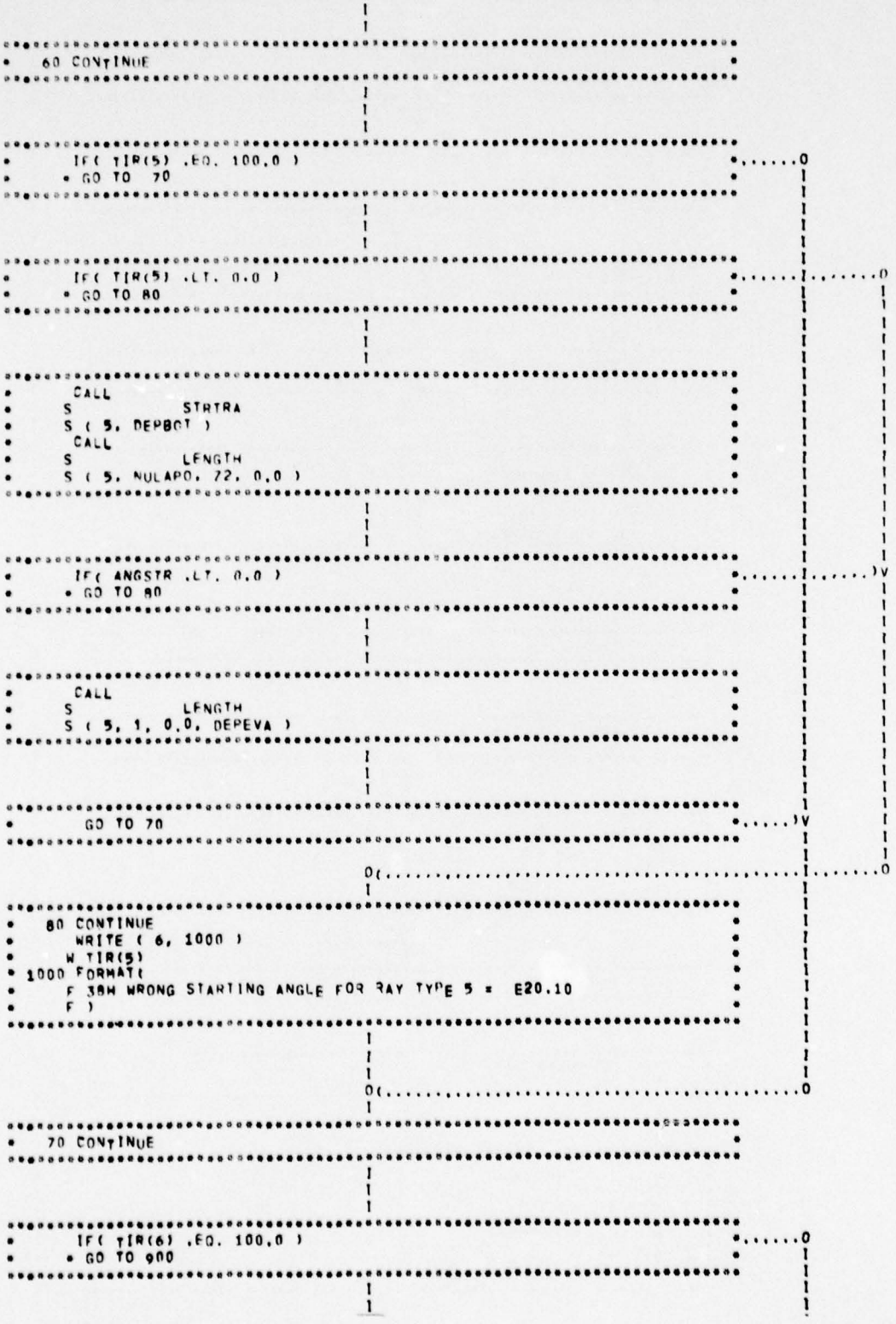


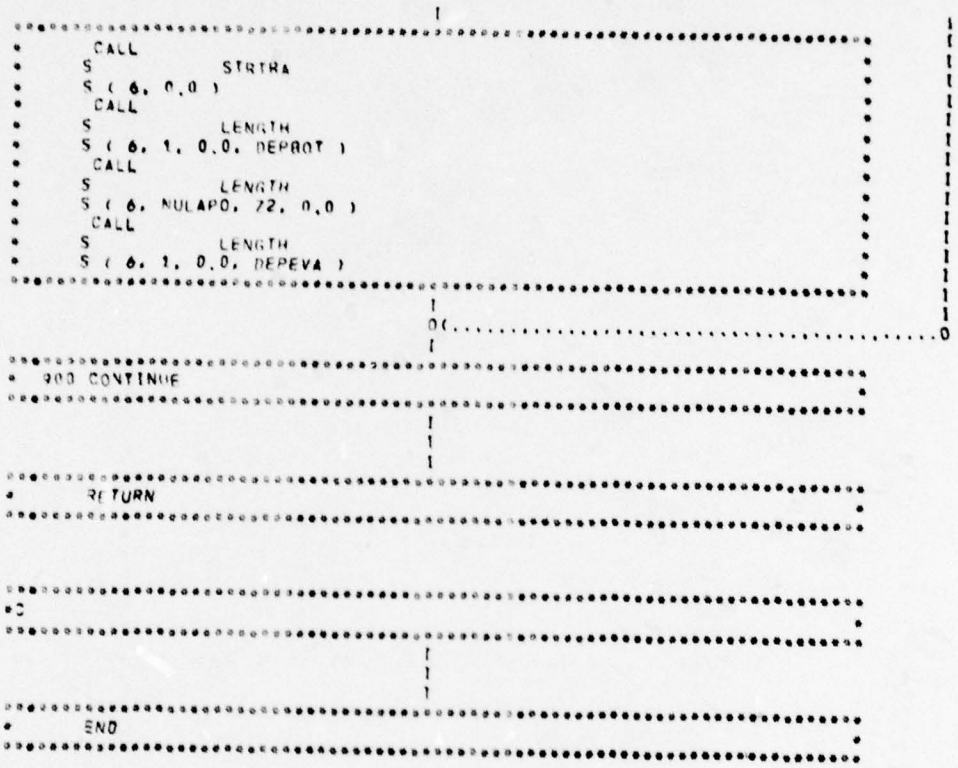
END
DATE
FILMED
7-79
DDC


(ENTRANCE)
C RAYNOW00 SUBROUTINE RAYNOW
C SRAYD000 SUBROUTINE FOR FINDING POSSIBLE RAYS AND PARAMETERS
SUBROUTINE
S RAYNOW
THIS FINDS RAYS CLOSE TO RANGE BETWEEN SHIPS
COMMON
C / CONSTN /
C DEPSER,
C NCONSK,
C SPATSF,
C DEPEVA,
C NCONSL,
C SPATEV
COMMON
C / LCONST /
C NOLAPO,
C DEPBOT,
D CONSG0(128) ,
D CONSG1(128) ,
D CONSG2(128) ,
D CONSV0(128) ,
D DELTAZ(128) ,
D DEPKYD(128) ,
D SLOPEJ(128) ,
D SPDKYD(128)
COMMON
C / RANGES /
C NUANMO,
C ANGMAX,
C DELANG,
C DELRAD,
D ANGINT(200)
D RNGMOD(6,200)
COMMON
C / RAYPAR /
C RANGEH,
D BOTLOS(6) ,
D DRDXDC(6) ,
D PATHLN(6) ,
D RANGEC(6) ,
D SPI(6) ,
D SPT(6) ,
D TIR(6) ,
D TTR (6)
COMMON
C / RAYTRA /
C NCONCI,
C INITLK,
C Z1 ,
C Z2 ,
C SPVRSO,
C ANGSTR,
C ANGARR,
C ANGBTM,
C ANGSUR,
C SPDVER,
C RANGET
DELRAD = INCREMENTAL ANGLE
DEPBOT = DEPTH OF BOTTOM
NUANMO = NUMBER OF ANGLES FOR RAYS MINUS ONE (DIMENS
TIR ( ) = INITIAL ANGLE OF RAY WITH RANGEH
RANGEH = HORIZONTAL RANGE BETWEEN SHIPS
DO 200
I I = 1,6
TIR(I) = 100.0
TTR(I) = 0.0
SPI(I) = 0.0
SPT(I) = 0.0
RANGEC(I) = 0.0
PATHLN(I) = 0.0
DRDXDC(I) = 0.0
BOTLOS(I) = 0.0


DO 100
J = 1,NUANMO
IF(RNGMOD(I,J) .EQ. 0.0)
GO TO 100
J1 = J + 1
R1 = RNGMOD(I,J1) - RANGEH
R2 = RNGMOD(I,J) - RANGEH
R3 = RNGMOD(I,J1) - RNGMOD(I,J)
IF(R3 .LE. 0.0)
GO TO 90
IF((R2 .LE. 0.0) .AND. (R1 .GT. 0.0))
GO TO 10
GO TO 100
C
90 CONTINUE
IF((R1 .LE. 0.0) .AND. (R2 .GT. 0.0))
GO TO 10
100 CONTINUE
GO TO 200
C
10 CONTINUE
TIR(I) = ANGINT(J) + DELRAD*ABS(R2/R3)


20 CONTINUE
IF(TIM(1) .EQ. 100.0)
GO TO 30
CALL
S STRIRA
S (1, DEPEVA)
30 CONTINUE
IF(TIM(2) .EQ. 100.0)
GO TO 40
CALL
S STRIRA
S (2, 0.0)
CALL
S LENGTH
S (2, 1, 0.0, DEPEVA)
40 CONTINUE
IF(TIM(3) .EQ. 100.0)
GO TO 50
CALL
S STRIRA
S (3, DEPBOT)
CALL
S LENGTH
S (3, NULAPO, 22, DEPEVA)
50 CONTINUE
IF(TIM(4) .EQ. 100.0)
GO TO 60
CALL
S STRIRA
S (4, 0.0)
CALL
S LENGTH
S (4, 1, 0.0, DEPBOT)
CALL
S LENGTH
S (4, NULAPO, 22, DEPEVA)


60 CONTINUE
IF(TIR(5) .EQ. 100.0)
GO TO 70
IF(TIR(5) .LT. 0.0)
GO TO 80
CALL
S STRTRA
S (5, DEPBOT)
CALL
S LENGTH
S (5, NULAPO, 72, 0.0)
IF(ANGSTR .LT. 0.0)
GO TO 80
CALL
S LENGTH
S (5, 1, 0.0, DEPEVA)
GO TO 70
80 CONTINUE
WRITE (6, 1000)
W TIR(5)
1000 FORMAT(
F 38H WRONG STARTING ANGLE FOR RAY TYPE 5 = E20.10
F)
70 CONTINUE
IF(TIR(6) .EQ. 100.0)
GO TO 900


CALL
S STRTRA
S (6, 0.0)
CALL
S LENGTH
S (6, 1, 0.0, DEPBOT)
CALL
S LENGTH
S (6, NULAPO, 22, 0.0)
CALL
S LENGTH
S (6, 1, 0.0, DEPEVA)
900 CONTINUE
RETURN
END

(ENTRANCE)

```
*RAYPTH00         SUBROUTINE RAYPTH
*SPATH000         SUBROUTINE CONTROLLED BY NCONCI
      SUBROUTINE
     S            RAYPTH

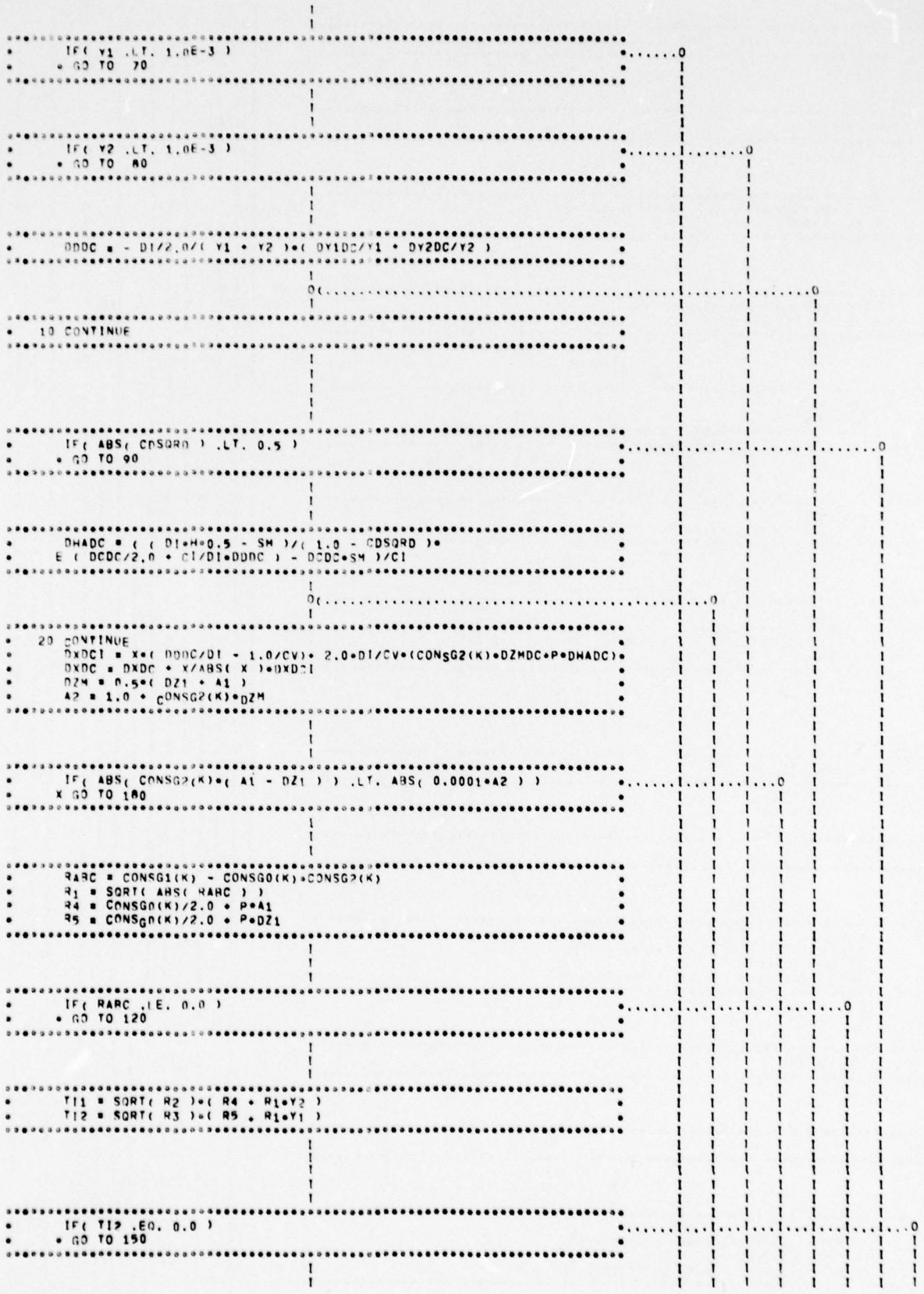
C
C     THIS COMPUTES REQUIRED RAY PARAMETERS SUCH AS TIME AND PATH LENGTH
C

      COMMON
     C / LCONST /
     C            NULAPO,
     C            DEPROT,
     D            CONSG0(128)      ,
     D            CONSG1(128)      ,
     D            CONSG2(128)      ,
     D            CONSYD(128)      ,
     D            DELTAZ(128)      ,
     D            DEPKYD(128)      ,
     D            SLOPEJ(128)      ,
     D            SPDKYD(128)
      COMMON
     C / PTHLNG /
     C            A1     ,
     C            B1     ,
     C            COSGRD,
     C            C1     ,
     C            D1     ,
     C            DXDC   ,
     C            DZ1    ,
     C            DZM    ,
     C            K      ,
     C            PL     ,
     C            SM     ,
     C            H      ,
     C            X      ,
     C            Y1     ,
     C            Y2     ,
     C            TIMCON
      COMMON
     C / RAYTRA /
     C            NCONCI,
     C            INITLK,
     C            Z1     ,
     C            Z2     ,
     C            SPVRSQ,
     C            ANGSTR,
     C            ANGARR,
     C            ANGRTM,
     C            ANGSUR,
     C            CV     ,
     C            RANGET

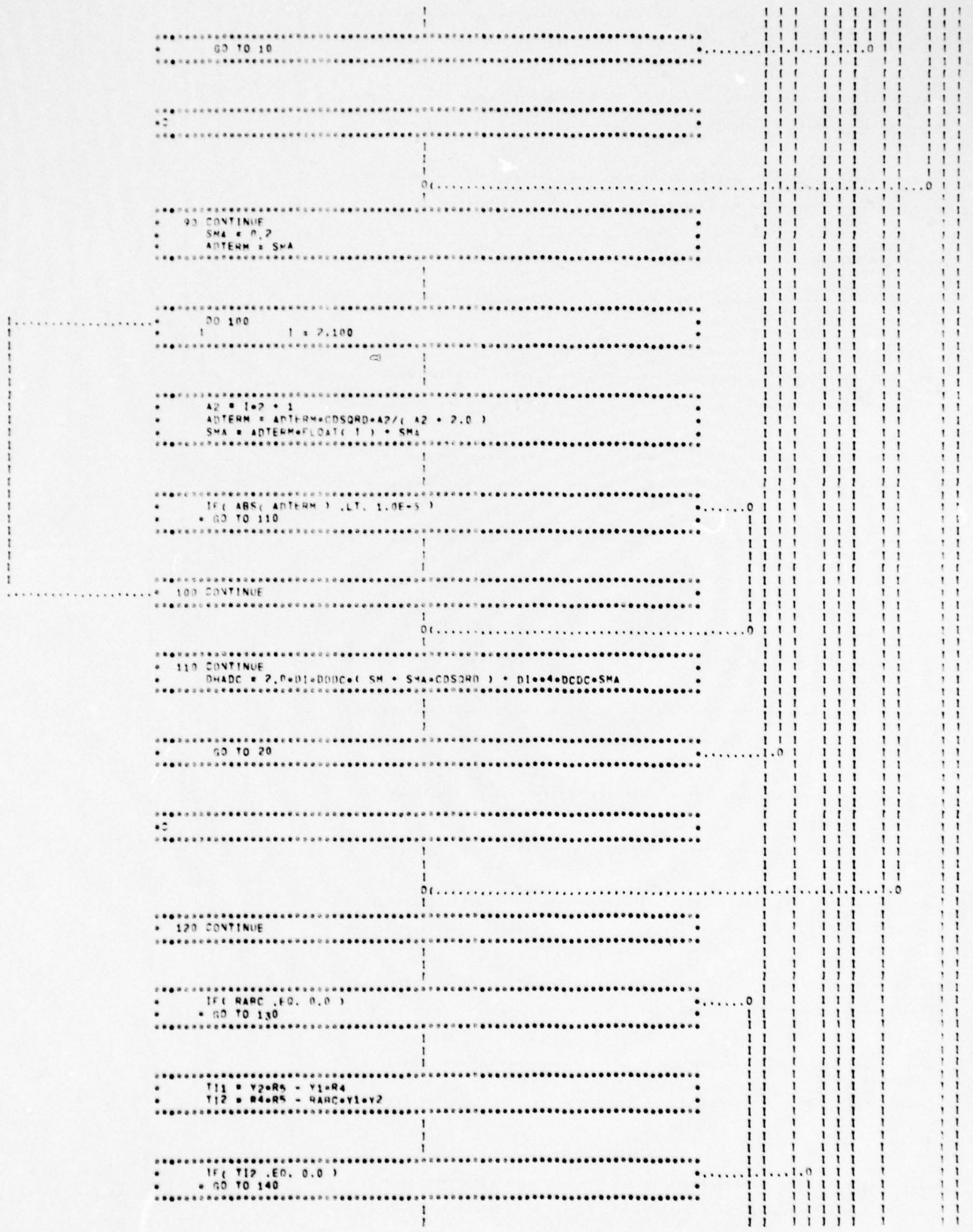
      A1           = DEPTH FROM  END  OF RAY PORTION TO TOP OF LAYER
                   ALWAYS POSITIVE OR ZERO
      DXDC         = RANGE DERIVATIVE
      DZ1          = DEPTH FROM START OF RAY PORTION TO TOP OF LAYER
                   ALWAYS POSITIVE OR ZERO
      TIMCON       = PROPAGATION TIME
      PL           = PATH LENGTH (IN K-YD) OF RAY
      CV           = VERTEX VELOCITY (IN K-YD/SEC)
      Y1           =
      Y2           =
      TI           = TIME INCREMENT FOR RAY TO GO X K-YD

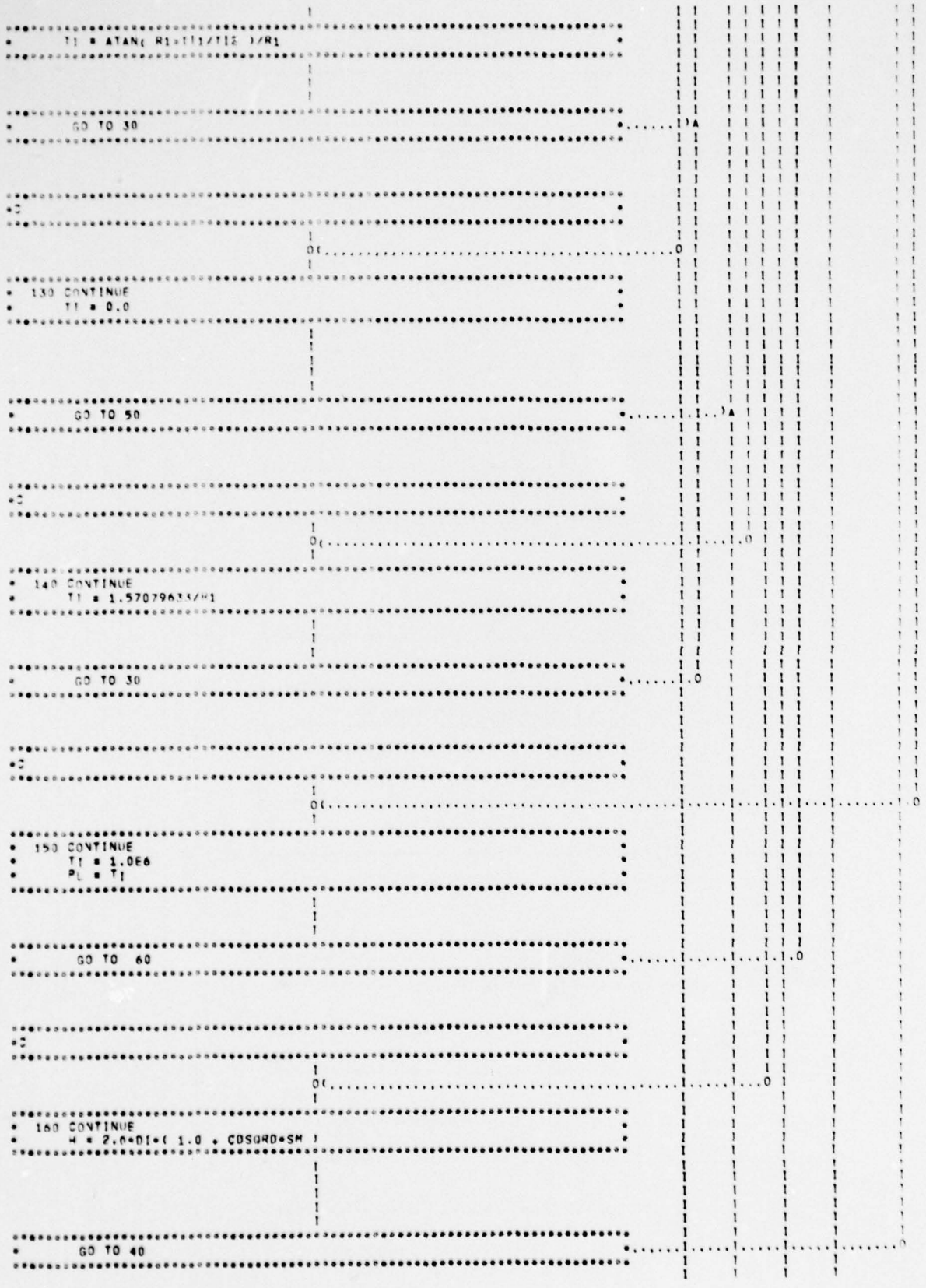
      NAMELIST
     N / TIMES
     N TOMUCH,
     NA1,A2,A3,ADTERM, A1, B1, COSGRD, C1, DADC, DCDC, DDDC, DHADC, D1,
     N DXDC, DXDC1, DY1DC, DY2DC, DZ1, DZ1DC, DZ2DC, DZMDC, H, K, P, PL,
     N Q1, R1, R2, R3, R4, R5, RADC, SM, SMA, SPDVER, SPVRSQ, TI, TI1,
     N TI2, TIMCON, TJ, TOMUCH, X, X1, X2, Y1, Y2

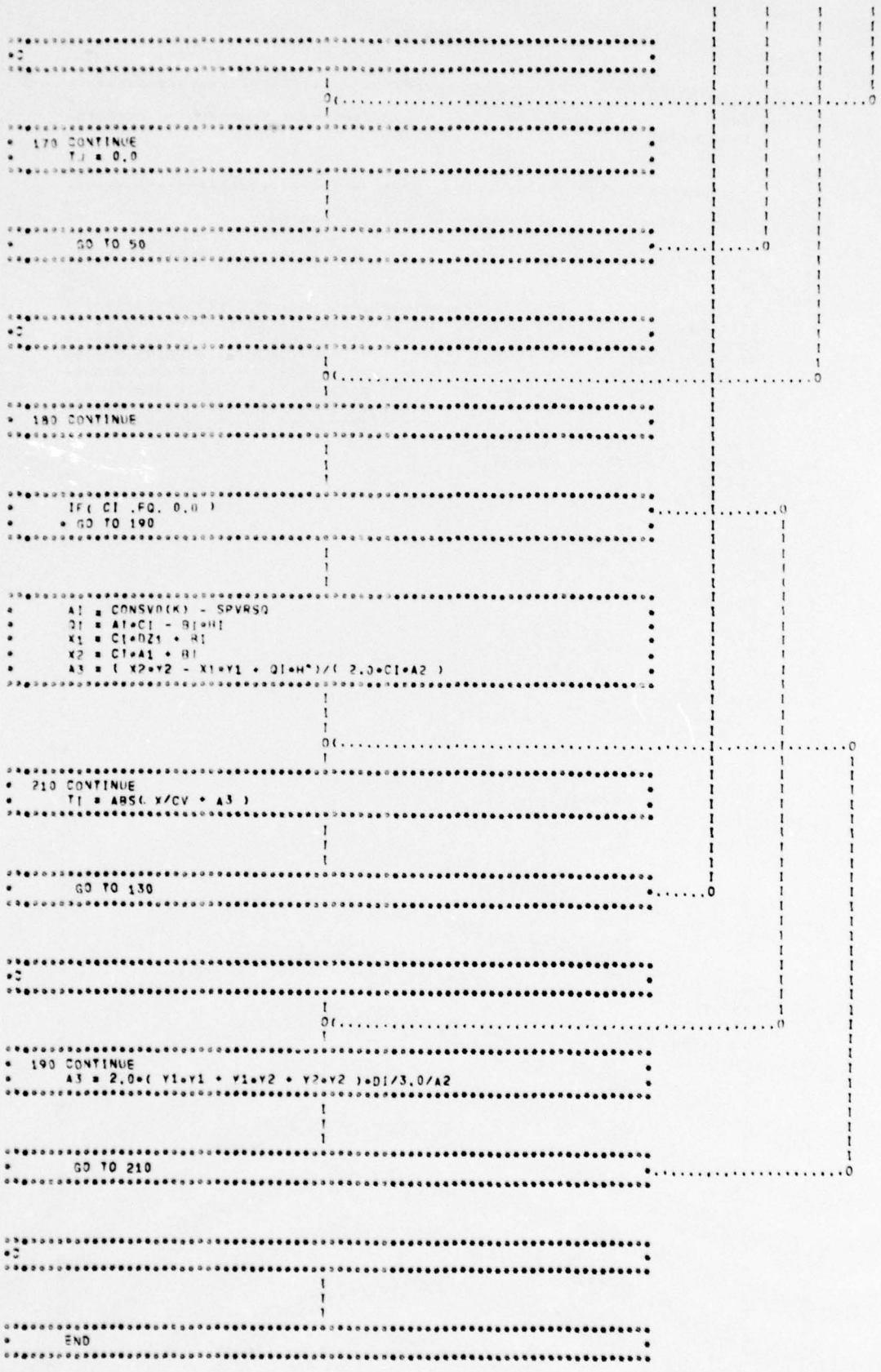
      IF( NCONCI .NE. 2 )
     * RETURN
      P = CONSG1(K) - 0.5*CONSG0(K)*CONSG2(K)
      DADC = 2.0/CV*SPVRSQ
      DCDC = CONSG2(K)**2*DADC
      R2 = (1.0 + CONSG2(K)*DZ1)**2
      DY1DC = R2*DADC
      R3 = ( 1.0 + CONSG2(K)*A1 )**2
      DY2DC = R3*DADC
      DZMDC = 0.0
```


IF(Y1 .LT. 1.0E-3)
GO TO 70
IF(Y2 .LT. 1.0E-3)
GO TO 80
DDDC = - DI/2.0/(Y1 + Y2)*(DY1DC/Y1 + DY2DC/Y2)
10 CONTINUE
IF(ABS(CDSQRD) .LT. 0.5)
GO TO 90
DHADC = ((DI*H*0.5 - SM)/(1.0 - CDSQRD)*
E (DCDC/2.0 + CI/DI*DDDC) - DCDC*SM)/CI
20 CONTINUE
DXDCI = X*(DDDC/DI - 1.0/CV)+ 2.0*DI/CV*(CONSG2(K)*DZMDC+P*DHADC)
DXDC = DXDC + X/ABS(X)*DXDCI
DZM = 0.5*(DZ1 + A1)
A2 = 1.0 + CONSG2(K)*DZM
IF(ABS(CONSG2(K)*(A1 - DZ1)) .LT. ABS(0.0001*A2))
X GO TO 180
RARC = CONSG1(K) - CONSG0(K)*CONSG2(K)
R1 = SQRT(ABS(RARC))
R4 = CONSG0(K)/2.0 + P*A1
R5 = CONSG0(K)/2.0 + P*DZ1
IF(RARC .LE. 0.0)
GO TO 120
TI1 = SQRT(R2)*(R4 + R1*Y2)
TI2 = SQRT(R3)*(R5 + R1*Y1)
IF(TI2 .EQ. 0.0)
GO TO 150

```
      TI = ALOG( TI1/TI2 )/R1
      GO TO 30
   30 CONTINUE
      IF( ABS( CONSQRD ) .LT. 0.5 )
     * GO TO 160
      IF( CONSG2(K) .EQ. 0.0 )
     * GO TO 170
   40 CONTINUE
      TOMUCH = TI
      TJ = ( CONSG1(K)*CV*X - 2.0*P*H + RARC*TI )/CONSG2(K)**2
      GO TO 50
   50 CONTINUE
      TI = ABS( CONSV0(K)*CV*X + TJ )
      GO TO 60
   60 CONTINUE
      TIMCON = TI + TIMCON
      PL = ( FVELDC( D/1, K ) + FVELDC( A1, K ) )/2.0*TI + PL
  900 CONTINUE
      RETURN
C
      GO TO 70
   70 CONTINUE
      DZ1DC = -0.5*R2/( C1*DZ1 + B1 )*DADC
      DZMDC = DZ1DC/2.0
      DDDC = DZ1DC/Y2 - D1/Y2/Y2*DY2DC/2.0
      GO TO 10
C
      GO TO 80
   80 CONTINUE
      DZ2DC = -0.5*R3/( C1*A1 + B1 )*DADC
      DZMDC = DZ2DC/2.0
      DDDC = DZ2DC/Y1 - 0.5/Y1*D1/Y1*DY1DC
```


GO TO 10
C
93 CONTINUE
SMA = 0.2
ADTERM = SMA
DO 100
I = 2,100
A2 = I*2 + 1
ADTERM = ADTERM*CDSQRD*A2/(A2 + 2.0)
SMA = ADTERM+FLOAT(I) + SMA
IF(ABS(ADTERM) .LT. 1.0E-5)
GO TO 110
100 CONTINUE
110 CONTINUE
DHADC = 2.0*D1*DDDC*(SM + SMA*CDSQRD) + D1**4*DCDC*SMA
GO TO 20
C
120 CONTINUE
IF(RARC .EQ. 0.0)
GO TO 130
T11 = Y2*R5 - Y1*R4
T12 = R4*R5 - RARC*Y1*Y2
IF(T12 .EQ. 0.0)
GO TO 140

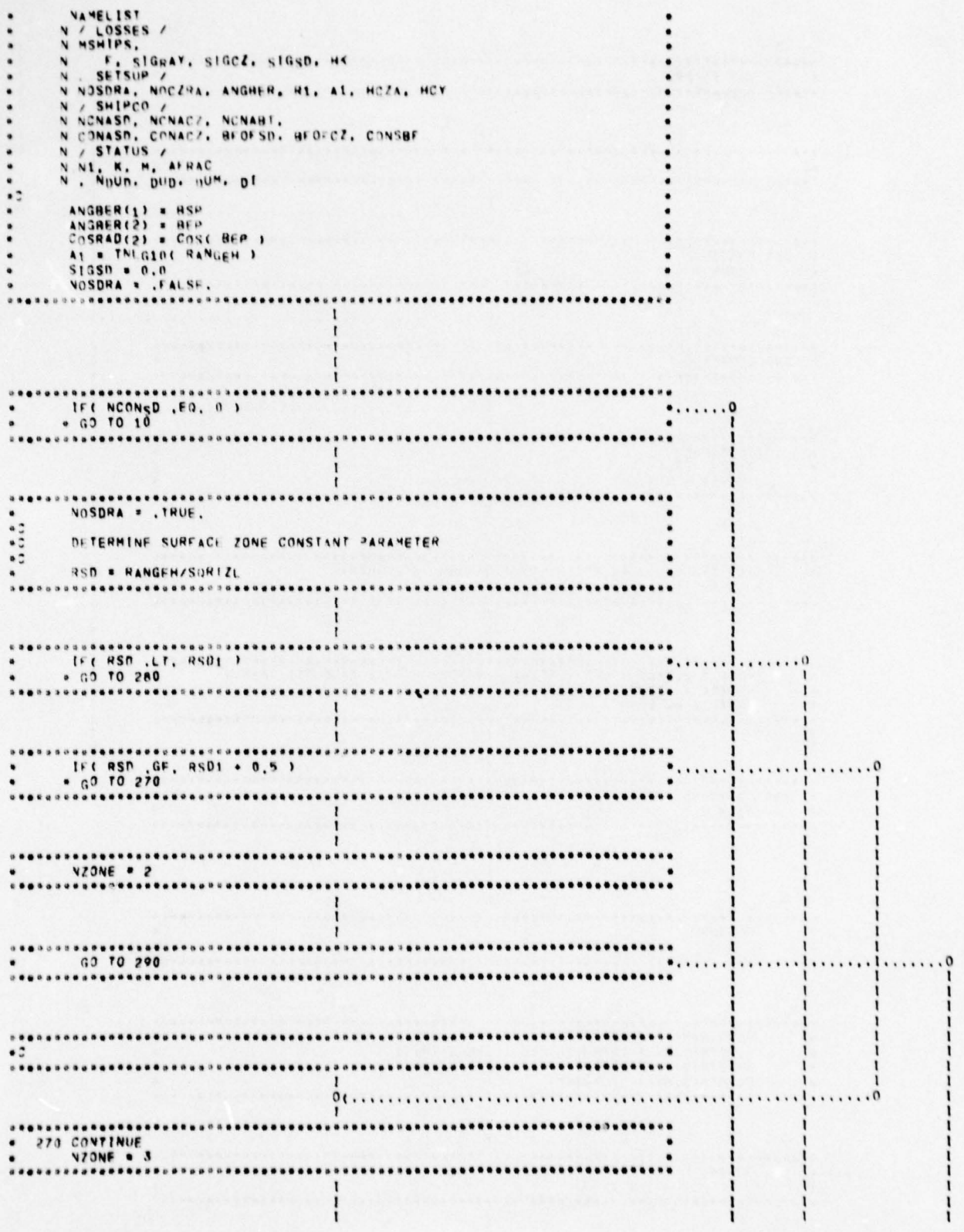

TI = ATAN(R1*TI1/TI2)/R1
GO TO 30
C
130 CONTINUE
TI = 0.0
GO TO 50
C
140 CONTINUE
TI = 1.57079633/R1
GO TO 30
C
150 CONTINUE
TI = 1.0E6
PL = TI
GO TO 60
C
160 CONTINUE
H = 2.0*DI*(1.0 + COSORD*SM)
GO TO 40

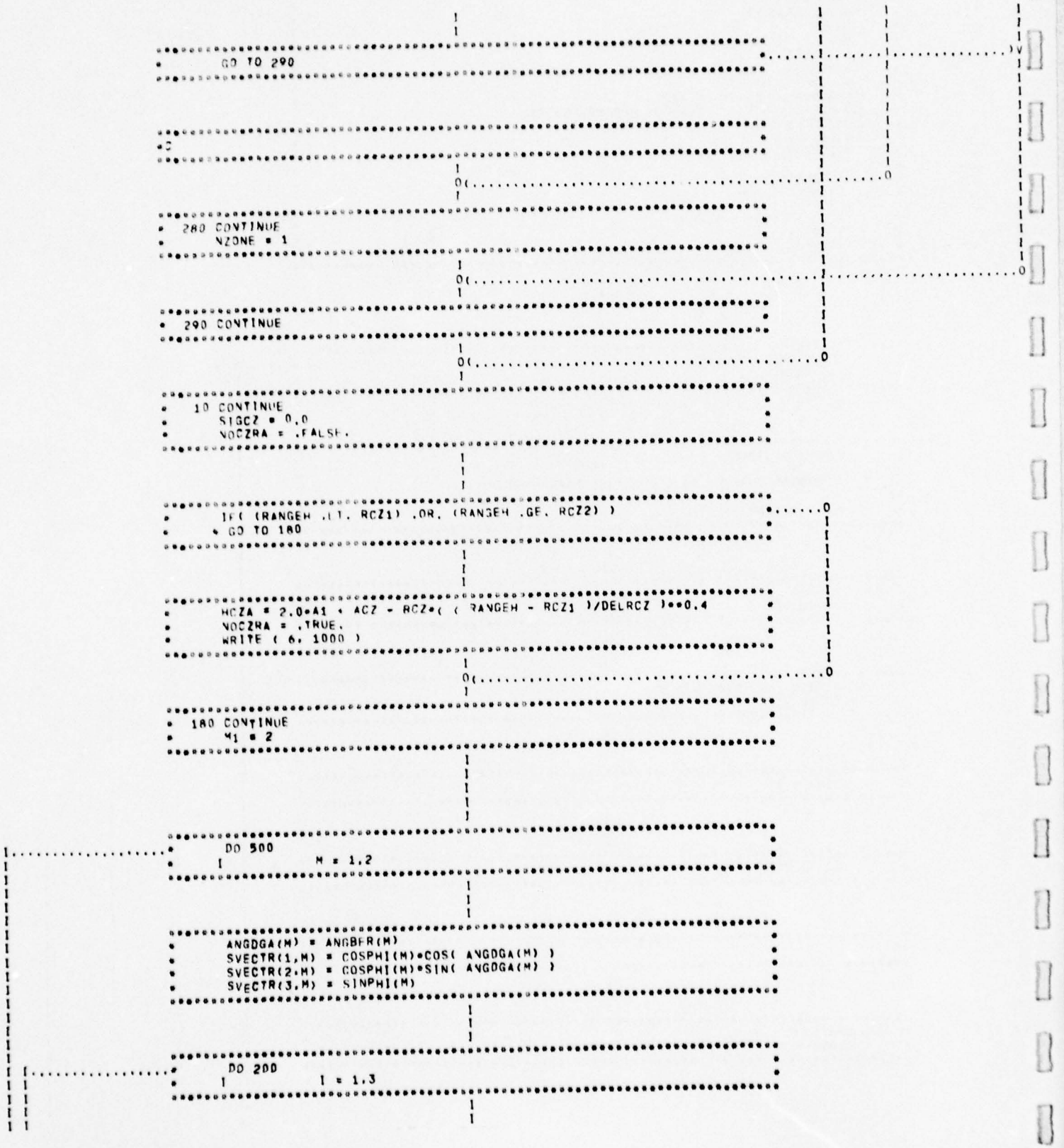

C
170 CONTINUE
TJ = 0.0
GO TO 50
C
180 CONTINUE
IF(CI .EQ. 0.0)
GO TO 190
AI = CONSVD(K) - SPVRSQ
QI = AI*CI - BI*HI
X1 = CI*DZ1 + BI
X2 = CI*A1 + BI
A3 = (X2*Y2 - X1*Y1 + QI*H**)/(2.0*CI*A2)
210 CONTINUE
TI = ABS(X/CV + A3)
GO TO 130
C
190 CONTINUE
A3 = 2.0*(Y1*Y1 + Y1*Y2 + Y2*Y2)*DI/3.0/A2
GO TO 210
C
END

```
                                (ENTRANCE)
                                    [
                                    1
*CRECEIV00          SUBROUTINE RECEIV                                          *
*SIGNAL000          THIS COMPUTES SIGNALS RECIEVED AS A FUNCTION OF FREQUEN*
*       SUBROUTINE                                                             *
*      S            RECIEV                                                     *
*
*
*       *                                                                     **
*       THIS COMPUTES REQUIRED NOISE AND SIGNAL SPECTRAE                      *
*       *                                                                     **
*       *                                                                     *
*
*       COMMON /LAREL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),PDE(3),PIE(3),CLPH,EVPH,RRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS,FRWS,FIS,PTS(128),FXS(128),FNS(128),FNS(128),FOE,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),FNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                     *
*       COMMON / ANGRAD / ANGBER(2)                                          *
*       COMMON                                                               *
*      C / ARRAYP /                                                          *
*      D            ANGDGA(2)          ,                                    *
*      D            ARRAYD(3,2)        ,                                    *
*      D            COSPHI(2)          ,                                    *
*      D            COSRAD(2)          ,                                    *
*      C            MSHIPS,                                                 *
*      D            SINPHI(2)          ,                                    *
*      D            SINRAD(2)          ,                                    *
*      D            TARRIV(3,2)        ,                                    *
*      D            TMATRX(3,3,2)      ,                                    *
*      D            TS01   (3)         ,                                    *
*      D            TS02   (3)         ,                                    *
*      DDUMMYS                                                              *
*       COMMON                                                              *
*      C / FREQUN /                                                          *
*      D            COSAVE(2)          ,                                    *
*      C            DELRCZ,                                                 *
*      D            DYHITH(2)          ,                                    *
*      D            ARAREA(2)          ,                                    *
*      C            CTWOPI,                                                 *
*      D            SQAREA(2)          ,                                    *
*      D            FRQSIG(128,2)      ,                                    *
*      D            FRQLCN(128,2)      ,                                    *
*      D            FRQLDB(128,2)      ,                                    *
*      D            ANGDEP(2)          ,                                    *
*      D            FRQTRN(128,2)      ,                                    *
*      D            FRQASD(128,2)      ,                                    *
*      D            FRQNCN(128,2)      ,                                    *
*      D            FRQNOS(128,2)      ,                                    *
*      D            NUMFRQ(2)          ,                                    *
*      DDUM9                                                                *
*       COMMON                                                              *
*      C / ARRAYC /                                                          *
*      D            NARRAY(2)          ,                                    *
*      C            ARRYH1,                                                 *
*      C            ARRYH2,                                                 *
*      D            ARWIDT(2),                                              *
*      C            DELF   ,                                                *
*      D            FRQRES(2)          ,                                    *
*      C            QTRAN1,                                                 *
*      C            QTRAN2,                                                 *
*      NDUM4                                                                *
*       COMMON                                                              *
*      C / CONSTN /                                                          *
*      C            DEPSER,                                                 *
*      C            NCONSK,                                                 *
*      C            SPATSE,                                                 *
*      C            DEPEVA,                                                 *
*      C            NCONSL,                                                 *
*      C            SPATEV                                                  *
```

```
      COMMON
     C / RAYPAR /
     C            RANGEH,
     D            BOTLOS(6)        ,
     D            DRDXDC(6)        ,
     D            PATHLN(6)        ,
     D            RANGEC(6)        ,
     D            SPRLOS(6,2)      ,
     D            ANGTER(6,2)
      COMMON
     C / INDEXS /
     D            ANDISO(16,2)     ,
     D            DIRSON(50,16,2)  ,
     D            DLDIAN(16,2)     ,
     D            NUDIAN(2)        ,
     CDUMMYA
      COMMON
     C / RCONST /
     C            ACZ    ,
     C            AMLSRD,
     C            BCZ    ,
     C            HCI    ,
     C            HZSD   ,
     C            NCONSD,
     C            RCZ1   ,
     C            RCZ2   ,
     C            SDCON  ,
     D            ANCZAV(2)        ,
     C            ZW     ,
     NDUM5
      COMMON
     C / SURDUC /
     C            BLA1   ,
     C            BLA2   ,
     C            BLA3   ,
     C            DTRAD  ,
     C            J      ,
     C            M      ,
     D            BAFFUN(128,2)    ,
     D            DELRAF(128,2)    ,
     D            FLONOS(128,2)    ,
     D            RADSPC(40,50,2)  ,
     C            NTIMEN
      COMMON
     C / SURFAC /
     C            A6     ,
     C            CONST2,
     C            CONST4,
     C            CZANGL,
     C            CZANDI,
     C            CZRANG,
     C            G1SD   ,
     C            G2SD   ,
     C            NCZRAS,
     C            NZONE  ,
     C            RSD    ,
     C            RSD1   ,
     C            SCSD   ,
     C            SQRTZL,
     C            SS     ,
     C            ZL     ,
     CDUMMYZ
C
      DIMENSION
     D            CONPHI(6)        ,
     D            CONSBF(6)        ,
     D            NCNABT(6)        ,
     D            POWSIG(128,2)    ,
     D            POWNOS(128,2)    ,
     D            SVECTR(3,2)      ,
     D            SO1    (3)       ,
     D            SO2    (3)       ,
     D            TVECTR(3,2)      ,
     DDUMMYS(1)
C
      LOGICAL
     L NOCZRA,
     L NOSDRA
C
```

```
      EQUIVALENCE                                                          *
     Q ( SVECTR, SO1 ),                                                    *
     Q ( SVECTR(1,2), SO2 ),                                               *
     Q ( TVECTR, TSO1 )                                                    *
C                                                                          *
C    AREAM1      = AZIMUTHAL ANGLE FOR SEARCHER STEERING                   (*
C    AREAM2      = AZIMUTHAL ANGLE FOR EVADER STEERING                     (*
C    ACZ         = CONVERGENT ZONE CONSTANT                                *
C    ANCZAV( )   = AVERAGE CONVERGENT ZONE ANGLES                          (*
C    ANDISO( , )= ANGLES OF DIFFERENT DIRECTIVITIES FOR SHIPS              (*
C    ANGAZM( )   = AZIMUTHAL STEERING ANGLES FOR EACH SHIP                 (*
C    ANGBER( )   = RELATIVE BEARING ANGLE FOR EACH SHIP                    (*
C    ARAD1       = AZIMUTHAL ANGLE OF SEARCHER MEASURED FROM EVADER        *
C    ARAD2       = AZIMUTHAL ANGLE OF EVADER MEASURED FROM SEARCHER        *
C    ARAREA( )   = ARRAY AREA AS A FUNCTION OF SHIP                        *
C    ARRAY2      = TYPE OF ARRAY CONTROL CONSTANT FOR EVADER      (DIMENS*
C    ARRAY1      = TYPE OF ARRAY CONTROL CONSTANT FOR SEARCHER    (DIMENS*
C    ARRAYD( , )= MODIFIED ARRAY DIMENSIONS FOR SHIPS                      *
C    ARRYH1      = ARRAY HEIGTH ON SEARCHER                                *
C    ARRYH2      = ARRAY HEIGTH ON EVADER                                  *
C    ARRAYW1     = ARRAY WIDTH ON SEARCHER                                 *
C    ARRAYW2     = ARRAY WIDTH ON EVADER                                   *
C    ARWIDT( )   = WIDTH OF ARRAY AS A FUNCTION OF SHIP                    *
C    ASNXOX      = FUNCTION TO COMPUTE SIN(X)/X                            *
C    BANGLE      = ABSOLUTE VALUE OF BOTTOM ANGLE                          (*
C    BCZ         = CONVERGENT ZONE CONSTANT                                *
C    BF1   ( )   = BAFFLING CORRECTION FACTOR FOR SEARCHER                 *
C                  AS A FUNCTION OF ANGLE                                  *
C    BF2   ( )   = BAFFLING CORRECTION FACTOR FOR EVADER                   *
C                  AS A FUNCTION OF ANGLE                                  *
C    BFOFCZ( )   = BAFFLING CONSTANT FOR CONVERGENT ZONE           (DIMENS*
C    BLA1        = BOTTOM LOSS CONSTANT                                    *
C    BLA2        = BOTTOM LOSS CONSTANT                                    *
C    BLA3        = BOTTOM LOSS CONSTANT                                    *
C    CDYOVV( )   = EFFECTIVE ARRAY HEIGTH AND SPEED CONSTANT               *
C    CONFBC      = FUNCTION TO COMPUTE CONSTANT FOR BC CALCULATIONS        *
C    COSAVE( )   = COSINE OF AVERAGE CONVERGENT ZONE ANGLES        (DIMENS*
C    DREAM1      = DEPRESSION ANGLE FOR SEARCHER STEERING                  (*
C    DREAM2      = DEPRESSION ANGLE FOR EVADER STEERING                    (*
C    DIRSON(,,) = DIRECTIVITY INDEX FOR SONAR                              *
C    DLDIAN( , )= DIFFERENCES IN ANGLES OF DIRECTIVITIES                   (*
C    DSIG        = DEPRESSION ANGLE OF RAY                                 (*
C    DTRAD       = ANGLE INCREMENT FOR RADIATED SIGNAL                     (*
C    DYHITH( )   = HIIGTH CONSTANT FOR ARRAY SURFACE DUCT          (DIMENS1*
C    FLN1  ( )   = PLANE WAVE FLOW NOISE FOR SEARCHER                      *
C    FLN2  ( )   = PLANE WAVE FLOW NOISE FOR EVADER                        *
C    FLONOS( , )= PLANE WAVE FLOW NOISE AS A FUNCTION OF FREQUENCY         *
C    FRAD11      = LOWER FREQUENCY LIMIT OF SEARCHER EQUIPMENT             *
C    FRAD12      = UPPER FREQUENCY LIMIT OF SEARCHER EQUIPMENT             *
C    FRAD21      = LOWER FREQUENCY LIMIT OF EVADER EQUIPMENT               *
C    FRAD22      = UPPER FREQUENCY LIMIT OF EVADER EQUIPMENT               *
C    FRES1       = TRANSDUCER RESONANT FREQUENCY ON SEARCHER               *
C    FRES2       = TRANSDUCER RESONANT FREQUENCY ON EVADER                 *
C    HZSD        = SURFACE DUCT CONSTANT                                   *
C    NARRAY( )   = CONTROL PARAMETER DEFINING TYPE OF ARRAY        (DIMENS*
C    NCNACZ( )   = BAFFLING ANGLE NUMBER AS A FUNCTION OF SHIP     (DIMENS*
C    NUDIAN( )   = NUMBER OF DIRECTIVITY ANGLES FOR SHIPS          (DIMENS*
C    NUFREV      = NUMBER OF FREQUENCY POINTS FOR EVADER           (DIMENS*
C    NUFRSE      = NUMBER OF FREQUENCY POINTS FOR SEARCHER         (DIMENS*
C    POR         = POROSITY OF BOTTOM                                      *
C    POWNOS( , )= NOISE POWER SPECTRUM AS A FUNCTION OF FREQUENCY          *
C    POWSIG( , )= SIGNAL POWER SPECTRUM AS A FUNCTION OF FREQUENCY         *
C    QTRAN1      = TRANSDUCER FIGURE OF MERIT ON SEARCHER                  *
C    QTRAN2      = TRANSDUCER FIGURE OF MERIT ON EVADER                    *
C    SDCON       = WAVE HEIGTH PARAMETER                                   *
C    SDCON       = SURFACE DUCT CONSTANT                                   *
C    SSD         =                                                         *
C    TCZAV1      =          AVERAGE OF CONVERGENT ZONE ANGLES              (*
C    TCZAV2      =          AVERAGE OF CONVERGENT ZONE ANGLES              (*
C    V           = PROPAGATION SPEED AT END POINT OF RAY                  (K*
C    XSD         = SEE -AMLSRD-                                          (SQR*
C    XSD         = SURFACE DUCT CONSTANT                                   *
C    ZW          = HEIGTH OF SURFACE WAVES                                 *
C                                                                          *
```


NAMELIST
N / LOSSES /
N NSHIPS,
N F, SIGRAY, SIGCZ, SIGSD, HK
N , SETSUP /
N NOSDRA, NOCZRA, ANGBER, R1, A1, HCZA, HCY
N / SHIPCO /
N NCNASD, NCNACZ, NCNABT,
N CONASD, CONACZ, BEOFSD, BEOFCZ, CONSBF
N / STATUS /
N N1, K, M, AFRAC
N , NDUD, DUD, DUM, DI
C
ANGBER(1) = BSP
ANGBER(2) = BEP
COSRAD(2) = COS(BEP)
A1 = ALOG10(RANGEH)
SIGSD = 0.0
NOSDRA = .FALSE.
IF(NCONSD .EQ. 0)
GO TO 10
NOSDRA = .TRUE.
C
C DETERMINE SURFACE ZONE CONSTANT PARAMETER
C
RSD = RANGEH/SQRT2L
IF(RSD .LT. RSD1)
GO TO 280
IF(RSD .GE. RSD1 + 0.5)
GO TO 270
NZONE = 2
GO TO 290
C
270 CONTINUE
NZONE = 3


GO TO 290
C
280 CONTINUE
NZONE = 1
290 CONTINUE
10 CONTINUE
SIGCZ = 0.0
NOCZRA = .FALSE.
IF((RANGEH .LT. RCZ1) .OR. (RANGEH .GE. RCZ2))
GO TO 180
HCZA = 2.0*A1 + ACZ - RCZ*((RANGEH - RCZ1)/DELRCZ)**0.4
NOCZRA = .TRUE.
WRITE (6, 1000)
180 CONTINUE
M1 = 2
DO 500
M = 1,2
ANGDGA(M) = ANGBER(M)
SVECTR(1,M) = COSPHI(M)*COS(ANGDGA(M))
SVECTR(2,M) = COSPHI(M)*SIN(ANGDGA(M))
SVECTR(3,M) = SINPHI(M)
DO 200
I = 1,3

```
                    TVECTR(I,M) = 0.0

                    DO 200
                    I           J = 1,3

                    TVECTR(I,M) = TMATRX(I,J,M)*SVECTR(J,M)
                   E + TVECTR(I,M)

200 CONTINUE

                    MSHIPS = M
                    SINRAD(M) = SIN( ANGBER(M) )
                    COSRAD(M) = COS( ANGBER(M) )
                    BEOFCZ = BAFFLE( NCNACZ, ACOS( COSRAD(M)*COSAVE(M) ), CONACZ)-HCZA
                    BEOFSD = BAFFLE( NCNASD, ANGBER(M), CONASD )

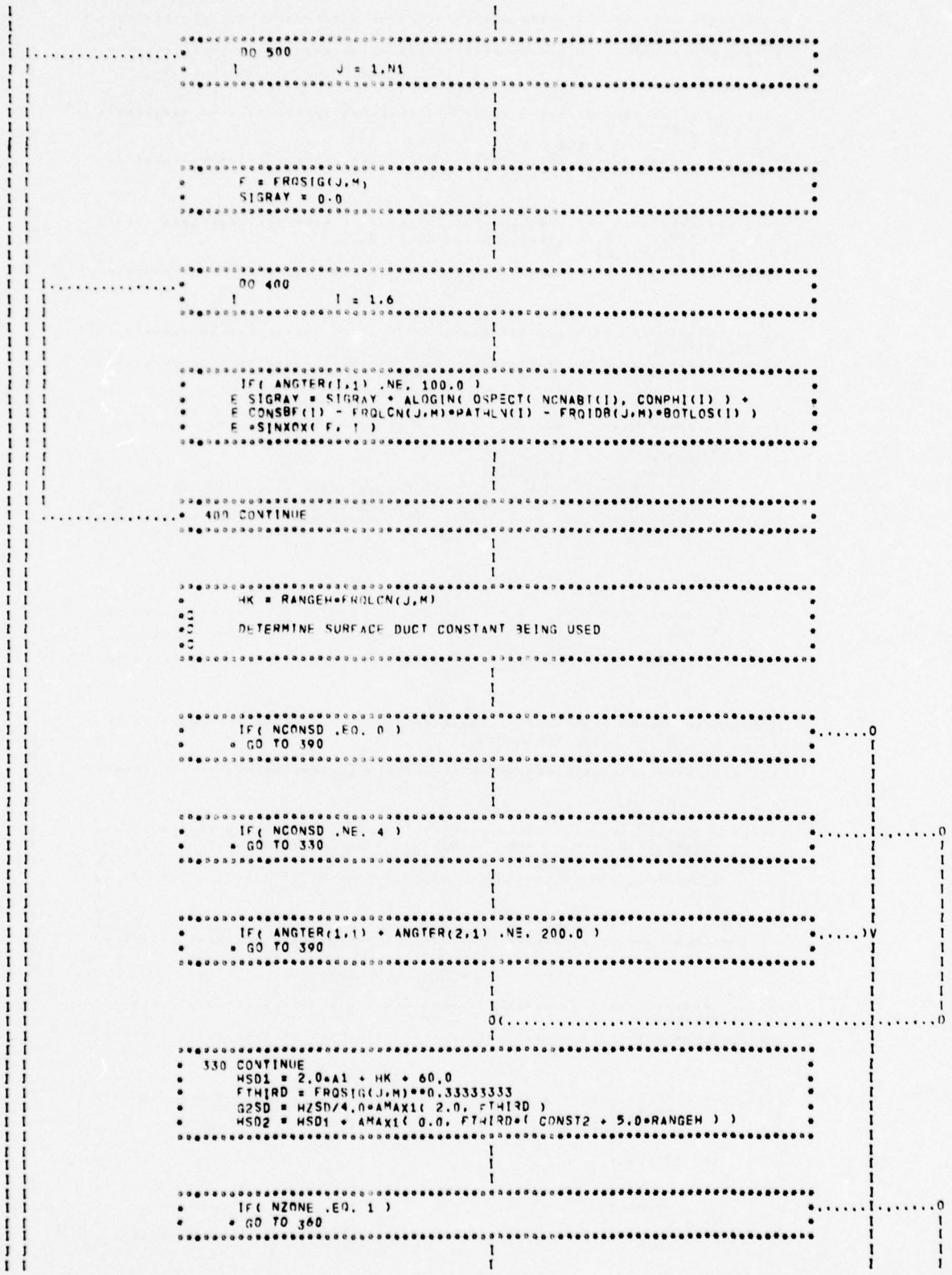
                    DO 600
                    I           I = 1,6

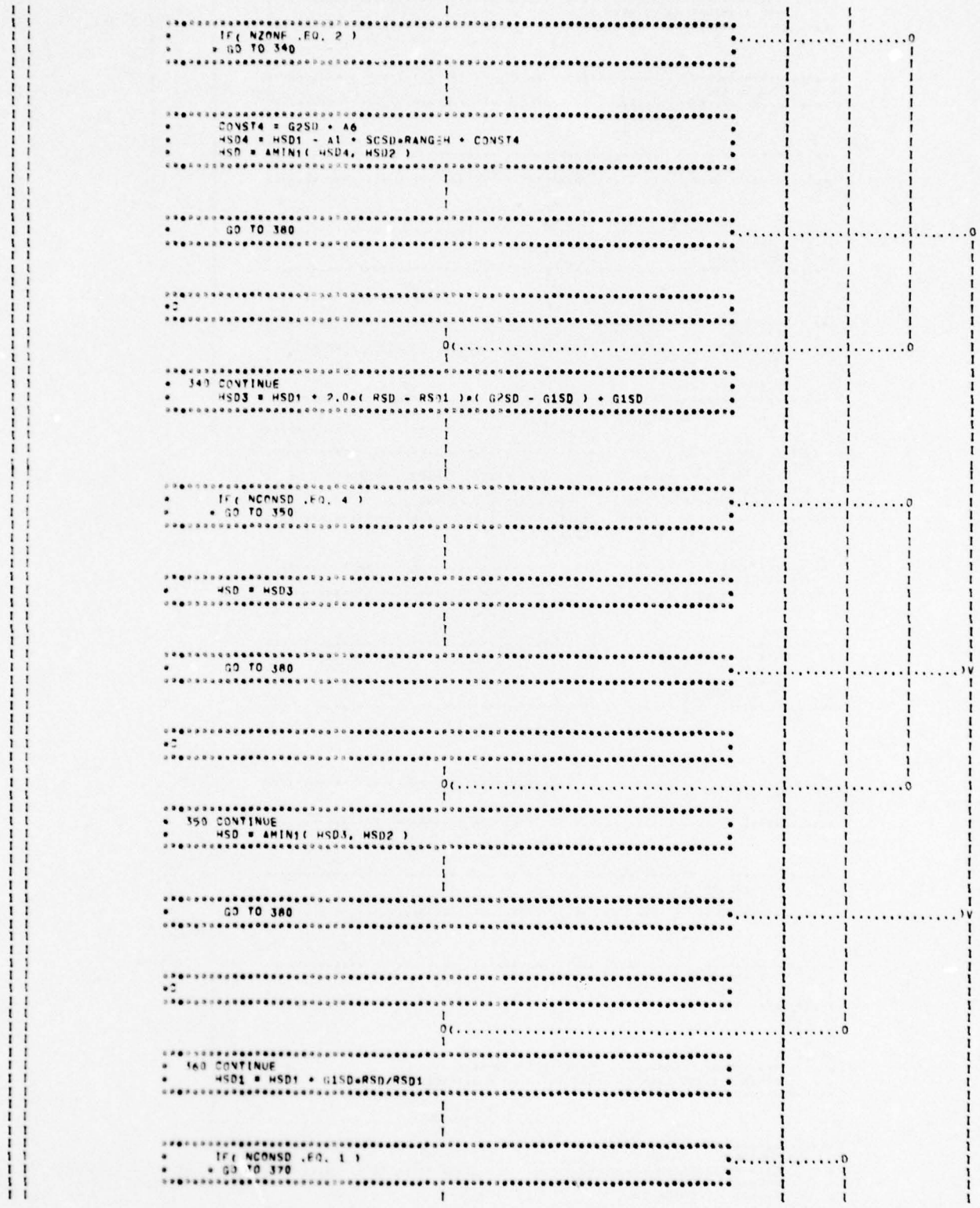
                    IF( ANGTER(I,1) .EQ. 100.0 )
                   * GO TO 600

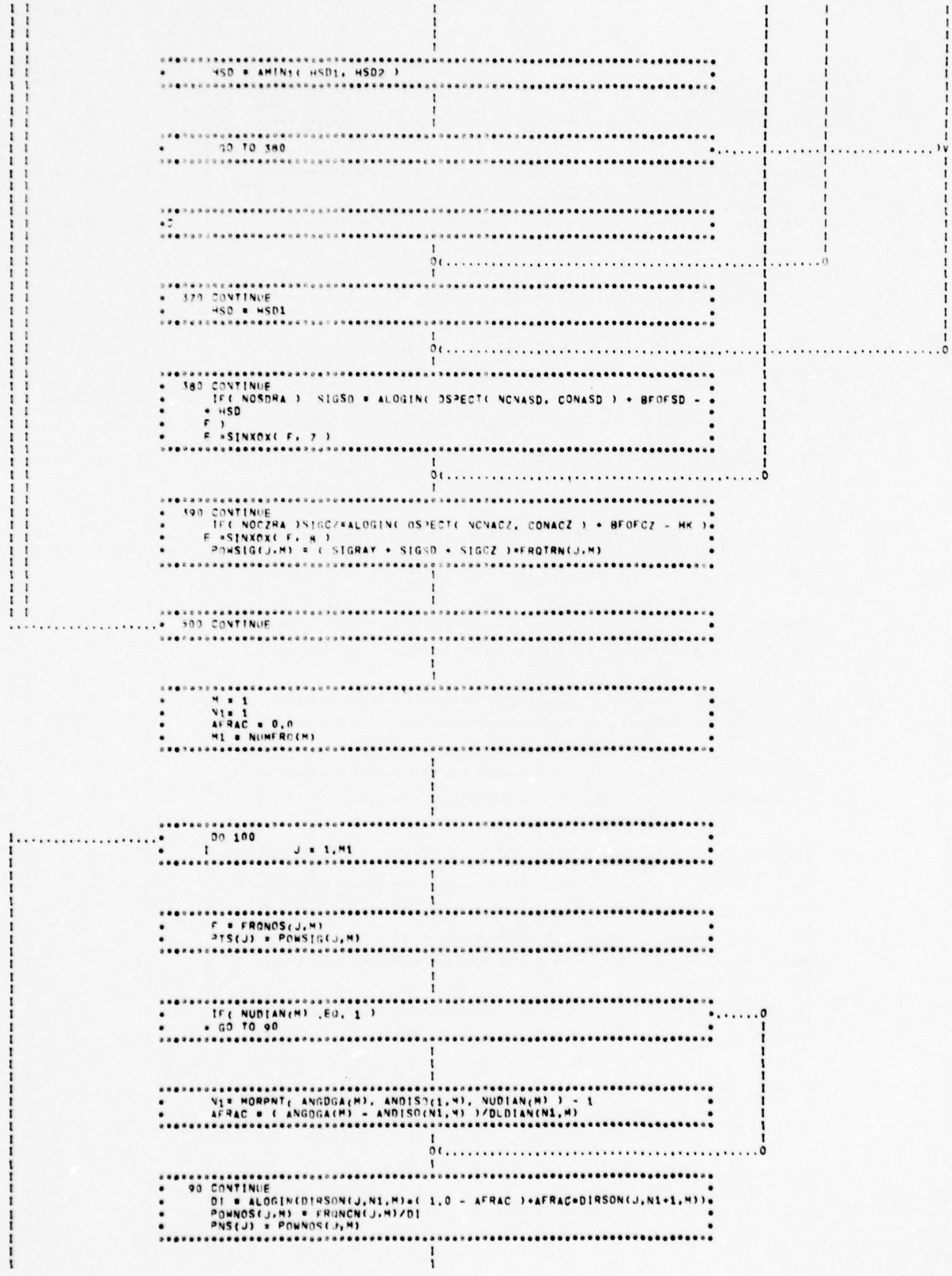
                    CONSBF(I) = BAFFLE( NDUD, ACOS( COSRAD(M)*COS( ANGTER(I,M) ) ),
                   E DUD ) + SPRLOS(I,M1)
                    DUM=BAFFLE(NCNABT(I),ACOS(COSRAD(M1)*COS(ANGTER(I,M1))),CONPHI(I))
                    CALL
                   S            CORREC
                   S            ( ANGTER(I,M), I, M )

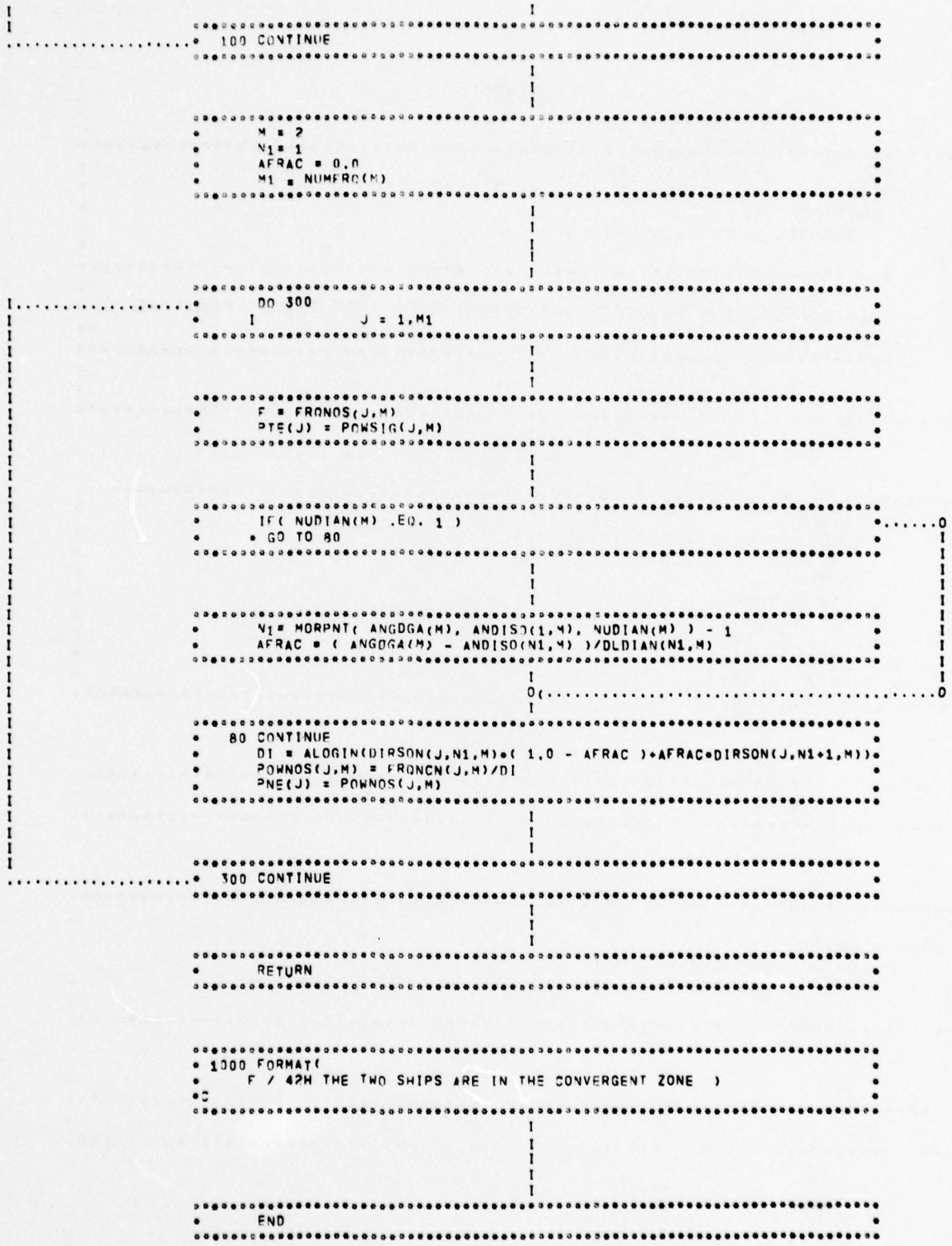
600 CONTINUE

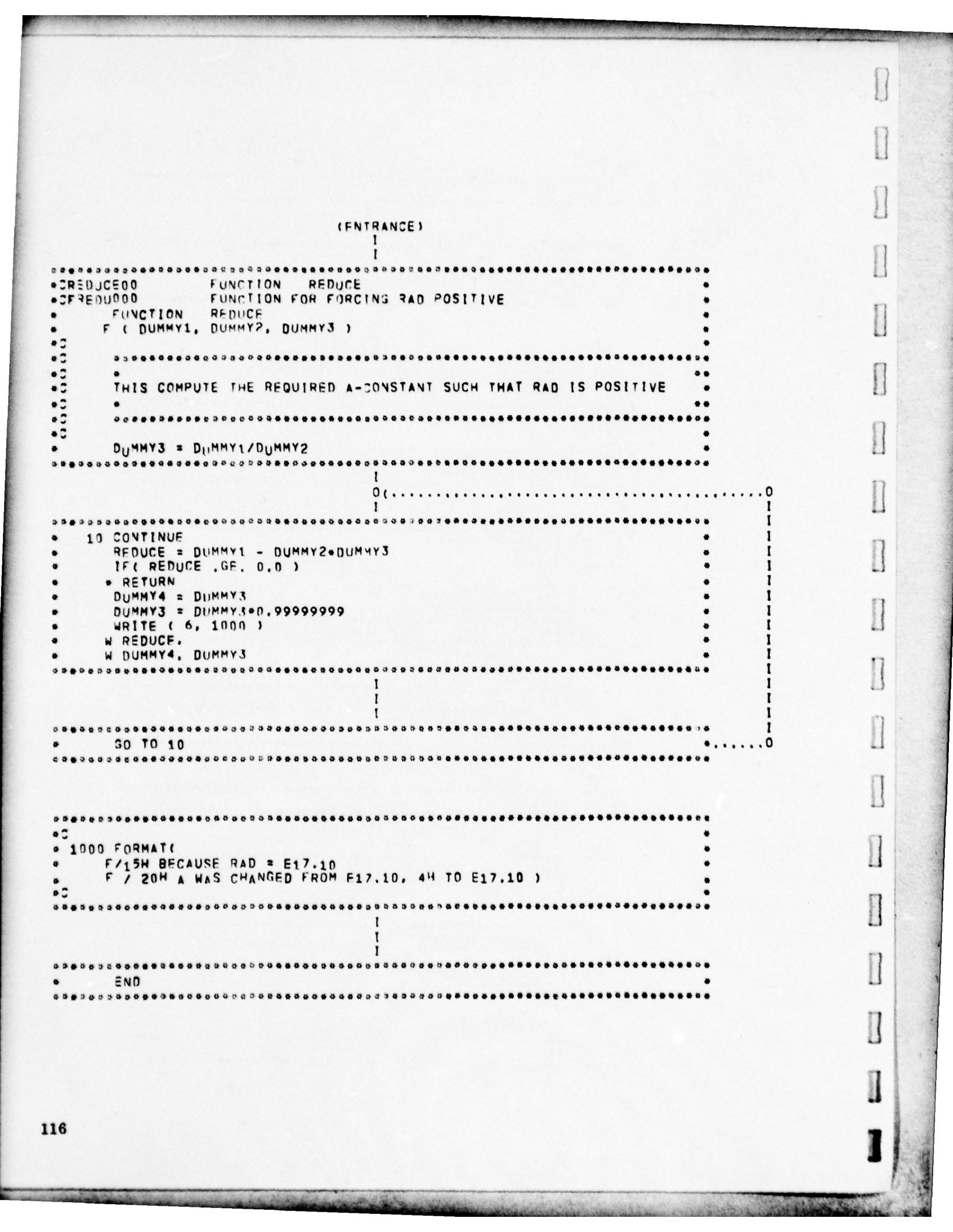
                    IF( NOSDRA )  CALL
                   S            CORREC
                   S            ( 0.0, 7, M )
                    IF( NOCZRA )  CALL
                   S            CORREC
                   S            ( ANCZAV(M), 8, M )
                    N1 = NUMFRQ(M)
                    M1 = 1
```

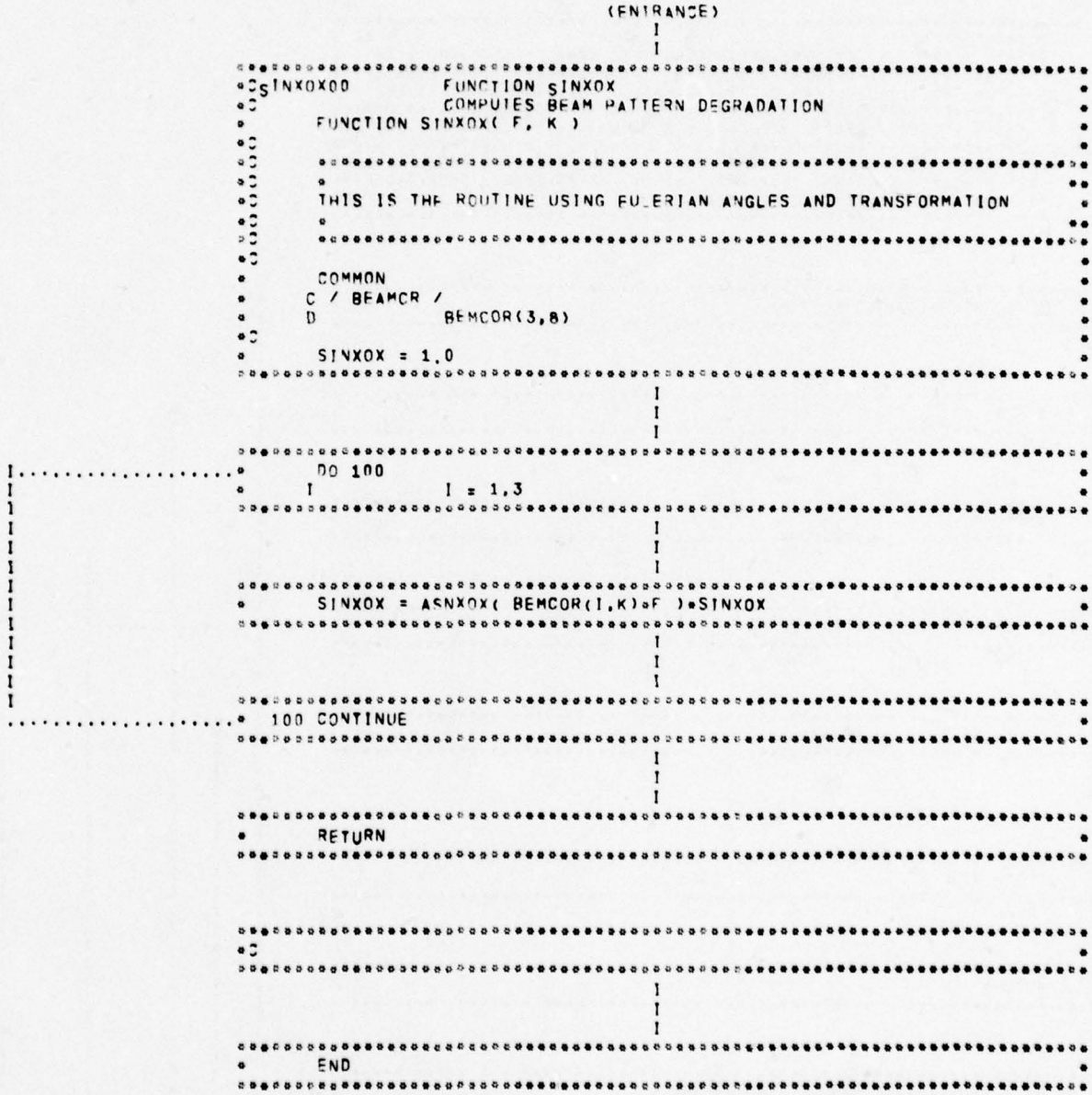

DO 500
J = 1,N1
F = FRQSIG(J,M)
SIGRAY = 0.0
DO 400
I = 1,6
IF(ANGTER(I,1) .NE. 100.0)
E SIGRAY = SIGRAY + ALOGIN(OSPECT(NCNABI(I), CONPHI(I)) +
E CONSBF(I) - FRQLCN(J,M)*PATHLN(I) - FRQIDB(J,M)*BOTLOS(I))
E *SINXOX(F, I)
400 CONTINUE
HK = RANGEH*FRQLCN(J,M)
DETERMINE SURFACE DUCT CONSTANT BEING USED
IF(NCONSD .EQ. 0)
GO TO 390
IF(NCONSD .NE. 4)
GO TO 330
IF(ANGTER(1,1) + ANGTER(2,1) .NE. 200.0)
GO TO 390
330 CONTINUE
HSD1 = 2.0*A1 + HK + 60.0
FTHIRD = FRQSIG(J,M)**0.33333333
G2SD = HZSD/4.0*AMAX1(2.0, FTHIRD)
HSD2 = HSD1 + AMAX1(0.0, FTHIRD*(CONST2 + 5.0*RANGEH))
IF(NZONE .EQ. 1)
GO TO 360

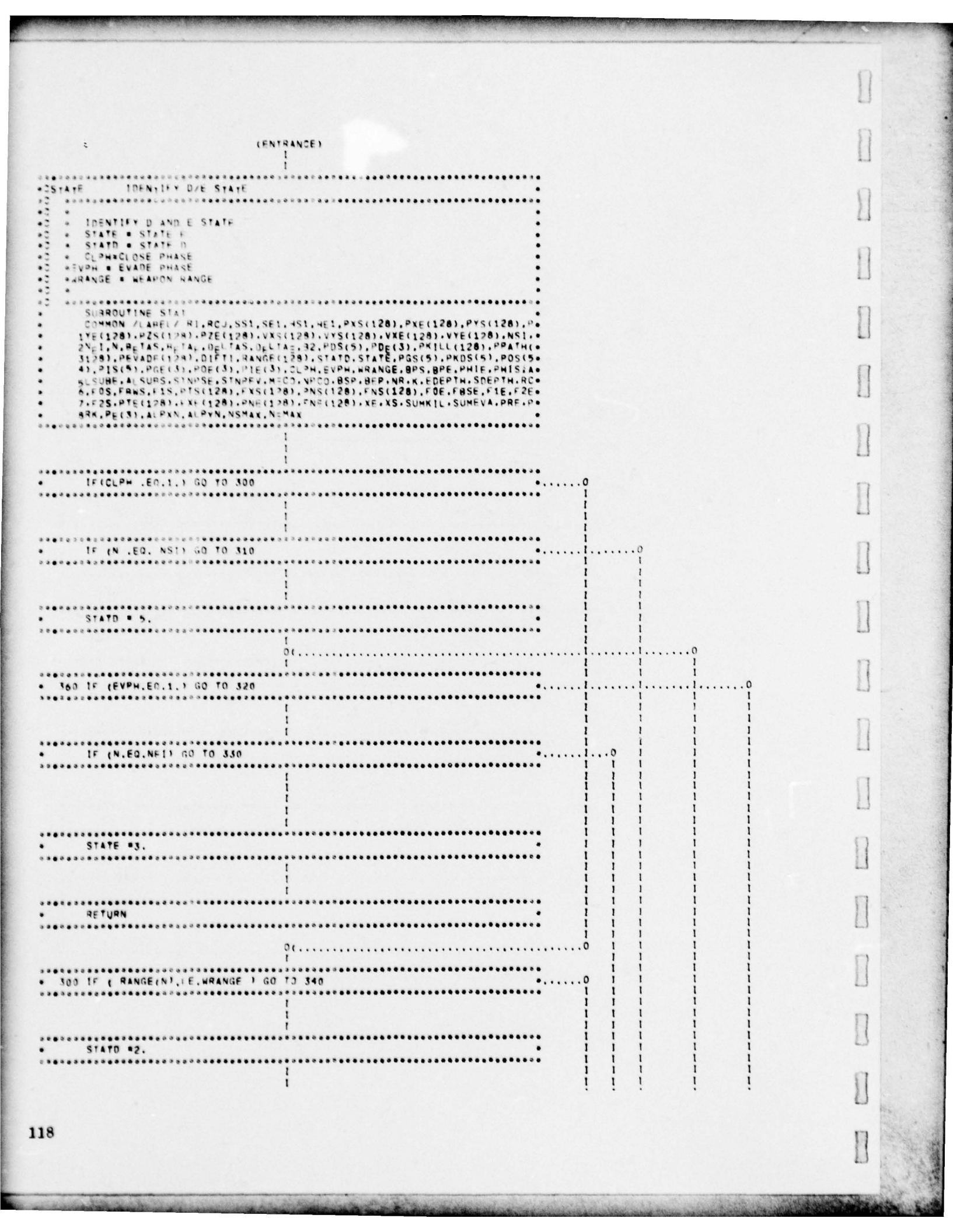

IF(NZONE .EQ. 2)
GO TO 340
CONST4 = G2SD + A6
HSD4 = HSD1 - A1 * SCSD*RANGEH + CONST4
HSD = AMIN1(HSD4, HSD2)
GO TO 380
C
340 CONTINUE
HSD3 = HSD1 + 2.0*(RSD - RSD1)*(G2SD - G1SD) + G1SD
IF(NCONSD .EQ. 4)
GO TO 350
HSD = HSD3
GO TO 380
C
350 CONTINUE
HSD = AMIN1(HSD3, HSD2)
GO TO 380
C
360 CONTINUE
HSD1 = HSD1 + G1SD*RSD/RSD1
IF(NCONSD .EQ. 1)
GO TO 370

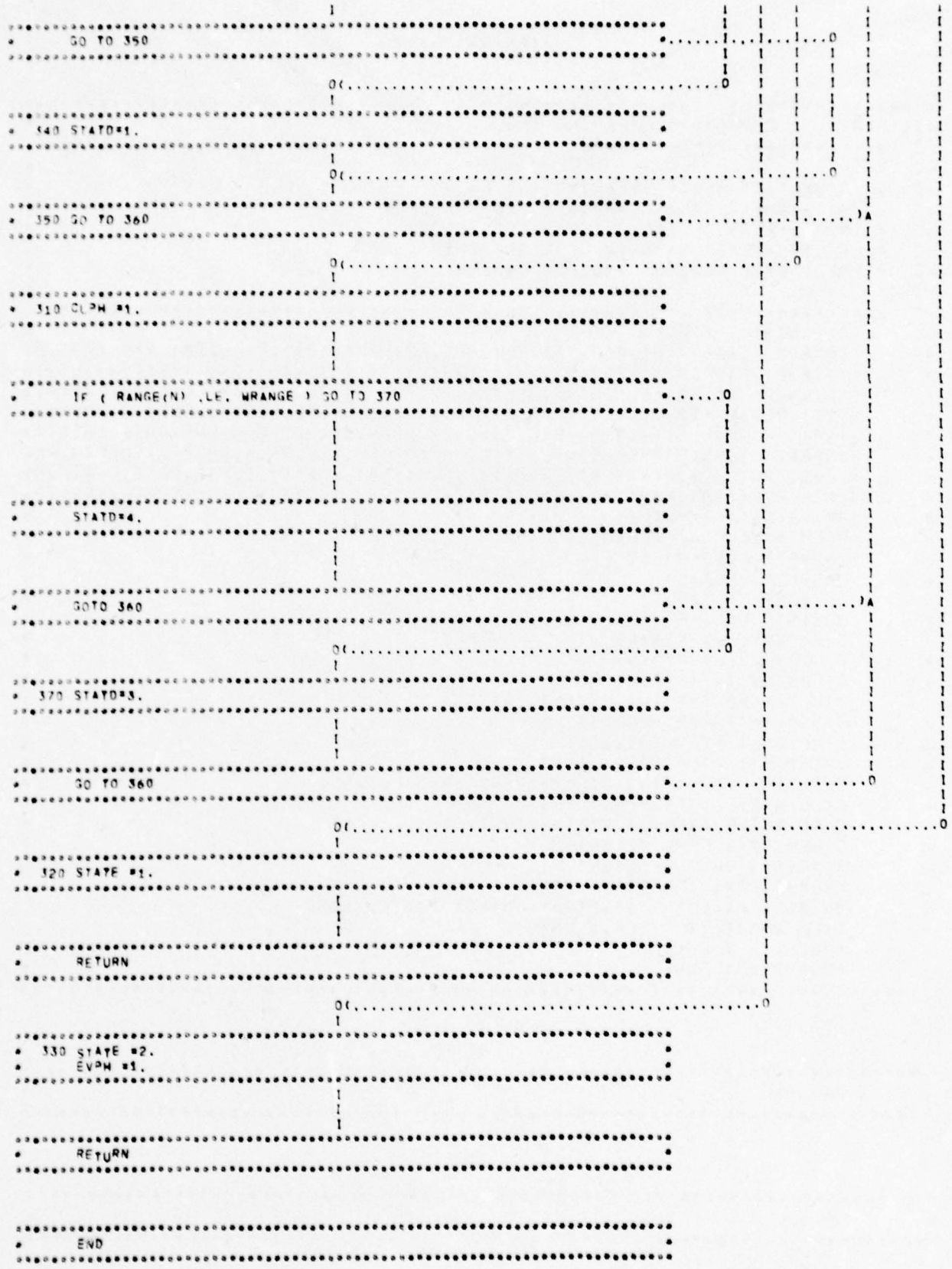

HSD = AMIN1(HSD1, HSD2)
GO TO 380
C
370 CONTINUE
HSD = HSD1
380 CONTINUE
IF(NOSDRA) SIGSD = ALOGIN(OSPECT(NCNASD, CONASD) + BFOFSD -
+ HSD
F)
F *SINXOX(F, 7)
390 CONTINUE
IF(NOCZRA)SIGCZ=ALOGIN(OSPECT(NCNACZ, CONACZ) + BFOFCZ - HK)
F *SINXOX(F, 8)
POWSIG(J,M) = (SIGRAY + SIGSD + SIGCZ)*FRQTRN(J,M)
390 CONTINUE
M = 1
N1= 1
AFRAC = 0.0
M1 = NUMFRQ(M)
DO 100
J = 1,M1
F = FRQNOS(J,M)
PTS(J) = POWSIG(J,M)
IF(NUDIAN(M) .EQ. 1)
GO TO 90
N1= MORPNT(ANGDGA(M), ANDISO(1,M), NUDIAN(M)) - 1
AFRAC = (ANGDGA(M) - ANDISO(N1,M))/DLDIAN(N1,M)
90 CONTINUE
DI = ALOGIN(DIRSON(J,N1,M)+(1.0 - AFRAC)+AFRAC+DIRSON(J,N1+1,M))
POWNOS(J,M) = FRQNCN(J,M)/DI
PNS(J) = POWNOS(J,M)


100 CONTINUE
M = 2
N1= 1
AFRAC = 0.0
M1 = NUMFRQ(M)
DO 300
J = 1,M1
F = FRQNOS(J,M)
PTE(J) = POWSIG(J,M)
IF(NUDIAN(M) .EQ. 1)
GO TO 80
N1= MORPNT(ANGDGA(M), ANDISO(1,M), NUDIAN(M)) - 1
AFRAC = (ANGDGA(M) - ANDISO(N1,M))/DLDIAN(N1,M)
80 CONTINUE
DI = ALOGIN(DIRSON(J,N1,M)*(1.0 - AFRAC)+AFRAC*DIRSON(J,N1+1,M))
POWNOS(J,M) = FRQNCN(J,M)/DI
PNE(J) = POWNOS(J,M)
300 CONTINUE
RETURN
1000 FORMAT(
F / 42H THE TWO SHIPS ARE IN THE CONVERGENT ZONE)
END


(ENTRANCE)
C REDUCE00 FUNCTION REDUCE
C FREDU000 FUNCTION FOR FORCING RAD POSITIVE
FUNCTION REDUCE
F (DUMMY1, DUMMY2, DUMMY3)
THIS COMPUTE THE REQUIRED A-CONSTANT SUCH THAT RAD IS POSITIVE
DUMMY3 = DUMMY1/DUMMY2
10 CONTINUE
REDUCE = DUMMY1 - DUMMY2*DUMMY3
IF(REDUCE .GE. 0.0)
* RETURN
DUMMY4 = DUMMY3
DUMMY3 = DUMMY3*0.99999999
WRITE (6, 1000)
W REDUCE,
W DUMMY4, DUMMY3
GO TO 10
1000 FORMAT(
F/15H BECAUSE RAD = E17.10
F / 20H A WAS CHANGED FROM E17.10, 4H TO E17.10)
END


(ENTRANCE)
C SINXOX00 FUNCTION SINXOX
C COMPUTES BEAM PATTERN DEGRADATION
FUNCTION SINXOX(F, K)
C
C
THIS IS THE ROUTINE USING EULERIAN ANGLES AND TRANSFORMATION
C
COMMON
C / BEAMCR /
D BEMCOR(3,8)
C
SINXOX = 1.0
DO 100
I I = 1,3
SINXOX = ASNXOX(BEMCOR(I,K)*F)*SINXOX
100 CONTINUE
RETURN
C
END


(ENTRANCE)
SUBROUTINE STATE
IDENTIFY D/E STATE
IDENTIFY D AND E STATE
STATE = STATE E
STATD = STATE D
CLPH=CLOSE PHASE
EVPH = EVADE PHASE
WRANGE = WEAPON RANGE
SUBROUTINE STAT
COMMON /LABEL/ R1,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),PYE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NS1,NE1,N,BETAS,BETAL,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(128),PEVADE(128),DIFT1,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5),PIS(5),PGE(3),PDE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,ALSUBE,ALSUBS,SINPSE,SINPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC,FOS,FBNS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,F8SE,F1E,F2E,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PREAK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX
IF(CLPH .EQ.1.) GO TO 300
IF (N .EQ. NS1) GO TO 310
STATD = 5.
360 IF (EVPH.EQ.1.) GO TO 320
IF (N.EQ.NE1) GO TO 330
STATE =3.
RETURN
300 IF (RANGE(N),LE.WRANGE) GO TO 340
STATD =2.


GO TO 350
340 STATD=1.
350 GO TO 360
310 CLPH =1.
IF (RANGE(N) .LE. WRANGE) GO TO 370
STATD=4.
GOTO 360
370 STATD=3.
GO TO 360
320 STATE =1.
RETURN
330 STATE =2.
EVPH =1.
RETURN
END

```
                                (ENTRANCE)
                                    I
                                    I
*****************************************************************************
*CSTEER      COMPUTING STEERING ANGLES                                        *
*C  ***************************************************************************
*C  *                                                                         *
*C  *  COMPUTING ARRAY STEERING ANGLES                                        *
*C  *  BETAS=BEARING ANG.WRT.BOW FOR SEARCHER                                 *
*C  *  BETAE=BEARING ANG.WRT.BOW FOR EVADER                                   *
*C  *  DELTAS=DEPRESSION ANG.FOR SEARCHER                                     *
*C  *  DELTAE=DEPRESSION ANG.FOR EVADER                                       *
*C  *                                                                         *
*C  ***************************************************************************
*       SUBROUTINE STEERA                                                     *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,F0S,F8WS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,F8SE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                     *
*       COMMON /ALPHA/ ABEAM1,ABEAM2                                          *
*       SDS=SIN(DELTAS)                                                       *
*       CDS=COS(DELTAS)                                                       *
*       CDE=COS(DELTAE)                                                       *
*       SDE=SIN(DELTAE)                                                       *
*       DIFPY=PYE(N)-PYS(N)                                                   *
*       DIFPZ=PZE(N)-PZS(N)                                                   *
*       DIFPX=PXE(N)-PXS(N)                                                   *
*       CALL ANGVE (VXS(N),VYS(N),THETHS)                                     *
*       STBS=SIN(THETHS+BETAS)                                                *
*       CTBS=COS(THETHS+BETAS)                                                *
*       XPS=DIFPX*CTBS-DIFPY*STBS                                             *
*       YPS=DIFPX*CDS*STBS+DIFPY*CDS*CTBS+DIFPZ*SDS                           *
*       CALL ANGVE (XPS,YPS,ALPHAS)                                           *
*       CALL ANGVE (VXE(N),VYE(N),THETHE)                                     *
*       CTBE=COS(THETHE+BETAE)                                                *
*       STBE=SIN(THETHE+BETAE)                                                *
*       XPE=-DIFPX*CTBE+DIFPY*STBE                                            *
*       YPE=-DIFPX*CDE*STBE-DIFPY*CDE*CTBE-DIFPZ*SDE                          *
*       CALL ANGVE (XPE,YPE,ALPHAE)                                           *
*       ABEAM1 = ALPHAS                                                       *
*       ABEAM2 = ALPHAE                                                       *
*****************************************************************************
                                    I
                                    I
                                    I
*****************************************************************************
*       RETURN                                                                *
*****************************************************************************


*****************************************************************************
*       END                                                                   *
*****************************************************************************
```

```
                                    (ENTRANCE)
                                        I
                                        I
*********************************************************************************
*CSTRTRA00          SUBROUTINE STRTRA                                            *
*CSSTRA000          SUBROUTINE FOR STARTING RAY TRACINGS                         *
*       SUBROUTINE                                                              *
*      S            STRTRA                                                      *
*      S ( N, DEPEND )                                                         *
*C                                                                              *
*C      ***************************************************************************
*C      *                                                                      **
*C      *                                                                      **
*C      ***************************************************************************
*C                                                                              *
*       COMMON                                                                  *
*      C / CONSTN /                                                            *
*      C            DEPSER,                                                    *
*      C            NCONSK,                                                    *
*      C            SPATSE,                                                    *
*      C            DEPEVA,                                                    *
*      C            NCONSL,                                                    *
*      C            SPATEV                                                     *
*       COMMON                                                                  *
*      C / PTHLNG /                                                            *
*      C            A1    ,                                                    *
*      C            BI    ,                                                    *
*      C            CDSORD,                                                    *
*      C            CI    ,                                                    *
*      C            DI    ,                                                    *
*      C            DXDC  ,                                                    *
*      C            DZ1   ,                                                    *
*      C            CV    ,                                                    *
*      C            K     ,                                                    *
*      C            PL    ,                                                    *
*      C            SM    ,                                                    *
*      C            V     ,                                                    *
*      C            X     ,                                                    *
*      C            Y1    ,                                                    *
*      C            Y2    ,                                                    *
*      C            TIMCON                                                     *
*       COMMON                                                                  *
*      C / RANGES /                                                            *
*      C            NUMANG,                                                    *
*      C            ANGMAX,                                                    *
*      C            DELANG,                                                    *
*      C            DELRAD,                                                    *
*      D            ANGINT(200)                                                *
*      D            RNGMOD(6,200)                                              *
```

```
*      COMMON                                                          *
*     C / RAYPAR /                                                     *
*     C           RANGEH,                                              *
*     D           BOTANG(6)       ,                                    *
*     D           DRDXDC(6)       ,                                    *
*     D           PATHLN(6)       ,                                    *
*     D           RANGEC(6)       ,                                    *
*     D           SPI(6)          ,                                    *
*     D           SPT(6)          ,                                    *
*     D           TIR(6)          ,                                    *
*     D           TTR   (6)                                            *
*      COMMON                                                          *
*     C / RAYTRA /                                                     *
*     C           NCONCI,                                              *
*     C           INITLK,                                              *
*     C           Z1    ,                                              *
*     C           Z2    ,                                              *
*     C           SPVRSQ,                                              *
*     C           ANGSTR,                                              *
*     C           ANGARR,                                              *
*     C           ANGRTM,                                              *
*     C           ANGSUR,                                              *
*     C           SPDVER,                                              *
*     C           RANGET                                               *
*C                                                                      *
*C                                                                      *
*      ANGSTR = TIR(N)                                                 *
*      SPDVER = SPATSE/COS( ANGSTR )                                   *
*      SPVRSQ = 1.0/SPDVER/SPDVER                                      *
*      DXDC = 0.0                                                      *
*      PL = 0.0                                                        *
*      RANGET = 0.0                                                    *
*      TIMCON = 0.0                                                    *
*      CALL                                                            *
*     S           LENGTH                                               *
*     S ( N, NCONSK, DEPSER, DEPEND )                                  *
*  900 CONTINUE                                                        *
**********************************************************************
                                      I
                                      I
                                      I
**********************************************************************
*      RETURN                                                          *
**********************************************************************



**********************************************************************
*C                                                                     *
**********************************************************************
                                      I
                                      I
                                      I
**********************************************************************
*      END                                                             *
**********************************************************************
```

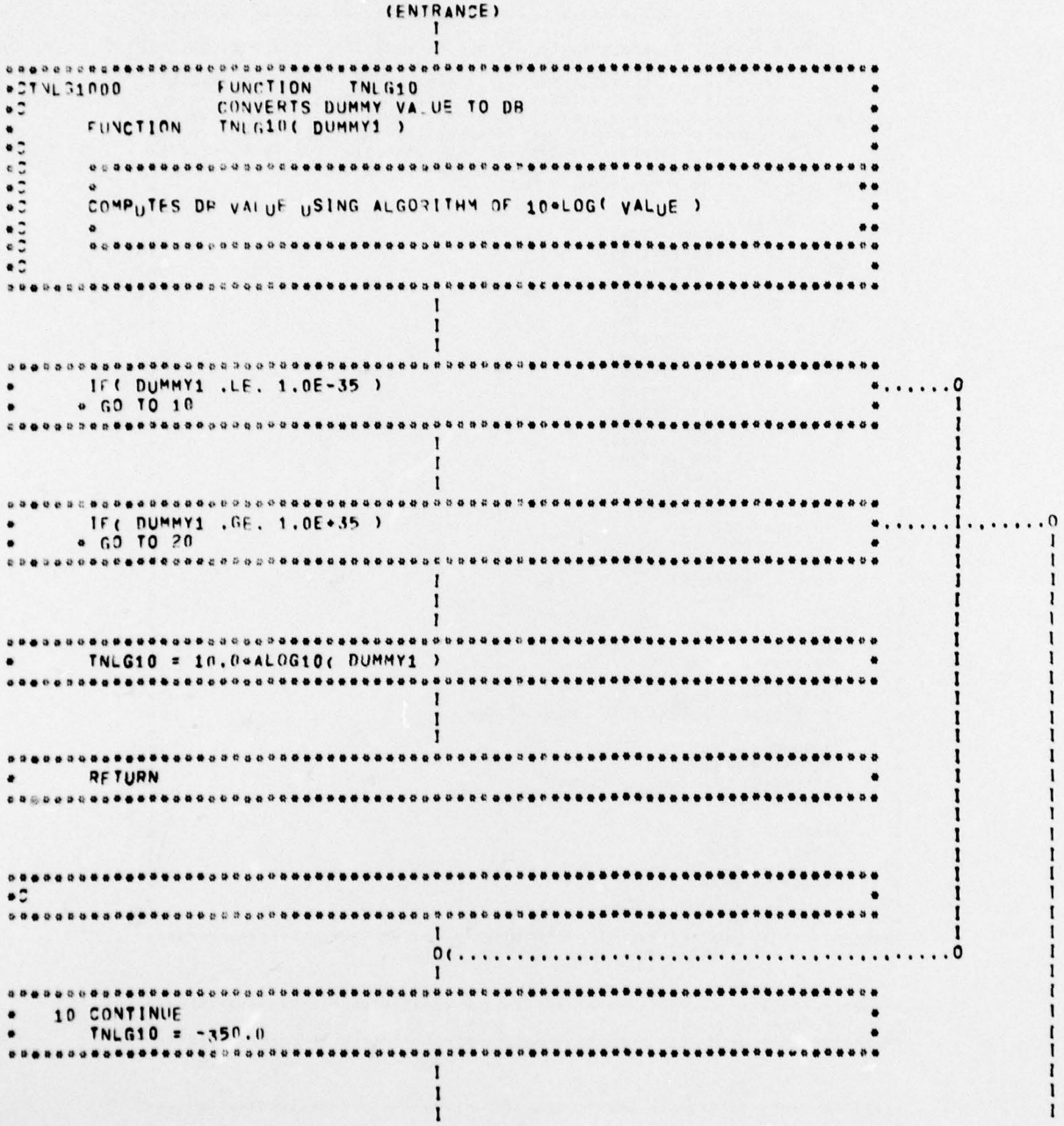
(ENTRANCE)

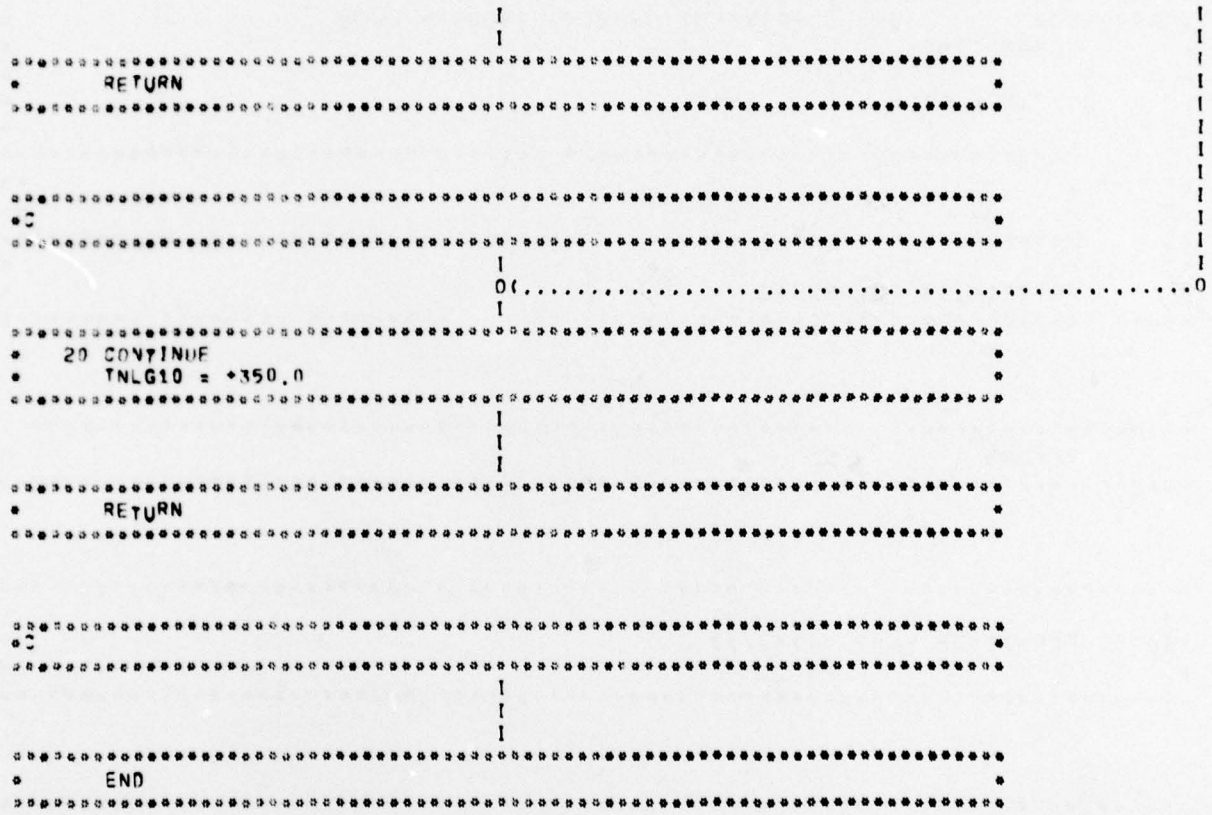
```
C*TAB_E00    D/E STATE TABLE
C*
C*  *
C*  *  D STATE -E STATE TABLE
C*  *
C*
      SUBROUTINE TABLE
      COMMON /LAREL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P
     1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,
     2NE1,N,BETAS,HETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(
     3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),ABC(5
     4),DEF(5),PGE(3),GHI(3),JKL(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A
     5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,VPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC
     6,FOS,FRWS,F1S,PTS(128),FXS(128),FNS(128),FNS(128),FOE,FBWE,F1E,F2E
     7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P
     8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX
      COMMON
     C / SIGNAL /
     D            PRYSEV(128)      ,
     D            PRYSSE(128)      ,
     D            PRNOEV(128)      ,
     D            PRNOSE(128)      ,
     D            PROEVA(128)      ,
     D            PROSER(128)      ,
     D            VAREVA(128)      ,
     D            VARSER(128)      ,
     D            GMUEVA(128)      ,
     D            GMUSER(128)      ,
     D            DEVAEV(128)      ,
     D            DEVASE(128)      ,
     D            DEMUEV(128)      ,
     D            DEMUSE(128)      ,
     C            THREVA,
     C            THRSER,
     C            NTIMEN
   10 ONE=1.-PROSER(N)
      TWO=1.-PROEVA(N)
      THREE=1.-PROSER(N)*PRK
      FOUR = PROSER(N)
      FIVE = PROEVA(N)
C
      PDE(1) = FIVE
      PDS(1) = FOUR
      PRNOEV(1) = TWO
      PRNOSE(1) = ONE/THREE
      PRYSEV(1) = FIVE
      PRYSSE(1) = FOUR*( 1.0 - PRK )/THREE
C
      PDS(2) = FOUR
      PGE(2) = FIVE
      PRNOSE(2)  =  1.0 - PROSER(N)
      PRYSSE(2) = FOUR
C
      PGE(3) = TWO
      PGS(3) = FOUR
C
      PGS(4) = FOUR
C
      PGS(5)=ONE
```

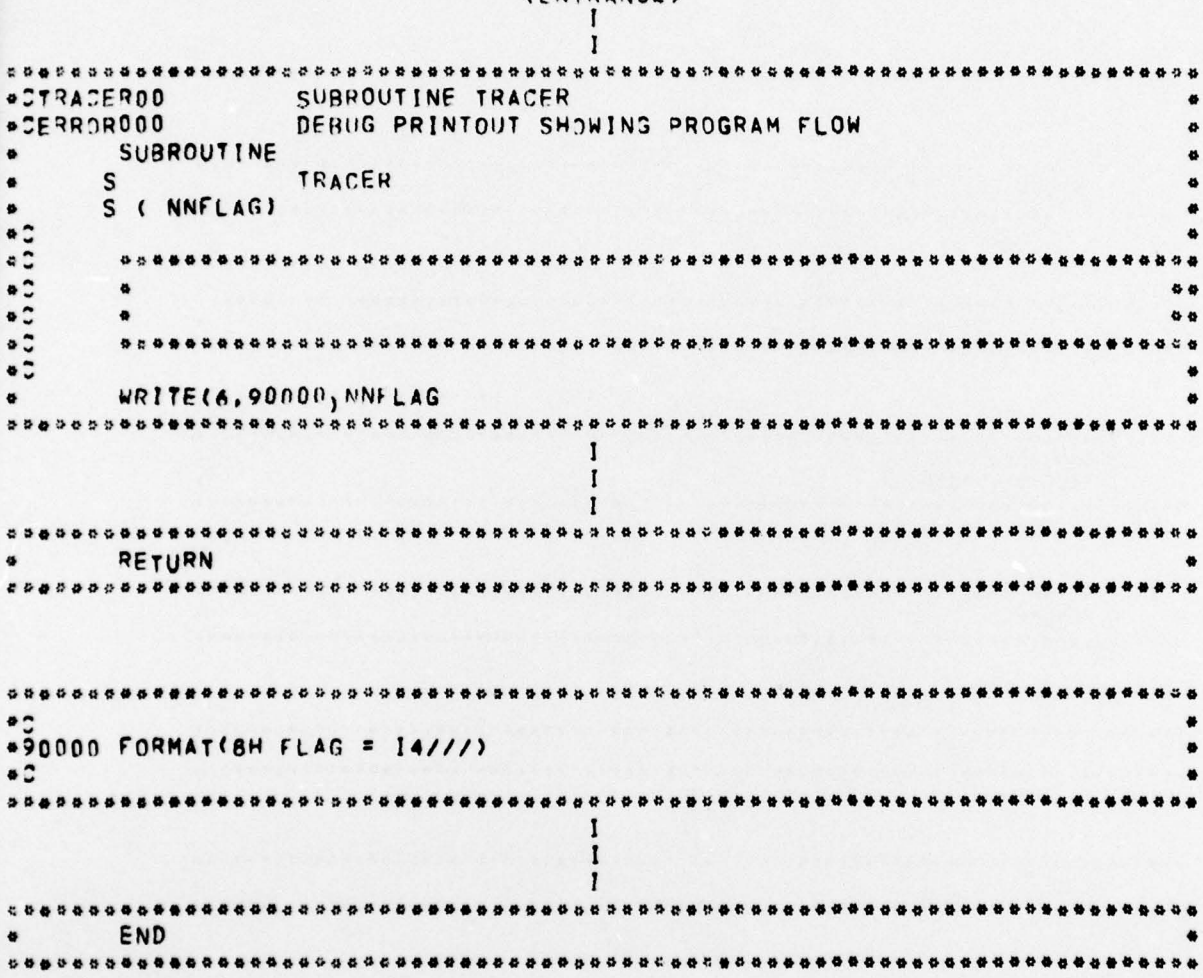
```
      RETURN
```

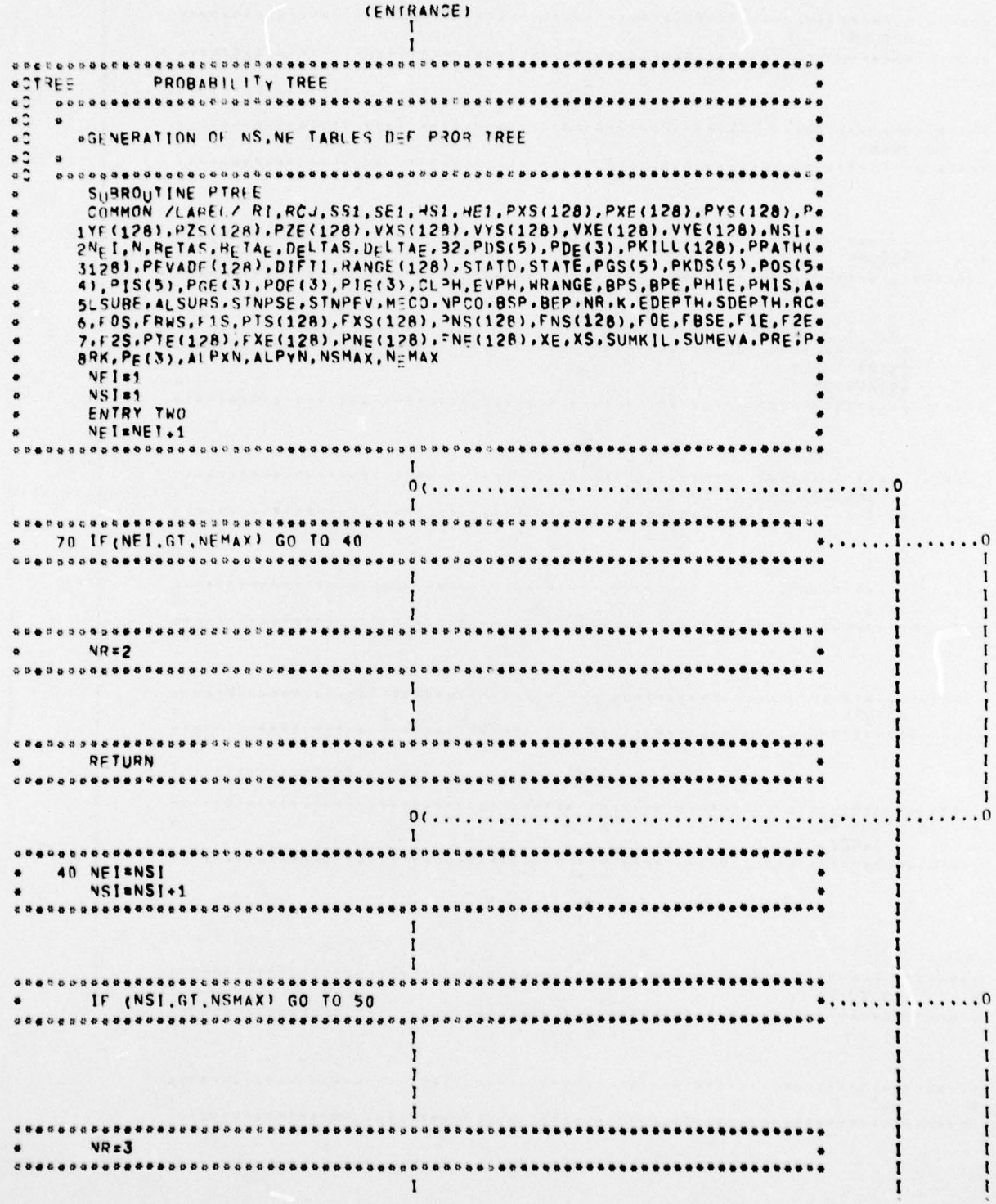
```
C*
```

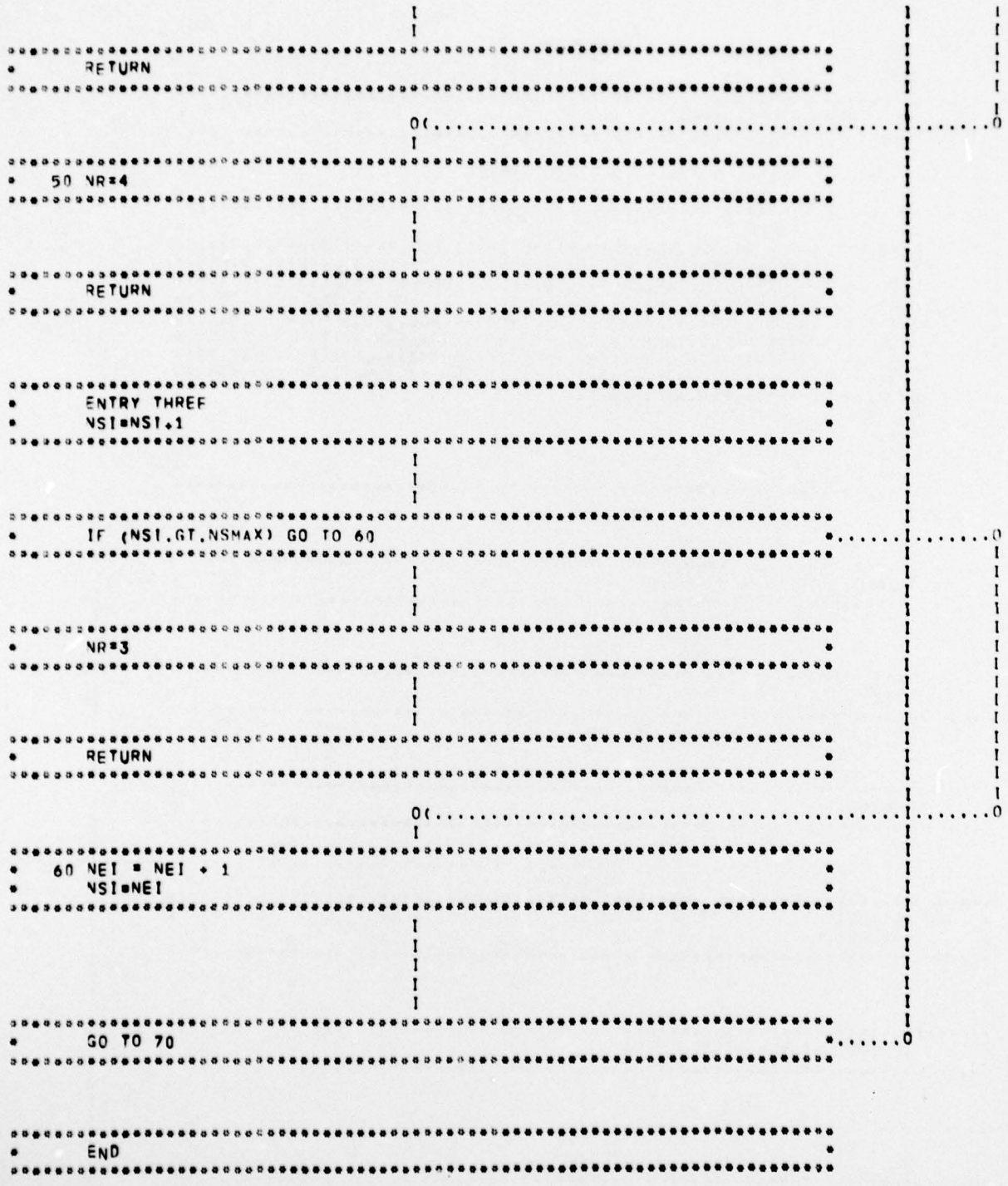
```
      END
```


(ENTRANCE)
C TNLG1000 FUNCTION TNLG10
C CONVERTS DUMMY VALUE TO DB
FUNCTION TNLG10(DUMMY1)
COMPUTES DB VALUE USING ALGORITHM OF 10*LOG(VALUE)
IF(DUMMY1 .LE. 1.0E-35)
GO TO 10
IF(DUMMY1 .GE. 1.0E+35)
GO TO 20
TNLG10 = 10.0*ALOG10(DUMMY1)
RETURN
C
10 CONTINUE
TNLG10 = -350.0

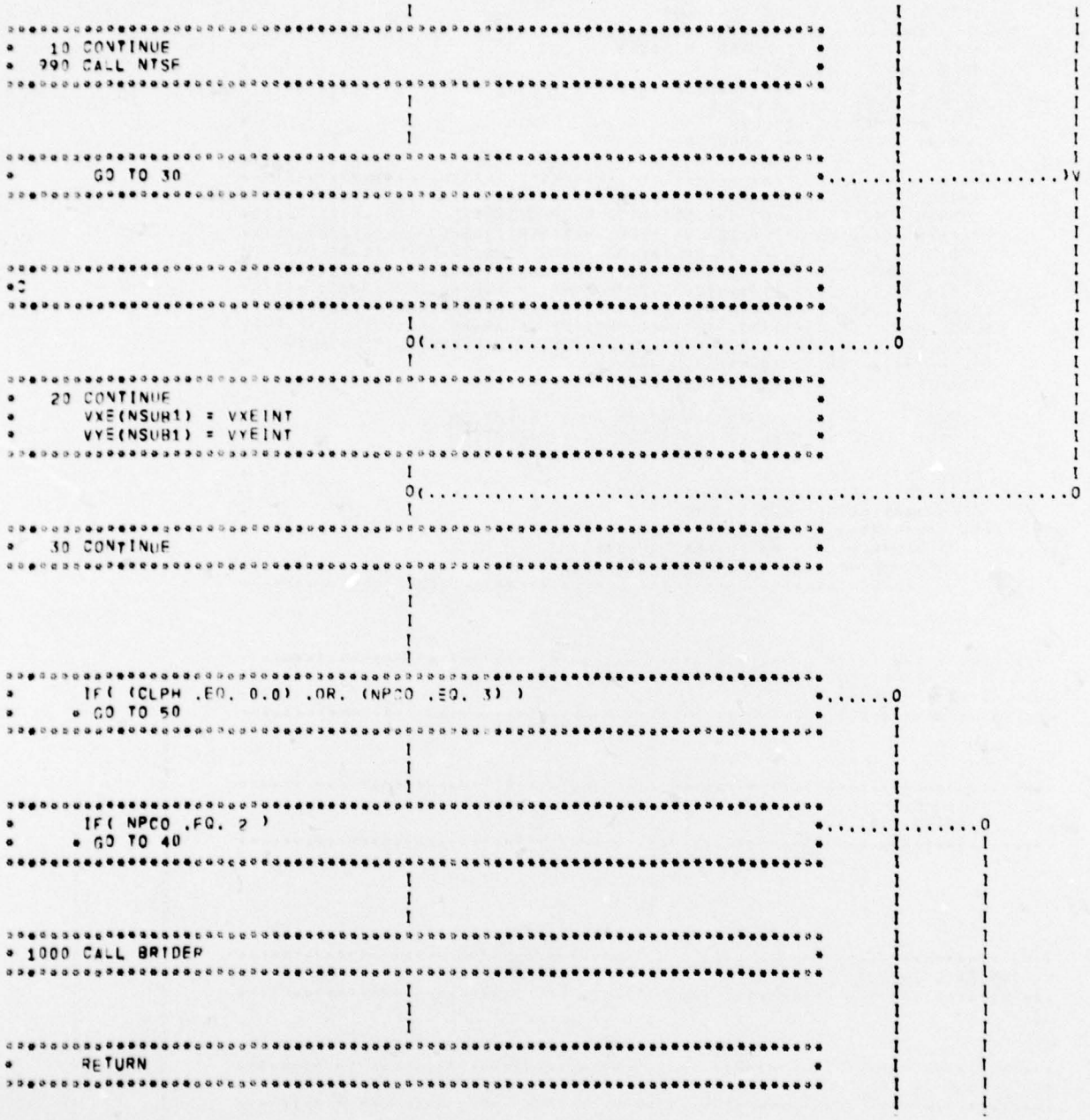

RETURN
C
20 CONTINUE
TNLG10 = +350.0
RETURN
C
END

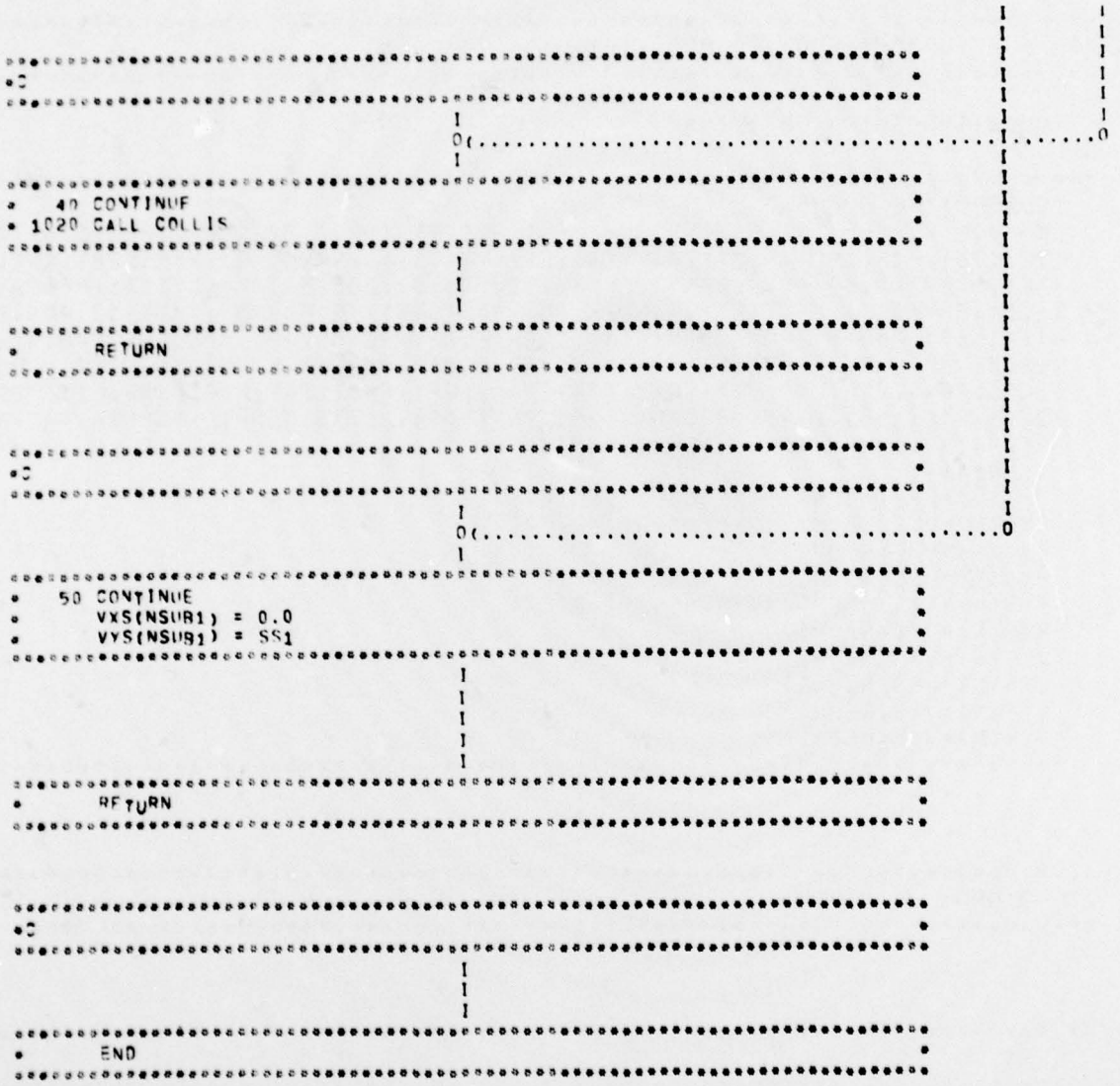

(ENTRANCE)
C TRACER00 SUBROUTINE TRACER
C ERROR000 DEBUG PRINTOUT SHOWING PROGRAM FLOW
SUBROUTINE
S TRACER
S (NNFLAG)
WRITE(6,90000) NNFLAG
RETURN
90000 FORMAT(8H FLAG = I4///)
END


(ENTRANCE)
CTREE PROBABILITY TREE
GENERATION OF NS,NE TABLES DEF PROB TREE
SUBROUTINE PTREE
COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),PYE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,ALSUBE,ALSURS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC,FOS,FRWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBSE,F1E,F2E,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,PBRK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX
NEI=1
NSI=1
ENTRY TWO
NEI=NEI+1
70 IF(NEI.GT.NEMAX) GO TO 40
NR=2
RETURN
40 NEI=NSI
NSI=NSI+1
IF (NSI.GT.NSMAX) GO TO 50
NR=3


RETURN
50 NR=4
RETURN
ENTRY THREF
NSI=NSI+1
IF (NSI.GT.NSMAX) GO TO 60
NR=3
RETURN
60 NEI = NEI + 1
NSI=NEI
GO TO 70
END


(ENTRANCE)
UPDAT UPDATE (MAIN SUB)
UPDATING SHIP MOTION TO NEXT TIME INTERVAL
MECO = ESCAPE COURSE OPTIONS
NPCO =PURSUIT COURSE OPTIONS
MECO =1 IMPLIES INVERSE B RIDER
MECO =2 IMPLIES NTSE
MECO =3 IMPLIES SAME COURSE
NPCO =1 IMPLIES B RIDER
NPCO =2 IMPLIES COLLIS
NPCO =3 IMPLIES SAME COURSE
SUBROUTINE UPDATE
COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P
1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NS1,
2VE1,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(
3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5
4),PIS(5),PGE(3),PDE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A
5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC
6,FDS,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FDE,FBSE,F1E,F2E
7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P
8BK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX
COMMON / INTSPD / VXEINT, VYEINT
VXEINT = INITIAL EVADER SPEED IN X-DIRECTION (K
VYEINT = INITIAL EVADER SPEED IN Y-DIRECTION (K
NSUB1=N-1
DIFPY=PYE(NSUB1)-PYS(NSUB1)
DIFPX=PXE(NSUB1)-PXS(NSUB1)
PESMAG=SQRT(DIFPX**2+DIFPY**2)
ALPXN=(PXE(NSUB1)-PXS(NSUB1))/PESMAG
ALPYN=(PYE(NSUB1)-PYS(NSUB1))/PESMAG
IF((EVPH .EQ. 0.0) .OR. (MECO .EQ. 3))
GO TO 20
IF(MECO .EQ. 2)
GO TO 10
240 CALL INVBRI
GO TO 30
0
0
0
0
0
0

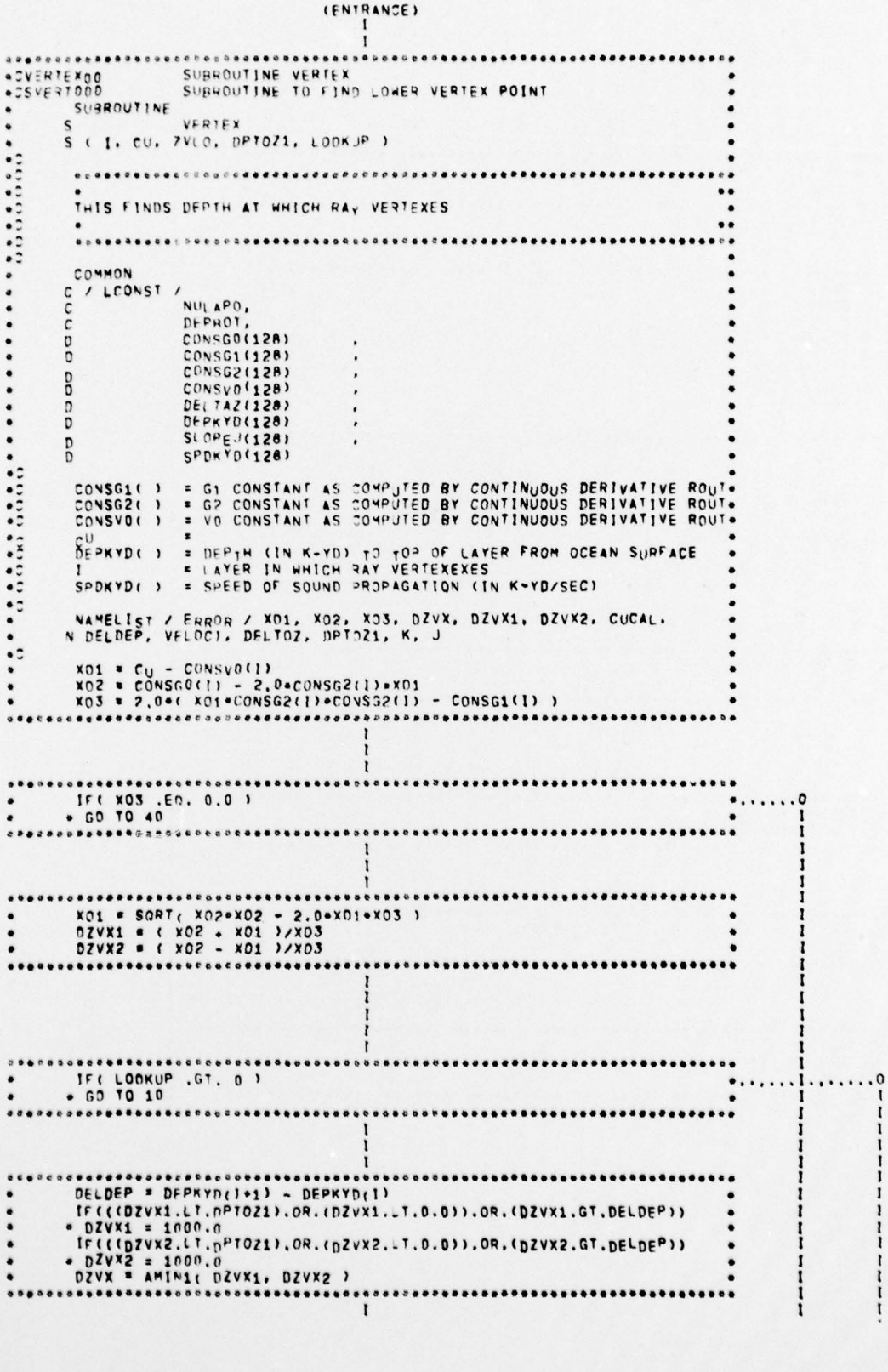

10 CONTINUE
990 CALL NTSF
GO TO 30
C
20 CONTINUE
VXE(NSUB1) = VXEINT
VYE(NSUB1) = VYEINT
30 CONTINUE
IF((CLPH .EQ. 0.0) .OR. (NPCO .EQ. 3))
GO TO 50
IF(NPCO .EQ. 2)
GO TO 40
1000 CALL BRIDER
RETURN

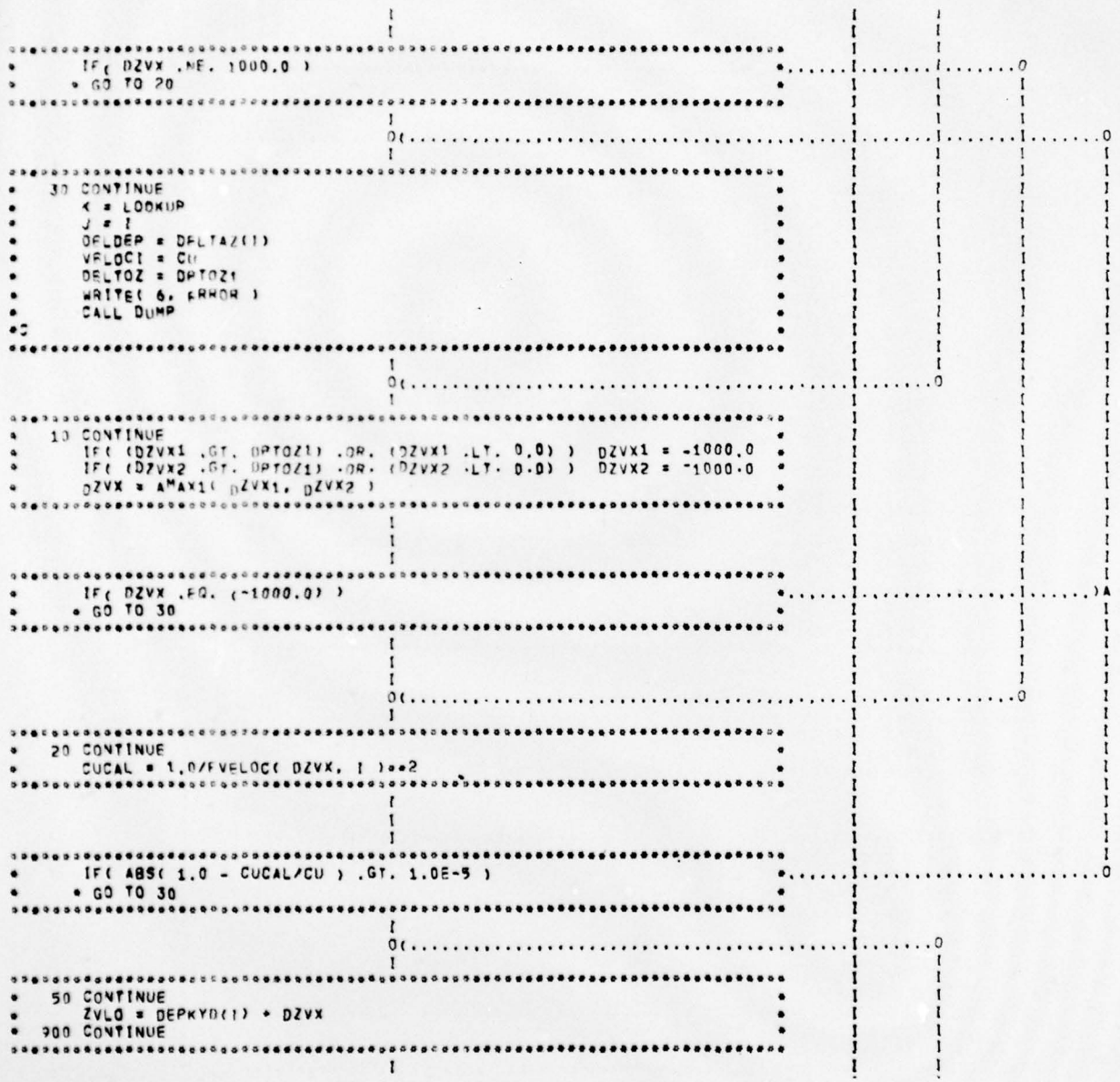

C
40 CONTINUE
1020 CALL COLLIS
RETURN
C
50 CONTINUE
VXS(NSUB1) = 0.0
VYS(NSUB1) = SS1
RETURN
C
END

```
                                    (ENTRANCE)
                                        I
                                        I
*****************************************************************************
*CUPPOS      UPPOS -UPDATES POSITIONS                                         *
*C  ***************************************************************************
*C  *                                                                        *
*C    *UPDATING SHIPS POSITION                                              *
*C  *                                                                        *
*C  ***************************************************************************
*       SUBROUTINE UPPOS                                                    *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS,FRWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                   *
*       NSUB1=N-1                                                          *
*       PXE(N)=PXE(NSUB1)+VXE(NSUB1)*DIFTI                                 *
*       PYE(N)=PYE(NSUB1)+VYE(NSUB1)*DIFTI                                 *
*       PZE(N)=PZE(NSUB1)                                                  *
*       PXS(N)=PXS(NSUB1)+VXS(NSUB1)*DIFTI                                 *
*       PYS(N)=PYS(NSUB1)+VYS(NSUB1)*DIFTI                                 *
*       PZS(N)=PZS(NSUB1)                                                  *
*       VXS(N)=VXS(NSUB1)                                                  *
*       VYS(N)=VYS(NSUB1)                                                  *
*       VXE(N)=VXE(NSUB1)                                                  *
*       VYE(N)=VYE(NSUB1)                                                  *
*****************************************************************************
                                        I
                                        I
                                        I
*****************************************************************************
*    25 RETURN                                                              *
*****************************************************************************



*****************************************************************************
*       END                                                                 *
*****************************************************************************
```


(ENTRANCE)
CVERTEX00 SUBROUTINE VERTEX
CSVERT000 SUBROUTINE TO FIND LOWER VERTEX POINT
SUBROUTINE
S VERTEX
S (I, CU, ZVLO, DPTOZ1, LOOKUP)
THIS FINDS DEPTH AT WHICH RAY VERTEXES
COMMON
C / LCONST /
C NULAPO,
C DEPROT,
D CONSG0(128) ,
D CONSG1(128) ,
D CONSG2(128) ,
D CONSV0(128) ,
D DELTAZ(128) ,
D DEPKYD(128) ,
D SLOPEJ(128) ,
D SPDKYD(128)
CONSG1( ) = G1 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT
CONSG2( ) = G2 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT
CONSV0( ) = V0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT
CU =
DEPKYD( ) = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE
I = LAYER IN WHICH RAY VERTEXEXES
SPDKYD( ) = SPEED OF SOUND PROPAGATION (IN K-YD/SEC)
NAMELIST / ERROR / XO1, XO2, XO3, DZVX, DZVX1, DZVX2, CUCAL,
N DELDEP, VELOCI, DELTOZ, DPTOZ1, K, J
XO1 = CU - CONSV0(I)
XO2 = CONSG0(I) - 2.0*CONSG2(I)*XO1
XO3 = 2.0*(XO1*CONSG2(I)*CONSG2(I) - CONSG1(I))
IF(XO3 .EQ. 0.0)
GO TO 40
XO1 = SQRT(XO2*XO2 - 2.0*XO1*XO3)
DZVX1 = (XO2 + XO1)/XO3
DZVX2 = (XO2 - XO1)/XO3
IF(LOOKUP .GT. 0)
GO TO 10
DELDEP = DEPKYD(I+1) - DEPKYD(I)
IF(((DZVX1.LT.DPTOZ1).OR.(DZVX1.LT.0.0)).OR.(DZVX1.GT.DELDEP))
DZVX1 = 1000.0
IF(((DZVX2.LT.DPTOZ1).OR.(DZVX2.LT.0.0)).OR.(DZVX2.GT.DELDEP))
DZVX2 = 1000.0
DZVX = AMIN1(DZVX1, DZVX2)


IF(DZVX .NE. 1000.0)
GO TO 20
30 CONTINUE
K = LOOKUP
J = I
DELDEP = DELTAZ(I)
VELOCI = CU
DELTOZ = DPTOZ1
WRITE(6, ERROR)
CALL DUMP
10 CONTINUE
IF((DZVX1 .GT. DPTOZ1) .OR. (DZVX1 .LT. 0.0)) DZVX1 = -1000.0
IF((DZVX2 .GT. DPTOZ1) .OR. (DZVX2 .LT. 0.0)) DZVX2 = -1000.0
DZVX = AMAX1(DZVX1, DZVX2)
IF(DZVX .EQ. (-1000.0))
GO TO 30
20 CONTINUE
CUCAL = 1.0/FVELOC(DZVX, I)**2
IF(ABS(1.0 - CUCAL/CU) .GT. 1.0E-5)
GO TO 30
50 CONTINUE
ZVLO = DEPKYD(I) + DZVX
900 CONTINUE


RETURN
C
40 CONTINUE
DZVX = X01/X02
GO TO 50
C
END