MITNE-195

DESCRIPTION OF THE COMPUTER CODE 2DTD

by

R.A. Shober

November 1976

Massachusetts Institute of Technology
Department of Nuclear Engineering
Cambridge, Massachusetts 02139

Electric Power Research Institute Report

DESCRIPTION OF THE COMPUTER CODE 2DTD

by

R.A. Shober

November, 1976

Massachusetts Institute of Technology
Department of Nuclear Engineering
Cambridge, Massachusetts 02139

Electric Power Research Institute Report

I. Introduction

The digital computer program 2DTD was written to test the flat leakage method developed in Ref. 1. It solves the static and transient diffusion equations in two spatial dimensions and two groups. The program uses an approximate analytical method to solve the neutron diffusion equation. The 2DTD program was written entirely in single precision on the IBM 370/168 at the M.I.T. Information Processing Center.

The 2DTD program always begins by solving the static, two-group diffusion equations for the eigenvalue and eigenfunction. After the steady state calculation has been completed, a transient solution may be calculated based on this initial condition. Initial criticality is insured by dividing $\nu\Sigma_f$ in every region by the eigenvalue λ.

2DTD requires that the reactor configuration be input as a regular array of rectangular assemblies. The borders of the reactor do not have to be rectangular, however. 2DTD allows the use of albedo boundary conditions on any border.

During a transient, 2DTD will edit the total reactor power and region powers integrated over user-defined regions. The code also edits the time required for various parts of the computation.

II. The Problem To Be Solved

The input to 2DTD describes an array of rectangular cells, each of which may contain a different composition. A map which defines the composition numbers is input into 2DTD. The composition numbers are defined by positive integers. The basic problem grid is the grid upon which the discrete problem will be formulated.

Around the outside of the composition map, one extra level of boxes must exist. This extra level is used to indicate the boundary condition at that edge of the reactor.

The 2DTD program has three symmetry options:

i) Full core

ii) Half core

iii) Quarter core

In the full-core option, only albedo boundary conditions can be applied. For the half-core option, the symmetry boundary is located along the left edge, and albedo conditions must be applied along all other boundaries. For the quarter-core option, the symmetry boundaries are located along the left and bottom edges of the problem; the other edges can only have albedo boundaries. The albedos are indicated by a negative integar in the material composition map. Each negative integer (each albedo set) has a set of both x-directed and y-directed albedos. If a problem has, for example, five albedo sets, these albedos sets are numbered -1 to -5. The symmetry options are defined by the input flags IBCL (controls the left boundary condition), and IBCB (controls the bottom boundary condition). If IBCL=1, the left boundary condition is symmetry; if IBCB=2, an albedo boundary condition is imposed. The same values hold for IBCB.

For each set of x-directed or y-directed albedos, the albedos are written

$$\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}$$

Therefore a zero flux boundary is easily imposed by setting all the α terms to zero. The zero flux is then imposed at the edge of the mesh rectangle.

We now give some examples of composition maps for various symmetry and albedo options:

1) This problem is quarter-core symmetric with only one albedo set

```
-1   -1   -1   -1   -1   -1
-1    1    2    1    2   -1
-1    2    1    2    1   -1
-1    1    2    1    2   -1
-1    2    1    2    1   -1
-1   -1   -1   -1   -1   -1
```

IBCL=IBCB=1

2) This problem is half-core symmetric with four different albedo sets:

```
-1   -1   -2   -3   -4   -4
-1    3    2    2    1   -4
-1    3    2    2    1   -4
-1    3    2    1    1   -4
-1    3    2    1    1   -4
-1    3    2    1    1   -4
-1    3    2    1    1   -4
-1    3    2    2    1   -4
-1    3    2    2    1   -4
-1   -1   -2   -3   -4   -4
```

IBCL=1, IBCB=2

We note that albedo set #1 is applied at the top and bottom of the first column; set #2 at top and bottom of the second column; set #3 at top and bottom of the third column; and set #4 at the top, bottom and right of the fourth column.

3) This quarter-core problem has a jagged boundary, and three albedo sets.

```
-1  -1  -1   0   0   0   0   0   0
-1   2   1  -4  -4   0   0   0   0
-1   1   2   1   2  -3   0   0   0
-1   2   1   2   1   2  -3   0   0
-1   1   2   1   2   1   2  -2   0
-1   2   1   2   1   2   1  -2   0
-1   1   2   1   2   1   2   1  -1
-1   2   1   2   1   2   1   2  -1
-1  -1  -1  -1  -1  -1  -1  -1  -1
```

IBCL=IBCB=1

We note that a solution only exists for regions above which have an integer greater than zero. Note also that although the left boundary condition is specified by IBCL, there must be a row of boxes on the left (above written as all -1's). These -1 entries are required by the code, but ignored. However, the -1 entries above the first and second columns, however, indicate albedo set #1.

The above composition map is indexed by the variables ND1 (x direction) and ND2 (y direction). These variables run

$$1 \leq ND1 \leq ND1X$$
$$1 \leq ND2 \leq ND2X.$$

Since there are always two extra boxes per row and per column, the solution is computed over an array of boxes indexed NP1 (x direction) and NP2 (y direction), where

$$1 \leq NP1 \leq NP1X$$
$$1 \leq NP2 \leq NP2X$$

and

$$NP1X=ND1X-2$$
$$NP2X=ND2X-2$$

Therefore when ND1=2, we have NP1=1. So

NP1=ND1-1

NP2=ND2-1

The 2DTD program allows editing at every edit time to be done over regions which may be as fine as the basic material composition mesh, or may integrate over several of these regions. The edit regions also must comprise a regular array of rectangular cells.

After the initial conditions have been calculated, the solution is normalized such that the mean power density is equal to a certain input value. The power in region k, written as P_k, can be defined as:

$$P_k = \frac{\varepsilon \sum_{g=1}^{2} \int_{\text{Region } k} \Sigma_{fg'}(\underline{r})\phi_{g'}(\underline{r})dV}{\int_{\text{Region } k} dV}$$

where ε=energy conversion factor, watt-seconds/fission.

The mean reactor power P_m' is then

$$P_m = \frac{\sum_{k=1}^{K} P_k A_k}{\sum_{k=1}^{K} A_k}$$

where

$$A_k = \int_{\text{Region } k} dV$$

and K is the total number of regions.

The total reactor power P_t is

$$P_t = \sum_{k=1}^{K} P_k A_k$$

The user must input the value of P_m he wishes to initialize the solution to.

The other equations of the neutronics and thermal-hydraulics equations in 2DTD are given in Ref. 1.

III. Machine Requirements for 2DTD

As currently programmed, 2DTD will execute on any IBM 360 or 370 with OS. The data is read in on data cards, and output written on a line printer. The core requirements are problem dependent.

The 2DTD program uses the dynamic storage allocation technique. This eliminates the need for large arrays reserved by DIMENSION or COMMON statements. The data management scheme used here is similar to that used in the MEKIN program, however 2DTD does not use any disk storage. The result is that the user need only request as much storage by the REGION parameter as needed by the particular problem.

The general formula for core requirements of 2DTD is (in bytes) $CORE = 120\ K + 4C_s + 4C_t$ where

$$C_s = 68(NPXY) + 9(NP1X) + 5(NP2X) + 8(NCORX) + 8(NALB) + NEDX + NEDY + 12$$

$$C_t = 2(NDEL)(NPXY) + 3(NPXY)$$

The variables are defined:

NPXY = NP1X * NP2X
NALB = the number of albedo sets
NCORX = the number of different material compositions
NEDX = number of edit regions in the x direction
NEDY = number of edit regions in the y direction
NDEL = number of precursor groups
C_t = zero if no transient is performed, equal to above expression if transient is performed.

This CORE requirement should be the minimum number entered on the

REGION parameter.

IV. Input Description:

Card 1: NINNER, XFEFF, DIFSS, DIFTD (I5, 3E10.3)

NINNER - the number of cyclic Chebyshev iterations per outer iteration for the steady state only. If NINNER is entered as zero, the code computes the number of inners to use.

XKEFF - input approximate eigenvalue

DIFSS - steady state convergence criterion - change in pointwise fluxes from one outer to another

DIFTD - transient convergence criterion - expressed as an error reduction.

Recommend: NINNER=0, XKEFF=1.0; If no transient is to be performed DIFSS=10^{-4} or 10^{-5}, if transient is to be performed DIFSS=10^{-6}. DIFTD=.05 unless prompt critical, then = 10^{-4}.

Card 2: ND1X, ND2X, IBCL, IBCB, NALB, ITHFB (6I5)

ND1X = number of x-regions +2

ND2X = number of y-regions +2

IBCL = left boundary condition

=1 symmetry

=2 albedo

IBCB = bottom boundary condition

NALB = number of albedo sets

ITHFB = thermal feedback flag (0=NO, 1=YES).

Card 3: XNU, WPCC, EPSIL (3E10.3)

XNU = number of neutrons per fission

WPCC = initial mean power density, watts/cc

EPSIL = energy conversion factor (watt-sec/fission).

Card(s) 4: Composition Box Map

Let NBOX(ND1, ND2) be the composition box map. Read in the order of - from top to bottom. For example

```
ND2 = ND2X + 1
DØ 10 ND2P=1, ND2X
ND2 = ND2-1
10    READ(5,20)(NBØX(ND1,ND2), ND1=1,ND1X)
20    FØRMAT(12I6)
```

So we enter ND2X cards, each with ND1X entries.

Card 5: NCPX Format(I5)

NCPX = number of material compositions.

Card(s) 6: x-Directed Mesh Spacings

(XMESH(NP1), NP1=1, NP1X)(6E10.5)

Read in the mesh spacings in the x-direction from NP1=1 to NP1X.

Card(s) 7: y-Directed Mesh Spacings

(YMESH(NP2), NP2=1,NP2X)(6E10.5)

Read in the mesh spacings in the y-direction from NP2=1 to NP2X.

Card(s) 8: x-Directed Albedos

(α11(I),α12(I),α21(I),α22(I), I=1, NALB)(4E10.3)

Read in one albedo set (4 numbers) per card for NALB albedo sets. These are the x-directed albedos.

Card(s) 9: y-Directed Albedos

(α11(I),α12(I),α21(I),α22(I), I=1, NALB)(4E10.3)

Read in the y-directed albedos in the same format as the x-directed albedos.

Card(s) 10: Cross Sections

Read in the cross sections for composition 1 first, then 2, etc. For each composition read in 8 numbers:

D_1, Σ_{T1}, Σ_{r1}, $\nu\Sigma_{f1}$, D_2, Σ_{T2}, Σ_{r2}, $\nu\Sigma_{f2}$

FORMAT(8E10.4)

D_1,D_2 are two-group diffusion coefficients

Σ_{T1}, Σ_{T2} are two-group total cross sections. Note

$$\Sigma_{T1} = \Sigma_{a1} + \Sigma_{r1}.$$

$\nu\Sigma_{f1}$ and $\nu\Sigma_{f2}$ are two group fission cross sections times nu.

Σ_{r1} = scattering from group 1 to 2.

Σ_{r2} = 0. (must be entered as zero).

Card 11: NEDX, NEDY Format(2I5)

NEDX = number of edit regions in x-direction
NEDY = number of edit regions in y-direction

Card(s) 12: x-Directed Edit Boundaries
(IEDX(I), I=1, NEDX)(12I5)

Each region of integration is over region (IEDX(I)+1) to (IEDX(I+1)). Therefore if NP1=1 to 10 and we desire to integrate over NP1 regions (1 and 2), (3 and 4), (5 and 6), (7 and 8), (9 and 10); then enter NEDX=5 and IEDX(I)=2,4,6,8,10.
Note the numbers input on IEDX correspond to the NP1 grid.

Card(s) 13: y-Directed Edit Boundaries

(IEDY(J), J=1, NEDY)(12I5)

Same input description as x-directed edit boundaries.

Card 14: ITRANS, NTD, NDEL, IEDTS(4I5)

ITRANS = transient flag

= 0 means no transient performed

= 1 means do transient

NTD = number of time domains ($\leq$5). A time domain is a region with a uniform time step.

NDEL = number of delayed precursor groups ($\geq$1)

IEDTS = edit flag. An edit is performed every IEDTS time steps.

If ITRANS=0, no more cards are read. The condition code is set to 1111.

Card 15: NUMS(I), DELT(I), I=1, NTD(I5,E10.5)

Read two numbers per card for NTD cards.

NUMS(I) = number of time steps in time domain I

DELT(I) = time step length (sec) in time domain I

Card 16: VIN(1), VIN(2)(2E10.3)

VIN(1) = $1/V_1$ fast inverse velocity

VIN(2) = $1/V_2$ thermal inverse velocity

Card 17: BETA(J), RLAM(J), J=1, NDEL(2E10.3)

BETA(J) = delayed neutron fraction for delayed family J.

RLAM(J) = delayed neutron decay constant for delayed family J.

(two numbers per card for NDEL cards).

Card 18: IPERT, NCØMP, TST, TFIN, SIG1, SIG2,(2I5,4E10.3)

IPERT = perturbation flag

= 1 step perturbation

= 2 ramp perturbation

NCØMP = composition number where perturbation is

TST = starting time for ramp or step

TFIN = final time for ramp

SIG1 = total change to Σ_{T1} for step or ramp(added to Σ_{T1})

SIG2 = total change to Σ_{T2} for step or ramp(added to Σ_{T2})

If ITHFB=0, no more cards read.

Card 19: ALFA, GAMMA, TREF(3E10.3)

ALFA = conversion factor for feedback, °K/cc

GAMMA = feedback constant, $°K^{-\frac{1}{2}}$

TREF = initial temperature (spatially independent), °K

Successful completion of a transient is followed by setting the condition code equal to 2222.

References

1) R. A. Shober, "Nonlinear Methods for Solving the Diffusion Equation", M.I.T. Department of Nuclear Engineering, Ph.D. Thesis, November, 1976.

APPENDIX A
LISTING OF COMPUTER PROGRAM 2DTD

```
C                                                                       MAIN0001
C                                                                       MAIN0002
C                                                                       MAIN0003
C       PROGRAM 2DTD    2 DIMENSIONAL, TIME DEPENDENT                   MAIN0004
C       I SOLVE THE NODAL EQUATIONS AFTER A SUBSTITUTION                MAIN0005
C       THIS VERSION HAS A CONSTANT LEAKAGE                             MAIN0006
C       THIS VERSION USES ONE CURRENT UPDATE PER OUTER ITERATION        MAIN0007
C       FLUXES ARE FOUND USING CYCLIC CHEBYSHEV                         MAIN0008
C       FULLY IMPLICIT IN TIME                                          MAIN0009
C       THIS VERSION HAS ADIABATIC THERMAL FEEDBACK (FAST DOPPLER)      MAIN0010
C                                                                       MAIN0011
C                                                                       MAIN0012
C       USES R. SIM'S SUBROUTINE CORE FOR DATA MANAGEMENT               MAIN0013
C                                                                       MAIN0014
C                                                                       MAIN0015
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      MAIN0016
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY  MAIN0017
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,MAIN0018
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,   MAIN0019
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                          MAIN0020
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    MAIN0021
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP, MAIN0022
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),  MAIN0023
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS   MAIN0024
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF            MAIN0025
C                                                                       MAIN0026
      COMMON DATA(1)                                                    MAIN0027
      DIMENSION AQ(1)                                                   MAIN0028
      EQUIVALENCE (WHX,AQ(1))                                           MAIN0029
      KWORD=1                                                           MAIN0030
      CALL GETCOR(KWORD)                                                MAIN0031
      CALL CLEAR                                                        MAIN0032
C                                                                       MAIN0033
C   ASSIGN BLOCK NAMES                                                  MAIN0034
C                                                                       MAIN0035
      DO 1 I=1,19                                                       MAIN0036
```

```
    1 AQ(I)=FLOAT(I)                                                    MAIN0037
C                                                                       MAIN0038
C   READ IN INPUT                                                       MAIN0039
C                                                                       MAIN0040
      CALL INPUTO                                                       MAIN0041
C                                                                       MAIN0042
C   CALCULATE N1,N2,N3,N4                                               MAIN0043
C                                                                       MAIN0044
      NN=NP1X-(NP1X/2)*2                                                MAIN0045
      IF(NN  .EQ.  0) GO TO 10                                          MAIN0046
      N3=NP1X-1                                                         MAIN0047
      N4=NP1X                                                           MAIN0048
      GO TO 11                                                          MAIN0049
   10 N3=NP1X                                                           MAIN0050
      N4=NP1X-1                                                         MAIN0051
   11 N2=N3                                                             MAIN0052
      N1=N4                                                             MAIN0053
C                                                                       MAIN0054
C   COMPUTE THE STEADY STATE SOLUTION                                   MAIN0055
C                                                                       MAIN0056
      CALL CALC                                                         MAIN0057
C                                                                       MAIN0058
C   COMPUTE THE TRANSIENT SOLUTION                                      MAIN0059
C                                                                       MAIN0060
      CALL TRANS                                                        MAIN0061
C                                                                       MAIN0062
      STOP 2222                                                         MAIN0063
C                                                                       MAIN0064
      END                                                               MAIN0065
```

```
      SUBROUTINE INPUTO                                                 INPT0001
C                                                                       INPT0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      INPT0003
     X       WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY   INPT0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX, INPT0005
     X       KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,    INPT0006
     X       KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                           INPT0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    INPT0008
     X       ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,  INPT0009
     X       TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   INPT0010
     X       VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS    INPT0011
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF            INPT0012
C                                                                       INPT0013
      COMMON         DATA(1)                                            INPT0014
      IZERO=0                                                           INPT0015
      IONE=1                                                            INPT0016
      ITWO=2                                                            INPT0017
      IFOUR=4                                                           INPT0018
      IN=5                                                              INPT0019
      IOUT=6                                                            INPT0020
      KTEMP=1                                                           INPT0021
C                                                                       INPT0022
C=====CARD TYPE 2                                                       INPT0023
C                                                                       INPT0024
      WRITE(IOUT,2001)                                                  INPT0025
      READ(IN,1010) NINNER,XKEFF,DIFSS,DIFTD                            INPT0026
      IF(NINNER  .EQ.  0) WRITE(IOUT,2006)                              INPT0027
      WRITE(IOUT,2005) NINNER,XKEFF,DIFSS,DIFTD                         INPT0028
      XKEFF=XKEFF*0.8                                                   INPT0029
C                                                                       INPT0030
C=====CARD TYPE 4                                                       INPT0031
C                                                                       INPT0032
      READ(IN,1020) ND1X,ND2X,IBCL,IBCB,NALB,ITHFB                      INPT0033
C                                                                       INPT0034
C                                                                       INPT0035
      WRITE(IOUT,2015)  ND1X,ND2X,IBCL,IBCB,NALB,ITHFB                  INPT0036
```

```
C                                                                       INPT0037
      READ(IN,1030) XNU,WPCC,EPSIL                                      INPT0038
C                                                                       INPT0039
      WRITE(IOUT,2020) XNU,WPCC,EPSIL                                   INPT0040
C                                                                       INPT0041
C=====READ REACTOR GEOMETRY DATA                                        INPT0042
C                                                                       INPT0043
      CALL IGEOM0                                                       INPT0044
C                                                                       INPT0045
C=====READ NEUTRONIC DATA                                               INPT0046
C                                                                       INPT0047
      CALL INEUT0                                                       INPT0048
C                                                                       INPT0049
C=====CONSTRUCT NEUTRONIC FINE MESH GEOMETRY DATA                       INPT0050
C                                                                       INPT0051
      CALL IFINE0                                                       INPT0052
C                                                                       INPT0053
C=====READ COMPOSITION DATA                                             INPT0054
C                                                                       INPT0055
      CALL ICOMP0                                                       INPT0056
C                                                                       INPT0057
C=====READ TRANSIENT DATA                                               INPT0058
C                                                                       INPT0059
      CALL ITRAN                                                        INPT0060
      RETURN                                                            INPT0061
C                                                                       INPT0062
C                                                                       INPT0063
C=====FORMATS                                                           INPT0064
C                                                                       INPT0065
 1010 FORMAT(I5,3E10.3)                                                 INPT0066
 1020 FORMAT(6I5)                                                       INPT0067
 1030 FORMAT(3E10.3)                                                    INPT0068
 2001 FORMAT(1H1,45X,'PROGRAM 2DTD OUTPUT',////)                        INPT0069
 2005 FORMAT(1H0,9X,'NO. OF INNERS PER OUTER IS ',I5,//,                INPT0070
     X  10X,'INITIAL EIGENVALUE ESTIMATE',E12.4,//,                     INPT0071
     X  10X,'STEADY STATE CONV. CRIT. ',E12.4,//,                       INPT0072
```

```
     X  10X,'TRANSIENT CONV. CRIT. ',E12.4)                                  INPT0073
 2006 FORMAT(1HO,9X,'CODE WILL COMPUTE NINNER FOR STEADY STATE')             INPT0074
 2015 FORMAT(1HO,9X,40HNO. OF HORIZONTAL (BOX) COORDINATES, X =,I5//         INPT0075
     1           10X,40HNO. OF HORIZONTAL (BOX) COORDINATES, Y =,I5//        INPT0076
     2           10X,'LEFT BOUNDARY CONDITION ',I5,//,                       INPT0077
     3           10X,'BOTTOM BOUNDARY CONDITION ',I5,//,                     INPT0078
     4           10X,'NUMBER OF ALBEDO SETS ',I5,//,                         INPT0079
     5           10X,'THERMAL FEEDBACK FLAG (0=NO, 1=YES) ',I5)              INPT0080
 2020 FORMAT(1HO,9X,'NUMBER OF NEUTRONS PER FISSION IS ',E12.4,//,           INPT0081
     X  10X,'INITIAL MEAN POWER LEVEL (W/CC) ',E12.4,//,                     INPT0082
     X  10X,'ENERGY CONVERSION FACTOR (WS/FISS) ',E12.4)                     INPT0083
C                                                                            INPT0084
      END                                                                    INPT0085
```

```
      SUBROUTINE IGEOMO                                                      IGOM0001
C                                                                            IGOM0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,           IGOM0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY       IGOM0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,     IGOM0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,       IGOM0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                              IGOM0007
      COMMON / FIXED / NINNER,NCUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,         IGOM0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,      IGOM0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),       IGOM0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS        IGOM0011
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF                 IGOM0012
C                                                                            IGOM0013
      COMMON         DATA(1)                                                 IGOM0014
      REAL*8 DDAT(1)                                                         IGOM0015
      INTEGER        IDAT(1)                                                 IGOM0016
      EQUIVALENCE    (DATA(1),DDAT(1),IDAT(1))                               IGOM0017
      IN=5                                                                   IGOM0018
      IOUT=6                                                                 IGOM0019
      IZERO=0                                                                IGOM0020
      IONE=1                                                                 IGOM0021
      ITWO=2                                                                 IGOM0022
C                                                                            IGOM0023
C==== CARDS TYPE G3                                                          IGOM0024
C                                                                            IGOM0025
      NP1X=ND1X-2                                                            IGOM0026
      NP2X=ND2X-2                                                            IGOM0027
      NPXY=NP1X*NP2X                                                         IGOM0028
      CALL ALOC(WNBOX,0,KNBOX,ND1X*ND2X)                                     IGOM0029
      KNBOX=KNBOX-1                                                          IGOM0030
      WRITE(IOUT,2017)                                                       IGOM0031
      ND2=ND2X+IONE                                                          IGOM0032
      DO 100 ND2P=1,ND2X                                                     IGOM0033
      ND2=ND2-IONE                                                           IGOM0034
      READ (IN,1030)          (IDAT(KNBOX+ND1+ND1X*(ND2-IONE)),ND1=1,ND1X)   IGOM0035
  100 WRITE(IOUT,2018)        (IDAT(KNBOX+ND1+ND1X*(ND2-IONE)),ND1=1,ND1X)   IGOM0036
```

```
      KNBOX=KNBOX+1                                                          IGOM0037
      RETURN                                                                 IGOM0038
C                                                                            IGOM0039
C==== FORMATS                                                                IGOM0040
C                                                                            IGOM0041
 1030 FORMAT(12I6)                                                           IGOM0042
C                                                                            IGOM0043
 2017 FORMAT(1H0,9X,'REACTOR BOX MAP (COMP. NUMBERS) VERSUS ND1,ND2',//)     IGOM0044
 2018 FORMAT(5X,30I4)                                                        IGOM0045
C                                                                            IGOM0046
      END                                                                    IGOM0047
```

```
      SUBROUTINE INEUTO                                                 INET0001
C                                                                       INET0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      INET0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY  INET0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX, INET0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,   INET0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                          INET0007
      COMMON / FIXED / NINNER,NCUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    INET0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP, INET0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   INET0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS   INET0011
C                                                                       INET0012
      COMMON         DATA(1)                                            INET0013
      REAL*8 DDAT(1)                                                    INET0014
      INTEGER        IDAT(1)                                            INET0015
      EQUIVALENCE    (DATA(1),DDAT(1),IDAT(1))                          INET0016
      IZERO=0                                                           INET0017
      IONE=1                                                            INET0018
      ITWO=2                                                            INET0019
      IN=5                                                              INET0020
      IOUT=6                                                            INET0021
C                                                                       INET0022
C=====CARD TYPE NO                                                      INET0023
C                                                                       INET0024
      READ(IN,1000) NCPX                                                INET0025
C                                                                       INET0026
      WRITE(IOUT,2002)  NCPX                                            INET0027
C                                                                       INET0028
C   RESERV SPACE AND READ IN MESH SPACINGS                              INET0029
C                                                                       INET0030
      CALL ALOC(WHX,0,KHX,NP1X)                                         INET0031
      CALL ALOC(WHY,0,KHY,NP2X)                                         INET0032
      KHX=KHX-1                                                         INET0033
      KHY=KHY-1                                                         INET0034
      READ(IN,1040)          (DATA(KHX+J),J=1,NP1X)                     INET0035
      READ(IN,1040)          (DATA(KHY+J),J=1,NP2X)                     INET0036
```

```
      WRITE(IOUT,2003)                                                  INET0037
      WRITE(IOUT,2005)     (DATA(KHX+J),J=1,NP1X)                       INET0038
      WRITE(IOUT,2004)                                                  INET0039
      WRITE(IOUT,2005)     (DATA(KHY+J),J=1,NP2X)                       INET0040
      KHX=KHX+1                                                         INET0041
      KHY=KHY+1                                                         INET0042
      CALL ALOC(WINTX,1,KISTRX,ND2X)                                    INET0043
      CALL ALOC(WINTX,2,KIENDX,ND2X)                                    INET0044
      CALL ALOC(WINTY,1,KISTRY,ND1X)                                    INET0045
      CALL ALOC(WINTY,2,KIENDY,ND1X)                                    INET0046
C   SET UP KISTRX AND KIENDX                                            INET0047
      ND2XX=ND2X-1                                                      INET0048
      DO 10 ND2=2,ND2XX                                                 INET0049
      DO 20 ND1=1,ND1X                                                  INET0050
      I=ND1                                                             INET0051
      K=IDAT(KNBOX+(ND2-1)*ND1X+ND1-1)                                  INET0052
      IF(K  .GT.  0) GO TO 25                                           INET0053
   20 CONTINUE                                                          INET0054
   25 IDAT(KISTRX+ND2-1)=I                                              INET0055
      DO 30 NDD1=1,ND1X                                                 INET0056
      ND1=ND1X+1-NDD1                                                   INET0057
      I=ND1                                                             INET0058
      K=IDAT(KNBOX+(ND2-1)*ND1X+ND1-1)                                  INET0059
      IF(K  .GT.  0) GO TO 35                                           INET0060
   30 CONTINUE                                                          INET0061
   35 IDAT(KIENDX+ND2-1)=I                                              INET0062
   10 CONTINUE                                                          INET0063
      ND1XX=ND1X-1                                                      INET0064
      DO 40 ND1=2,ND1XX                                                 INET0065
      DO 50 ND2=1,ND2X                                                  INET0066
      I=ND2                                                             INET0067
      K=IDAT(KNBOX+(ND2-1)*ND1X+ND1-1)                                  INET0068
      IF(K  .GT.  0) GO TO 55                                           INET0069
   50 CONTINUE                                                          INET0070
   55 IDAT(KISTRY+ND1-1)=I                                              INET0071
      DO 60 NDD2=1,ND2X                                                 INET0072
```

```
      ND2=ND2X+1-NDD2                                                    INET0073
      I=ND2                                                              INET0074
      K=IDAT(KNBOX+(ND2-1)*ND1X+ND1-1)                                   INET0075
      IF(K  .GT.  0) GO TO 65                                            INET0076
   60 CONTINUE                                                           INET0077
   65 IDAT(KIENDY+ND1-1)=I                                               INET0078
   40 CONTINUE                                                           INET0079
      WRITE(6,2006)                                                      INET0080
      DO 70 J=1,ND2X                                                     INET0081
   70 WRITE(6,2007) J,IDAT(KISTRX+J-1),IDAT(KIENDX+J-1)                  INET0082
      WRITE(6,2008)                                                      INET0083
      DO 80 J=1,ND1X                                                     INET0084
   80 WRITE(6,2007) J,IDAT(KISTRY+J-1),IDAT(KIENDY+J-1)                  INET0085
      CALL ALOC(WALBX,0,KALBX,NALB*4)                                    INET0086
      CALL ALOC(WALBY,0,KALBY,NALB*4)                                    INET0087
      KALBX=KALBX-1                                                      INET0088
      KALBY=KALBY-1                                                      INET0089
      WRITE(6,2010)                                                      INET0090
      DO 22 I=1,NALB                                                     INET0091
      J=KALBX+(I-1)*4                                                    INET0092
      READ(5,1070)  (DATA(J+K),K=1,4)                                    INET0093
   22 WRITE(6,2011)  I,(DATA(J+K),K=1,4)                                 INET0094
      WRITE(6,2012)                                                      INET0095
      DO 21 I=1,NALB                                                     INET0096
      J=KALBY+(I-1)*4                                                    INET0097
      READ(5,1070)  (DATA(J+K),K=1,4)                                    INET0098
   21 WRITE(6,2011)  I,(DATA(J+K),K=1,4)                                 INET0099
      KALBX=KALBX+1                                                      INET0100
      KALBY=KALBY+1                                                      INET0101
      RETURN                                                             INET0102
C                                                                        INET0103
C==== FORMATS                                                            INET0104
C                                                                        INET0105
 1000 FORMAT(2I5)                                                        INET0106
 1040 FORMAT(6E10.5)                                                     INET0107
 1050 FORMAT(6E10.5)                                                     INET0108
```

```
1070 FORMAT(4E10.3)                                                    INET0109
2002 FORMAT(1H0,9X,'NUMBER OF COMPOSITIONS IS ',I5)                    INET0110
2003 FORMAT(1H0,9X,'X DIRECTION MESH SPACINGS',/)                      INET0111
2004 FORMAT(1H0,9X,'Y DIRECTION MESH SPACINGS',/)                      INET0112
2005 FORMAT(15X,6E12.4)                                                INET0113
2006 FORMAT(1H0,9X,'ND2   ISTART   IEND',//)                           INET0114
2007 FORMAT(11X,I3,5X,I3,5X,I3)                                        INET0115
2008 FORMAT(1H0,9X,'ND1   ISTART   IEND',//)                           INET0116
2010 FORMAT(1H0,9X,'X DIRECTED ALBEDOS',/,10X,'NALB    ALBEDOS',//)    INET0117
2011 FORMAT(11X,I3,3X,4E12.4)                                          INET0118
2012 FORMAT(1H0,9X,'Y DIRECTED ALBEDOS',/,10X,'NALB    ALBEDOS',//)    INET0119
     END                                                               INET0120
```

```
      SUBROUTINE IFINEC                                                 IFIN0001
C                                                                       IFIN0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      IFIN0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY  IFIN0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,IFIN0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,   IFIN0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                          IFIN0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    IFIN0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP, IFIN0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),  IFIN0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS   IFIN0011
C                                                                       IFIN0012
C                                                                       IFIN0013
C   ALLOCATE BLOCKS FOR DATA STORAGE                                    IFIN0014
C                                                                       IFIN0015
      NEED=NP1X*(NP2X+2)*2                                              IFIN0016
      CALL ALOC(WFLUX,0,KFLUX,NEED)                                     IFIN0017
      KFLUX=KFLUX+NP1X+NP1X                                             IFIN0018
      NEED=NPXY*52                                                      IFIN0019
      CALL ALOC(WMAT,0,KMAT,NEED)                                       IFIN0020
      NEED=NPXY*2                                                       IFIN0021
      CALL ALOC(WFISS,0,KFISS,NEED)                                     IFIN0022
      NEED=NPXY*4                                                       IFIN0023
      CALL ALOC(WRHS,0,KRHS,NEED)                                       IFIN0024
      NEED=NPXY                                                         IFIN0025
      CALL ALOC(WS,1,KS1,NEED)                                          IFIN0026
      CALL ALOC(WS,2,KS2,NEED)                                          IFIN0027
      CALL ALOC(WS,3,KS3,NEED)                                          IFIN0028
      NEED=NPXY*4                                                       IFIN0029
      CALL ALOC(WB,0,KB,NEED)                                           IFIN0030
      FEED=50.0                                                         IFIN0031
      CALL ALOC(FEED,1,KSA1,NPXY)                                       IFIN0032
      RETURN                                                            IFIN0033
C                                                                       IFIN0034
      END                                                               IFIN0035
```

```
      SUBROUTINE ICCMPO                                                 ICMP0001
C                                                                       ICMP0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      ICMP0003
     X        WFISS,WFHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY  ICMP0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,ICMP0005
     X        KFLUX,KMAT,KFISS,KFHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,   ICMP0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                          ICMP0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    ICMP0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP, ICMP0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   ICMP0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS   ICMP0011
C                                                                       ICMP0012
      COMMON         DATA(1)                                            ICMP0013
      REAL*8 DDAT(1)                                                    ICMP0014
      INTEGER        IDAT(1)                                            ICMP0015
      EQUIVALENCE    (DATA(1),DDAT(1),IDAT(1))                          ICMP0016
C                                                                       ICMP0017
C==== SET SCALARS                                                       ICMP0018
C                                                                       ICMP0019
      IZERO=0                                                           ICMP0020
      IONE=1                                                            ICMP0021
      ITWO=2                                                            ICMP0022
      IN=5                                                              ICMP0023
      IOUT=6                                                            ICMP0024
C                                                                       ICMP0025
C==== CARDS TYPE N8                                                     ICMP0026
C                                                                       ICMP0027
      NCORX=8                                                           ICMP0028
      CALL ALOC(WCORR,0,KCORR,NCORX*NCPX)                               ICMP0029
      KCORR=KCORR-1                                                     ICMP0030
      DO 200 NCP=1,NCPX                                                 ICMP0031
      READ(IN,2000) (DATA(KCORR+NCOR+NCORX*(NCP-1)),NCOR=1,NCORX)       ICMP0032
  200 WRITE(IOUT,1002)NCP,(DATA(KCORR+NCOR+NCORX*(NCP-IONE)),           ICMP0033
     X     NCOR=1,NCORX)                                                ICMP0034
      KCORR=KCORR+1                                                     ICMP0035
C                                                                       ICMP0036
```

```
C    READ IN EDIT STUFF                                                    ICMP0037
C                                                                          ICMP0038
       READ(IN,2001) NEDX,NEDY                                             ICMP0039
       CALL ALOC(WEDX,0,KEDX,NEDX)                                         ICMP0040
       CALL ALOC(WEDY,0,KEDY,NEDY)                                         ICMP0041
       KEDX=KEDX-1                                                         ICMP0042
       KEDY=KEDY-1                                                         ICMP0043
       WRITE(IOUT,1003)                                                    ICMP0044
       READ(IN,2001)        (IDAT(J+KEDX),J=1,NEDX)                        ICMP0045
       WRITE(IOUT,1004)     (IDAT(J+KEDX),J=1,NEDX)                        ICMP0046
       READ(IN,2001)        (IDAT(J+KEDY),J=1,NEDY)                        ICMP0047
       WRITE(IOUT,1004)     (IDAT(J+KEDY),J=1,NEDY)                        ICMP0048
       KEDX=KEDX+1                                                         ICMP0049
       KEDY=KEDY+1                                                         ICMP0050
       RETURN                                                              ICMP0051
C                                                                          ICMP0052
 1002 FORMAT(28H           COMPOSITION   DATA/11X,11(1H-),3X,90(1H-)/      ICMP0053
     X      (11X,I11,3X,1P4E15.5) / (25X,4E15.5))                         ICMP0054
 1003 FORMAT(1H0,9X,'X AND Y DIRECTED EDIT BOUNDS',//)                     ICMP0055
 1004 FORMAT(10X,12I5)                                                     ICMP0056
C                                                                          ICMP0057
 2000 FORMAT(8E10.4)                                                       ICMP0058
 2001 FORMAT(12I5)                                                         ICMP0059
C                                                                          ICMP0060
       END                                                                 ICMP0061
```

```
      SUBROUTINE ITRAN                                                  ITRN0001
C                                                                       ITRN0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      ITRN0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY  ITRN0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,ITRN0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,   ITRN0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                          ITRN0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    ITRN0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP, ITRN0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),  ITRN0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS   ITRN0011
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF            ITRN0012
C                                                                       ITRN0013
      COMMON A(1)                                                       ITRN0014
      DIMENSION IDAT(1)                                                 ITRN0015
      EQUIVALENCE (A(1),IDAT(1))                                        ITRN0016
      READ(5,1010) ITRANS,NTD,NDEL,IEDTS                                ITRN0017
      IF(ITRANS  .EQ.  0) GO TO 500                                     ITRN0018
      WRITE(6,2010) NTD,NDEL,IEDTS                                      ITRN0019
      DO 10 I=1,NTD                                                     ITRN0020
      READ(5,1020) NUMS(I),DELT(I)                                      ITRN0021
   10 WRITE(6,2020) I,NUMS(I),DELT(I)                                   ITRN0022
      READ(5,1030) VIN(1),VIN(2)                                        ITRN0023
      WRITE(6,2030) VIN(1),VIN(2)                                       ITRN0024
      WRITE(6,2040)                                                     ITRN0025
      DO 20 J=1,NDEL                                                    ITRN0026
      READ(5,1030)      BETA(J),RLAM(J)                                 ITRN0027
   20 WRITE(6,2050)  J,BETA(J),RLAM(J)                                  ITRN0028
      READ(5,1040) IPERT,NCOMP,TST,TFIN,SIG1,SIG2                       ITRN0029
      IF(IPERT  .EQ.  1) WRITE(6,2060)                                  ITRN0030
      IF(IPERT  .EQ.  2) WRITE(6,2070)                                  ITRN0031
      WRITE(6,2080) NCOMP,TST,TFIN,SIG1,SIG2                            ITRN0032
      NEED=NDEL*NPXY                                                    ITRN0033
      CALL ALOC(WPREC,0,KPREC,NEED)                                     ITRN0034
      NEED=NPXY*2                                                       ITRN0035
      CALL ALOC(WOMGP,0,KOMGP,NEED)                                     ITRN0036
```

```
      NEED=NDEL*NPXY                                                    ITRN0037
      CALL ALOC(WOMGD,0,KOMGD,NEED)                                     ITRN0038
      IF(ITHFB  .EQ.  0) RETURN                                         ITRN0039
      FEED=50.0                                                         ITRN0040
      READ(5,1050) ALFA,GAMMA,TREF                                      ITRN0041
      WRITE(6,2090) ALFA,GAMMA,TREF                                     ITRN0042
      CALL ALOC(FEED,2,KTEMP,NPXY)                                      ITRN0043
      DO 50 NP=1,NPXY                                                   ITRN0044
   50 A(KTEMP+NP-1)=TREF                                                ITRN0045
      RETURN                                                            ITRN0046
  500 WRITE(6,2005)                                                     ITRN0047
      RETURN                                                            ITRN0048
C                                                                       ITRN0049
 1010 FORMAT(4I5)                                                       ITRN0050
 1020 FORMAT(I5,E10.5)                                                  ITRN0051
 1030 FORMAT(2E10.3)                                                    ITRN0052
 1040 FORMAT(2I5,4E10.3)                                                ITRN0053
 1050 FORMAT(3E10.3)                                                    ITRN0054
C                                                                       ITRN0055
 2005 FORMAT(1H0,9X,'DO NOT DO TRANSIENT CALCULATION')                  ITRN0056
 2010 FORMAT(1H1, 9X,'NUMBER OF TIME DOMAINS ',I5,/,10X,                ITRN0057
     X  'NUMBER OF DELAYED FAMILIES',I5,/,10X,                          ITRN0058
     X  'NUMBER OF TIME STEPS PER EDIT',I5,//,                          ITRN0059
     X  10X,'TIME DOMAIN   NO. OF STEPS   DELTA T',//)                  ITRN0060
 2020 FORMAT(15X,I2,9X,I4,4X,E12.4)                                     ITRN0061
 2030 FORMAT(1H0,9X,'FAST INVERSE VELOCITY',E12.4,/,10X,'SLOW INVERSE VE ITRN0062
     XLOCITY',E12.4)                                                    ITRN0063
 2040 FORMAT(1H0,9X,'DEL. FAM.       BETA        LAMDA',//)             ITRN0064
 2050 FORMAT(12X,I2,3X,2E12.4)                                          ITRN0065
 2060 FORMAT(1H0,9X,'STEP PERTURBATION')                                ITRN0066
 2070 FORMAT(1H0,9X,'RAMP PERTURBATION')                                ITRN0067
 2080 FORMAT(1H0,9X,'PERTURBATION IN COMPOSITION',I5,/,10X,             ITRN0068
     X   'STARTING TIME ',E12.4,/,10X,'FINISH TIME ',E12.4,/,10X,       ITRN0069
     X   'CHANGE IN SIG-T-1 ',E12.4,/,10X,                              ITRN0070
     X   'CHANGE IN SIG-T-2 ',E12.4)                                    ITRN0071
 2090 FORMAT(1H0,9X,'CONVERSION FACTOR FOR FEEDBACK (K/CC) ',E12.4,//,  ITRN0072
```

```
     X   10X,'FEEDBACK CONSTANT ',E12.4,//,                              ITRN0073
     X   10X,'REFERENCE TEMPERATURE (K) ',E12.4)                        ITRN0074
C                                                                       ITRN0075
      END                                                               ITRN0076
```

```
      SUBROUTINE CALC                                                    CALC0001
C                                                                        CALC0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,       CALC0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY   CALC0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX, CALC0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,    CALC0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                           CALC0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,     CALC0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,  CALC0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   CALC0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS    CALC0011
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF             CALC0012
C                                                                        CALC0013
      COMMON A(1)                                                        CALC0014
      DIMENSION IDAT(1)                                                  CALC0015
      EQUIVALENCE (A(1),IDAT(1))                                         CALC0016
      LOGICAL JSW,JJJ,JLAST                                              CALC0017
      JSW=.FALSE.                                                        CALC0018
      JJJ=.FALSE.                                                        CALC0019
      JLAST=.FALSE.                                                      CALC0020
      RHO=0.9                                                            CALC0021
      ITS=0                                                              CALC0022
      TIME=0.0                                                           CALC0023
      CALL TFER(A(KISTRX),A(KIENDX),NP1X,NP2X,A(KNBOX),A(KCORR),         CALC0024
     X  A(KSA1),ND1X)                                                    CALC0025
      CALL CPUT0                                                         CALC0026
      CALL MATRX                                                         CALC0027
      CALL CPUT(CTIME)                                                   CALC0028
      CSTOR=CTIME                                                        CALC0029
      WRITE(6,201) CTIME                                                 CALC0030
  201 FORMAT(1H0,5X,'TIME FOR MATRX IS ',E12.4)                          CALC0031
C                                                                        CALC0032
C   INITIALIZE FLUXES                                                    CALC0033
C                                                                        CALC0034
      DO 20 NP2=1,NP2X                                                   CALC0035
      IS=IDAT(KISTRX+NP2)                                                CALC0036
```

```
      IE=IDAT(KIENDX+NP2)                                                CALC0037
      DO 21 ND1=IS,IE                                                    CALC0038
      NP1=ND1-1                                                          CALC0039
      NPP=(NP2-1)*NP1X+NP1                                               CALC0040
      DO 23 NG=1,2                                                       CALC0041
   23 A(KFLUX-1+(NPP-1)*2+NG)=1.0                                        CALC0042
   21 CONTINUE                                                           CALC0043
   20 CONTINUE                                                           CALC0044
      CALL CPUTO                                                         CALC0045
      CALL DORPES                                                        CALC0046
      CALL CPUT(CTIME)                                                   CALC0047
      WRITE(6,202) CTIME                                                 CALC0048
  202 FORMAT(1HO,5X,'TIME FOR DORPES IS ',E12.4)                         CALC0049
      CALL CPUTO                                                         CALC0050
      DO 170 NP=1,NPXY                                                   CALC0051
      NP1=(NP-1)*2                                                       CALC0052
      NP2=NP1+1                                                          CALC0053
      X=A(KFLUX+NP1)*A(KFISS+NP1)+A(KFLUX+NP2)*A(KFISS+NP2)              CALC0054
  170 A(KS1+NP-1)=X                                                      CALC0055
      DO 110 NP=1,NPXY                                                   CALC0056
      NPP=NP-1                                                           CALC0057
      A(KS2+NPP)=A(KS1+NPP)                                              CALC0058
  110 A(KS3+NPP)=A(KS1+NPP)                                              CALC0059
      IT=0                                                               CALC0060
 1000 IT=IT+1                                                            CALC0061
C                                                                        CALC0062
C   CALCULATE RIGHT HAND SIDE                                            CALC0063
C                                                                        CALC0064
      DO 120 NP=1,NPXY                                                   CALC0065
      NP1=(NP-1)*4                                                       CALC0066
      NP2=NP1+1                                                          CALC0067
      X=A(KS1+NP-1)                                                      CALC0068
      A(KRHS+NP1)=X / XKEFF                                              CALC0069
  120 A(KRHS+NP2)=0.0                                                    CALC0070
      CALL INNERS(JSW)                                                   CALC0071
      DO 160 NP=1,NPXY                                                   CALC0072
```

```
      NP1=(NP-1)*2                                                      CALC0073
      NP2=NP1+1                                                         CALC0074
      X=A(KFLUX+NP1)*A(KFISS+NP1)+A(KFLUX+NP2)*A(KFISS+NP2)             CALC0075
  160 A(KS1+NP-1)=X                                                     CALC0076
      CALL OUTERS(IT,A(KS1),A(KS2),A(KS3),NPXY,IRET,DIFSS,              CALC0077
     X   DIFLAM,XKEFF)                                                  CALC0078
      IF(IRET  .EQ.  1) GO TO 2000                                      CALC0079
      IF(IT  .GE.  200) GO TO 2000                                      CALC0080
      IF(JJJ) GO TO 1000                                                CALC0081
      IF((DIFLAM/10.0)  .GT.  DIFSS) GO TO 1000                         CALC0082
      CALL MATRX                                                        CALC0083
      WRITE(6,175)                                                      CALC0084
  175 FORMAT(1H0)                                                       CALC0085
      JJJ=.TRUE.                                                        CALC0086
      GO TO 1000                                                        CALC0087
 2000 IF(JLAST) GO TO 2001                                              CALC0088
      CALL MATRX                                                        CALC0089
      JLAST=.TRUE.                                                      CALC0090
      GO TO 1000                                                        CALC0091
 2001 CALL CPUT(CTIME)                                                  CALC0092
      CTIME=CTIME+CSTOR                                                 CALC0093
      WRITE(6,203) CTIME                                                CALC0094
  203 FORMAT(1H0,5X,'TIME TO DO OUTER ITERATIONS ',E12.4)               CALC0095
      CALL EDIT(NP1X,NP2X,NEDX,NEDY,ITS,NPXY,A(KFLUX),                  CALC0096
     X   A(KFISS),A(KS3),A(KEDX),A(KEDY),A(KS2),A(KB),TIME,            CALC0097
     X  A(KHX),A(KHY),A(KTEMP))                                         CALC0098
      RETURN                                                            CALC0099
C                                                                       CALC0100
      END                                                               CALC0101
```

```
      SUBROUTINE MATRX                                                  MTRX0001
C                                                                       MTRX0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      MTRX0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY  MTRX0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,MTRX0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,   MTRX0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                          MTRX0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    MTRX0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP, MTRX0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   MTRX0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS   MTRX0011
C                                                                       MTRX0012
      DIMENSION AA(4,75),BB(4,75),COEF(4,75)                            MTRX0013
      COMMON A(1)                                                       MTRX0014
C                                                                       MTRX0015
      CALL XSWEEP(A(KMAT),A(KFISS),A(KNBOX),A(KHX),A(KHY),A(KCORR),     MTRX0016
     X  AA,BB,COEF,A(KALBX),A(KOMGP),A(KOMGD),A(KSA1))                  MTRX0017
C                                                                       MTRX0018
      CALL YSWEEP(A(KMAT),A(KFISS),A(KNBOX),A(KHX),A(KHY),A(KCORR),     MTRX0019
     X  AA,BB,COEF,A(KALBY),A(KOMGP),A(KOMGD),A(KSA1))                  MTRX0020
C                                                                       MTRX0021
      RETURN                                                            MTRX0022
      END                                                               MTRX0023
```

```
      SUBROUTINE XSWEEP(XMAT,FISS,NBOX,HX,HY,CORR,A,B,C,ALB,OMP,OMD,SA1)        XSWP0001
C                                                                              XSWP0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,             XSWP0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY          XSWP0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,       XSWP0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,          XSWP0006
     X        KALBX,KALBY,KEDI,KEDY,KTEMP,KSA1                                 XSWP0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,           XSWP0008
     X        ND2X,NP1X,NP2X,NEL,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,          XSWP0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),          XSWP0010
     X        VIN(2),DELT(5),TIMES(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS          XSWP0011
C                                                                              XSWP0012
      DIMENSION XMAT(52,1),FISS(2,1),HX(1),HY(1),NBOX(ND1X,1),SA1(1),          XSWP0013
     X  CORR(8,1),A(4,1),B(4,1),C(4,1),ALB(4,1),OMP(2,1),OMD(NDEL,1)           XSWP0014
      DIMENSION D(4,1)                                                         XSWP0015
      COMMON IDAT(1)                                                           XSWP0016
      ZERO=0.0                                                                 XSWP0017
      HALF=0.5                                                                 XSWP0018
      RL=XKEFF                                                                 XSWP0019
      IF(ITS  .EQ.  0) GO TO 1                                                 XSWP0020
      SMM=1.0-BTOT                                                             XSWP0021
      DO 2 ND=1,NDEL                                                           XSWP0022
    2 SMM=SMM+(BETA(ND)*RLAM(ND))/(DTI+RLAM(ND))                               XSWP0023
C                                                                              XSWP0024
C   COMPUTE FISS VECTOR                                                        XSWP0025
C                                                                              XSWP0026
    1 DO 51 NP2=1,NP2X                                                         XSWP0027
      ND2=NP2+1                                                                XSWP0028
      IS=IDAT(KISTRX+NP2)                                                      XSWP0029
      IE=IDAT(KIENDX+NP2)                                                      XSWP0030
      DO 51 ND1=IS,IE                                                          XSWP0031
      NP1=ND1-1                                                                XSWP0032
      NPP=(NP2-1)*NP1X+NP1                                                     XSWP0033
      K=NBOX(ND1,ND2)                                                          XSWP0034
      FISS(1,NPP)=HX(NP1)*HY(NP2)*CORR(4,K)                                    XSWP0035
      FISS(2,NPP)=HX(NP1)*HY(NP2)*CORR(8,K)                                    XSWP0036
```

```
   51 CONTINUE                                                          XSWP0037
   15 DO 40 NP2=1,NP2X                                                  XSWP0038
      ND2=NP2+1                                                         XSWP0039
      IS=IDAT(KISTRX+NP2)                                               XSWP0040
      IE=IDAT(KIENDX+NP2)                                               XSWP0041
      DO 52 ND1=IS,IE                                                   XSWP0042
      NP1=ND1-1                                                         XSWP0043
      NPP=(NP2-1)*NP1X+NP1                                              XSWP0044
      I=NP1                                                             XSWP0045
      K=NBOX(ND1,ND2)                                                   XSWP0046
      D1=CORR(1,K)                                                      XSWP0047
      D2=CORR(5,K)                                                      XSWP0048
      SR=CORR(3,K)                                                      XSWP0049
      IF(ITS  .EQ.  0) GO TO 18                                         XSWP0050
      SUM=1.0-BTOT                                                      XSWP0051
      DO 17 ND=1,NDEL                                                   XSWP0052
   17 SUM=SUM+(BETA(ND)*RLAM(ND))/(OMD(ND,NPP)+RLAM(ND))                XSWP0053
      S1=SA1(NPP)-SUM*CORR(4,K)/RL+VIN(1)*OMP(1,NPP)                    XSWP0054
      S2=CORR(6,K)+OMP(2,NPP)*VIN(2)                                    XSWP0055
      V2=SUM*CORR(8,K)                                                  XSWP0056
      GO TO 19                                                          XSWP0057
   18 S1=SA1(NPP)-CORR(4,K)/RL                                          XSWP0058
      S2=CORR(6,K)                                                      XSWP0059
      V2=CORR(8,K)                                                      XSWP0060
   19 C1=S1/D1+S2/D2                                                    XSWP0061
      C2=(HALF*(S2/D2-S1/D1))**2  +  (V2*SR)/(RL*D1*D2)                 XSWP0062
      C3=SQRT(C2)                                                       XSWP0063
      XKS=-HALF*C1+C3                                                   XSWP0064
      XMS=HALF*C1+C3                                                    XSWP0065
      IF(V2  .GT.  ZERO) GO TO 81                                       XSWP0066
      XS=1.0E+20                                                        XSWP0067
      GO TO 82                                                          XSWP0068
   81 XS=SR/(-D2*XMS+S2)                                                XSWP0069
   82 XR=SR/(D2*XKS+S2)                                                 XSWP0070
      IF(XKS  .LE.  ZERO) GO TO 61                                      XSWP0071
C                                                                       XSWP0072
```

```
C    KAPPA IS REAL                                                          XSWP0073
C                                                                           XSWP0074
       X=SQRT(XKS)                                                          XSWP0075
       XK=X                                                                 XSWP0076
       IF(X   .LT.   1.0E-07) GO TO 31                                      XSWP0077
       XCSC=X/SIN(X*HX(I))                                                  XSWP0078
       XTAN=TAN(X*HX(I)*HALF)/X                                             XSWP0079
       XCDK=XCSC/XKS                                                        XSWP0080
       GO TO 62                                                             XSWP0081
    31 XCSC=1.0/HX(I)                                                       XSWP0082
       XTAN=HX(I)*HALF                                                      XSWP0083
       XCDK=XCSC/XKS                                                        XSWP0084
       GO TO 62                                                             XSWP0085
C                                                                           XSWP0086
C    KAPPA IS IMAGINARY                                                     XSWP0087
C                                                                           XSWP0088
    61 X=SQRT(ABS(XKS))                                                     XSWP0089
       XK=X                                                                 XSWP0090
       IF(X   .LT.   1.0E-07) GO TO 31                                      XSWP0091
       XCSC=X/SINH(X*HX(I))                                                 XSWP0092
       XTAN=TANH(X*HX(I)*HALF)/X                                            XSWP0093
       XCDK=-XCSC/(X*X)                                                     XSWP0094
    62 CONTINUE                                                             XSWP0095
       XM=SQRT(XMS)                                                         XSWP0096
       XCSCH=XM/SINH(XM*HX(I))                                              XSWP0097
       XTANH=TANH(XM*HX(I)*HALF)/XM                                         XSWP0098
       XCHDK=XCSCH/(XM*XM)                                                  XSWP0099
C                                                                           XSWP0100
C    NOW FIND ENTRIES OF A AND B MATRICES   AND C MATRX ALSO                XSWP0101
C                                                                           XSWP0102
       IF(V2   .GT.   ZERO) GO TO 91                                        XSWP0103
C                                                                           XSWP0104
C    WE ARE IN A REFLECTOR REGION - USE THE EXACT FORMULAS                  XSWP0105
C                                                                           XSWP0106
       A(1,I)=XCSC                                                          XSWP0107
       A(2,I)=ZERO                                                          XSWP0108
```

```
      A(3,I)=XR*XCSC          - XP*XCSCH                                    XSWP0109
      A(4,I)=XCSCH                                                          XSWP0110
      B(1,I)=XTAN/D1                                                        XSWP0111
      B(2,I)=ZERO                                                           XSWP0112
      B(3,I)=(XP/D1)*(XTAN-XTANH)                                           XSWP0113
      B(4,I)=XTANH/D2                                                       XSWP0114
      HI=HX(I)                                                              XSWP0115
      DET=S1*S2                                                             XSWP0116
      C(1,I)=(XCDK*HI)/D1+S2/DET                                            XSWP0117
      C(2,I)=ZERO                                                           XSWP0118
      C(3,I)=(XR*XCDK*HI+XP*XCHDK*HI)/D1+SR/DET                             XSWP0119
      C(4,I)=-(XCHDK*HI)/D2+S1/DET                                          XSWP0120
      GO TO 52                                                              XSWP0121
   91 XMULT=1.0/(XS-XR)                                                     XSWP0122
      A(1,I)=(    XS*XCSC-XP*XCSCH)*XMULT                                   XSWP0123
      A(2,I)=(-    XCSC+XCSCH)*XMULT                                        XSWP0124
      A(3,I)=XS*XR*XMULT*(XCSC-XCSCH)                                       XSWP0125
      A(4,I)=(-    XR*XCSC+XS*XCSCH)*XMULT                                  XSWP0126
      XMM=XS-XR                                                             XSWP0127
      XM1=XTAN/(D1*XMM)                                                     XSWP0128
      XM2=XTAN/(D2*XMM)                                                     XSWP0129
      XM3=XTANH/(D1*XMM)                                                    XSWP0130
      XM4=XTANH/(D2*XMM)                                                    XSWP0131
      B(1,I)=XS*XM1-XR*XM3                                                  XSWP0132
      B(2,I)=-XM2+XM4                                                       XSWP0133
      B(3,I)=XR*XS*(XM1-XM3                                                 XSWP0134
      B(4,I)=-XR*XM2+XS*XM4                                                 XSWP0135
      HI=HX(I)                                                              XSWP0136
      T1=-XCDK                                                              XSWP0137
      T2=XCHDK                                                              XSWP0138
      T3=XR*T1                                                              XSWP0139
      T4=XS*T2                                                              XSWP0140
      T5=-XS/D1                                                             XSWP0141
      T6=1.0/D2                                                             XSWP0142
      T7=XR/D1                                                              XSWP0143
      T8=-T6                                                                XSWP0144
```

```
      C(1,I)=T1*T5+T2*T7                                                XSWP0145
      C(2,I)=T1*T6+T2*T8                                                XSWP0146
      C(3,I)=T3*T5+T4*T7                                                XSWP0147
      C(4,I)=T3*T6+T4*T8                                                XSWP0148
      TI=HI/(XS-XR)                                                     XSWP0149
      DO 92 K=1,4                                                       XSWP0150
   92 C(K,I)=C(K,I)*TI                                                  XSWP0151
      DET=S1*S2-(V2*SR)/RL                                              XSWP0152
      C(1,I)=C(1,I)+S2/DET                                              XSWP0153
      C(2,I)=C(2,I)+V2/(RL*DET)                                         XSWP0154
      C(3,I)=C(3,I)+SR/DET                                              XSWP0155
      C(4,I)=C(4,I)+S1/DET                                              XSWP0156
   52 CONTINUE                                                          XSWP0157
C                                                                       XSWP0158
C    NOW FIND COUPLING COEFFICIENTS                                     XSWP0159
C    LEFT BOUNDARY IS ZERO J OR ALBEDO                                  XSWP0160
C    RIGHT BOUNDARY IS ALBEDO                                           XSWP0161
C                                                                       XSWP0162
      DO 55 ND1=IS,IE                                                   XSWP0163
      NP1=ND1-1                                                         XSWP0164
      NPP=(NP2-1)*NP1X+NP1                                              XSWP0165
      I=NP1                                                             XSWP0166
C                                                                       XSWP0167
C    COMPUTE COUPLING TO LEFT                                           XSWP0168
C                                                                       XSWP0169
      IF(ND1  .NE.  IS) GO TO 85                                        XSWP0170
C                                                                       XSWP0171
C    IBCL=1  SYMMETRY BOUNDARY ON LEFT                                  XSWP0172
C    IBCL=2  ALBEDO BOUNDARY ON LEFT                                    XSWP0173
C                                                                       XSWP0174
      IF(IBCL  .EQ.  2) GO TO 86                                        XSWP0175
      XL1=ZERO                                                          XSWP0176
      XL2=ZERO                                                          XSWP0177
      XL3=ZERO                                                          XSWP0178
      XL4=ZERO                                                          XSWP0179
      GO TO 56                                                          XSWP0180
```

```
   96 KLEFT=IABS(NBOX(ND1-1,ND2))                                    XSWP0181
      DO 11 J4=1,4                                                   XSWP0182
   11 D(J4,1)=B(J4,1)+ALB(J4,KLEFT)                                  XSWP0183
      CALL BINV(D,1,0,XL1,XL2,XL3,XL4)                               XSWP0184
      GO TO 56                                                       XSWP0185
   85 CALL BINV(B,I-1,I,X1,X2,X3,X4)                                 XSWP0186
      XL1=X1                                                         XSWP0187
      XL2=X2                                                         XSWP0188
      XL3=X3                                                         XSWP0189
      XL4=X4                                                         XSWP0190
      F1=X1                                                          XSWP0191
      F2=X2                                                          XSWP0192
      F3=X3                                                          XSWP0193
      F4=X4                                                          XSWP0194
      CALL BMULT(X1,X2,X3,X4,A,I-1)                                  XSWP0195
      CALL BMULT(F1,F2,F3,F4,C,I-1)                                  XSWP0196
      HH=HX(I-1)*HY(NP2)                                             XSWP0197
      XMAT(5,NPP)=-HH*X1                                             XSWP0198
      XMAT(6,NPP)=-HH*X2                                             XSWP0199
      XMAT(7,NPP)=-HH*X3                                             XSWP0200
      XMAT(8,NPP)=-HH*X4                                             XSWP0201
      XMAT(29,NPP)=F1                                                XSWP0202
      XMAT(30,NPP)=F2                                                XSWP0203
      XMAT(31,NPP)=F3                                                XSWP0204
      XMAT(32,NPP)=F4                                                XSWP0205
C                                                                    XSWP0206
C    COMPUTE CENTER POINT COUPLING                                   XSWP0207
C                                                                    XSWP0208
   56 CONTINUE                                                       XSWP0209
      J1=I                                                           XSWP0210
      J2=I+1                                                         XSWP0211
      IF(ND1  .NE.  IE) GO TO 12                                     XSWP0212
      KRIGHT=IABS(NBOX(ND1+1,ND2))                                   XSWP0213
      DO 13 J4=1,4                                                   XSWP0214
   13 B(J4,I)=B(J4,I)+ALB(J4,KRIGHT)                                 XSWP0215
      J2=0                                                           XSWP0216
```

```
12 CALL BINV(B,J1,J2,X1,X2,X3,X4)                                              XSWP0217
   XR1=X1                                                                      XSWP0218
   XR2=X2                                                                      XSWP0219
   XR3=X3                                                                      XSWP0220
   XR4=X4                                                                      XSWP0221
   X1=XR1+XL1                                                                  XSWP0222
   X2=XR2+XL2                                                                  XSWP0223
   X3=XR3+XL3                                                                  XSWP0224
   X4=XR4+XL4                                                                  XSWP0225
   F1=X1                                                                       XSWP0226
   F2=X2                                                                       XSWP0227
   F3=X3                                                                       XSWP0228
   F4=X4                                                                       XSWP0229
   CALL BMULT(X1,X2,X3,X4,A,I)                                                 XSWP0230
   CALL BMULT(F1,F2,F3,F4,C,I)                                                 XSWP0231
   HH=HX(I)*HY(NP2)                                                            XSWP0232
   K=NBOX(ND1,ND2)                                                             XSWP0233
   ST1=SA1(NPP)*HH                                                             XSWP0234
   ST2=CORR(6,K)*HH                                                            XSWP0235
   SR1=CORR(3,K)*HH                                                            XSWP0236
   XMAT(1,NPP)=ST1+HH*X1                                                       XSWP0237
   XMAT(2,NPP)=ZERO+HH*X2                                                      XSWP0238
   XMAT(3,NPP)=-SR1+HH*X3                                                      XSWP0239
   XMAT(4,NPP)=ST2+HH*X4                                                       XSWP0240
   XMAT(21,NPP)=HH*X1                                                          XSWP0241
   XMAT(22,NPP)=HH*X2                                                          XSWP0242
   XMAT(23,NPP)=HH*X3                                                          XSWP0243
   XMAT(24,NPP)=HH*X4                                                          XSWP0244
   XMAT(33,NPP)=-F1                                                            XSWP0245
   XMAT(34,NPP)=-F2                                                            XSWP0246
   XMAT(35,NPP)=-F3                                                            XSWP0247
   XMAT(36,NPP)=-F4                                                            XSWP0248
   IF(ITS  .EQ.  0) GO TO 57                                                   XSWP0249
   XMAT(1,NPP)=XMAT(1,NPP)+DTI*VIN(1)*HH-SMM*HH*CORR(4,K)/RL                   XSWP0250
   XMAT(2,NPP)=XMAT(2,NPP)-SMM*HH*CORR(8,K)/RL                                 XSWP0251
   XMAT(4,NPP)=XMAT(4,NPP)+DTI*VIN(2)*HH                                       XSWP0252
```

```
C                                                                       XSWP0253
C    COMPUTE PIGHT COUPLING                                             XSWP0254
C                                                                       XSWP0255
   57 IF(ND1  .EQ.  IE) GO TO 55                                        XSWP0256
      F1=XR1                                                            XSWP0257
      F2=XR2                                                            XSWP0258
      F3=XR3                                                            XSWP0259
      F4=XR4                                                            XSWP0260
      CALL BMULT(XR1,XR2,XR3,XR4,A,I+1)                                 XSWP0261
      CALL BMULT(F1,F2,F3,F4,C,I+1)                                     XSWP0262
      HH=HX(I+1)*HY(NP2)                                                XSWP0263
      XMAT(9,NPP)=-HH*XR1                                               XSWP0264
      XMAT(10,NPP)=-HH*XR2                                              XSWP0265
      XMAT(11,NPP)=-HH*XR3                                              XSWP0266
      XMAT(12,NPP)=-HH*XR4                                              XSWP0267
      XMAT(37,NPP)=F1                                                   XSWP0268
      XMAT(38,NPP)=F2                                                   XSWP0269
      XMAT(39,NPP)=F3                                                   XSWP0270
      XMAT(40,NPP)=F4                                                   XSWP0271
   55 CONTINUE                                                          XSWP0272
   40 CONTINUE                                                          XSWP0273
      RETURN                                                            XSWP0274
      END                                                               XSWP0275
```

```
      SUBROUTINE YSWEEP(XMAT,FISS,NBOX,HX,HY,CORR,A,B,C,ALB,OMP,OMD,SA1)          YSWP0001
C                                                                                YSWP0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,               YSWP0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY           YSWP0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,         YSWP0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,           YSWP0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                                  YSWP0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,             YSWP0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,          YSWP0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),           YSWP0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS            YSWP0011
C                                                                                YSWP0012
      DIMENSION XMAT(52,1),FISS(2,1),HX(1),HY(1),NBOX(ND1X,1),SA1(1),            YSWP0013
     X  CORR(8,1),A(4,1),B(4,1),C(4,1),ALB(4,1),OMP(2,1),OMD(NDEL,1)            YSWP0014
      DIMENSION D(4,1)                                                           YSWP0015
      COMMON IDAT(1)                                                             YSWP0016
      ZERO=0.0                                                                   YSWP0017
      HALF=0.5                                                                   YSWP0018
      RL=XKEFF                                                                   YSWP0019
      IF(ITS  .EQ.  0) GO TO 1                                                   YSWP0020
      SMM=1.0-BTOT                                                               YSWP0021
      DO 2 ND=1,NDEL                                                             YSWP0022
    2 SMM=SMM+(BETA(ND)*RLAM(ND))/(DTI+RLAM(ND))                                 YSWP0023
    1 DO 40 NP1=1,NP1X                                                           YSWP0024
      ND1=NP1+1                                                                  YSWP0025
      IS=IDAT(KISTRY+NP1)                                                        YSWP0026
      IE=IDAT(KIENDY+NP1)                                                        YSWP0027
      DO 52 ND2=IS,IE                                                            YSWP0028
      NP2=ND2-1                                                                  YSWP0029
      NPP=(NP2-1)*NP1X+NP1                                                       YSWP0030
      I=NP2                                                                      YSWP0031
      K=NBOX(ND1,ND2)                                                            YSWP0032
      D1=CORR(1,K)                                                               YSWP0033
      D2=CORR(5,K)                                                               YSWP0034
      SR=CORR(3,K)                                                               YSWP0035
      IF(ITS  .EQ.  0) GO TO 18                                                  YSWP0036
```

```
      SUM=1.0-PTOT                                                      YSWP0037
      DO 17 ND=1,NDEL                                                   YSWP0038
   17 SUM=SUM+(BETA(ND)*RLAM(ND))/(CMD(ND,NPP)+RLAM(ND))                YSWP0039
      S1=SA1(NPP)-SUM*CORR(4,K)/RL+VIN(1)*OMP(1,NPP)                    YSWP0040
      S2=CORR(6,K)+OMP(2,NPP)*VIN(2)                                    YSWP0041
      V2=SUM*CORR(8,K)                                                  YSWP0042
      GO TO 19                                                          YSWP0043
   18 S1=SA1(NPP)-CORR(4,K)/RL                                          YSWP0044
      S2=CORR(6,K)                                                      YSWP0045
      V2=CORR(8,K)                                                      YSWP0046
   19 C1=S1/D1+S2/D2                                                    YSWP0047
      C2=(HALF*(S2/D2-S1/D1))**2  +  (V2*SR)/(RL*D1*D2)                 YSWP0048
      C3=SQRT(C2)                                                       YSWP0049
      XKS=-HALF*C1+C3                                                   YSWP0050
      XMS=HALF*C1+C3                                                    YSWP0051
      IF(V2  .GT.  ZERO) GO TO 81                                       YSWP0052
      XS=1.0E+20                                                        YSWP0053
      GO TO 82                                                          YSWP0054
   81 XS=SR/(-D2*XMS+S2)                                                YSWP0055
   82 XR=SR/(D2*XKS+S2)                                                 YSWP0056
      IF(XKS  .LE.  ZERO) GO TO 61                                      YSWP0057
C                                                                       YSWP0058
C   KAPPA IS REAL                                                       YSWP0059
C                                                                       YSWP0060
      X=SQRT(XKS)                                                       YSWP0061
      XK=X                                                              YSWP0062
      IF(X  .LT.  1.0E-07) GO TO 31                                     YSWP0063
      XCSC=X/SIN(X*HY(I))                                               YSWP0064
      XTAN=TAN(X*HY(I)*HALF)/X                                          YSWP0065
      XCDK=XCSC/XKS                                                     YSWP0066
      GO TO 62                                                          YSWP0067
   31 XCSC=1.0/HY(I)                                                    YSWP0068
      XTAN=HY(I)*HALF                                                   YSWP0069
      XCDK=XCSC/XKS                                                     YSWP0070
      GO TO 62                                                          YSWP0071
C                                                                       YSWP0072
```

```
C   KAPPA IS IMAGINARY                                                    YSWP0073
C                                                                         YSWP0074
   61 X=SQRT(ABS(XKS))                                                    YSWP0075
      XK=X                                                                YSWP0076
      IF(X  .LT.  1.0E-07) GO TO 31                                       YSWP0077
      XCSC=X/SINH(X*HY(I))                                                YSWP0078
      XTAN=TANH(X*HY(I)*HALF)/X                                           YSWP0079
      XCDK=-XCSC/(X*X)                                                    YSWP0080
   62 CONTINUE                                                            YSWP0081
      XM=SQRT(XMS)                                                        YSWP0082
      XCSCH=XM/SINH(XM*HY(I))                                             YSWP0083
      XTANH=TANH(XM*HY(I)*HALF)/XM                                        YSWP0084
      XCHDK=XCSCH/(XM*XM)                                                 YSWP0085
C                                                                         YSWP0086
C   NOW FIND ENTRIES OF A AND B MATRICES   AND C MATRX ALSO               YSWP0087
C                                                                         YSWP0088
      IF(V2  .GT.  ZERO) GO TO 91                                         YSWP0089
C                                                                         YSWP0090
C   WE ARE IN A REFLECTOR REGION - USE THE EXACT FORMULAS                 YSWP0091
C                                                                         YSWP0092
      A(1,I)=XCSC                                                         YSWP0093
      A(2,I)=ZERO                                                         YSWP0094
      A(3,I)=XR*XCSC          - XR*XCSCH                                  YSWP0095
      A(4,I)=XCSCH                                                        YSWP0096
      B(1,I)=XTAN/D1                                                      YSWP0097
      B(2,I)=ZERO                                                         YSWP0098
      B(3,I)=(XR/D1)*(XTAN-XTANH)                                         YSWP0099
      B(4,I)=XTANH/D2                                                     YSWP0100
      HI=HY(I)                                                            YSWP0101
      DET=S1*S2                                                           YSWP0102
      C(1,I)=(XCDK*HI)/D1+S2/DET                                          YSWP0103
      C(2,I)=ZERO                                                         YSWP0104
      C(3,I)=(XR*XCDK*HI+XR*XCHDK*HI)/D1+SR/DET                           YSWP0105
      C(4,I)=-(XCHDK*HI)/D2+S1/DET                                        YSWP0106
      GO TO 52                                                            YSWP0107
   91 XMULT=1.0/(XS-XR)                                                   YSWP0108
```

```
      A(1,I)=(    XS*XCSC-XR*XCSCH)*XMULT                                    YSWP0109
      A(2,I)=(-     XCSC+XCSCH)*XMULT                                        YSWP0110
      A(3,I)=XS*XR*XMULT*(XCSC-XCSCH)                                        YSWP0111
      A(4,I)=(-     XR*XCSC+XS*XCSCH)*XMULT                                  YSWP0112
      XMM=XS-XR                                                              YSWP0113
      XM1=XTAN/(D1*XMM)                                                      YSWP0114
      XM2=XTAN/(D2*XMM)                                                      YSWP0115
      XM3=XTANH/(D1*XMM)                                                     YSWP0116
      XM4=XTANH/(D2*XMM)                                                     YSWP0117
      B(1,I)=XS*XM1-XR*XM3                                                   YSWP0118
      B(2,I)=-XM2+XM4                                                        YSWP0119
      B(3,I)=XR*XS*(XM1-XM3)                                                 YSWP0120
      B(4,I)=-XR*XM2+XS*XM4                                                  YSWP0121
      HI=HY(I)                                                               YSWP0122
      T1=-XCDK                                                               YSWP0123
      T2=XCHDK                                                               YSWP0124
      T3=XR*T1                                                               YSWP0125
      T4=XS*T2                                                               YSWP0126
      T5=-XS/D1                                                              YSWP0127
      T6=1.0/D2                                                              YSWP0128
      T7=XR/D1                                                               YSWP0129
      T8=-T6                                                                 YSWP0130
      C(1,I)=T1*T5+T2*T7                                                     YSWP0131
      C(2,I)=T1*T6+T2*T8                                                     YSWP0132
      C(3,I)=T3*T5+T4*T7                                                     YSWP0133
      C(4,I)=T3*T6+T4*T8                                                     YSWP0134
      TI=HI/(XS-XR)                                                          YSWP0135
      DO 92 K=1,4                                                            YSWP0136
   92 C(K,I)=C(K,I)*TI                                                       YSWP0137
      DET=S1*S2-(V2*SR)/RL                                                   YSWP0138
      C(1,I)=C(1,I)+S2/DET                                                   YSWP0139
      C(2,I)=C(2,I)+V2/(RL*DET)                                              YSWP0140
      C(3,I)=C(3,I)+SR/DET                                                   YSWP0141
      C(4,I)=C(4,I)+S1/DET                                                   YSWP0142
   52 CONTINUE                                                               YSWP0143
C                                                                            YSWP0144
```

```
C   NOW FIND COUPLING COEFFICIENTS                                    YSWP0145
C   LEFT BOUNDARY IS ZERO J OR ALBEDO                                 YSWP0146
C   RIGHT BOUNDARY IS ALBEDO                                          YSWP0147
C                                                                     YSWP0148
      DO 55 ND2=IS,IE                                                 YSWP0149
      NP2=ND2-1                                                       YSWP0150
      NPP=(NP2-1)*NP1X+NP1                                            YSWP0151
      I=NP2                                                           YSWP0152
C                                                                     YSWP0153
C   COMPUTE COUPLING TO LEFT                                          YSWP0154
C                                                                     YSWP0155
      IF(ND2  .NE.  IS) GO TO 85                                      YSWP0156
C                                                                     YSWP0157
C   IBCB=1  SYMMETRY BOUNDARY ON BOTTOM                               YSWP0158
C   IBCB=2  ALBEDO BOUNDARY ON BOTTOM                                 YSWP0159
C                                                                     YSWP0160
      IF(IBCB  .EQ.  2) GO TO 86                                      YSWP0161
      XL1=ZERO                                                        YSWP0162
      XL2=ZERO                                                        YSWP0163
      XL3=ZERO                                                        YSWP0164
      XL4=ZERO                                                        YSWP0165
      GO TO 56                                                        YSWP0166
   86 KDN=IABS(NBOX(ND1,ND2-1))                                       YSWP0167
      DO 11 J4=1,4                                                    YSWP0168
   11 D(J4,1)=B(J4,1)+ALB(J4,KDN)                                     YSWP0169
      CALL BINV(D,1,0,XL1,XL2,XL3,XL4)                                YSWP0170
      GO TO 56                                                        YSWP0171
   85 CALL BINV(B,I-1,I,X1,X2,X3,X4)                                  YSWP0172
      XL1=X1                                                          YSWP0173
      XL2=X2                                                          YSWP0174
      XL3=X3                                                          YSWP0175
      XL4=X4                                                          YSWP0176
      F1=X1                                                           YSWP0177
      F2=X2                                                           YSWP0178
      F3=X3                                                           YSWP0179
      F4=X4                                                           YSWP0180
```

```
      CALL BMULT(X1,X2,X3,X4,A,I-1)                                     YSWP0181
      CALL BMULT(F1,F2,F3,F4,C,I-1)                                     YSWP0182
      HH=HY(I-1)*HX(NP1)                                                YSWP0183
      XMAT(13,NPP)=-HH*X1                                               YSWP0184
      XMAT(14,NPP)=-HH*X2                                               YSWP0185
      XMAT(15,NPP)=-HH*X3                                               YSWP0186
      XMAT(16,NPP)=-HH*X4                                               YSWP0187
      XMAT(41,NPP)=F1                                                   YSWP0188
      XMAT(42,NPP)=F2                                                   YSWP0189
      XMAT(43,NPP)=F3                                                   YSWP0190
      XMAT(44,NPP)=F4                                                   YSWP0191
C                                                                       YSWP0192
CC    COMPUTE CENTER POINT COUPLING                                     YSWP0193
C                                                                       YSWP0194
   56 CONTINUE                                                          YSWP0195
      J1=I                                                              YSWP0196
      J2=I+1                                                            YSWP0197
      IF(ND2  .NE.  IE) GO TO 12                                        YSWP0198
      KUP=IABS(NBOX(ND1,ND2+1))                                         YSWP0199
      DO 13 J4=1,4                                                      YSWP0200
   13 B(J4,I)=B(J4,I)+ALB(J4,KUP)                                       YSWP0201
      J2=0                                                              YSWP0202
   12 CALL BINV(B,J1,J2,X1,X2,X3,X4)                                    YSWP0203
      XR1=X1                                                            YSWP0204
      XR2=X2                                                            YSWP0205
      XR3=X3                                                            YSWP0206
      XR4=X4                                                            YSWP0207
      X1=XR1+XL1                                                        YSWP0208
      X2=XR2+XL2                                                        YSWP0209
      X3=XR3+XL3                                                        YSWP0210
      X4=XR4+XL4                                                        YSWP0211
      F1=X1                                                             YSWP0212
      F2=X2                                                             YSWP0213
      F3=X3                                                             YSWP0214
      F4=X4                                                             YSWP0215
      CALL BMULT(X1,X2,X3,X4,A,I)                                       YSWP0216
```

```
      CALL BMULT(E1,E2,E3,E4,C,I)                                       YSWP0217
      HH=HY(I)*HX(NP1)                                                  YSWP0218
      XMAT(1,NPP)=XMAT(1,NPP) + HH*X1                                   YSWP0219
      XMAT(2,NPP)=XMAT(2,NPP) + HH*X2                                   YSWP0220
      XMAT(3,NPP)=XMAT(3,NPP) + HH*X3                                   YSWP0221
      XMAT(4,NPP)=XMAT(4,NPP) + HH*X4                                   YSWP0222
      XMAT(25,NPP)=HH*X1                                                YSWP0223
      XMAT(26,NPP)=HH*X2                                                YSWP0224
      XMAT(27,NPP)=HH*X3                                                YSWP0225
      XMAT(28,NPP)=HH*X4                                                YSWP0226
      XMAT(45,NPP)=-F1                                                  YSWP0227
      XMAT(46,NPP)=-F2                                                  YSWP0228
      XMAT(47,NPP)=-F3                                                  YSWP0229
      XMAT(48,NPP)=-F4                                                  YSWP0230
C                                                                       YSWP0231
C     COMPUTE RIGHT COUPLING                                            YSWP0232
C                                                                       YSWP0233
   57 IF(ND2  .EQ.  IE) GO TO 55                                        YSWP0234
      F1=XR1                                                            YSWP0235
      F2=XR2                                                            YSWP0236
      F3=XR3                                                            YSWP0237
      F4=XR4                                                            YSWP0238
      CALL BMULT(XR1,XR2,XR3,XR4,A,I+1)                                 YSWP0239
      CALL BMULT(F1,F2,F3,F4,C,I+1)                                     YSWP0240
      HH=HY(I+1)*HX(NP1)                                                YSWP0241
      XMAT(17,NPP)=-HH*XR1                                              YSWP0242
      XMAT(18,NPP)=-HH*XR2                                              YSWP0243
      XMAT(19,NPP)=-HH*XR3                                              YSWP0244
      XMAT(20,NPP)=-HH*XR4                                              YSWP0245
      XMAT(49,NPP)=F1                                                   YSWP0246
      XMAT(50,NPP)=F2                                                   YSWP0247
      XMAT(51,NPP)=F3                                                   YSWP0248
      XMAT(52,NPP)=F4                                                   YSWP0249
   55 CONTINUE                                                          YSWP0250
   40 CONTINUE                                                          YSWP0251
      DO 400 ND2=1,NP2X                                                 YSWP0252
```

```
      IS=IDAT(KISTRX+NP2)                                              YSWP0253
      IE=IDAT(KIENDX+NP2)                                              YSWP0254
      DO 400 ND1=IS,IE                                                 YSWP0255
      NP1=ND1-1                                                        YSWP0256
      NPP=(NP2-1)*NP1X+NP1                                             YSWP0257
      F1=XMAT(1,NPP)                                                   YSWP0258
      F2=XMAT(2,NPP)                                                   YSWP0259
      F3=XMAT(3,NPP)                                                   YSWP0260
      F4=XMAT(4,NPP)                                                   YSWP0261
      DET=F1*F4-F2*F3                                                  YSWP0262
      XMAT(1,NPP)=F4/DET                                               YSWP0263
      XMAT(2,NPP)=-F2/DET                                              YSWP0264
      XMAT(3,NPP)=-F3/DET                                              YSWP0265
      XMAT(4,NPP)=F1/DET                                               YSWP0266
  400 CONTINUE                                                         YSWP0267
      RETURN                                                           YSWP0268
      END                                                              YSWP0269
```

```
      SUBROUTINE BINV(B,I1,I2,X1,X2,X3,X4)                          BINV0001
      DIMENSION B(4,1)                                              BINV0002
      IF(I2  .EQ.  0) GO TO 1                                       BINV0003
      Y1=B(1,I1)+B(1,I2)                                            BINV0004
      Y2=B(2,I1)+B(2,I2)                                            BINV0005
      Y3=B(3,I1)+B(3,I2)                                            BINV0006
      Y4=B(4,I1)+B(4,I2)                                            BINV0007
    2 DET=1.0/(Y1*Y4-Y2*Y3)                                         BINV0008
      X1=Y4*DET                                                     BINV0009
      X2=-Y2*DET                                                    BINV0010
      X3=-Y3*DET                                                    BINV0011
      X4=Y1*DET                                                     BINV0012
      RETURN                                                        BINV0013
    1 Y1=B(1,I1)                                                    BINV0014
      Y2=B(2,I1)                                                    BINV0015
      Y3=B(3,I1)                                                    BINV0016
      Y4=B(4,I1)                                                    BINV0017
      GO TO 2                                                       BINV0018
      END                                                           BINV0019
```

```
      SUBROUTINE BMULT(X1,X2,X3,X4,A,I)                                 BMLT0001
      DIMENSION A(4,1)                                                  BMLT0002
      Y1=X1*A(1,I)+X2*A(3,I)                                            BMLT0003
      Y2=X1*A(2,I)+X2*A(4,I)                                            BMLT0004
      Y3=X3*A(1,I)+X4*A(3,I)                                            BMLT0005
      Y4=X3*A(2,I)+X4*A(4,I)                                            BMLT0006
      X1=Y1                                                             BMLT0007
      X2=Y2                                                             BMLT0008
      X3=Y3                                                             BMLT0009
      X4=Y4                                                             BMLT0010
      RETURN                                                            BMLT0011
      END                                                               BMLT0012
```

```
      SUBROUTINE DORPES                                                 DRPS0001
C                                                                       DRPS0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      DRPS0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY  DRPS0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,DRPS0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,   DRPS0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                          DRPS0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    DRPS0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP, DRPS0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   DRPS0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS   DRPS0011
C                                                                       DRPS0012
      COMMON A(1)                                                       DRPS0013
      DIMENSION IDAT(1)                                                 DRPS0014
      EQUIVALENCE(A(1),IDAT(1))                                         DRPS0015
      LOGICAL JSW                                                       DRPS0016
      DATA INMIN/2/                                                     DRPS0017
      JSW=.TRUE.                                                        DRPS0018
      PSINRM=DIFTD                                                      DRPS0019
      IF(ITS  .EQ.  0) PSINRM=.02                                       DRPS0020
      IOUT=6                                                            DRPS0021
C                                                                       DRPS0022
C   ZERO OUT THE FILE WRHS                                              DRPS0023
C                                                                       DRPS0024
      IT=NPXY*4                                                         DRPS0025
      DO 50 NPP=1,IT                                                    DRPS0026
   50 A(KRHS+NPP-1)=0.0                                                 DRPS0027
C                                                                       DRPS0028
C   STORE GROUP 1 FLUX AWAY IN KS1                                      DRPS0029
C                                                                       DRPS0030
      DO 550 NP=1,NPXY                                                  DRPS0031
      NPP=NP*2-2                                                        DRPS0032
  550 A(KS1+NP-1)=A(KFLUX+NPP)                                          DRPS0033
C                                                                       DRPS0034
C   STORE GROUP 2 FLUX AWAY IN KS3                                      DRPS0035
C                                                                       DRPS0036
```

```
      DO 560 NP=1,NPXY                                                  DRPS0037
      NPP=NP*2-1                                                        DRPS0038
  560 A(KS3+NP-1)=A(KFLUX+NPP)                                          DRPS0039
      ICYCIT=1                                                          DRPS0040
  120 CONTINUE                                                          DRPS0041
      PDIFD=0.0                                                         DRPS0042
      PDIFN=0.0                                                         DRPS0043
      PLAMUP=0.0                                                        DRPS0044
      PLAMLO=1.0E+20                                                    DRPS0045
C                                                                       DRPS0046
C   COPY FLUX INTO KS2                                                  DRPS0047
C                                                                       DRPS0048
      DO 110 NP=1,NPXY                                                  DRPS0049
      NPP=NP*2-1                                                        DRPS0050
  110 A(KS2+NP-1)=A(KFLUX+NPP)                                          DRPS0051
      CALL SOLV(JSW,ICYCIT)                                             DRPS0052
C                                                                       DRPS0053
C==== THIS CALL TO  SOLV  WILL DO ONE INNER ITERATION WITH OMEGA=1.0    DRPS0054
C     AND WITH THE RIGHT HAND SIDE SOURCE SET EQUAL TO ZERO.            DRPS0055
C                                                                       DRPS0056
      DO 130 NP2=1,NP2X                                                 DRPS0057
      IS=IDAT(KISTRX+NP2)                                               DRPS0058
      IE=IDAT(KIENDX+NP2)                                               DRPS0059
      DO 131 ND1=IS,IE                                                  DRPS0060
      NP1=ND1-1                                                         DRPS0061
      NPP=(NP2-1)*NP1X+NP1                                              DRPS0062
      PSIN=A(KFLUX+NPP*2-1)                                             DRPS0063
      IF(ABS(PSIN)  .LT.  1.0E-20) GO TO 130                            DRPS0064
      PSIO=A(KS2+NPP-1)                                                 DRPS0065
      RATO=ABS(PSIN/PSIO)                                               DRPS0066
      PLAMLO=AMIN1(PLAMLO,RATO)                                         DRPS0067
      PLAMUP=AMAX1(PLAMUP,RATO)                                         DRPS0068
      PDIFN=PSIN*PSIN+PDIFN                                             DRPS0069
      PDIFD=PSIN*PSIO+PDIFD                                             DRPS0070
  131 CONTINUE                                                          DRPS0071
  130 CONTINUE                                                          DRPS0072
```

```
C                                                                       DRPS0073
C==== COMPUTE OMEGA AND NUMDO                                           DRPS0074
C                                                                       DRPS0075
      XMULT=PDIFD/PDIFN                                                 DRPS0076
      IT=NPXY*2                                                         DRPS0077
      DO 135 NPP=1,IT                                                   DRPS0078
  135 A(KFLUX+NPP-1)=A(KFLUX+NPP-1) * XMULT                             DRPS0079
C                                                                       DRPS0080
      ICYCIT=ICYCIT+1                                                   DRPS0081
      IF(ICYCIT  .LT.   8) GO TO 120                                    DRPS0082
      RHOEST=PDIFN/PDIFD                                                DRPS0083
      OMEGBL=2.0/(1.0+SQRT(1.0-PLAMLO))                                 DRPS0084
      X=1.0-PLAMUP                                                      DRPS0085
      IF(X  .LE.  0.0) OMEGBU=2.0                                       DRPS0086
      IF(X  .GT.  0.0) OMEGBU=2.0/(1.0+SQRT(X))                         DRPS0087
      OMEGM=2.0/(1.0+SQRT(1.0-RHOEST))                                  DRPS0088
      IF(ABS(OMEGBU-OMEGBL)  .LE.  ((2.0-OMEGM)/5.0)) GO TO 240         DRPS0089
      IF(ICYCIT  .LT.  50) GO TO 120                                    DRPS0090
  240 CONTINUE                                                          DRPS0091
      RHO=RHOEST                                                        DRPS0092
      X=2.0*SQRT(OMEGM/(OMEGM-1.0))                                     DRPS0093
      Y=OMEGM-1.0                                                       DRPS0094
      NING=0                                                            DRPS0095
  260 X=X*Y                                                             DRPS0096
      NING=NING+1                                                       DRPS0097
      IF(X  .GT.  PSINRM) GO TO 260                                     DRPS0098
      NUMDO=NING                                                        DRPS0099
      IF(NUMDO  .LT.  INMIN) NUMDO=INMIN                                DRPS0100
      IF(NINNER  .EQ.  0) NINNER=NUMDO                                  DRPS0101
      IF(ITS  .NE.  0) NINNER=NUMDO                                     DRPS0102
      WRITE(IOUT,1000) NINNER,RHO,OMEGM                                 DRPS0103
 1000 FORMAT(1H1,5X,'DO ',I5,3X,'INNERS WITH RHO = ',E12.4,3X,          DRPS0104
     X  'AND OMEGA = ',E12.4)                                           DRPS0105
      IF(ITS  .NE.  0) GO TO 500                                        DRPS0106
C                                                                       DRPS0107
C   SET FLUXES EQUAL TO ONE                                             DRPS0108
```

```
C                                                                       DRPS0109
      DO 340 NP2=1,NP2X                                                 DRPS0110
      IS=IDAT(KISTRX+NP2)                                               DRPS0111
      IE=IDAT(KIENDX+NP2)                                               DRPS0112
      DO 340 ND1=IS,IE                                                  DRPS0113
      NP1=ND1-1                                                         DRPS0114
      NPP=(NP2-1)*NP1X+NP1                                              DRPS0115
      DO 340 NG=1,2                                                     DRPS0116
  340 A(KFLUX-1+NG+(NPP-1)*2)=1.0                                       DRPS0117
      IT=NPXY*4                                                         DRPS0118
      DO 341 NPP=1,IT                                                   DRPS0119
  341 A(KB+NPP-1)=0.0                                                   DRPS0120
      GO TO 600                                                         DRPS0121
C                                                                       DRPS0122
  500 DO 570 NP=1,NPXY                                                  DRPS0123
      NPP=NP*2-2                                                        DRPS0124
  570 A(KFLUX+NPP)=A(KS1+NP-1)                                          DRPS0125
C                                                                       DRPS0126
      DO 580 NP=1,NPXY                                                  DRPS0127
      NPP=NP*2-1                                                        DRPS0128
  580 A(KFLUX+NPP)=A(KS3+NP-1)                                          DRPS0129
C                                                                       DRPS0130
  600 RETURN                                                            DRPS0131
      END                                                               DRPS0132
```

```
      SUBROUTINE INNERS(JSW)                                                    INRS0001
C                                                                               INRS0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,              INRS0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY           INRS0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,        INRS0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,           INRS0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                                  INRS0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,            INRS0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,          INRS0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),           INRS0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS            INRS0011
C                                                                               INRS0012
      COMMON A(1)                                                               INRS0013
C                                                                               INRS0014
C        DO NINNER NUMBER OF INNER (FLUX) ITERATIONS                            INRS0015
C                                                                               INRS0016
      LOGICAL JSW                                                               INRS0017
      CALL RHS(A(KMAT),A(KB),A(KFLUX),A(KRHS))                                  INRS0018
      DO 10 IN=1,NINNER                                                         INRS0019
   10 CALL SOLV(JSW,IN)                                                         INRS0020
      CALL BUCK(A(KMAT),A(KB),A(KFLUX),A(KHX),A(KHY))                           INRS0021
      RETURN                                                                    INRS0022
      END                                                                       INRS0023
```

```
      SUBROUTINE RHS(XMAT,B,FLUX,R)                                     RTHS0001
C                                                                       RTHS0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      RTHS0003
     X       WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY   RTHS0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX, RTHS0005
     X       KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,    RTHS0006
     X       KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                           RTHS0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    RTHS0008
     X       ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,  RTHS0009
     X       TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   RTHS0010
     X       VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS    RTHS0011
C                                                                       RTHS0012
      COMMON IDAT(1)                                                    RTHS0013
      DIMENSION XMAT(1),B(4,1),FLUX(2,1),R(4,1)                         RTHS0014
      ZERO=0.0                                                          RTHS0015
      DO 10 NP2=1,NP2X                                                  RTHS0016
      IS=IDAT(KISTRX+NP2)                                               RTHS0017
      IE=IDAT(KIENDX+NP2)                                               RTHS0018
      DO 11 ND1=IS,IE                                                   RTHS0019
      NP1=ND1-1                                                         RTHS0020
      SUM1=ZERO                                                         RTHS0021
      SUM2=ZERO                                                         RTHS0022
      SUM5=ZERO                                                         RTHS0023
      SUM6=ZERO                                                         RTHS0024
      SM1=ZERO                                                          RTHS0025
      SM2=ZERO                                                          RTHS0026
      SM5=ZERO                                                          RTHS0027
      SM6=ZERO                                                          RTHS0028
      NPP=(NP2-1)*NP1X+NP1                                              RTHS0029
      NPPL=NPP-1                                                        RTHS0030
      NPPR=NPP+1                                                        RTHS0031
      J=(NPP-1)*52                                                      RTHS0032
      IF(ND1  .EQ.  IS) GO TO 15                                        RTHS0033
      SUM1=XMAT(29+J)*B(3,NPPL)+XMAT(30+J)*B(4,NPPL)                    RTHS0034
      SUM2=XMAT(31+J)*B(3,NPPL)+XMAT(32+J)*B(4,NPPL)                    RTHS0035
   15 SUM3=XMAT(33+J)*B(3,NPP)+XMAT(34+J)*B(4,NPP)                      RTHS0036
```

```
      SUM4=XMAT(35+J)*B(3,NPP)+XMAT(36+J)*B(4,NPP)                      RTHS0037
      IF(ND1  .EQ.  IF) GO TO 16                                        RTHS0038
      SUM5=XMAT(37+J)*B(3,NPPR)+XMAT(38+J)*B(4,NPPR)                    RTHS0039
      SUM6=XMAT(39+J)*B(3,NPPR)+XMAT(40+J)*B(4,NPPR)                    RTHS0040
   16 NPPL=NPP-NP1X                                                     RTHS0041
      NPPR=NPP+NP1X                                                     RTHS0042
      IF(NP2  .EQ.  1) GO TO 17                                         RTHS0043
      SM1=XMAT(41+J)*B(1,NPPL)+XMAT(42+J)*B(2,NPPL)                     RTHS0044
      SM2=XMAT(43+J)*B(1,NPPL)+XMAT(44+J)*B(2,NPPL)                     RTHS0045
   17 SM3=XMAT(45+J)*B(1,NPP)+XMAT(46+J)*B(2,NPP)                       RTHS0046
      SM4=XMAT(47+J)*B(1,NPP)+XMAT(48+J)*B(2,NPP)                       RTHS0047
      IF(NP2  .EQ.  NP2X) GO TO 18                                      RTHS0048
      SM5=XMAT(49+J)*B(1,NPPR)+XMAT(50+J)*B(2,NPPR)                     RTHS0049
      SM6=XMAT(51+J)*B(1,NPPR)+XMAT(52+J)*B(2,NPPR)                     RTHS0050
   18 G1SUM=(SUM1+SUM3+SUM5+SM1+SM3+SM5)                                RTHS0051
      G2SUM=(SUM2+SUM4+SUM6+SM2+SM4+SM6)                                RTHS0052
      B(3,NPP)=G1SUM                                                    RTHS0053
      B(4,NPP)=G2SUM                                                    RTHS0054
   11 CONTINUE                                                          RTHS0055
   10 CONTINUE                                                          RTHS0056
      RETURN                                                            RTHS0057
      END                                                               RTHS0058
```

```
      SUBROUTINE SOLV(JSW,MM)                                                  SOLV0001
C                                                                              SOLV0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,             SOLV0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY          SOLV0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,       SOLV0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,          SOLV0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                                 SOLV0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,           SOLV0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,         SOLV0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),          SOLV0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS           SOLV0011
C                                                                              SOLV0012
      LOGICAL JSW                                                              SOLV0013
      COMMON A(1)                                                              SOLV0014
C                                                                              SOLV0015
C   IF JSW IS TRUE, SET W TO ONE FOR ALL INNERS                                SOLV0016
C   MM IS THE INNER ITERATION COUNTER                                          SOLV0017
C   FIRST GO THROUGH RED SQUARES, THEN BLACK                                   SOLV0018
C                                                                              SOLV0019
      IF(MM  .NE.  1) GO TO 5                                                  SOLV0020
      W=1.0                                                                    SOLV0021
      GO TO 6                                                                  SOLV0022
    5 W=1.0/(1.0-RHO*WLAST*0.25)                                               SOLV0023
    6 IF(JSW)  W=1.0                                                           SOLV0024
      DO 20 NP2=1,NP2X                                                         SOLV0025
      NN=NP2-(NP2/2)*2                                                         SOLV0026
      IF(NN  .EQ.  0) GO TO 21                                                 SOLV0027
      NPX1=2                                                                   SOLV0028
      NPX2=N3                                                                  SOLV0029
      GO TO 30                                                                 SOLV0030
   21 NPX1=1                                                                   SOLV0031
      NPX2=N4                                                                  SOLV0032
   30 CALL ROW(NPX1,NPX2,NP2,W,A(KFLUX),A(KMAT),A(KRHS),NP1X,                  SOLV0033
     X  A(KISTRX+NP2),A(KIENDX+NP2))                                           SOLV0034
   20 CONTINUE                                                                 SOLV0035
      WLAST=W                                                                  SOLV0036
```

```
      IF(MM  .NE.  1) GO TO 7                                          SOLV0037
      W=2.0/(2.0-RHO)                                                  SOLV0038
      GO TO 8                                                          SOLV0039
    7 W=1.0/(1.0-RHO*WLAST*0.25)                                       SOLV0040
    8 IF(JSW)  W=1.0                                                   SOLV0041
      DO 40 NP2=1,NP2X                                                 SOLV0042
      NN=NP2-(NP2/2)*2                                                 SOLV0043
      IF(NN  .EQ.  0) GO TO 41                                         SOLV0044
      NPX1=1                                                           SOLV0045
      NPX2=N1                                                          SOLV0046
      GO TO 50                                                         SOLV0047
   41 NPX1=2                                                           SOLV0048
      NPX2=N2                                                          SOLV0049
   50 CALL ROW(NPX1,NPX2,NP2,W,A(KFLUX),A(KMAT),A(KRHS),NP1X,          SOLV0050
     X  A(KISTRX+NP2),A(KIENDX+NP2))                                   SOLV0051
   40 CONTINUE                                                         SOLV0052
      WLAST=W                                                          SOLV0053
      RETURN                                                           SOLV0054
      END                                                              SOLV0055
```

```
      SUBROUTINE BUCK(XMAT,B,F,HX,HY)                                   BUCK0001
C                                                                       BUCK0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      BUCK0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY  BUCK0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,BUCK0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,   BUCK0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                          BUCK0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    BUCK0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP, BUCK0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   BUCK0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS   BUCK0011
C                                                                       BUCK0012
      COMMON IDAT(1)                                                    BUCK0013
      DIMENSION XMAT(1),B(4,1),F(2,1),HX(1),HY(1)                       BUCK0014
C                                                                       BUCK0015
      ZERO=0.0                                                          BUCK0016
C                                                                       BUCK0017
C   FIRST CALCULATE X DIRECTED LEAKAGES                                 BUCK0018
C                                                                       BUCK0019
      DO 100 NP2=1,NP2X                                                 BUCK0020
      IS=IDAT(KISTRX+NP2)                                               BUCK0021
      IE=IDAT(KIENDX+NP2)                                               BUCK0022
      DO 100 ND1=IS,IE                                                  BUCK0023
      NP1=ND1-1                                                         BUCK0024
      NPP=(NP2-1)*NP1X+NP1                                              BUCK0025
      J=(NPP-1)*52                                                      BUCK0026
   51 NPPL=NPP-1                                                        BUCK0027
      NPPR=NPP+1                                                        BUCK0028
      X1=ZERO                                                           BUCK0029
      X2=ZERO                                                           BUCK0030
      Z1=ZERO                                                           BUCK0031
      Z2=ZERO                                                           BUCK0032
      X5=ZERO                                                           BUCK0033
      X6=ZERO                                                           BUCK0034
      Z5=ZERO                                                           BUCK0035
      Z6=ZERO                                                           BUCK0036
```

```
      IF(ND1  .EQ.  IS) GO TO 10                                        BUCK0037
      X1=XMAT(5+J)*F(1,NPPL)+XMAT(6+J)*F(2,NPPL)                        BUCK0038
      X2=XMAT(7+J)*F(1,NPPL)+XMAT(8+J)*F(2,NPPL)                        BUCK0039
      Z1=XMAT(29+J)*B(3,NPPL)+XMAT(30+J)*B(4,NPPL)                      BUCK0040
      Z2=XMAT(31+J)*B(3,NPPL)+XMAT(32+J)*B(4,NPPL)                      BUCK0041
   10 X3=XMAT(21+J)*F(1,NPP)+XMAT(22+J)*F(2,NPP)                        BUCK0042
      X4=XMAT(23+J)*F(1,NPP)+XMAT(24+J)*F(2,NPP)                        BUCK0043
      Z3=XMAT(33+J)*B(3,NPP)+XMAT(34+J)*B(4,NPP)                        BUCK0044
      Z4=XMAT(35+J)*B(3,NPP)+XMAT(36+J)*B(4,NPP)                        BUCK0045
      IF(ND1  .EQ.  IE) GO TO 11                                        BUCK0046
      X5=XMAT(9+J)*F(1,NPPR)+XMAT(10+J)*F(2,NPPR)                       BUCK0047
      X6=XMAT(11+J)*F(1,NPPR)+XMAT(12+J)*F(2,NPPR)                      BUCK0048
      Z5=XMAT(37+J)*B(3,NPPR)+XMAT(38+J)*B(4,NPPR)                      BUCK0049
      Z6=XMAT(39+J)*B(3,NPPR)+XMAT(40+J)*B(4,NPPR)                      BUCK0050
   11 S1=(X1+Z1+X3+Z3+X5+Z5)/HY(NP2)                                    BUCK0051
      S2=(X2+Z2+X4+Z4+X6+Z6)/HY(NP2)                                    BUCK0052
      B(1,NPP)=S1                                                       BUCK0053
      B(2,NPP)=S2                                                       BUCK0054
  100 CONTINUE                                                          BUCK0055
C                                                                       BUCK0056
C   NOW DO LEAKAGES IN THE Y DIRECTION                                  BUCK0057
C                                                                       BUCK0058
      DO 200 NP2=1,NP2X                                                 BUCK0059
      IS=IDAT(KISTRX+NP2)                                               BUCK0060
      IE=IDAT(KIENDX+NP2)                                               BUCK0061
      DO 200 ND1=IS,IE                                                  BUCK0062
      NP1=ND1-1                                                         BUCK0063
      NPP=(NP2-1)*NP1X+NP1                                              BUCK0064
      J=(NPP-1)*52                                                      BUCK0065
   50 NPPL=NPP-NP1X                                                     BUCK0066
      NPPR=NPP+NP1X                                                     BUCK0067
      X1=ZERO                                                           BUCK0068
      X2=ZERO                                                           BUCK0069
      Z1=ZERO                                                           BUCK0070
      Z2=ZERO                                                           BUCK0071
      X5=ZERO                                                           BUCK0072
```

```
      X6=ZERO                                                          BUCK0073
      Z5=ZERO                                                          BUCK0074
      Z6=ZERO                                                          BUCK0075
      IF(NP2  .EQ.  1) GO TO 12                                        BUCK0076
      X1=XMAT(13+J)*F(1,NPPL)+XMAT(14+J)*F(2,NPPL)                     BUCK0077
      X2=XMAT(15+J)*F(1,NPPL)+XMAT(16+J)*F(2,NPPL)                     BUCK0078
      Z1=XMAT(41+J)*B(1,NPPL)+XMAT(42+J)*B(2,NPPL)                     BUCK0079
      Z2=XMAT(43+J)*B(1,NPPL)+XMAT(44+J)*B(2,NPPL)                     BUCK0080
   12 X3=XMAT(25+J)*F(1,NPP)+XMAT(26+J)*F(2,NPP)                       BUCK0081
      X4=XMAT(27+J)*F(1,NPP)+XMAT(28+J)*F(2,NPP)                       BUCK0082
      Z3=XMAT(45+J)*B(1,NPP)+XMAT(46+J)*B(2,NPP)                       BUCK0083
      Z4=XMAT(47+J)*B(1,NPP)+XMAT(48+J)*B(2,NPP)                       BUCK0084
      IF(NP2  .EQ.  NP2X) GO TO 13                                     BUCK0085
      X5=XMAT(17+J)*F(1,NPPR)+XMAT(18+J)*F(2,NPPR)                     BUCK0086
      X6=XMAT(19+J)*F(1,NPPR)+XMAT(20+J)*F(2,NPPR)                     BUCK0087
      Z5=XMAT(49+J)*B(1,NPPR)+XMAT(50+J)*B(2,NPPR)                     BUCK0088
      Z6=XMAT(51+J)*B(1,NPPR)+XMAT(52+J)*B(2,NPPR)                     BUCK0089
   13 S1=(X1+Z1+X3+Z3+X5+Z5)/HX(NP1)                                   BUCK0090
      S2=(X2+Z2+X4+Z4+X6+Z6)/HX(NP1)                                   BUCK0091
      B(3,NPP)=S1                                                      BUCK0092
      B(4,NPP)=S2                                                      BUCK0093
  200 CONTINUE                                                         BUCK0094
      RETURN                                                           BUCK0095
      END                                                              BUCK0096
```

```
      SUBROUTINE ROW(NPX1,NPX2,NP2,W,FLUX,XMAT,RHS,NP1X,ISTR,IEND)          ROWS0001
C                                                                           ROWS0002
      DIMENSION FLUX(2,1),RHS(4,1),XMAT(1)                                  ROWS0003
C                                                                           ROWS0004
      NP=(NP2-1)*NP1X                                                       ROWS0005
      DO 10 I=NPX1,NPX2,2                                                   ROWS0006
      K=I+1                                                                 ROWS0007
      IF(K .LT. ISTR  .OR.  K .GT. IEND) GO TO 10                           ROWS0008
      NPP=NP+I                                                              ROWS0009
      NUP=NPP+NP1X                                                          ROWS0010
      NDN=NPP-NP1X                                                          ROWS0011
      NL=NPP-1                                                              ROWS0012
      NR=NPP+1                                                              ROWS0013
      J=(NPP-1)*52                                                          ROWS0014
      SUM1=XMAT(J+17)*FLUX(1,NUP)+XMAT(J+18)*FLUX(2,NUP)                    ROWS0015
      SUM2=XMAT(J+19)*FLUX(1,NUP)+XMAT(J+20)*FLUX(2,NUP)                    ROWS0016
      SUM3=XMAT(J+9)*FLUX(1,NR)+XMAT(J+10)*FLUX(2,NR)                       ROWS0017
      SUM4=XMAT(J+11)*FLUX(1,NR)+XMAT(J+12)*FLUX(2,NR)                      ROWS0018
      SUM5=XMAT(J+5)*FLUX(1,NL)+XMAT(J+6)*FLUX(2,NL)                        ROWS0019
      SUM6=XMAT(J+7)*FLUX(1,NL)+XMAT(J+8)*FLUX(2,NL)                        ROWS0020
      SUM7=XMAT(J+13)*FLUX(1,NDN)+XMAT(J+14)*FLUX(2,NDN)                    ROWS0021
      SUM8=XMAT(J+15)*FLUX(1,NDN)+XMAT(J+16)*FLUX(2,NDN)                    ROWS0022
      RHS1=RHS(1,NPP)-SUM1-SUM3-SUM5-SUM7-RHS(3,NPP)                        ROWS0023
      RHS2=RHS(2,NPP)-SUM2-SUM4-SUM6-SUM8-RHS(4,NPP)                        ROWS0024
      F1=XMAT(J+1)                                                          ROWS0025
      F2=XMAT(J+2)                                                          ROWS0026
      F3=XMAT(J+3)                                                          ROWS0027
      F4=XMAT(J+4)                                                          ROWS0028
      E1=F1*RHS1+F2*RHS2                                                    ROWS0029
      E2=F3*RHS1+F4*RHS2                                                    ROWS0030
      FLUX(1,NPP)=W*E1+(1.0-W)*FLUX(1,NPP)                                  ROWS0031
      FLUX(2,NPP)=W*E2+(1.0-W)*FLUX(2,NPP)                                  ROWS0032
   10 CONTINUE                                                              ROWS0033
      RETURN                                                                ROWS0034
      END                                                                   ROWS0035
```

```
      SUBROUTINE CUTERS(IOT,SORC,SORC1,SORC2,NPXY,IRETRN,DIF,          OTRS0001
     X   DIFLAM,XKEFF)                                                 OTRS0002
      DIMENSION SORC(1),SORC1(1),SORC2(1)                              OTRS0003
      ISIX=6                                                           OTRS0004
      ININ=9                                                           OTRS0005
      ITW=12                                                           OTRS0006
      NOUTBA=4                                                         OTRS0007
      FISMON=1.0E-06                                                   OTRS0008
      IF(IOT  .NE.  1) GO TO 10                                        OTRS0009
      WRITE(6,660)                                                     OTRS0010
  660 FORMAT(1H0,5X,'OUTER ITERATION OUTPUT ',//)                      OTRS0011
      EFFK=1.0                                                         OTRS0012
      SIGMA=0.0                                                        OTRS0013
      SIGBAR=0.0                                                       OTRS0014
      FISMIN=FISMON                                                    OTRS0015
      FLAMDA=1.0                                                       OTRS0016
      FISLNN=0.0                                                       OTRS0017
      FISLNO=0.0                                                       OTRS0018
      ERRATN=1.0                                                       OTRS0019
      ERRAT=1.0                                                        OTRS0020
   10 CONTINUE                                                         OTRS0021
      DIFKEF=0.0                                                       OTRS0022
      GAMMAN=0.0                                                       OTRS0023
      GAMMAD=0.0                                                       OTRS0024
      FLAMUP=0.0                                                       OTRS0025
      FLAMLO=1.0E+20                                                   OTRS0026
      ERRATD=ERRATN                                                    OTRS0027
      FISLNO=FISLNN                                                    OTRS0028
      FISLNN=0.0                                                       OTRS0029
      ERRATN=0.0                                                       OTRS0030
      IF(IOT  .GT.  NOUTBA) GO TO 110                                  OTRS0031
  100 CONTINUE                                                         OTRS0032
      NORDCP=0                                                         OTRS0033
      NEWCP=1                                                          OTRS0034
      SIGMA=0.0                                                        OTRS0035
      ALPHAC=0.0                                                       OTRS0036
```

```
      BETAC=0.0                                                          OTRS0037
      GO TO 130                                                          OTRS0038
  110 CONTINUE                                                           OTRS0039
      IF(NEWCP  .EQ.  1) GO TO 120                                       OTRS0040
      NORDCP=NORDCP+1                                                    OTRS0041
      CALL CHEBE(ALPHAC,BETAC,NORDCP,SIGMA)                              OTRS0042
      GO TO 130                                                          OTRS0043
  120 ERPROD=1.0                                                         OTRS0044
      NORDCP=1                                                           OTRS0045
      SIGMA=SIGBAR                                                       OTRS0046
      IF(SIGMA  .GT.  1.0  .OR.  SIGMA  .LT.  0.4) GO TO 100             OTRS0047
      IF(IOT  .LE.  ISIX) SIGMA=AMIN1(SIGMA,0.9)                         OTRS0048
      IF(IOT  .LE.  ININ) SIGMA=AMIN1(SIGMA,0.95)                        OTRS0049
      IF(IOT  .LE.  ITW) SIGMA=AMIN1(SIGMA,0.985)                        OTRS0050
      SIGMA=AMIN1(SIGMA,0.999)                                           OTRS0051
      NEWCP=0                                                            OTRS0052
      CALL CHEBE(ALPHAC,BETAC,NORDCP,SIGMA)                              OTRS0053
  130 CONTINUE                                                           OTRS0054
      ASSIGN 150 TO NS1                                                  OTRS0055
      IF(BETAC  .LE.  0.0) ASSIGN 160 TO NS1                             OTRS0056
      IF(NORDCP  .EQ.  0) ASSIGN 170 TO NS1                              OTRS0057
C                                                                        OTRS0058
C==== BEGIN THE SWEEP OVER THE MESH                                      OTRS0059
C                                                                        OTRS0060
      DO 310 NPP=1,NPXY                                                  OTRS0061
      FISNS=SORC(NPP)                                                    OTRS0062
      FISNM=FISNS                                                        OTRS0063
      FROM1S=SORC1(NPP)                                                  OTRS0064
      IF(IOT  .EQ.  1) FISLNO=FISLNO+FROM1S                              OTRS0065
      FISM1S=FROM1S                                                      OTRS0066
      FISDEL=FISNS-FROM1S                                                OTRS0067
      ERRATN=ERRATN+FISDEL*FISDEL                                        OTRS0068
      FISM2S=SORC2(NPP)                                                  OTRS0069
      SORC2(NPP)=FROM1S                                                  OTRS0070
      IF(FISM1S  .LE.  FISMIN) GO TO 140                                 OTRS0071
      RATO=FISNS/FISM1S                                                  OTRS0072
```

```
      FLAMUP=AMAX1(FLAMUP,RATC)                                          OTRS0073
      FLAMLO=AMIN1(FLAMLC,RATC)                                          OTRS0074
  140 GAMMAN=GAMMAN+FISNS*FISNS                                          OTRS0075
      GAMMAD=GAMMAD+FISM1S*FISNS                                         OTRS0076
      IF(NORDCP  .LE.  0) GO TO 180                                      OTRS0077
      GO TO NS1,(150,160,170)                                            OTRS0078
  150 FISNE=FISM1S+ALPHAC*(FISNS-FISM1S)+BETAC*(FISM1S-FISM2S)           OTRS0079
      GO TO 170                                                          OTRS0080
  160 FISNE=FISM1S+ALPHAC*(FISNS-FISM1S)                                 OTRS0081
  170 CONTINUE                                                           OTRS0082
      IF(FISNE  .LE.  0.0) FISNE=ABS(FISNE)                              OTRS0083
  180 CONTINUE                                                           OTRS0084
      SORC(NPP)=FISNE                                                    OTRS0085
      SORC1(NPP)=FISNE                                                   OTRS0086
      FISLNN=FISLNN+FISNE                                                OTRS0087
  190 CONTINUE                                                           OTRS0088
  310 CONTINUE                                                           OTRS0089
      DUM2=FLAMDA*GAMMAN/GAMMAD                                          OTRS0090
      DIFKEF=ABS(DUM2-EFFK)                                              OTRS0091
      EFFK=DUM2                                                          OTRS0092
      FLAMDA=FLAMDA*FISLNN/FISLNO                                        OTRS0093
      DIFLAM=(FLAMUP-FLAMLO)/2.0                                         OTRS0094
      DIFFIS=SQRT(ERRATN/GAMMAD)                                         OTRS0095
      IF(IOT  .GT.  1) ERRAT=SQRT(ERRATN/ERRATD)                         OTRS0096
      IF(NORDCP-1) 220,230,240                                           OTRS0097
  220 SIGBAR=ERRAT                                                       OTRS0098
      GO TO 250                                                          OTRS0099
  230 ERPROD=1.0                                                         OTRS0100
      GO TO 250                                                          OTRS0101
  240 ERPROD=ERPROD*ERRAT                                                OTRS0102
      NPM1=NORDCP-1                                                      OTRS0103
      DUM3=FLOAT(NPM1)                                                   OTRS0104
      DUM4=(2.0-SIGMA)/SIGMA                                             OTRS0105
      CPM1=COSH1(DUM3*DACOSH(DUM4))                                      OTRS0106
      IF(ERPROD*CPM1 .LT. 1.0) SIGBAR=SIGMA*(COS(ARCOS(CPM1*ERPROD)/     OTRS0107
     X  DUM3)+1.0)/2.0                                                   OTRS0108
```

```
      IF(ERPROD*CPM1  .GE.  1.0)                                        OTRS0109
     X  SIGBAR=SIGMA*(COSH1(DACOSH(ERPROD*CPM1)/DUM3)+1.0)/2.0          OTRS0110
      IF(NORDCP  .LT.  3) GO TO 250                                     OTRS0111
      IF(ERPROD  .LT.  (1.0/CPM1)) GO TO 250                            OTRS0112
      NEWCP=1                                                           OTRS0113
  250 IRETRN=0                                                          OTRS0114
      IF(IOT  .EQ.  1) GO TO 270                                        OTRS0115
      IF(DIFLAM  .LE.  DIF) IRETRN=1                                    OTRS0116
  270 XKEFF=FLAMDA                                                      OTRS0117
      IF(IRETRN  .EQ.  1) XKEFF=EFFK                                    OTRS0118
      WRITE(6,661) IOT,NORDCP,SIGMA,EFFK,XKEFF                          OTRS0119
  661 FORMAT(6X,2I10,5X,E12.4,5X,E15.7,5X,E15.7)                        OTRS0120
      RETURN                                                            OTRS0121
      END                                                               OTRS0122
```

```
      SUBROUTINE CHEBE(ALPHAC,BETAC,NORDCP,SIGMA)                          CHEB0001
      IF(NORDCP  .GT.  1) GO TO 100                                        CHEB0002
      ALPHAC=2.0/(2.0-SIGMA)                                               CHEB0003
      BETAC=0.0                                                            CHEB0004
      RETURN                                                               CHEB0005
  100 COSHGM=(2.0-SIGMA)/SIGMA                                             CHEB0006
      GAMMA=DACOSH(COSHGM)                                                 CHEB0007
      ALPHAC=4.0*COSH1((NORDCP-1)*GAMMA)/(SIGMA*COSH1(NORDCP*GAMMA))       CHEB0008
      BETAC=(1.0-0.5*SIGMA)*ALPHAC-1.0                                     CHEB0009
      RETURN                                                               CHEB0010
      END                                                                  CHEB0011
```

```
FUNCTION CCSH1(X)                                                    COSH0001
CCSH1=CCSH(X)                                                        COSH0002
RETURN                                                               COSH0003
END                                                                  COSH0004
```

```
      FUNCTION DACOSH(X)                                              DACH0001
      DACOSH=ALOG(X+SQRT(X*X-1.0))                                    DACH0002
      RETURN                                                          DACH0003
      END                                                             DACH0004
```

```
      SUBROUTINE CPUT(CTIME)                                        CPUT0001
      CALL TIMING(IT)                                               CPUT0002
      CTIME=FLOAT(IT-ITT)*.01                                       CPUT0003
      RETURN                                                        CPUT0004
      ENTRY CPUTO                                                   CPUT0005
      CALL TIMING(ITT)                                              CPUT0006
      RETURN                                                        CPUT0007
      END                                                           CPUT0008
```

```
      SUBROUTINE EDIT(ND1X,ND2X,NEDX,NEDY,ITS,NPXY,FLUX,FISS,S,NX,      EDIT0001
     X   NY,STOR,BUCK,TIME,HX,HY,TEMP)                                  EDIT0002
C                                                                       EDIT0003
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF            EDIT0004
C                                                                       EDIT0005
      DIMENSION FLUX(2,ND1X,1),FISS(2,ND1X,1),S(ND1X,1),TEMP(ND1X,1)    EDIT0006
      DIMENSION NX(1),NY(1),STOR(NEDX,NEDY),BUCK(4,ND1X,1),HX(1),HY(1)  EDIT0007
      IF(ITS  .NE.  0) GO TO 26                                         EDIT0008
      XTOT=0.0                                                          EDIT0009
      HTOT=0.0                                                          EDIT0010
      DO 210 NP2=1,ND2X                                                 EDIT0011
      DO 210 NP1=1,ND1X                                                 EDIT0012
      X=FLUX(1,NP1,NP2)*FISS(1,NP1,NP2)+FLUX(2,NP1,NP2)*FISS(2,NP1,NP2) EDIT0013
      X=X/XNU                                                           EDIT0014
      S(NP1,NP2)=X                                                      EDIT0015
      IF(X  .LE.  0.0) GO TO 210                                        EDIT0016
      XTOT=XTOT+X                                                       EDIT0017
      HH=HX(NP1)*HY(NP2)                                                EDIT0018
      HTOT=HTOT+HH                                                      EDIT0019
  210 CONTINUE                                                          EDIT0020
      PTOT=(HTOT*WPCC)/(XTOT*EPSIL)                                     EDIT0021
      DO 220 NP2=1,ND2X                                                 EDIT0022
      DO 220 NP1=1,ND1X                                                 EDIT0023
      FLUX(1,NP1,NP2)=FLUX(1,NP1,NP2) * PTOT                            EDIT0024
      FLUX(2,NP1,NP2)=FLUX(2,NP1,NP2) * PTOT                            EDIT0025
  220 S(NP1,NP2)=S(NP1,NP2) * PTOT                                      EDIT0026
      DO 221 NP2=1,ND2X                                                 EDIT0027
      DO 221 NP1=1,ND1X                                                 EDIT0028
      DO 221 L=1,4                                                      EDIT0029
      BUCK(L,NP1,NP2)=BUCK(L,NP1,NP2)*PTOT                              EDIT0030
  221 CONTINUE                                                          EDIT0031
      GO TO 26                                                          EDIT0032
   25 WRITE(6,260)                                                      EDIT0033
  260 FORMAT(1H0,5X,                                                    EDIT0034
     X '  NP1',5X,'NP2',5X,'FLUX1',8X,'FLUX2',8X,'LEAKAGES',46X,        EDIT0035
     X  'POWER FRACTION',//)                                            EDIT0036
```

```
      DO 230 NP2=1,ND2X                                                  EDIT0037
      DO 230 NP1=1,ND1X                                                  EDIT0038
  230 WRITE(6,250) NP1,NP2,FLUX(1,NP1,NP2),FLUX(2,NP1,NP2),              EDIT0039
     X  (BUCK(K,NP1,NP2),K=1,4),S(NP1,NP2)                               EDIT0040
  250 FORMAT(6X,2I5,3X,7E14.6)                                           EDIT0041
   27 DO 10 J=1,NEDY                                                     EDIT0042
      DO 10 I=1,NEDX                                                     EDIT0043
   10 STOR(I,J)=0.0                                                      EDIT0044
      DO 11 J=1,NEDY                                                     EDIT0045
      IF(J  .NE.  1) JJ1=NY(J-1)+1                                       EDIT0046
      IF(J  .EQ.  1) JJ1=1                                               EDIT0047
      JJ2=NY(J)                                                          EDIT0048
      DO 11 I=1,NEDX                                                     EDIT0049
      IF(I  .NE.  1) II1=NX(I-1)+1                                       EDIT0050
      IF(I  .EQ.  1) II1=1                                               EDIT0051
      II2=NX(I)                                                          EDIT0052
      TOT=0.0                                                            EDIT0053
      DO 12 JJ=JJ1,JJ2                                                   EDIT0054
      DO 12 II=II1,II2                                                   EDIT0055
   12 TOT=TOT+S(II,JJ)                                                   EDIT0056
   11 STOR(I,J)=TOT                                                      EDIT0057
      WRITE(6,15)                                                        EDIT0058
   15 FORMAT(1HO,5X,'BOX POWERS',//)                                     EDIT0059
      DO 16 JJ=1,NEDY                                                    EDIT0060
      J=NEDY+1-JJ                                                        EDIT0061
   16 WRITE(6,17)  J,(STOR(I,J),I=1,NEDX)                                EDIT0062
   17 FORMAT(1HO,3X,I5,2X,3(2X,10E12.4,/))                               EDIT0063
      IF(ITHFB  .EQ.  0) RETURN                                          EDIT0064
      WRITE(6,140)                                                       EDIT0065
  140 FORMAT(1HO,5X,'BOX TEMPERATURES',//)                               EDIT0066
      DO 141 JJ=1,ND2X                                                   EDIT0067
      J=ND2X+1-JJ                                                        EDIT0068
  141 WRITE(6,142) J,(TEMP(I,J),I=1,ND1X)                                EDIT0069
  142 FORMAT(3X,I3,3(2X,11E11.4,/))                                      EDIT0070
      RETURN                                                             EDIT0071
   26 XTOT=0.0                                                           EDIT0072
```

```
      HTOT=0.0                                                          EDIT0073
      DO 170 NP2=1,ND2X                                                 EDIT0074
      DO 170 NP1=1,ND1X                                                 EDIT0075
      X=FLUX(1,NP1,NP2)*FISS(1,NP1,NP2)+FLUX(2,NP1,NP2)*FISS(2,NP1,NP2) EDIT0076
      X=X/XNU                                                           EDIT0077
      IF(X  .LE.  0.0) GO TO 175                                        EDIT0078
      XTOT=XTOT+X                                                       EDIT0079
      HH=HX(NP1)*HY(NP2)                                                EDIT0080
      HTOT=HTOT+HH                                                      EDIT0081
      X=X/HH                                                            EDIT0082
      S(NP1,NP2)=X*EPSIL                                                EDIT0083
      GO TO 170                                                         EDIT0084
  175 S(NP1,NP2)=0.0                                                    EDIT0085
  170 CONTINUE                                                          EDIT0086
      PTOT=XTOT*EPSIL                                                   EDIT0087
      PM=PTOT/HTOT                                                      EDIT0088
      WRITE(6,28) ITS,TIME,PTOT,PM                                      EDIT0089
   28 FORMAT(1HC,5X,'TIME STEP',I5,3X,'TIME IS ' ,E12.4,3X,             EDIT0090
     X  'TOTAL POWER ',E14.6,/,10X,'MEAN POWER ',E14.6,//)              EDIT0091
      IF(ITS  .EQ.  0) GO TO 25                                         EDIT0092
      GO TO 27                                                          EDIT0093
      END                                                               EDIT0094
```

```
      SUBROUTINE TRANS                                                  TRNS0001
C                                                                       TRNS0002
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,      TRNS0003
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY  TRNS0004
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,TRNS0005
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,   TRNS0006
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1                          TRNS0007
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    TRNS0008
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP, TRNS0009
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   TRNS0010
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS   TRNS0011
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF            TRNS0012
C                                                                       TRNS0013
      COMMON A(1)                                                       TRNS0014
      LOGICAL JSW,JST,JEND                                              TRNS0015
      LOGICAL JFIRST                                                    TRNS0016
      IF(ITRANS  .EQ.  0) STOP 1111                                     TRNS0017
      CALL CPUTO                                                        TRNS0018
      WRITE(6,450)                                                      TRNS0019
  450 FORMAT(1H1,10X,'START TRANSIENT CALCULATIONS',//)                 TRNS0020
      JSW=.FALSE.                                                       TRNS0021
      JST=.FALSE.                                                       TRNS0022
      JEND=.FALSE.                                                      TRNS0023
      BTOT=0.0                                                          TRNS0024
      DO 10 ND=1,NDEL                                                   TRNS0025
   10 BTOT=BTOT+BETA(ND)                                                TRNS0026
      IT=NPXY*2                                                         TRNS0027
      DO 20 NP=1,IT                                                     TRNS0028
   20 A(KOMGP+NP-1)=0.0                                                 TRNS0029
      IT=NPXY*NDEL                                                      TRNS0030
      DO 30 NP=1,IT                                                     TRNS0031
   30 A(KOMGD+NP-1)=0.0                                                 TRNS0032
      TIME=0.0                                                          TRNS0033
      ITS=0                                                             TRNS0034
      CALL PERTO(A(KCORR))                                              TRNS0035
      CALL PRECO(A(KPREC),A(KFLUX),A(KISTRX),A(KIENDX),A(KFISS),        TRNS0036
```

```
     X  BETA,PLAM,NP1X,NP2X,NDEL,XKEFF)                                    TRNS0037
      DO 100 NTTD=1,NTD                                                    TRNS0038
      NSTEP=NUMS(NTTD)                                                     TRNS0039
      DT=DELT(NTTD)                                                        TRNS0040
      DTI=1.0/DT                                                           TRNS0041
      DO 200 NST=1,NSTEP                                                   TRNS0042
      TIME=TIME+DT                                                         TRNS0043
      ITS=ITS+1                                                            TRNS0044
      CALL PERT(A(KCORR),JST,JEND)                                         TRNS0045
      CALL TFER(A(KISTRX),A(KIENDX),NP1X,NP2X,A(KNBOX),A(KCORR),           TRNS0046
     X  A(KSA1),ND1X)                                                      TRNS0047
      CALL FDBK(A(KSA1),A(KTEMP),A(KS3),A(KISTRX),A(KIENDX),NP1X,NP2X,     TRNS0048
     X  A(KNBOX),A(KCORR),ND1X)                                            TRNS0049
      CALL MATRX                                                           TRNS0050
      JFIRST=.FALSE.                                                       TRNS0051
      IF(NST  .GT.  1  .OR.  NTTD  .GT.  1) GO TO 50                       TRNS0052
      CALL DORPES                                                          TRNS0053
      JFIRST=.TRUE.                                                        TRNS0054
   50 CONTINUE                                                             TRNS0055
      CALL RHST(A(KRHS),A(KFLUX),A(KPREC),A(KS1),A(KS2),RLAM,A(KISTRX),    TRNS0056
     X  A(KIENDX),A(KHX),A(KHY),NP1X,NP2X,NDEL,VIN,DT)                     TRNS0057
      CALL FEXT(A(KFLUX),A(KB),A(KOMGP),A(KISTRX),A(KIENDX),NP1X,NP2X,     TRNS0058
     X  DT)                                                                TRNS0059
      CALL INNERS(JSW)                                                     TRNS0060
      CALL PREC1(A(KPREC),A(KFLUX),A(KS1),A(KS2),A(KOMGP),A(KOMGD),        TRNS0061
     X  A(KISTRX),A(KIENDX),A(KFISS),BETA,RLAM,NP1X,NP2X,NDEL,XKEFF,DTI)   TRNS0062
      CALL TEMP(A(KFLUX),A(KFISS),A(KHX),A(KHY),A(KISTRX),                 TRNS0063
     X  A(KIENDX),NP1X,NP2X,A(KTEMP),A(KS3),A(KS1),DT,JFIRST)              TRNS0064
      I=(ITS/IEDTS)*IEDTS-ITS                                              TRNS0065
      IF(I .EQ. 0) CALL EDIT(NP1X,NP2X,NEDX,NEDY,ITS,NPXY,A(KFLUX),        TRNS0066
     X   A(KFISS),A(KS1),A(KEDX),A(KEDY),A(KS2),A(KB),TIME,                TRNS0067
     X   A(KHX),A(KHY),A(KTEMP))                                           TRNS0068
  200 CONTINUE                                                             TRNS0069
  100 CONTINUE                                                             TRNS0070
      CALL CPUT(CTIME)                                                     TRNS0071
      WRITE(6,1) CTIME                                                     TRNS0072
```

```
    1 FORMAT(1H0,5X,'TIME TO DO TRANSIENT IS',E12.4)                    TRNS0073
      RETURN                                                           TRNS0074
C                                                                      TRNS0075
      END                                                              TRNS0076
```

```
      SUBROUTINE TFER(ISTR,IEND,NP1X,NP2X,NBOX,CORR,SA1,ND1X)          TFER0001
      DIMENSION ISTR(1),IEND(1),NBOX(ND1X,1),SA1(1),CORR(8,1)          TFER0002
      DO 10 NP2=1,NP2X                                                 TFER0003
      ND2=NP2+1                                                        TFER0004
      IS=ISTR(ND2)                                                     TFER0005
      IE=IEND(ND2)                                                     TFER0006
      DO 20 ND1=IS,IE                                                  TFER0007
      NP1=ND1-1                                                        TFER0008
      K=NBOX(ND1,ND2)                                                  TFER0009
      NPP=(NP2-1)*NP1X+NP1                                             TFER0010
      SA1(NPP)=CORR(2,K)                                               TFER0011
   20 CONTINUE                                                         TFER0012
   10 CONTINUE                                                         TFER0013
      RETURN                                                           TFER0014
      END                                                              TFER0015
```

```
      SUBROUTINE PRECO(PREC,FLUX,ISTR,IEND,FISS,BETA,RLAM,NP1X,NP2X,     PRC00001
     X  NDEL,RL)                                                        PRC00002
      DIMENSION PREC(NDEL,1),ISTR(1),IEND(1),FISS(2,1),FLUX(2,1),       PRC00003
     X  BETA(1),RLAM(1)                                                 PRC00004
      DO 10 NP2=1,NP2X                                                  PRC00005
      IS=ISTR(NP2+1)                                                    PRC00006
      IE=IEND(NP2+1)                                                    PRC00007
      DO 20 ND1=IS,IE                                                   PRC00008
      NP1=ND1-1                                                         PRC00009
      NPP=(NP2-1)*NP1X+NP1                                              PRC00010
      X=(FISS(1,NPP)*FLUX(1,NPP)+FISS(2,NPP)*FLUX(2,NPP))/RL            PRC00011
      DO 30 ND=1,NDEL                                                   PRC00012
   30 PREC(ND,NPP)=(BETA(ND)/RLAM(ND))*X                                PRC00013
   20 CONTINUE                                                          PRC00014
   10 CONTINUE                                                          PRC00015
      RETURN                                                            PRC00016
      END                                                               PRC00017
```

```
      SUBROUTINE PERT(CORR,JST,JEND)                                    PERT0001
C                                                                       PERT0002
      COMMON / PIXEL / NINNER,NCUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,    PERT0003
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,  PERT0004
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),   PERT0005
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS    PERT0006
C                                                                       PERT0007
      DIMENSION CORR(8,1)                                               PERT0008
      LOGICAL JST,JEND                                                  PERT0009
      IF(IPERT  .EQ.  1) GO TO 10                                       PERT0010
      IF(JEND) RETURN                                                   PERT0011
      IF(JST) GO TO 20                                                  PERT0012
   12 IF(TIME  .LE.  TST) RETURN                                        PERT0013
      JST=.TRUE.                                                        PERT0014
   20 IF(TIME  .GT.  TFIN) GO TO 21                                     PERT0015
      TOT=TFIN-TST                                                      PERT0016
      FRAC=(TIME-TST)/TOT                                               PERT0017
      CORR(2,NCOMP)=SIGB1+SIG1*FRAC                                     PERT0018
      CORR(6,NCOMP)=SIGB2+SIG2*FRAC                                     PERT0019
      RETURN                                                            PERT0020
   10 IF(JST) RETURN                                                    PERT0021
      IF(TIME  .LE.  TST) RETURN                                        PERT0022
      JST=.TRUE.                                                        PERT0023
      CORR(2,NCOMP)=CORR(2,NCOMP)+SIG1                                  PERT0024
      CORR(6,NCOMP)=CORR(6,NCOMP)+SIG2                                  PERT0025
      RETURN                                                            PERT0026
   21 JEND=.TRUE.                                                       PERT0027
      RETURN                                                            PERT0028
      ENTRY PERTO(CORR)                                                 PERT0029
      SIGB1=CORR(2,NCOMP)                                               PERT0030
      SIGB2=CORR(6,NCOMP)                                               PERT0031
      RETURN                                                            PERT0032
      END                                                               PERT0033
```

```
      SUBROUTINE FEXT(FLUX,BUCK,OMP,ISTR,IEND,NP1X,NP2X,DT)                FEXT0001
      DIMENSION FLUX(2,1),BUCK(4,1),OMP(2,1),ISTR(1),IEND(1)               FEXT0002
      DO 10 NP2=1,NP2X                                                     FEXT0003
      IS=ISTR(NP2+1)                                                       FEXT0004
      IE=IEND(NP2+1)                                                       FEXT0005
      DO 20 ND1=IS,IE                                                      FEXT0006
      NP1=ND1-1                                                            FEXT0007
      NPP=(NP2-1)*NP1X+NP1                                                 FEXT0008
      FAC1=EXP(OMP(1,NPP)*DT)                                              FEXT0009
      FAC2=EXP(OMP(2,NPP)*DT)                                              FEXT0010
      FLUX(1,NPP)=FLUX(1,NPP)*FAC1                                         FEXT0011
      FLUX(2,NPP)=FLUX(2,NPP)*FAC2                                         FEXT0012
      DO 30 ND=1,4                                                         FEXT0013
   30 BUCK(ND,NPP)=BUCK(ND,NPP)*FAC2                                       FEXT0014
   20 CONTINUE                                                             FEXT0015
   10 CONTINUE                                                             FEXT0016
      RETURN                                                               FEXT0017
      END                                                                  FEXT0018
```

```
      SUBROUTINE RHST(RHS,FLUX,PREC,SORC1,SORC2,RLAM,ISTR,IEND,          RHST0001
     X   HX,HY,NP1X,NP2X,NDEL,VIN,DT)                                    RHST0002
      DIMENSION RHS(4,1),FLUX(2,1),PREC(NDEL,1),RLAM(1),ISTR(1),         RHST0003
     X   IEND(1),HX(1),HY(1),VIN(1),SORC1(1),SORC2(1)                    RHST0004
      DO 10 NP2=1,NP2X                                                   RHST0005
      IS=ISTR(NP2+1)                                                     RHST0006
      IE=IEND(NP2+1)                                                     RHST0007
      DO 20 ND1=IS,IE                                                    RHST0008
      NP1=ND1-1                                                          RHST0009
      NPP=(NP2-1)*NP1X+NP1                                               RHST0010
      HH=HX(NP1)*HY(NP2)/DT                                              RHST0011
      SUM1=FLUX(1,NPP)*HH*VIN(1)                                         RHST0012
      SUM2=FLUX(2,NPP)*HH*VIN(2)                                         RHST0013
      DO 30 ND=1,NDEL                                                    RHST0014
   30 SUM1=SUM1+(PREC(ND,NPP)*RLAM(ND))/(1.0+DT*RLAM(ND))                RHST0015
      RHS(1,NPP)=SUM1                                                    RHST0016
      RHS(2,NPP)=SUM2                                                    RHST0017
      SORC1(NPP)=FLUX(1,NPP)                                             RHST0018
      SORC2(NPP)=FLUX(2,NPP)                                             RHST0019
   20 CONTINUE                                                           RHST0020
   10 CONTINUE                                                           RHST0021
      RETURN                                                             RHST0022
      END                                                                RHST0023
```

```
      SUBROUTINE FDBK(SA1,TEMP,SORC,ISTR,IEND,NP1X,NP2X,                FDBK0001
     X   NBOX,CORR,ND1X)                                                FDBK0002
C                                                                       FDBK0003
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF            FDBK0004
C                                                                       FDBK0005
      DIMENSION SA1(1),TEMP(1),SORC(1),ISTR(1),IEND(1)                  FDBK0006
      DIMENSION NBOX(ND1X,1),CORR(3,1)                                  FDBK0007
      IF(ITHFB  .EQ.  0) RETURN                                         FDBK0008
      SR=SQRT(TREF)                                                     FDBK0009
      DO 10 NP2=1,NP2X                                                  FDBK0010
      ND2=NP2+1                                                         FDBK0011
      IS=ISTR(ND2)                                                      FDBK0012
      IE=IEND(ND2)                                                      FDBK0013
      DO 20 ND1=IS,IE                                                   FDBK0014
      NP1=ND1-1                                                         FDBK0015
      NPP=(NP2-1)*NP1X+NP1                                              FDBK0016
      IF(SORC(NPP)  .LE.  0.0) GO TO 20                                 FDBK0017
      K=NBOX(ND1,ND2)                                                   FDBK0018
      SR1=CORR(3,K)                                                     FDBK0019
      TOT=SA1(NPP)-SR1                                                  FDBK0020
      TOT=TOT*(1.0+GAMMA*(SQRT(TEMP(NPP))-SR))                          FDBK0021
      SA1(NPP)=TOT+SR1                                                  FDBK0022
   20 CONTINUE                                                          FDBK0023
   10 CONTINUE                                                          FDBK0024
      RETURN                                                            FDBK0025
      END                                                               FDBK0026
```

```
      SUBROUTINE PREC1(PREC,FLUX,SORC1,SORC2,OMP,OMD,ISTR,IEND,FISS,      PRC10001
     X  BETA,RLAM,NP1X,NP2X,NDEL,RL,DTI)                                  PRC10002
      DIMENSION PREC(NDEL,1),FLUX(2,1),SORC1(1),SORC2(1),OMP(2,1),        PRC10003
     X  OMD(NDEL,1),ISTR(1),IEND(1),FISS(2,1),BETA(1),RLAM(1)             PRC10004
      DO 10 NP2=1,NP2X                                                    PRC10005
      IS=ISTR(NP2+1)                                                      PRC10006
      IE=IEND(NP2+1)                                                      PRC10007
      DO 20 ND1=IS,IE                                                     PRC10008
      NP1=ND1-1                                                           PRC10009
      NPP=(NP2-1)*NP1X+NP1                                                PRC10010
      S1=FLUX(1,NPP)                                                      PRC10011
      S2=FLUX(2,NPP)                                                      PRC10012
      F1=S1/SORC1(NPP)                                                    PRC10013
      F2=S2/SORC2(NPP)                                                    PRC10014
      IF(F1  .LE.  1.0  .OR.  F2  .LE.  1.0) GO TO 15                     PRC10015
      OMP(1,NPP)=DTI*ALOG(F1)                                             PRC10016
      OMP(2,NPP)=DTI*ALOG(F2)                                             PRC10017
      GO TO 50                                                            PRC10018
   15 OMP(1,NPP)=0.0                                                      PRC10019
      OMP(2,NPP)=0.0                                                      PRC10020
   50 X1=(FISS(1,NPP)*FLUX(1,NPP)+FISS(2,NPP)*FLUX(2,NPP))/RL             PRC10021
      DO 30 ND=1,NDEL                                                     PRC10022
      X=(X1*BETA(ND)+DTI*PREC(ND,NPP))/(DTI+RLAM(ND))                     PRC10023
      IF(X  .LT.  1.0E-25) GO TO 31                                       PRC10024
      XOMG=(-RLAM(ND)*X+X1*BETA(ND))/X                                    PRC10025
      OMD(ND,NPP)=XOMG                                                    PRC10026
   31 PREC(ND,NPP)=X                                                      PRC10027
   30 CONTINUE                                                            PRC10028
   20 CONTINUE                                                            PRC10029
   10 CONTINUE                                                            PRC10030
      RETURN                                                              PRC10031
      END                                                                 PRC10032
```

```
      SUBROUTINE TEMP(FLUX,FISS,HX,HY,ISTR,IEND,NP1X,NP2X,TMPP,SORC1,     TEMP0001
     X  SORC2,DT,JFIRST)                                                  TEMP0002
C                                                                         TEMP0003
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF              TEMP0004
C                                                                         TEMP0005
      DIMENSION FLUX(2,1),FISS(2,1),HX(1),HY(1),ISTR(1),IEND(1),TMPP(1),  TEMP0006
     X  SORC1(1),SORC2(1)                                                 TEMP0007
      LOGICAL JFIRST                                                      TEMP0008
      IF(ITHFB  .EQ.  0) RETURN                                           TEMP0009
      DO 10 NP2=1,NP2X                                                    TEMP0010
      IS=ISTR(NP2+1)                                                      TEMP0011
      IE=IEND(NP2+1)                                                      TEMP0012
      DO 20 ND1=IS,IE                                                     TEMP0013
      NP1=ND1-1                                                           TEMP0014
      NPP=(NP2-1)*NP1X+NP1                                                TEMP0015
      X=FLUX(1,NPP)*FISS(1,NPP)+FLUX(2,NPP)*FISS(2,NPP)                   TEMP0016
      HH=HX(NP1)*HY(NP2)*XNU                                              TEMP0017
      SNEW=X/HH                                                           TEMP0018
      SOLD=SORC1(NPP)                                                     TEMP0019
      IF(JFIRST) SOLD=SNEW                                                TEMP0020
      SHALF=(SNEW+SOLD)*0.5                                               TEMP0021
      SORC1(NPP)=SNEW                                                     TEMP0022
      TNEW=TMPP(NPP)+ALFA*DT*SHALF                                        TEMP0023
      TMPP(NPP)=TNEW                                                      TEMP0024
   20 CONTINUE                                                            TEMP0025
   10 CONTINUE                                                            TEMP0026
      RETURN                                                              TEMP0027
      END                                                                 TEMP0028
```

```
      SUBROUTINE CORE                                                    CORE0001
C                                                                        CORE0002
C                                                                        CORE0003
C                                                                        CORE0004
C=========STORAGE                                                        CORE0005
C                                                                        CORE0006
      COMMON DATA(1)                                                     CORE0007
      REAL       BT(999),AT(99),A(1)                                     CORE0008
      INTEGER    DATA,AV,KOT(999),ITYP(1),KXT(999)                       CORE0009
      INTEGER    NT(999),ITYPET(99)                                      CORE0010
C                                                                        CORE0011
C                                                                        CORE0012
C                                                                        CORE0013
C                                                                        CORE0014
C                                                                        CORE0015
C=========GETCOR                                                         CORE0016
C                                                                        CORE0017
      ENTRY GETCOR(KMAX)                                                 CORE0018
C                                                                        CORE0019
       CALL ZIGET(DATA(1),KMAX,4,KS,5000,&905)                           CORE0020
      KFREE=KS                                                           CORE0021
      KMX =KMAX+KS-1                                                     CORE0022
      KS=KS+MOD(KS+1,2)                                                  CORE0023
      KMAX=KMX-KS+1                                                      CORE0024
      GO TO 900                                                          CORE0025
C                                                                        CORE0026
C                                                                        CORE0027
C=========CLEAR                                                          CORE0028
C                                                                        CORE0029
      ENTRY CLEAR                                                        CORE0030
C                                                                        CORE0031
      IN=5                                                               CORE0032
      IOUT=6                                                             CORE0033
      IERFL=0                                                            CORE0034
      IXX=999                                                            CORE0035
      DO 110 K=KS,KMX                                                    CORE0036
```

```
  110 DATA(K)=0                                                    CORE0037
      DO 120 I=1,IXX                                               CORE0038
      BT(I)=0.0                                                    CORE0039
      NT(I)=0                                                      CORE0040
      KOT(I)=0                                                     CORE0041
  120 KXT(I)=0                                                     CORE0042
      IX=0                                                         CORE0043
      KLST=0                                                       CORE0044
      JX=0                                                         CORE0045
      IP=0                                                         CORE0046
      JXX=99                                                       CORE0047
      DO 125 I=1,JXX                                               CORE0048
      AT(I)=0.0                                                    CORE0049
  125 ITYPET(I)=0                                                  CORE0050
      GO TO 900                                                    CORE0051
C                                                                  CORE0052
C                                                                  CORE0053
C==========FREE DATA SPACE                                         CORE0054
C                                                                  CORE0055
      ENTRY FRECOR                                                 CORE0056
      CALL ZIFREE (DATA(KFREE),&907)                               CORE0057
      GO TO 900                                                    CORE0058
C                                                                  CORE0059
C                                                                  CORE0060
C=========ALLOCATE                                                 CORE0061
C                                                                  CORE0062
      ENTRY ALOC(B,N,KORG,KN)                                      CORE0063
      KY=KN+MOD(KN,2)                                              CORE0064
      IF(KY.LE.0) GO TO 900                                        CORE0065
      IF(N.LT.0) GO TO 900                                         CORE0066
      KAY=2                                                        CORE0067
      ASSIGN 150 TO KAYP                                           CORE0068
      GO TO 300                                                    CORE0069
C                                                                  CORE0070
  150 IF(IX.EQ.0) KORG=KS                                          CORE0071
      IF(IX.GT.0) KORG=KOT(IX)+KXT(IX)                             CORE0072
```

```
      KXP=KORG+KY-1                                                    CORE0073
      IF(KXP.GT.KMX) GO TO 901                                         CORE0074
      IF(KXP.GT.KLST) KLST=KXP                                         CORE0075
      IX=IX+1                                                          CORE0076
      IF(IX.GT.IXX) GO TO 902                                          CORE0077
      BT(IX)=B                                                         CORE0078
      NT(IX)=N                                                         CORE0079
      KOT(IX)=KORG                                                     CORE0080
      KXT(IX)=KY                                                       CORE0081
      IF(KAY .EQ. 7) GO TO 840                                         CORE0082
      GO TO 900                                                        CORE0083
C                                                                      CORE0084
C                                                                      CORE0085
C=========LENGTH                                                       CORE0086
C                                                                      CORE0087
      ENTRY LENGTH(B,N,KXXX, *  )                                      CORE0088
C                                                                      CORE0089
C                                                                      CORE0090
C=========FIND                                                         CORE0091
C                                                                      CORE0092
      ENTRY FIND(B,N,KORG, *  )                                        CORE0093
      KEY=1                                                            CORE0094
      ASSIGN 230 TO KEYP                                               CORE0095
      ASSIGN 900 TO KOYP                                               CORE0096
200   IS=MAX0(IP,1)                                                    CORE0097
      IF (IS.GT.IX) IS=1                                               CORE0098
      IE=IX                                                            CORE0099
205   DO 210 I=IS,IE                                                   CORE0100
      IF(B.NE.BT(I)) GO TO 210                                         CORE0101
      IF(N.NE.NT(I).AND.N.GE.0) GO TO 210                              CORE0102
      IP=I                                                             CORE0103
      GO TO 220                                                        CORE0104
C                                                                      CORE0105
  210 CONTINUE                                                         CORE0106
      IE=IP-1                                                          CORE0107
      IP=0                                                             CORE0108
```

```
      IF(IS.EQ.1) GO TO 220                                                  CORE0109
      IS=1                                                                   CORE0110
      GO TO 205                                                              CORE0111
C                                                                            CORE0112
  220 GO TO KEYP, (230,250,310,        560,620,670,    830,900)              CORE0113
C                                                                            CORE0114
  230 IF(IP.NE.0) GO TO 231                                                  CORE0115
      GO TO 999                                                              CORE0116
C                                                                            CORE0117
  231 KXXX=KXT(IP)                                                           CORE0118
      KORG=KOT(IP)                                                           CORE0119
      GO TO 900                                                              CORE0120
C                                                                            CORE0121
C                                                                            CORE0122
C=========FILL                                                               CORE0123
C                                                                            CORE0124
      ENTRY FILL(B,N,KORG,AV)                                                CORE0125
      KEY=2                                                                  CORE0126
      ASSIGN 250 TO KEYP                                                     CORE0127
      ASSIGN 900 TO KOYP                                                     CORE0128
      GO TO 200                                                              CORE0129
  250 IF(IP.EQ.0) GO TO 900                                                  CORE0130
      KORG=KOT(IP)                                                           CORE0131
      KX=KXT(IP)                                                             CORE0132
      DO 260 K=1,KX                                                          CORE0133
  260 DATA(KORG+K-1)=AV                                                      CORE0134
      GO TO 900                                                              CORE0135
C                                                                            CORE0136
C                                                                            CORE0137
C=========DROP                                                               CORE0138
C                                                                            CORE0139
      ENTRY DROP(B,N)                                                        CORE0140
  290 KAY=1                                                                  CORE0141
      ASSIGN 900 TO KAYP                                                     CORE0142
  300 KEY=3                                                                  CORE0143
      ASSIGN 310 TO KEYP                                                     CORE0144
```

```
      ASSIGN 900 TO KOYP                                                CORE0145
      GO TO 200                                                         CORE0146
C                                                                       CORE0147
  310 IF(IP.EQ.0) GO TO 380                                             CORE0148
      IF(KAY  .NE.  2) GO TO 315                                        CORE0149
      KK=KXT(IP)                                                        CORE0150
      IF (KY .NE. KK)  GO TO 315                                        CORE0151
      AV=0                                                              CORE0152
      GO TO 250                                                         CORE0153
  315 CONTINUE                                                          CORE0154
C                                                                       CORE0155
      IF(IP.EQ.IX) GO TO 350                                            CORE0156
      KL=KOT(IP+1)                                                      CORE0157
      KH=KOT(IX)+KXT(IX)-1                                              CORE0158
      KX=KXT(IP)                                                        CORE0159
      DO 320 K=KL,KH                                                    CORE0160
  320 DATA(K-KX)=DATA(K)                                                CORE0161
      KL=KH-KX+1                                                        CORE0162
      DO 330 K=KL,KH                                                    CORE0163
  330 DATA(K)=0                                                         CORE0164
      IL=IP+1                                                           CORE0165
      DO 340 I=IL,IX                                                    CORE0166
      BT(I-1)=BT(I)                                                     CORE0167
      KOT(I-1)=KOT(I)-KX                                                CORE0168
      NT(I-1)=NT(I)                                                     CORE0169
      KXT(I-1)=KXT(I)                                                   CORE0170
  340 CONTINUE                                                          CORE0171
      GO TO 370                                                         CORE0172
C                                                                       CORE0173
  350 KO=KOT(IP)                                                        CORE0174
      KX=KXT(IP)                                                        CORE0175
      DO 360 K=1,KX                                                     CORE0176
  360 DATA(KO+K-1)=0                                                    CORE0177
  370 BT(IX)=0.0                                                        CORE0178
      NT(IX)=0                                                          CORE0179
      KOT(IX)=0                                                         CORE0180
```

```
      KXT(IX)=0                                                          CORE0181
      IX=IX-1                                                            CORE0182
C                                                                        CORE0183
  380 GO TO KAYP, (150,510,     900)                                     CORE0184
C                                                                        CORE0185
C                                                                        CORE0186
C=========LIST                                                           CORE0187
C                                                                        CORE0188
      ENTRY LIST                                                         CORE0189
  385 IF(IX.LE.0) GO TO 900                                              CORE0190
      WRITE(IOUT,390) IX,(BT(I),NT(I),KOT(I),KXT(I),I=1,IX)              CORE0191
  390 FORMAT(1H0,I3,7H BLOCKS/5H NAME,8X,13HORIGIN LENGTH/               CORE0192
     $       1X,10H----------,2(2X,6H------)/(1X,A4,I6,2I8))             CORE0193
      IF(IERFL.EQ.1) GO TO 386                                           CORE0194
      GO TO 900                                                          CORE0195
C                                                                        CORE0196
C                                                                        CORE0197
C=========COUNT                                                          CORE0198
C                                                                        CORE0199
      ENTRY COUNT                                                        CORE0200
      WRITE(IOUT,500) KLST                                               CORE0201
  500 FORMAT(28H0LAST DATA LOCATION USED WAS,I6)                         CORE0202
      GO TO 900                                                          CORE0203
C                                                                        CORE0204
C                                                                        CORE0205
C=========OPEN                                                           CORE0206
C                                                                        CORE0207
      ENTRY OPEN(B,N,KORG)                                               CORE0208
      IF(N.LT.0) GO TO 900                                               CORE0209
      KAY=3                                                              CORE0210
      ASSIGN 510 TO KAYP                                                 CORE0211
      GO TO 300                                                          CORE0212
C                                                                        CORE0213
  510 IF(IX.EQ.0) KORG=KS                                                CORE0214
      IF(IX.GT.0) KORG=KOT(IX)+KXT(IX)                                   CORE0215
      IF(IX.GE.IXX) GO TO 902                                            CORE0216
```

```
      BT(IX+1)=B                                                        CORE0217
      NT(IX+1)=N                                                        CORE0218
      KOT(IX+1)=KORG                                                    CORE0219
      GO TO 900                                                         CORE0220
C                                                                       CORE0221
C                                                                       CORE0222
C=========CLOSE                                                         CORE0223
C                                                                       CORE0224
      ENTRY CLOSE(B,N,KXN)                                              CORE0225
      IX=IX+1                                                           CORE0226
      IF(IX.GT.IXX) GO TO 902                                           CORE0227
      IF(B.NE.BT(IX).OR.N.NE.NT(IX)) GO TO 903                          CORE0228
      KXX=KXN+MOD(KXN,2)+KOT(IX)-1                                      CORE0229
      IF(KXX.GT.KMX) GO TO 901                                          CORE0230
      KXT(IX)=KXX-KOT(IX)+1                                             CORE0231
      IF(KXX.GT.KLST) KLST=KXX                                          CORE0232
      GO TO 900                                                         CORE0233
C                                                                       CORE0234
C                                                                       CORE0235
C=========COPYB                                                         CORE0236
C                                                                       CORE0237
      ENTRY COPYB(AA,M,BP,NP,KORG, *  )                                 CORE0238
      NCOPYB=1                                                          CORE0239
      B=AA                                                              CORE0240
      N=M                                                               CORE0241
      KEY=21                                                            CORE0242
      ASSIGN 830 TO KEYP                                                CORE0243
      ASSIGN 900 TO KOYP                                                CORE0244
      GO TO 200                                                         CORE0245
C                                                                       CORE0246
 830  IF(IP.NE.0)  GO TO 831                                            CORE0247
      GO TO 999                                                         CORE0248
C                                                                       CORE0249
  831 IF(NCOPYB.EQ.2) GO TO 832                                         CORE0250
      B=BP                                                              CORE0251
      N=NP                                                              CORE0252
```

```
      KY=KXT(IP)                                                        CORE0253
      KAY=7                                                             CORE0254
      ASSIGN 150 TO KAYP                                                CORE0255
      GO TO 300                                                         CORE0256
C                                                                       CORE0257
  840 K2=KORG                                                           CORE0258
      NCOPYB=2                                                          CORE0259
      B=AA                                                              CORE0260
      N=M                                                               CORE0261
      KEY=21                                                            CORE0262
      ASSIGN 830 TO KEYP                                                CORE0263
      ASSIGN 900 TO KOYP                                                CORE0264
      GO TO 200                                                         CORE0265
C                                                                       CORE0266
  832 K1=KOT(IP)                                                        CORE0267
      KX1=KXT(IP)                                                       CORE0268
      DO 850  K=1,KX1                                                   CORE0269
  850 DATA(K2-1+K)=DATA(K1-1+K)                                         CORE0270
      GO TO 900                                                         CORE0271
C                                                                       CORE0272
C                                                                       CORE0273
C=========CLIP                                                          CORE0274
C                                                                       CORE0275
      ENTRY CLIP(B,N,KN)                                                CORE0276
      IF (KN.LE.0) GO TO 290                                            CORE0277
      KEY=18                                                            CORE0278
      ASSIGN 560 TO KEYP                                                CORE0279
      ASSIGN 900 TO KOYP                                                CORE0280
      GO TO 200                                                         CORE0281
C                                                                       CORE0282
  560 IF(IP.LE.0) GO TO 900                                             CORE0283
      IF(IP.LT.IX) GO TO 565                                            CORE0284
      IF (KN.GT.KXT(IP)) GO TO 904                                      CORE0285
      KH=KOT(IX)+KXT(IX)-1                                              CORE0286
      KXT(IX)=KN+MOD(KN,2)                                              CORE0287
      KL=KOT(IX)+KXT(IX)                                                CORE0288
```

```
      DO 564 K=KL,KH                                                    CORE0289
      DATA(K)=0                                                         CORE0290
564   CONTINUE                                                          CORE0291
      GO TO 900                                                         CORE0292
C                                                                       CORE0293
  565 KL=KOT(IP+1)                                                      CORE0294
      KH=KOT(IX)+KXT(IX)-1                                              CORE0295
      KX=KXT(IP)-KN-MOD(KN,2)                                           CORE0296
      IF(KX.LT.1) GO TO 904                                             CORE0297
      DO 570 K=KL,KH                                                    CORE0298
  570 DATA(K-KX)=DATA(K)                                                CORE0299
      KXT(IP)=KN+MOD(KN,2)                                              CORE0300
      IL=IP+1                                                           CORE0301
      DO 575 I=IL,IX                                                    CORE0302
  575 KOT(I)=KOT(I)-KX                                                  CORE0303
      KO=KOT(IX)+KXT(IX)-1                                              CORE0304
      DO 580 K=1,KX                                                     CORE0305
  580 DATA(KO+K)=0                                                      CORE0306
      GO TO 900                                                         CORE0307
C                                                                       CORE0308
C                                                                       CORE0309
C=========SETNAM                                                        CORE0310
C                                                                       CORE0311
      ENTRY SETNAM(A,ITYP,JXP)                                          CORE0312
      IF(JXP.LT.1) GO TO 900                                            CORE0313
      KEY=6                                                             CORE0314
      ASSIGN 900 TO KEYP                                                CORE0315
      ASSIGN 600 TO KOYP                                                CORE0316
      DO 610 I=1,JXP                                                    CORE0317
      AP=A(I)                                                           CORE0318
      GO TO 620                                                         CORE0319
C                                                                       CORE0320
600   IF(JP.GT.0) GO TO 610                                             CORE0321
      IF (JX+1.GT.JXX) GO TO 902                                        CORE0322
      JX=JX+1                                                           CORE0323
      JP=JX                                                             CORE0324
```

```
      AT(JP)=AP                                                         CORE0325
      ITYPET(JP)=ITYP(I)                                                CORE0326
610   CONTINUE                                                          CORE0327
      GO TO 900                                                         CORE0328
C                                                                       CORE0329
C---------SEARCH TABLE FOR AP.  SET JP                                  CORE0330
C                                                                       CORE0331
620   JP=0                                                              CORE0332
      DO 630 J=1,JX                                                     CORE0333
      IF (AP.NE.AT(J)) GO TO 630                                        CORE0334
      JP=J                                                              CORE0335
      GO TO 640                                                         CORE0336
C                                                                       CORE0337
630   CONTINUE                                                          CORE0338
C                                                                       CORE0339
  640 GO TO KOYP, (600,    680,           750,                          CORE0340
     X             760,780,790,           900)                         CORE0341
C                                                                       CORE0342
C                                                                       CORE0343
C=========PRINT BLOCK                                                   CORE0344
C                                                                       CORE0345
      ENTRY PRINT(B,N,I1X)                                              CORE0346
      KEY=8                                                             CORE0347
      ASSIGN 670 TO KEYP                                                CORE0348
      ASSIGN 680 TO KOYP                                                CORE0349
      GO TO 200                                                         CORE0350
C                                                                       CORE0351
670   IF(IP.EQ.0) GO TO 900                                             CORE0352
      KORG=KOT(IP)                                                      CORE0353
      LEN=KXT(IP)                                                       CORE0354
      AP=B                                                              CORE0355
      GO TO 620                                                         CORE0356
C                                                                       CORE0357
  680 IF(JP.EQ.0) GO TO 900                                             CORE0358
      IF(JP.NE.0) ITYPE=ITYPET(JP)                                      CORE0359
      CALL GPRNT(B,N,LEN,     KORG ,ITYPE,I1X)                          CORE0360
```

```
      GO TO 900                                                         CORE0361
C                                                                       CORE0362
C                                                                       CORE0363
C=========EXCHANGE NAME                                                 CORE0364
C                                                                       CORE0365
      ENTRY EXNAME(AA,M,BP,NP, *  )                                     CORE0366
      KEY=14                                                            CORE0367
      ASSIGN 620 TO KEYP                                                CORE0368
      ASSIGN 780 TO KOYP                                                CORE0369
      N=NP                                                              CORE0370
      B=BP                                                              CORE0371
      AP=BP                                                             CORE0372
      GO TO 200                                                         CORE0373
C                                                                       CORE0374
750   IF(IP.GT.0) GO TO 751                                             CORE0375
      GO TO 999                                                         CORE0376
C                                                                       CORE0377
  751 I1P=IP                                                            CORE0378
      J1P=JP                                                            CORE0379
      AP=AA                                                             CORE0380
      KEY=15                                                            CORE0381
      ASSIGN 620 TO KEYP                                                CORE0382
      ASSIGN 760 TO KOYP                                                CORE0383
      B=AA                                                              CORE0384
      N=M                                                               CORE0385
      GO TO 200                                                         CORE0386
C                                                                       CORE0387
760   IF (IP.GT.0) GO TO 761                                            CORE0388
      GO TO 999                                                         CORE0389
C                                                                       CORE0390
  761 BT(IP)=BT(I1P)                                                    CORE0391
      NT(IP)=NT(I1P)                                                    CORE0392
      BT(I1P)=AA                                                        CORE0393
      NT(I1P)=M                                                         CORE0394
      IF(J1P.GT.0)  AT(J1P)=AA                                          CORE0395
      IF(JP.GT.0)  AT(JP)=BP                                            CORE0396
```

```
      GO TO 900                                                         CORE0397
C                                                                       CORE0398
C                                                                       CORE0399
C=========RE-NAME                                                       CORE0400
C                                                                       CORE0401
      ENTRY RENAME (AA,M,BP,NP, *  )                                    CORE0402
      KEY=16                                                            CORE0403
      ASSIGN 620 TO KEYP                                                CORE0404
      ASSIGN 780 TO KOYP                                                CORE0405
      B=BP                                                              CORE0406
      N=NP                                                              CORE0407
      AP=B                                                              CORE0408
      GO TO 200                                                         CORE0409
C                                                                       CORE0410
 780  CONTINUE                                                          CORE0411
      IF(IP.LE.0) GO TO 781                                             CORE0412
      GO TO 999                                                         CORE0413
C                                                                       CORE0414
  781 IF(JP.NE.0) GO TO 782                                             CORE0415
      JX=JX+1                                                           CORE0416
      IF(JX.GT.JXX)  GO TO 902                                          CORE0417
      JP=JX                                                             CORE0418
      AT(JX)=AP                                                         CORE0419
C                                                                       CORE0420
  782 KEY=17                                                            CORE0421
      ASSIGN 620 TO KEYP                                                CORE0422
      ASSIGN 790 TO KOYP                                                CORE0423
      AP=AA                                                             CORE0424
      N=M                                                               CORE0425
      B=AA                                                              CORE0426
      GO TO 200                                                         CORE0427
C                                                                       CORE0428
 790  CONTINUE                                                          CORE0429
      IF(IP.GE.1) GO TO 791                                             CORE0430
      GO TO 999                                                         CORE0431
C                                                                       CORE0432
```

```
  791 BT(IP)=BP                                                          CORE0433
      NT(IP)=NP                                                          CORE0434
      GO TO 900                                                          CORE0435
C                                                                        CORE0436
C                                                                        CORE0437
C=========COMPUTE CORE REMAINING                                         CORE0438
C                                                                        CORE0439
      ENTRY CREM (KXA)                                                   CORE0440
      KXA=KMX-(KOT(IX)+KXT(IX))+1                                        CORE0441
C                                                                        CORE0442
C                                                                        CORE0443
C=========FINISHED                                                       CORE0444
C                                                                        CORE0445
  900 RETURN                                                             CORE0446
C                                                                        CORE0447
C                                                                        CORE0448
  999 RETURN 1                                                           CORE0449
C                                                                        CORE0450
C                                                                        CORE0451
C                                                                        CORE0452
C=========ERROR STOPS                                                    CORE0453
C                                                                        CORE0454
  901 WRITE(IOUT,9010)                                                   CORE0455
 9010 FORMAT(33H0*** CORE *** DATA ARRAY OVERFLOW)                       CORE0456
      IERFL=1                                                            CORE0457
      GO TO 385                                                          CORE0458
 386  CONTINUE                                                           CORE0459
      KXA=KMX-(KOT(IX)+KXT(IX))+1                                        CORE0460
      WRITE(IOUT,162)  KXA                                               CORE0461
  162 FORMAT(30X,29H0UNUSED CORE (  IN WORDS  ) =,I10)                   CORE0462
      STOP 1                                                             CORE0463
  902 WRITE(IOUT,9020)                                                   CORE0464
 9020 FORMAT(32H0*** CORE *** CATALOGUE OVERFLOW)                        CORE0465
      KXA=KMX-(KOT(IX)+KXT(IX))+1                                        CORE0466
      WRITE(IOUT,162)  KXA                                               CORE0467
      STOP 1                                                             CORE0468
```

```
  903 WRITE(IOUT,9030)                                                  CORE0469
 9030 FORMAT(38H0*** CORE *** CLOSING BLOCK IMPROPERLY)                 CORE0470
      KXA=KMX-(KOT(IX)+KXT(IX))+1                                       CORE0471
      WRITE(IOUT,162)  KXA                                              CORE0472
      STOP 1                                                            CORE0473
  904 WRITE(IOUT,9040)                                                  CORE0474
 9040 FORMAT(33H0*** CORE *** IMPROPER CLIP ENTRY)                      CORE0475
      KXA=KMX-(KOT(IX)+KXT(IX))+1                                       CORE0476
      WRITE(IOUT,162)  KXA                                              CORE0477
      STOP 1                                                            CORE0478
  905 WRITE(IOUT,9050)                                                  CORE0479
 9050 FORMAT(41H0********** PROBLEM WITH ZIGET **********)              CORE0480
      STOP 1                                                            CORE0481
  907 WRITE(IOUT,9070)                                                  CORE0482
 9070 FORMAT(42H0********** PROBLEM WITH ZIFREE **********)             CORE0483
      STOP 1234                                                         CORE0484
      END                                                               CORE0485
```

```
      SUBROUTINE GPRNT(BL,NBL,LX,KORG   ,IDT,I1X)                       GPRN0001
C                                                                       GPRN0002
      COMMON         DATA(1)                                            GPRN0003
      INTEGER    I4DATA(1)                                             GPRN0004
      REAL    R4DATA(1)                                                GPRN0005
      DOUBLE PRECISION R8DATA(1)                                        GPRN0006
      EQUIVALENCE (R8DATA(1),R4DATA(1),I4DATA(1),DATA(1))               GPRN0007
C                                                                       GPRN0008
C                                                                       GPRN0009
       INTEGER    FMTI4(5),FMTR48( 7),INTGR(20)                         GPRN0010
      DATA   FMTI4 / 20H(1H+,12X,        I6/) /,                       GPRN0011
     $       FMTR48 / 28H(1H+,11X,         (1PE12.4)/)  /,             GPRN0012
     $       INTGR /  80H    1   2   3   4   5   6   7   8   9  10  11  12  GPRN0013
     $  13   14   15  16  17  18  19  20  /                             GPRN0014
C                                                                       GPRN0015
C                                                                       GPRN0016
      IN=5                                                              GPRN0017
      IOUT=6                                                            GPRN0018
C                                                                       GPRN0019
C                                                                       GPRN0020
      LO=0                                                              GPRN0021
      LOX=0                                                             GPRN0022
      NPL=10                                                            GPRN0023
       LIX=I1X                                                          GPRN0024
C                                                                       GPRN0025
C                                                                       GPRN0026
      IF((IDT.GE.1).AND.(IDT.LE.3)) GO TO 109                           GPRN0027
      WRITE(IOUT,9020) BL,NBL,IDT                                       GPRN0028
      GO TO 900                                                         GPRN0029
C                                                                       GPRN0030
C                                                                       GPRN0031
 109  GO TO (120,110,120                          ),IDT                GPRN0032
  110 NPL=20                                                            GPRN0033
C                                                                       GPRN0034
C                                                                       GPRN0035
  120 IF(LIX.GT.1) GO TO 130                                            GPRN0036
```

```
      GO TO (121,121,122                                ), IDT      GPRN0037
  121 LIX=LX                                                        GPRN0038
      GO TO 300                                                     GPRN0039
  122 LIX=LX/2                                                      GPRN0040
      GO TO 300                                                     GPRN0041
C                                                                   GPRN0042
C                                                                   GPRN0043
  130 GO TO (140,140,150                                ), IDT      GPRN0044
  140 LOX=LX/LIX                                                    GPRN0045
      GO TO 160                                                     GPRN0046
  150 LX=LX/2                                                       GPRN0047
      LOX=LX/LIX                                                    GPRN0048
  160 CONTINUE                                                      GPRN0049
C                                                                   GPRN0050
C                                                                   GPRN0051
  300 WRITE(IOUT,6010) BL,NBL                                       GPRN0052
      IF(LOX.EQ.0) GO TO 330                                        GPRN0053
      WRITE(IOUT,7010)                                              GPRN0054
  330 IPL=NPL/10                                                    GPRN0055
      GO TO (340,350),IPL                                           GPRN0056
  340 WRITE(IOUT,7021)                                              GPRN0057
      GO TO 360                                                     GPRN0058
  350 WRITE(IOUT,7022)                                              GPRN0059
  360 IF(LOX.EQ.0) GO TO 370                                        GPRN0060
      WRITE(IOUT,7030)                                              GPRN0061
  370 WRITE(IOUT,7040)                                              GPRN0062
C                                                                   GPRN0063
C                                                                   GPRN0064
  200 LO=LO+1                                                       GPRN0065
C                                                                   GPRN0066
C                                                                   GPRN0067
      GO TO (204,204,205                                ), IDT      GPRN0068
  204 IR=(LO-1)*LIX+KORG-1                                          GPRN0069
      GO TO 206                                                     GPRN0070
  205 IR=(LO-1)*LIX+(KORG-1)/2                                      GPRN0071
C                                                                   GPRN0072
```

```
C                                                                       GPRN0073
  206 IF(LOX.EQ.C) GO TO 202                                            GPRN0074
      IF(LO.EQ.1) GO TO 201                                             GPRN0075
      IF(LIX.LE.NPL) GO TO 201                                          GPRN0076
      WRITE(IOUT,7050)                                                  GPRN0077
  201 WRITE(IOUT,8030) LO                                               GPRN0078
C                                                                       GPRN0079
C                                                                       GPRN0080
  202 L1=0                                                              GPRN0081
      L2=0                                                              GPRN0082
  207 L1=L2+1                                                           GPRN0083
      LR=L1-1                                                           GPRN0084
      L2=LR+NPL                                                         GPRN0085
      IF(L2.GT.LIX) L2=LIX                                              GPRN0086
      IF(LOX.EQ.0) GO TO 203                                            GPRN0087
      WRITE(IOUT,7060)                                                  GPRN0088
  203 WRITE(IOUT,8010) LR                                               GPRN0089
C                                                                       GPRN0090
      GO TO (210,230,220                                  ), IDT        GPRN0091
  210 FMTR48(4)=INTGR(L2-L1+1)                                          GPRN0092
      WRITE(IOUT,FMTR48)                                                GPRN0093
     $                (R4DATA(IR+I),I=L1,L2)                           GPRN0094
      GO TO 240                                                         GPRN0095
  220 FMTR48(4)=INTGR(L2-L1+1)                                          GPRN0096
      WRITE(IOUT,FMTR48)                                                GPRN0097
     $                (R8DATA(IR+I),I=L1,L2)                           GPRN0098
      GO TO 240                                                         GPRN0099
  230 FMTI4(4)=INTGR(L2-L1+1)                                           GPRN0100
      WRITE(IOUT,FMTI4)                                                 GPRN0101
     $               (I4DATA(IR+I),I=L1,L2)                           GPRN0102
C                                                                       GPRN0103
  240 IF(L2.EQ.LIX) GO TO 250                                           GPRN0104
      GO TO 207                                                         GPRN0105
C                                                                       GPRN0106
  250 IF(LOX.EQ.0) GO TO 900                                            GPRN0107
      IF(LO.LT.LOX) GO TO 200                                           GPRN0108
```

```
C                                                                       GPRN0109
C                                                                       GPRN0110
  900 RETURN                                                            GPRN0111
C                                                                       GPRN0112
C                                                                       GPRN0113
 6010 FORMAT(1HC,60X,16H================/                              GPRN0114
     $       1X,60X,3H||  ,A4,1H.,I5,3H ||/                             GPRN0115
     $       1X,60X,16H================///)                             GPRN0116
 7010 FORMAT(1H+,1X,2HI2,1X,1H|)                                        GPRN0117
 7021 FORMAT(1H+,7X,2HI1,1X,1H/,                                        GPRN0118
     $       6X,2H+1,10X,2H+2,10X,2H+3,10X,2H+4,10X,2H+5,               GPRN0119
     $       10X,2H+6,10X,2H+7,10X,2H+8,10X,2H+9,10X,3H+10/)            GPRN0120
 7022 FORMAT(1H+,7X,2HI1,1X,1H/,                                        GPRN0121
     $       4X,2H+1,4X,2H+2,4X,2H+3,4X,2H+4,4X,2H+5,                   GPRN0122
     $       4X,2H+6,4X,2H+7,4X,2H+8,4X,2H+9,3X,3H+10,                  GPRN0123
     $       3X,3H+11,3X,3H+12,3X,3H+13,3X,3H+14,3X,3H+15,              GPRN0124
     $       3X,3H+16,3X,3H+17,3X,3H+18,3X,3H+19,3X,3H+20/)             GPRN0125
 7030 FORMAT(1H+,5(1H-)/1H+,4X,1H|)                                     GPRN0126
 7040 FORMAT(1H+,5X,127(1H-)/1H+,10X,1H|/)                              GPRN0127
 7050 FORMAT(1H+,132(1H-)/1H+,4X,1H|,5X,1H|/)                           GPRN0128
 7060 FORMAT(1H+,4X,1H|)                                                GPRN0129
 8010 FORMAT(1H+,5X,I4,1X,1H|)                                          GPRN0130
 8030 FORMAT(1H+,I3)                                                    GPRN0131
 9020 FORMAT(//1X,45HDATA TYPE INTEGER QUALIFIER OUT OF RANGE FOR ,A4,I5 GPRN0132
     $      /1X,10HDATA TYPE ,I4//)                                     GPRN0133
      END                                                               GPRN0134
```

```
*      CALLING SEQUENCE                                                    ZIGT0001
*           CALL ZIGET (A,NDIM,LENGTH,ISUB,OVHD,*) WHERE                   ZIGT0002
*           A = DUMMY DIMENSIONED VARIABLE FROM CALLING PROGRAM             ZIGT0003
*          NDIMS = TOTAL DIMENSION SIZE OF A.                              ZIGT0004
*          LENGTH = LENGTH SPECIFICATION FOR FIRST ARGUMENT                ZIGT0005
*          OVHD = OVERHEAD SPACE TO BE RESERVED                            ZIGT0006
*          ISUB = INCREMENTAL SUBSCRIPT VALUE (FOR NORMAL RETURN)          ZIGT0007
*       FOR ERROR RETURN                                                   ZIGT0008
*          ISUB = +,  MAX. NUMBER OF WORDS AVAILABLE                       ZIGT0009
*          ISUB = -999, DYNAMIC STORAGE AREA NOT ACCESSIBLE                ZIGT0010
*                                                                          ZIGT0011
*          IF THE WRONG NO. OF ARGUMENTS ARE PASSED FROM THE CALLING       ZIGT0012
*          PROGRAM, A FORTRAN TRACEBACK IS GIVEN (IHC230I)                 ZIGT0013
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ZIGT0014
ZIGET    CSECT                                                             ZIGT0015
R0       EQU   0                                                           ZIGT0016
R1       EQU   1                                                           ZIGT0017
R2       EQU   2                                                           ZIGT0018
R3       EQU   3                                                           ZIGT0019
R4       EQU   4                                                           ZIGT0020
R5       EQU   5                                                           ZIGT0021
R6       EQU   6                                                           ZIGT0022
R7       EQU   7                                                           ZIGT0023
R8       EQU   8                                                           ZIGT0024
R9       EQU   9                                                           ZIGT0025
RA       EQU   10                                                          ZIGT0026
RB       EQU   11                                                          ZIGT0027
RC       EQU   12                                                          ZIGT0028
RD       EQU   13                                                          ZIGT0029
RE       EQU   14                                                          ZIGT0030
RF       EQU   15                                                          ZIGT0031
R10      EQU   10                                                          ZIGT0032
R11      EQU   11                                                          ZIGT0033
R12      EQU   12                                                          ZIGT0034
R13      EQU   13                                                          ZIGT0035
R14      EQU      14                                                       ZIGT0036
```

```
R15      EQU   15                                                    ZIGT0037
         BC    15,12(15)                                             ZIGT0038
         DC    X'7'                                                  ZIGT0039
         DC    CL7'ZIGET  '                                          ZIGT0040
GET1     STM   14,12,12(13)          SAVE REGISTERS.                 ZIGT0041
         LR    12,15                                                 ZIGT0042
         USING ZIGET,12                                              ZIGT0043
*                                                                    ZIGT0044
         SR    15,15                                                 ZIGT0045
         SR    7,7                                                   ZIGT0046
         ST    7,LENGTH              MINIMUM LENGTH                  ZIGT0047
         LM    2,5,0(1)              LOAD ADDRESS OF ARGUMENTS       ZIGT0048
         TM    12(1),X'80'           TEST FOR CORRECT NUMBER OF ARGS.ZIGT0049
         BNZ   PRESET IF RIGHT NO. OF ARGS.,BRANCH                   ZIGT0050
         L     R6,16(R1)                                             ZIGT0051
         L     R6,0(0,R6)                                            ZIGT0052
         SLL   R6,2          GET  VALUE IN WRDS UNITS                ZIGT0053
*        OPEN  (SNAPDCB,(OUTPUT))      FOR DEBUGGING PURPOSE         ZIGT0054
*---------GO GET ALL CORE AVAILABLE                                   ZIGT0055
         BAL   R10,ALLGET                                            ZIGT0056
*--------MAKE SURE SYSTEM OVHD REQUEST IS ON A 2K BOUNDARY           ZIGT0057
         LA    R6,2047(0,R6)                                         ZIGT0058
         SRL   R6,11                                                 ZIGT0059
         SLL   R6,11                                                 ZIGT0060
*---------COMPUTE SYSTEM OVERHEAD                                     ZIGT0061
         L     R9,LENA                                               ZIGT0062
         SR    R9,R6                                                 ZIGT0063
         BNP   RETREQ                                                ZIGT0064
         A     R9,ADDR                                               ZIGT0065
         ST    R9,RESADD                                             ZIGT0066
         ST    R6,RESLGTH                                            ZIGT0067
*---------                                                           ZIGT0068
         BAL   R10,FREEOVHD                                          ZIGT0069
*        SNAP  DCB=SNAPDCB,ID=0,SDATA=(CB),PDATA=REGS      DEBUGG    ZIGT0070
*---------COMPUTE AMT OF AREA TO BE RETURNED TO USER                  ZIGT0071
         BAL   R10,RETALL                                            ZIGT0072
```

```
         ST    R7,0(0,R3)                                                ZIGT0073
*        CLOSE (SNAPDCB)              FOR DEBUGGING PURPOSE             ZIGT0074
         B     CNTRLINF                                                  ZIGT0075
*                                                                        ZIGT0076
         MVC   ISN+2(2),2(14)       PICK UP INTERNAL STATEMENT NUMBER   ZIGT0077
         LA    1,PLIST                                                   ZIGT0078
         L     15,IBER                                                   ZIGT0079
         BALR  14,15                BRANCH TO TRACEBACK ROUTINE        ZIGT0080
*                                                                        ZIGT0081
PRESET   EQU   *                                                        ZIGT0082
         L     7,0(3)               LOAD NO. OF WORDS REQUESTED.       ZIGT0083
         M     6,0(4)               MULTIPLY BY LENGTH SPECIFICATION.  ZIGT0084
         LPR   7,7                  R7= NUMBER OF BYTES TO REQUEST     ZIGT0085
SVE      ST    2,REG2                                                    ZIGT0086
         LA    7,15(0,7)        MAKE SURE REQUEST IS ON 8 BYTE BOUNDARY ZIGT0087
         SRL   7,3              AND ADD 8 BYTES FOR CONTROL WORDS.      ZIGT0088
         SLL   7,3                                                      ZIGT0089
         ST    7,LENGTH+4           REQUESTED LENGTH (MAX)             ZIGT0090
         SPACE 1                                                        ZIGT0091
*---------FIRST, GET ALL AVAILIABLE AREA                                 ZIGT0092
         BAL   R10,ALLGET                                               ZIGT0093
CONT1    EQU   *                                                        ZIGT0094
         C     R7,LENA              COMPARE RET. LGTH WITH REQ. LGTH   ZIGT0095
         BH    RETREQ               CALLER REQUESTED TOO MUCH, GO ADJUST ZIGT0096
         SPACE 1                                                        ZIGT0097
*---------GET REQUEST ONTO 8K BOUNDARY                                   ZIGT0098
         LA    R7,2047(0,R7)                                            ZIGT0099
         SRL   R7,11                                                    ZIGT0100
         SLL   R7,11                                                    ZIGT0101
         ST    R7,ORGLGTH                                               ZIGT0102
*---------COMPUTE SYSTEM OVERHEAD AREA                                   ZIGT0103
         A     R7,ADDR              COMPUTE RESERVE AREA               ZIGT0104
         ST    R7,RESADD            SAVE VALUE                         ZIGT0105
         L     R7,LENA              LOAD LGTH OF AREA OBTAINED         ZIGT0106
         S     R7,ORGLGTH           SUB LGTH OF AREA WANTED            ZIGT0107
         C     R7,ZERO              CHECK FOR NO SYSTEM AREA AVAILIABLE ZIGT0108
```

```
         BNH   CNTRLINF           IF NOT,                                  ZIGT0109
         ST    R7,RESLGTH                                                 ZIGT0110
         SPACE 1                                                          ZIGT0111
*---------FREE SYSTEM OVERHEAD AREA                                       ZIGT0112
         BAL   R10,FREEOVHD                                               ZIGT0113
         SPACE 1                                                          ZIGT0114
*--------SET UP CONTROL INFORMATION INTO OBTAINED AREA                    ZIGT0115
CNTRLINF EQU   *                                                         ZIGT0116
         L     R6,ADDR            GET ADDR OF CORE ALLOCATED            ZIGT0117
         MVC   0(8,6),ADDR         STORE CONTROL WORDS AT BEGINNING     ZIGT0118
         LA    6,8(0,6)            ADD 8 BYTES FOR CONTROL WORDS        ZIGT0119
*                    COMPUTE SUBSCRIPT                                   ZIGT0120
         ST    6,REG6                                                     ZIGT0121
         SR    6,2                 SUBTRACT ADDRESS OF FIRST ARGUMENT   ZIGT0122
         BM    ERROR               IF NEGATIVE, RETURN                  ZIGT0123
         XR    R15,R15                                                    ZIGT0124
         BAL   R10,DIV                                                    ZIGT0125
         B     FINISH                                                     ZIGT0126
ERROR    MVC   0(4,5),ECODE        RETURN CODE-STORAGE NOT ACCESSIBLE   ZIGT0127
ERROR1   LA    15,4(0,0)           ERROR RETURN  (RETURN 1)             ZIGT0128
         BC    15,RETURN                                                  ZIGT0129
FINISH   ST    7,0(0,5)                                                   ZIGT0130
         LA    R11,ZIFREE                                                 ZIGT0131
         B     RETURN                                                     ZIGT0132
         SPACE 1                                                          ZIGT0133
RETREQ   EQU   *                                                         ZIGT0134
*---------CALLER REQUESTED TOO MUCH CORE, FREE OBTAINED AREA AND          ZIGT0135
*             PASS BACK VALUE OF MAX AREA THAT CAN BE OBTAINED           ZIGT0136
         FREEMAIN V,A=ADDR,SP=0                                           ZIGT0137
         LA    R15,4               SET ERROR CODE                       ZIGT0138
         LA    R10,FINISH                                                 ZIGT0139
         L     R6,LENA                                                    ZIGT0140
         S     R6,SYSOVHD                                                 ZIGT0141
         S     R6,EIGHT            ADJUST FOR CONTRL INFORMATION        ZIGT0142
         BNP   RETALL                                                     ZIGT0143
         B     DIV                                                        ZIGT0144
```

```
         SPACE 2                                                        ZIGT0145
*==== ROUTINES FOR ZIGET ==========================================     ZIGT0146
         SPACE 1                                                        ZIGT0147
*--------ROUTINE TO GET ALL CORE AVAILABLE                             ZIGT0148
ALLGET   EQU   *                                                        ZIGT0149
         GETMAIN VC,LA=GETALL,A=ADDR,SP=0                               ZIGT0150
         LTR   R15,R15                                                  ZIGT0151
         BZ    0(0,R10)                                                 ZIGT0152
         ABEND 1020,DUMP                                                ZIGT0153
         SPACE 1                                                        ZIGT0154
*--------ROUTINE TO FREE SYSTEM OVERHEAD                               ZIGT0155
FREEOVHD EQU   *                                                        ZIGT0156
         FREEMAIN V,A=RESADD,SP=0                                       ZIGT0157
         SPACE 1                                                        ZIGT0158
*---------ADJUST LENA FOR CONTROL INFORMATION                           ZIGT0159
         L     R7,LENA                                                  ZIGT0160
         S     R7,RESLGTH                                               ZIGT0161
         ST    R7,LENA                                                  ZIGT0162
         BR    R10                                                      ZIGT0163
*---------ADJUST RETURN REQ FOR CONTRL INFO                             ZIGT0164
RETALL   EQU   *                                                        ZIGT0165
         L     R6,LENA                                                  ZIGT0166
         S     R6,EIGHT                                                 ZIGT0167
*---------COMPUTE  NUMBER OF DIMENSIONS                                 ZIGT0168
DIV      SRDA  6,32(0)                                                  ZIGT0169
         D     6,0(4)                                                   ZIGT0170
         BR    R10                                                      ZIGT0171
         SPACE 2                                                        ZIGT0172
*==== ENTRY POINT FOR ZIFREE ======================================     ZIGT0173
*                                                                       ZIGT0174
*         CALL ZIFREE (A,*) WHERE,                                      ZIGT0175
*         A = ADDRESS OF MAIN STORAGE TO BE FREED                       ZIGT0176
*         * = ERROR RETURN STATEMENT NUMBER                             ZIGT0177
         ENTRY ZIFREE                                                   ZIGT0178
ZIFREE   BC    15,12(0,15)                                              ZIGT0179
         DC    X'7'                                                     ZIGT0180
```

```
         DC    CL7'ZIFREE '                                              ZIGT0181
FREE1    STM   14,12,12(13)        SAVE REGISTERS.                       ZIGT0182
         DROP  12                                                        ZIGT0183
         LR    11,15                                                     ZIGT0184
         USING ZIFREE,11                                                 ZIGT0185
         SR    15,15                                                     ZIGT0186
         TM    0(1),X'80'          TEST FOR RIGHT NO. OF ARGUMENTS       ZIGT0187
         BZ    ERRTR                                                     ZIGT0188
         L     2,0(0,1)            LOAD ADDR OF AREA TO BE FREED         ZIGT0189
         LA    2,0(0,2)            GET RID OF FIRST BYTE                 ZIGT0190
         S     2,EIGHT             SUBTRACT 8 TO GET TO CNTRL WDS.       ZIGT0191
         LA    6,5                 LOOP 5 TIMES TO FIND CONTROL WORDS.   ZIGT0192
         LA    7,1                 R7= ADDRESS MODIFIER                  ZIGT0193
         ST    2,REG2              SAVE INITIAL ADDRESS POINTER          ZIGT0194
LOOP     L     2,REG2              LOAD INITIAL ADDRESS                  ZIGT0195
         AR    2,7                  R7 = 1,2,4,8, OR 16                  ZIGT0196
         MVC   TEST(4),0(2)        PREPARE TO TEST (WORD BOUNDARY PROB)  ZIGT0197
         C     2,TEST              CHECK FOR CORRECT ADDRESS             ZIGT0198
         BE    REL                 YOU FOUND IT!                         ZIGT0199
         SLL   7,1                 TRY AGAIN-MULTIPLY R7 BY 2            ZIGT0200
         BCT   6,LOOP                                                    ZIGT0201
*                                                                        ZIGT0202
ERR2     LA    15,4(0,0)           ERROR RETURN  (RETURN 1)              ZIGT0203
         B     RETURN                                                    ZIGT0204
*          TRANSFER TO FORTRAN TRACEBACK ROUTINE                         ZIGT0205
ERRTR    MVC   ISN+2(2),2(14)      PICK UP INTERNAL STATEMENT NUMBER     ZIGT0206
         LA    1,PLIST                                                   ZIGT0207
         L     15,IBER                                                   ZIGT0208
         BALR  14,15                                                     ZIGT0209
REL      L     3,4(0,2)            R3= NO OF BYTES TO BE RELEASED        ZIGT0210
         STM   2,3,ADDR                                                  ZIGT0211
         FREEMAIN V,A=ADDR,SP=0                                          ZIGT0212
*        OPEN  (SNAPDCB,(OUTPUT))      FOR DEBUGGING PURPOSE             ZIGT0213
*        SNAP  DCB=SNAPDCB,ID=2,SDATA=(CB),PDATA=REGS      DEBUGG        ZIGT0214
*        CLOSE (SNAPDCB)               FOR DEBUGGING PURPOSE             ZIGT0215
RETURN   EQU   *                                                        ZIGT0216
```

```
EOJ      EQU   *                                                        ZIGT0217
         L     14,12(13)                                                ZIGT0218
         L     R1,ADDR                                                  ZIGT0219
         LM    2,12,28(13)                                              ZIGT0220
         MVI   12(13),X'FF'                                             ZIGT0221
         BCR   15,14                 RETURN                             ZIGT0222
*NAPDCB  DCB   DSORG=PS,RECFM=VBA,MACRF=(W),BLKSIZE=1632,LRECL=125,      ZIGT0223
*              DDNAME=SNAPCARD           FOR DEBUGGING PURPOSE          ZIGT0224
*                                                                       ZIGT0225
PLIST    DC    A(ISN)                                                   ZIGT0226
ISN      DC    F'0'                  INTERNAL STATEMENT NUMBER          ZIGT0227
LENGTH   DC    F'0'                  LENGTH OF STORAGE REQUESTED. (MIN) ZIGT0228
ORGLGTH  DC    F'0'                  LENGTH OF STORAGE REQUESTED. (MAX) ZIGT0229
ADDR     DC    F'0'                  ADDRESS OF STORAGE ALLOCATED       ZIGT0230
LENA     DC    F'0'                  LENGTH OF STORAGE ALLOCATED        ZIGT0231
REG2     DS    1F                    ADDR. OF DUMMY VARIABLE(ARG1)      ZIGT0232
REG3     DS    1F                    ADDRESS OF AREA TO BE RELEASED     ZIGT0233
REG6     DS    1F                    STARTING ADDR. OF DYNAMIC CORE ALLOC ZIGT0234
EIGHT    DC    F'8'                                                     ZIGT0235
ECODE    DC    F'-999'               ERROR CODE                         ZIGT0236
TEST     DC    F'0'                  TEST WORD                          ZIGT0237
IBER     DC    V(IBERH#)             ADDRESS OF FORTRAN TRACEBACK ROUTINE ZIGT0238
ZERO     DC    F'0'                                                     ZIGT0239
ONE      DC    F'1'                                                     ZIGT0240
RESADD   DC    F'0'                                                     ZIGT0241
RESLGTH  DC    F'0'                                                     ZIGT0242
GETALL   DC    F'0'                                                     ZIGT0243
         DC    F'1638400'                                               ZIGT0244
SYSOVHD  DC    F'10240'                                                 ZIGT0245
         END                                                            ZIGT0246
```

APPENDIX B
LISTING OF INPUT FOR BWR TEST PROBLEM

```
    0     1    1.0E-06    1.0E-02                                                    TP570001
   13    13      1     1     1     1                                                 TP570002
 2.43          1.0E-06 .3204E-10                                                    TP570003
   -1    -1     -1    -1    -1    -1    -1    -1    -1    -1    -1    -1          TP570004
   -1                                                                              TP570005
   -1     5      5     5     5     5     5     5     5     5     5     5          TP570006
   -1                                                                              TP570007
   -1     5      5     5     5     5     5     5     5     5     5     5          TP570008
   -1                                                                              TP570009
   -1     3      3     3     3     3     3     3     5     5     5     5          TP570010
   -1                                                                              TP570011
   -1     3      3     3     3     3     3     3     4     5     5     5          TP570012
   -1                                                                              TP570013
   -1     2      1     1     1     1     2     2     6     6     5     5          TP570014
   -1                                                                              TP570015
   -1     2      1     1     1     1     2     2     6     6     5     5          TP570016
   -1                                                                              TP570017
   -1     1      1     1     1     1     1     1     3     3     5     5          TP570018
   -1                                                                              TP570019
   -1     1      1     1     1     1     1     1     3     3     5     5          TP570020
   -1                                                                              TP570021
   -1     1      1     1     1     1     1     1     3     3     5     5          TP570022
   -1                                                                              TP570023
   -1     1      1     1     1     1     1     1     3     3     5     5          TP570024
   -1                                                                              TP570025
   -1     2      1     1     1     1     2     2     3     3     5     5          TP570026
   -1                                                                              TP570027
   -1    -1     -1    -1    -1    -1    -1    -1    -1    -1    -1    -1          TP570028
   -1                                                                              TP570029
    6                                                                              TP570030
 15.0       15.0       15.0       15.0       15.0       15.0                       TP570031
 15.0       15.0       15.0       15.0       15.0                                  TP570032
 15.0       15.0       15.0       15.0       15.0       15.0                       TP570033
 15.0       15.0       15.0       15.0       15.0                                  TP570034
  0.0        0.0        0.0        0.0                                           TP570035
  0.0        0.0        0.0        0.0                                           TP570036
```

```
 1.255     .0337075   .02533     .004602     .211      .1003211    0.0        .1091      TP570037
  1.268    .0349778    .02767     .004609     .1902    .07048902   0.0        .08675     TP570038
 1.259     .0342979    .02617     .004663    .2091     .0834609    0.0        .1021      TP570039
  1.234    .0352954    .02805     .004668     .1935    .06553935     0.0       .08792    TP570040
  1.257    .0482691    .04754      0.0       .1592     .01912592     0.0       _0.0       TP570041
 1.259     .0342979    .02617     .004663    .2091     .0834609    0.0        .1021      TP570042
    9     9                                                                              TP570043
    1     2     3     4     5     6     7     8     9                                    TP570044
    1     2     3     4     5     6     7     8     9                                    TP570045
    1     5     2    10                                                                 TP570046
  100   .01                                                                            TP570047
  300   .001                                                                           TP570048
  600   .0005                                                                          TP570049
  200   .002                                                                           TP570050
  100   .01                                                                            TP570051
3.3333E-083.3333E-06                                                                   TP570052
 .0054      .0654                                                                      TP570053
  .001087   1.35                                                                       TP570054
    2     6    0.0      2.0          0.0      -.010116                                TP570055
  3.83E-11 3.034E-03   300.                                                           TP570056
```