MITNE-195

C. 3

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)
$\text{CORE} = 120\text{ K} + 4C_s + 4C_t$ where

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

$$C_t = 2(\text{NDEL})(\text{NPXY}) + 3(\text{NPXY})$$

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

```
    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
      READ(IN,1030) XNU,WPCC,EPSIL
C
      WRITE(IOUT,2020) XNU,WPCC,EPSIL
C
C=====READ REACTOR GEOMETRY DATA
C
      CALL IGEOMO
C
C=====READ NEUTRONIC DATA
C
      CALL INEUTO
C
C=====CONSTRUCT NEUTRONIC FINE MESH GEOMETRY DATA
C
      CALL IFINEO
C
C=====READ COMPOSITION DATA
C
      CALL ICOMPO
C
C=====READ TRANSIENT DATA
C
      CALL ITRAN
      RETURN
C
C
C=====FORMATS
C
 1010 FORMAT(I5,3E10.3)
 1020 FORMAT(6I5)
 1030 FORMAT(3E10.3)
 2001 FORMAT(1H1,45X,'PROGRAM 2DTD OUTPUT',////)
 2005 FORMAT(1H0,9X,'NO. OF INNERS PER OUTER IS ',I5,//,
     X  10X,'INITIAL EIGENVALUE ESTIMATE',E12.4,//,
     X  10X,'STEADY STATE CONV. CRIT. ',E12.4,//,
     X  10X,'TRANSIENT CONV. CRIT. ',E12.4)                                INPT0073
 2006 FORMAT(1H0,9X,'CODE WILL COMPUTE NINNER FOR STEADY STATE')            INPT0074
 2015 FORMAT(1H0,9X,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(1H0,9X,'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
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X       WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X       KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X       KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NCUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X       ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X       TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X       VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF
C
      COMMON       DATA(1)
      REAL*8 DDAT(1)
      INTEGER      IDAT(1)
      EQUIVALENCE  (DATA(1),DDAT(1),IDAT(1))
      IN=5
      IOUT=6
      IZERO=0
      IONE=1
      ITWO=2
C
C==== CARDS TYPE G3
C
      NP1X=ND1X-2
      NP2X=ND2X-2
      NPXY=NP1X*NP2X
      CALL ALOC(WNBOX,0,KNBOX,ND1X*ND2X)
      KNBOX=KNBOX-1
      WRITE(IOUT,2017)
      ND2=ND2X+IONE
      DO 100 ND2P=1,ND2X
      ND2=ND2-IONE
      READ (IN,1030)        (IDAT(KNBOX+ND1+ND1X*(ND2-IONE)),ND1=1,ND1X)
  100 WRITE(IOUT,2018)      (IDAT(KNBOX+ND1+ND1X*(ND2-IONE)),ND1=1,ND1X)
```

```
      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
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X       WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X       KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X       KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NCUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X       ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X       TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X       VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
      COMMON       DATA(1)
      REAL*8 DDAT(1)
      INTEGER      IDAT(1)
      EQUIVALENCE  (DATA(1),DDAT(1),IDAT(1))
      IZERO=0
      IONE=1
      ITWO=2
      IN=5
      IOUT=6
C
C=====CARD TYPE N0
C
      READ(IN,1000) NCPX
C
      WRITE(IOUT,2002)  NCPX
C
C    RESERV SPACE AND READ IN MESH SPACINGS
C
      CALL ALOC(WHX,0,KHX,NP1X)
      CALL ALOC(WHY,0,KHY,NP2X)
      KHX=KHX-1
      KHY=KHY-1
      READ(IN,1040)      (DATA(KHX+J),J=1,NP1X)
      READ(IN,1040)      (DATA(KHY+J),J=1,NP2X)
      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
      I=ND2
      K=IDAT(KNBOX+(ND2-1)*ND1X+ND1-1)
      IF(K  .GT.  0) GO TO 65
   60 CONTINUE
   65 IDAT(KIENDY+ND1-1)=I
   40 CONTINUE
      WRITE(6,2006)
      DO 70 J=1,ND2X
   70 WRITE(6,2007) J,IDAT(KISTRX+J-1),IDAT(KIENDX+J-1)
      WRITE(6,2008)
      DO 80 J=1,ND1X
   80 WRITE(6,2007) J,IDAT(KISTRY+J-1),IDAT(KIENDY+J-1)
      CALL ALOC(WALBX,0,KALBX,NALB*4)
      CALL ALOC(WALBY,0,KALBY,NALB*4)
      KALBX=KALBX-1
      KALBY=KALBY-1
      WRITE(6,2010)
      DO 22 I=1,NALB
      J=KALBX+(I-1)*4
      READ(5,1070)  (DATA(J+K),K=1,4)
   22 WRITE(6,2011) I,(DATA(J+K),K=1,4)
      WRITE(6,2012)
      DO 21 I=1,NALB
      J=KALBY+(I-1)*4
      READ(5,1070)  (DATA(J+K),K=1,4)
   21 WRITE(6,2011) I,(DATA(J+K),K=1,4)
      KALBX=KALBX+1
      KALBY=KALBY+1
      RETURN
C
C==== FORMATS
C
 1000 FORMAT(2I5)
 1040 FORMAT(6E10.5)
 1050 FORMAT(6E10.5)
```

```
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 IFINEO
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
C
C    ALLOCATE BLOCKS FOR DATA STORAGE
C
      NEED=NP1X*(NP2X+2)*2
      CALL ALOC(WFLUX,0,KFLUX,NEED)
      KFLUX=KFLUX+NP1X+NP1X
      NEED=NPXY*52
      CALL ALOC(WMAT,0,KMAT,NEED)
      NEED=NPXY*2
      CALL ALOC(WFISS,0,KFISS,NEED)
      NEED=NPXY*4
      CALL ALOC(WRHS,0,KRHS,NEED)
      NEED=NPXY
      CALL ALOC(WS,1,KS1,NEED)
      CALL ALOC(WS,2,KS2,NEED)
      CALL ALOC(WS,3,KS3,NEED)
      NEED=NPXY*4
      CALL ALOC(WB,0,KB,NEED)
      FEED=50.0
      CALL ALOC(FEED,1,KSA1,NPXY)
      RETURN
C
      END
      SUBROUTINE ICOMP0
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X         WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X         KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X         KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X         ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X         TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X         VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
      COMMON          DATA(1)
      REAL*8 DDAT(1)
      INTEGER         IDAT(1)
      EQUIVALENCE     (DATA(1),DDAT(1),IDAT(1))
C
C==== SET SCALARS
C
      IZERO=0
      IONE=1
      ITWO=2
      IN=5
      IOUT=6
C
C==== CARDS TYPE N8
C
      NCORX=8
      CALL ALOC(WCORR,0,KCORR,NCORX*NCPX)
      KCORR=KCORR-1
      DO 200  NCP=1,NCPX
      READ(IN,2000)  (DATA(KCORR+NCOR+NCORX*(NCP-1)),NCOR=1,NCORX)
  200 WRITE(IOUT,1002) NCP,(DATA(KCORR+NCOR+NCORX*(NCP-IONE)),
     X    NCOR=1,NCORX)
      KCORR=KCORR+1
C
```

```
C   READ IN EDIT STUFF
C
      READ(IN,2001) NEDX,NEDY
      CALL ALOC(WEDX,0,KEDX,NEDX)
      CALL ALOC(WEDY,0,KEDY,NEDY)
      KEDX=KEDX-1
      KEDY=KEDY-1
      WRITE(IOUT,1003)
      READ(IN,2001)        (IDAT(J+KEDX),J=1,NEDX)
      WRITE(IOUT,1004)     (IDAT(J+KEDX),J=1,NEDX)
      READ(IN,2001)        (IDAT(J+KEDY),J=1,NEDY)
      WRITE(IOUT,1004)     (IDAT(J+KEDY),J=1,NEDY)
      KEDX=KEDX+1
      KEDY=KEDY+1
      RETURN
C
 1002 FORMAT(28H          COMPOSITION   DATA/11X,11(1H-),3X,90(1H-)/
     X     (11X,I11,3X,1P4E15.5) / (25X,4E15.5))
 1003 FORMAT(1H0,9X,'X AND Y DIRECTED EDIT BOUNDS',//)
 1004 FORMAT(10X,12I5)
C
 2000 FORMAT(8E10.4)
 2001 FORMAT(12I5)
C
      END
      SUBROUTINE ITRAN
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X       WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X       KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X       KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X       ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X       TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X       VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF
C
      COMMON A(1)
      DIMENSION IDAT(1)
      EQUIVALENCE (A(1),IDAT(1))
      READ(5,1010) ITRANS,NTD,NDEL,IEDTS
      IF(ITRANS  .EQ.  0) GO TO 500
      WRITE(6,2010) NTD,NDEL,IEDTS
      DO 10 I=1,NTD
      READ(5,1020) NUMS(I),DELT(I)
   10 WRITE(6,2020) I,NUMS(I),DELT(I)
      READ(5,1030) VIN(1),VIN(2)
      WRITE(6,2030) VIN(1),VIN(2)
      WRITE(6,2040)
      DO 20 J=1,NDEL
      READ(5,1030)      BETA(J),RLAM(J)
   20 WRITE(6,2050)  J,BETA(J),RLAM(J)
      READ(5,1040) IPERT,NCOMP,TST,TFIN,SIG1,SIG2
      IF(IPERT  .EQ.  1) WRITE(6,2060)
      IF(IPERT  .EQ.  2) WRITE(6,2070)
      WRITE(6,2080) NCOMP,TST,TFIN,SIG1,SIG2
      NEED=NDEL*NPXY
      CALL ALOC(WPREC,0,KPREC,NEED)
      NEED=NPXY*2
      CALL ALOC(WOMGP,0,KOMGP,NEED)
      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 VEITRN0062
     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
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NCUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
      COMMON / THFEED / ITHFE,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF
C
      COMMON A(1)
      DIMENSION IDAT(1)
      EQUIVALENCE (A(1),IDAT(1))
      LOGICAL JSW,JJJ,JLAST
      JSW=.FALSE.
      JJJ=.FALSE.
      JLAST=.FALSE.
      RHO=0.9
      ITS=0
      TIME=0.0
      CALL TFER(A(KISTRX),A(KIENDX),NP1X,NP2X,A(KNBOX),A(KCORR),
     X  A(KSA1),ND1X)
      CALL CPUTO
      CALL MATRX
      CALL CPUT(CTIME)
      CSTOR=CTIME
      WRITE(6,201) CTIME
  201 FORMAT(1H0,5X,'TIME FOR MATRX IS ',E12.4)
C
C   INITIALIZE FLUXES
C
      DO 20 NP2=1,NP2X
      IS=IDAT(KISTRX+NP2)
      IE=IDAT(KIENDX+NP2)
      DO 21 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      DO 23 NG=1,2
   23 A(KFLUX-1+(NPP-1)*2+NG)=1.0
   21 CONTINUE
   20 CONTINUE
      CALL CPUTO
      CALL DORPES
      CALL CPUT(CTIME)
      WRITE(6,202) CTIME
  202 FORMAT(1H0,5X,'TIME FOR DORPES IS ',E12.4)
      CALL CPUTO
      DO 170 NP=1,NPXY
      NP1=(NP-1)*2
      NP2=NP1+1
      X=A(KFLUX+NP1)*A(KFISS+NP1)+A(KFLUX+NP2)*A(KFISS+NP2)
  170 A(KS1+NP-1)=X
      DO 110 NP=1,NPXY
      NPP=NP-1
      A(KS2+NPP)=A(KS1+NPP)
  110 A(KS3+NPP)=A(KS1+NPP)
      IT=0
 1000 IT=IT+1
C
C    CALCULATE RIGHT HAND SIDE
C
      DO 120 NP=1,NPXY
      NP1=(NP-1)*4
      NP2=NP1+1
      X=A(KS1+NP-1)
      A(KRHS+NP1)=X / XKEFF
  120 A(KRHS+NP2)=0.0
      CALL INNERS(JSW)
      DO 160 NP=1,NPXY
      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
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NCUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
      DIMENSION AA(4,75),BB(4,75),COEF(4,75)
      COMMON A(1)
C
      CALL XSWEEP(A(KMAT),A(KFISS),A(KNBOX),A(KHX),A(KHY),A(KCORR),
     X  AA,BB,COEF,A(KALBX),A(KOMGP),A(KOMGD),A(KSA1))
C
      CALL YSWEEP(A(KMAT),A(KFISS),A(KNBOX),A(KHX),A(KHY),A(KCORR),
     X  AA,BB,COEF,A(KALBY),A(KOMGP),A(KOMGD),A(KSA1))
C
      RETURN
      END
      SUBROUTINE XSWEEP(XMAT,FISS,NBOX,HX,HY,CORR,A,B,C,ALB,OMP,OMD,SA1)
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X         WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X         KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X         KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X         ND2X,NP1X,NP2X,NPRT,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X         TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X         VIN(2),DELT(5),NMAS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
      DIMENSION XMAT(52,1),FISS(2,1),HX(1),HY(1),NBOX(ND1X,1),SA1(1),
     X  CORR(8,1),A(4,1),B(4,1),C(4,1),ALB(4,1),OMP(2,1),OMD(NDEL,1)
      DIMENSION D(4,1)
      COMMON IDAT(1)
      ZERO=0.0
      HALF=0.5
      RL=XKEFF
      IF(ITS  .EQ.  0) GO TO 1
      SMM=1.0-BTOT
      DO 2 ND=1,NDEL
    2 SMM=SMM+(BETA(ND)*RLAM(ND))/(DTI+RLAM(ND))
C
C   COMPUTE FISS VECTOR
C
    1 DO 51 NP2=1,NP2X
      ND2=NP2+1
      IS=IDAT(KISTRX+NP2)
      IE=IDAT(KIENDX+NP2)
      DO 51 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      K=NBOX(ND1,ND2)
      FISS(1,NPP)=HX(NP1)*HY(NP2)*CORR(4,K)
      FISS(2,NPP)=HX(NP1)*HY(NP2)*CORR(8,K)
```

```
   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
C
      X=SQRT(XKS)
      XK=X
      IF(X  .LT.  1.0E-07) GO TO 31
      XCSC=X/SIN(X*HX(I))
      XTAN=TAN(X*HX(I)*HALF)/X
      XCDK=XCSC/XKS
      GO TO 62
   31 XCSC=1.0/HX(I)
      XTAN=HX(I)*HALF
      XCDK=XCSC/XKS
      GO TO 62
C
C     KAPPA IS IMAGINARY
C
   61 X=SQRT(ABS(XKS))
      XK=X
      IF(X  .LT.  1.0E-07) GO TO 31
      XCSC=X/SINH(X*HX(I))
      XTAN=TANH(X*HX(I)*HALF)/X
      XCDK=-XCSC/(X*X)
   62 CONTINUE
      XM=SQRT(XMS)
      XCSCH=XM/SINH(XM*HX(I))
      XTANH=TANH(XM*HX(I)*HALF)/XM
      XCHDK=XCSCH/(XM*XM)
C
C     NOW FIND ENTRIES OF A AND B MATRICES   AND C MATRX ALSO
C
      IF(V2  .GT.  ZERO) GO TO 91
C
C     WE ARE IN A REFLECTOR REGION - USE THE EXACT FORMULAS
C
      A(1,I)=XCSC
      A(2,I)=ZERO
      A(3,I)=XR*XCSC          - XS*XCSCH                              XSWP0109
      A(4,I)=XCSCH                                                    XSWP0110
      B(1,I)=XTAN/D1                                                  XSWP0111
      B(2,I)=ZERO                                                     XSWP0112
      B(3,I)=(XR/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+XS*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-XR*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
      C(2,I)=T1*T6+T2*T8
      C(3,I)=T3*T5+T4*T7
      C(4,I)=T3*T6+T4*T8
      TI=HI/(XS-XR)
      DO 92 K=1,4
   92 C(K,I)=C(K,I)*TI
      DET=S1*S2-(V2*SR)/RL
      C(1,I)=C(1,I)+S2/DET
      C(2,I)=C(2,I)+V2/(RL*DET)
      C(3,I)=C(3,I)+SR/DET
      C(4,I)=C(4,I)+S1/DET
   52 CONTINUE
C
C    NOW FIND COUPLING COEFFICIENTS
C    LEFT BOUNDARY IS ZERO J OR ALBEDO
C    RIGHT BOUNDARY IS ALBEDO
C
      DO 55 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      I=NP1
C
C    COMPUTE COUPLING TO LEFT
C
      IF(ND1  .NE.  IS) GO TO 85
C
C    IBCL=1  SYMMETRY BOUNDARY ON LEFT
C    IBCL=2  ALBEDO BOUNDARY ON LEFT
C
      IF(IBCL  .EQ.  2) GO TO 86
      XL1=ZERO
      XL2=ZERO
      XL3=ZERO
      XL4=ZERO
      GO TO 56
   86 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
C     COMPUTE PIGHT COUPLING
C
   57 IF(ND1  .EQ.  IE) GO TO 55
      F1=XR1
      F2=XR2
      F3=XR3
      F4=XR4
      CALL BMULT(XR1,XR2,XR3,XR4,A,I+1)
      CALL BMULT(F1,F2,F3,F4,C,I+1)
      HH=HX(I+1)*HY(NP2)
      XMAT(9,NPP)=-HH*XR1
      XMAT(10,NPP)=-HH*XR2
      XMAT(11,NPP)=-HH*XR3
      XMAT(12,NPP)=-HH*XR4
      XMAT(37,NPP)=F1
      XMAT(38,NPP)=F2
      XMAT(39,NPP)=F3
      XMAT(40,NPP)=F4
   55 CONTINUE
   40 CONTINUE
      RETURN
      END
      SUBROUTINE YSWEEP(XMAT,FISS,NBOX,HX,HY,CORR,A,B,C,ALB,OMP,OMD,SA1)
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXFD / NINNER,NCUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X        TST,TFIN,TIMF,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
      DIMENSION XMAT(52,1),FISS(2,1),HX(1),HY(1),NBOX(ND1X,1),SA1(1),
     X  CORR(8,1),A(4,1),B(4,1),C(4,1),ALB(4,1),OMP(2,1),OMD(NDEL,1)
      DIMENSION D(4,1)
      COMMON IDAT(1)
      ZERO=0.0
      HALF=0.5
      RL=XKEFF
      IF(ITS  .EQ.  0) GO TO 1
      SMM=1.0-BTOT
      DO 2 ND=1,NDEL
    2 SMM=SMM+(BETA(ND)*RLAM(ND))/(DTI+RLAM(ND))
    1 DO 40 NP1=1,NP1X
      ND1=NP1+1
      IS=IDAT(KISTRY+NP1)
      IE=IDAT(KIENDY+NP1)
      DO 52 ND2=IS,IE
      NP2=ND2-1
      NPP=(NP2-1)*NP1X+NP1
      I=NP2
      K=NBOX(ND1,ND2)
      D1=CORR(1,K)
      D2=CORR(5,K)
      SR=CORR(3,K)
      IF(ITS  .EQ.  0) GO TO 18
      SUM=1.0-BTOT
      DO 17 ND=1,NDEL
   17 SUM=SUM+(BETA(ND)*RLAM(ND))/(OMD(ND,NPP)+RLAM(ND))
      S1=SA1(NPP)-SUM*CORR(4,K)/RL+VIN(1)*OMP(1,NPP)
      S2=CORR(6,K)+OMP(2,NPP)*VIN(2)
      V2=SUM*CORR(8,K)
      GO TO 19
   18 S1=SA1(NPP)-CORR(4,K)/RL
      S2=CORR(6,K)
      V2=CORR(8,K)
   19 C1=S1/D1+S2/D2
      C2=(HALF*(S2/D2-S1/D1))**2  +  (V2*SR)/(RL*D1*D2)
      C3=SQRT(C2)
      XKS=-HALF*C1+C3
      XMS=HALF*C1+C3
      IF(V2  .GT.  ZERO) GO TO 81
      XS=1.0E+20
      GO TO 82
   81 XS=SR/(-D2*XMS+S2)
   82 XR=SR/(D2*XKS+S2)
      IF(XKS  .LE.  ZERO) GO TO 61
C
C    KAPPA IS REAL
C
      X=SQRT(XKS)
      XK=X
      IF(X  .LT.  1.0E-07) GO TO 31
      XCSC=X/SIN(X*HY(I))
      XTAN=TAN(X*HY(I)*HALF)/X
      XCDK=XCSC/XKS
      GO TO 62
   31 XCSC=1.0/HY(I)
      XTAN=HY(I)*HALF
      XCDK=XCSC/XKS
      GO TO 62
C
```

```
C    KAPPA IS IMAGINARY
C
  61 X=SQRT(ABS(XKS))
     XK=X
     IF(X  .LT.  1.0E-07) GO TO 31
     XCSC=X/SINH(X*HY(I))
     XTAN=TANH(X*HY(I)*HALF)/X
     XCDK=-XCSC/(X*X)
  62 CONTINUE
     XM=SQRT(XMS)
     XCSCH=XM/SINH(XM*HY(I))
     XTANH=TANH(XM*HY(I)*HALF)/XM
     XCHDK=XCSCH/(XM*XM)
C
C    NOW FIND ENTRIES OF A AND B MATRICES   AND C MATRX ALSO
C
     IF(V2  .GT.  ZERO) GO TO 91
C
C    WE ARE IN A REFLECTOR REGION - USE THE EXACT FORMULAS
C
     A(1,I)=XCSC
     A(2,I)=ZERO
     A(3,I)=XR*XCSC         - XR*XCSCH
     A(4,I)=XCSCH
     B(1,I)=XTAN/D1
     B(2,I)=ZERO
     B(3,I)=(XR/D1)*(XTAN-XTANH)
     B(4,I)=XTANH/D2
     HI=HY(I)
     DET=S1*S2
     C(1,I)=(XCDK*HI)/D1+S2/DET
     C(2,I)=ZERO
     C(3,I)=(XR*XCDK*HI+XR*XCHDK*HI)/D1+SR/DET
     C(4,I)=-(XCHDK*HI)/D2+S1/DET
     GO TO 52
  91 XMULT=1.0/(XS-XR)
```

```
      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)
      CALL BMULT(F1,F2,F3,F4,C,I-1)
      HH=HY(I-1)*HX(NP1)
      XMAT(13,NPP)=-HH*X1
      XMAT(14,NPP)=-HH*X2
      XMAT(15,NPP)=-HH*X3
      XMAT(16,NPP)=-HH*X4
      XMAT(41,NPP)=F1
      XMAT(42,NPP)=F2
      XMAT(43,NPP)=F3
      XMAT(44,NPP)=F4
C
C   COMPUTE CENTER POINT COUPLING
C
   56 CONTINUE
      J1=I
      J2=I+1
      IF(ND2  .NE.  IE) GO TO 12
      KUP=IABS(NBOX(ND1,ND2+1))
      DO 13 J4=1,4
   13 B(J4,I)=B(J4,I)+ALB(J4,KUP)
      J2=0
   12 CALL BINV(B,J1,J2,X1,X2,X3,X4)
      XR1=X1
      XR2=X2
      XR3=X3
      XR4=X4
      X1=XR1+XL1
      X2=XR2+XL2
      X3=XR3+XL3
      X4=XR4+XL4
      F1=X1
      F2=X2
      F3=X3
      F4=X4
      CALL BMULT(X1,X2,X3,X4,A,I)
      CALL BMULT(F1,F2,F3,F4,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 NP2=1,NP2X                                                 YSWP0252
      IS=IDAT(KISTRX+NP2)
      IE=IDAT(KIENDX+NP2)
      DO 400 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      F1=XMAT(1,NPP)
      F2=XMAT(2,NPP)
      F3=XMAT(3,NPP)
      F4=XMAT(4,NPP)
      DET=F1*F4-F2*F3
      XMAT(1,NPP)=F4/DET
      XMAT(2,NPP)=-F2/DET
      XMAT(3,NPP)=-F3/DET
      XMAT(4,NPP)=F1/DET
  400 CONTINUE
      RETURN
      END
      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)
DIMENSION A(4,1)
Y1=X1*A(1,I)+X2*A(3,I)
Y2=X1*A(2,I)+X2*A(4,I)
Y3=X3*A(1,I)+X4*A(3,I)
Y4=X3*A(2,I)+X4*A(4,I)
X1=Y1
X2=Y2
X3=Y3
X4=Y4
RETURN
END
```

```
      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
      EQUIVALENCF(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
      NPP=NP*2-1
  560 A(KS3+NP-1)=A(KFLUX+NPP)
      ICYCIT=1
  120 CONTINUE
      PDIFD=0.0
      PDIFN=0.0
      PLAMUP=0.0
      PLAMLO=1.0E+20
C
C    COPY FLUX INTO KS2
C
      DO 110 NP=1,NPXY
      NPP=NP*2-1
  110 A(KS2+NP-1)=A(KFLUX+NPP)
      CALL SOLV(JSW,ICYCIT)
C
C==== THIS CALL TO  SOLV  WILL DO ONE INNER ITERATION WITH OMEGA=1.0
C     AND WITH THE RIGHT HAND SIDE SOURCE SET EQUAL TO ZERO.
C
      DO 130 NP2=1,NP2X
      IS=IDAT(KISTRX+NP2)
      IE=IDAT(KIENDX+NP2)
      DO 131 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      PSIN=A(KFLUX+NPP*2-1)
      IF(ABS(PSIN)  .LT.  1.0E-20) GO TO 130
      PSIO=A(KS2+NPP-1)
      RATO=ABS(PSIN/PSIO)
      PLAMLO=AMIN1(PLAMLO,RATO)
      PLAMUP=AMAX1(PLAMUP,RATO)
      PDIFN=PSIN*PSIN+PDIFN
      PDIFD=PSIN*PSIO+PDIFD
  131 CONTINUE
  130 CONTINUE
```

```
C
C==== COMPUTE CMEGA AND NUMDO
C
      XMULT=PDIFD/PDIFN
      IT=NPXY*2
      DO 135 NPP=1,IT
  135 A(KFLUX+NPP-1)=A(KFLUX+NPP-1) * XMULT
C
      ICYCIT=ICYCIT+1
      IF(ICYCIT  .LT.   8) GO TC 120
      RHOEST=PDIFN/PDIFD
      OMEGBL=2.0/(1.0+SQRT(1.0-PLAMLO))
      X=1.0-PLAMUP
      IF(X  .LE.  0.C) OMEGBU=2.0
      IF(X  .GT.  0.0) OMEGBU=2.0/(1.0+SQRT(X))
      OMEGM=2.C/(1.0+SQRT(1.0-RHOEST))
      IF(ABS(OMEGBU-OMEGBL)  .LE.  ((2.0-OMEGM)/5.0)) GO TO 240
      IF(ICYCIT  .LT.  50) GO TC 120
  240 CONTINUE
      RHO=RHOEST
      X=2.0*SQRT(OMEGM/(OMEGM-1.0))
      Y=OMEGM-1.0
      NING=0
  260 X=X*Y
      NING=NING+1
      IF(X  .GT.  PSINRM) GO TO 260
      NUMDO=NING
      IF(NUMDO  .LT.  INMIN) NUMDO=INMIN
      IF(NINNER  .EQ.  0) NINNER=NUMDO
      IF(ITS  .NE.  0) NINNER=NUMDO
      WRITE(IOUT,1000) NINNER,RHO,OMEGM
 1000 FORMAT(1H1,5X,'DO ',I5,3X,'INNERS WITH RHO = ',E12.4,3X,
     X  'AND OMEGA = ',E12.4)
      IF(ITS  .NE.  0) GO TO 500
C
C   SET FLUXES EQUAL TO ONE
```

```
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)
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
      COMMON A(1)
C
C        DO NINNER NUMBER OF INNER (FLUX) ITERATIONS
C
      LOGICAL JSW
      CALL RHS(A(KMAT),A(KB),A(KFLUX),A(KRHS))
      DO 10 IN=1,NINNER
   10 CALL SOLV(JSW,IN)
      CALL BUCK(A(KMAT),A(KB),A(KFLUX),A(KHX),A(KHY))
      RETURN
      END
      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)
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X       WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X       KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X       KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X       ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X       TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X       VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
      LOGICAL JSW
      COMMON A(1)
C
C    IF JSW IS TRUE, SET W TO ONE FOR ALL INNERS
C    MM IS THE INNER ITERATION COUNTER
C    FIRST GO THROUGH RED SQUARES, THEN BLACK
C
      IF(MM  .NE.  1) GO TO 5
      W=1.0
      GO TO 6
    5 W=1.0/(1.0-RHO*WLAST*0.25)
    6 IF(JSW) W=1.0
      DO 20 NP2=1,NP2X
      NN=NP2-(NP2/2)*2
      IF(NN  .EQ.  0) GO TO 21
      NPX1=2
      NPX2=N3
      GO TO 30
   21 NPX1=1
      NPX2=N4
   30 CALL ROW(NPX1,NPX2,NP2,W,A(KFLUX),A(KMAT),A(KRHS),NP1X,
     X  A(KISTRX+NP2),A(KIENDX+NP2))
   20 CONTINUE
      WLAST=W
      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)
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X        WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X        KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X        KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
      COMMON IDAT(1)
      DIMENSION XMAT(1),B(4,1),F(2,1),HX(1),HY(1)
C
      ZERO=0.0
C
C    FIRST CALCULATE X DIRECTED LEAKAGES
C
      DO 100 NP2=1,NP2X
      IS=IDAT(KISTRX+NP2)
      IE=IDAT(KIENDX+NP2)
      DO 100 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      J=(NPP-1)*52
   51 NPPL=NPP-1
      NPPR=NPP+1
      X1=ZERO
      X2=ZERO
      Z1=ZERO
      Z2=ZERO
      X5=ZERO
      X6=ZERO
      Z5=ZERO
      Z6=ZERO
      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)
C
      DIMENSION FLUX(2,1),RHS(4,1),XMAT(1)
C
      NP=(NP2-1)*NP1X
      DO 10 I=NPX1,NPX2,2
      K=I+1
      IF(K .LT. ISTR  .OR.  K .GT. IEND) GO TO 10
      NPP=NP+I
      NUP=NPP+NP1X
      NDN=NPP-NP1X
      NL=NPP-1
      NR=NPP+1
      J=(NPP-1)*52
      SUM1=XMAT(J+17)*FLUX(1,NUP)+XMAT(J+18)*FLUX(2,NUP)
      SUM2=XMAT(J+19)*FLUX(1,NUP)+XMAT(J+20)*FLUX(2,NUP)
      SUM3=XMAT(J+9)*FLUX(1,NR)+XMAT(J+10)*FLUX(2,NR)
      SUM4=XMAT(J+11)*FLUX(1,NR)+XMAT(J+12)*FLUX(2,NR)
      SUM5=XMAT(J+5)*FLUX(1,NL)+XMAT(J+6)*FLUX(2,NL)
      SUM6=XMAT(J+7)*FLUX(1,NL)+XMAT(J+8)*FLUX(2,NL)
      SUM7=XMAT(J+13)*FLUX(1,NDN)+XMAT(J+14)*FLUX(2,NDN)
      SUM8=XMAT(J+15)*FLUX(1,NDN)+XMAT(J+16)*FLUX(2,NDN)
      RHS1=RHS(1,NPP)-SUM1-SUM3-SUM5-SUM7-RHS(3,NPP)
      RHS2=RHS(2,NPP)-SUM2-SUM4-SUM6-SUM8-RHS(4,NPP)
      F1=XMAT(J+1)
      F2=XMAT(J+2)
      F3=XMAT(J+3)
      F4=XMAT(J+4)
      E1=F1*RHS1+F2*RHS2
      E2=F3*RHS1+F4*RHS2
      FLUX(1,NPP)=W*E1+(1.0-W)*FLUX(1,NPP)
      FLUX(2,NPP)=W*E2+(1.0-W)*FLUX(2,NPP)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE OUTERS(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
      GO TO 130
  110 CONTINUE
      IF(NEWCP  .EQ.  1) GO TO 120
      NORDCP=NORDCP+1
      CALL CHEBE(ALPHAC,BETAC,NORDCP,SIGMA)
      GO TO 130
  120 ERPROD=1.0
      NORDCP=1
      SIGMA=SIGBAR
      IF(SIGMA  .GT.  1.0  .OR.  SIGMA  .LT.  0.4) GO TO 100
      IF(IOT  .LE.  ISIX) SIGMA=AMIN1(SIGMA,0.9)
      IF(IOT  .LE.  ININ) SIGMA=AMIN1(SIGMA,0.95)
      IF(IOT  .LE.  ITW) SIGMA=AMIN1(SIGMA,0.985)
      SIGMA=AMIN1(SIGMA,0.999)
      NEWCP=0
      CALL CHEBE(ALPHAC,BETAC,NORDCP,SIGMA)
  130 CONTINUE
      ASSIGN 150 TO NS1
      IF(BETAC  .LE.  0.0) ASSIGN 160 TO NS1
      IF(NORDCP  .EQ.  0) ASSIGN 170 TO NS1
C
C==== BEGIN THE SWEEP OVER THE MESH
C
      DO 310 NPP=1,NPXY
      FISNS=SORC(NPP)
      FISNE=FISNS
      FROM1S=SORC1(NPP)
      IF(IOT  .EQ.  1) FISLNO=FISLNO+FROM1S
      FISM1S=FROM1S
      FISDEL=FISNS-FROM1S
      ERRATN=ERRATN+FISDEL*FISDEL
      FISM2S=SORC2(NPP)
      SORC2(NPP)=FROM1S
      IF(FISM1S  .LE.  FISMIN) GO TO 140
      RATO=FISNS/FISM1S
      FLAMUP=AMAX1(FLAMUP,RATC)                                         OTRS0073
      FLAMLO=AMIN1(FLAMLO,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)
      IF(NORDCP  .GT.  1) GO TO 100
      ALPHAC=2.0/(2.0-SIGMA)
      BETAC=0.0
      RETURN
  100 COSHGM=(2.0-SIGMA)/SIGMA
      GAMMA=DACOSH(COSHGM)
      ALPHAC=4.0*COSH1((NORDCP-1)*GAMMA)/(SIGMA*COSH1(NORDCP*GAMMA))
      BETAC=(1.0-0.5*SIGMA)*ALPHAC-1.0
      RETURN
      END
      FUNCTION COSH1(X)
      COSH1=COSH(X)
      RETURN
      END
      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,
     X   NY,STOR,BUCK,TIME,HX,HY,TEMP)
C
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF
C
      DIMENSION FLUX(2,ND1X,1),FISS(2,ND1X,1),S(ND1X,1),TEMP(ND1X,1)
      DIMENSION NX(1),NY(1),STOR(NEDX,NEDY),BUCK(4,ND1X,1),HX(1),HY(1)
      IF(ITS  .NE.  0) GO TO 26
      XTOT=0.0
      HTOT=0.0
      DO 210 NP2=1,ND2X
      DO 210 NP1=1,ND1X
      X=FLUX(1,NP1,NP2)*FISS(1,NP1,NP2)+FLUX(2,NP1,NP2)*FISS(2,NP1,NP2)
      X=X/XNU
      S(NP1,NP2)=X
      IF(X  .LE.  0.0) GO TO 210
      XTOT=XTOT+X
      HH=HX(NP1)*HY(NP2)
      HTOT=HTOT+HH
  210 CONTINUE
      PTOT=(HTOT*WPCC)/(XTOT*EPSIL)
      DO 220 NP2=1,ND2X
      DO 220 NP1=1,ND1X
      FLUX(1,NP1,NP2)=FLUX(1,NP1,NP2) * PTOT
      FLUX(2,NP1,NP2)=FLUX(2,NP1,NP2) * PTOT
  220 S(NP1,NP2)=S(NP1,NP2) * PTOT
      DO 221 NP2=1,ND2X
      DO 221 NP1=1,ND1X
      DO 221 L=1,4
      BUCK(L,NP1,NP2)=BUCK(L,NP1,NP2)*PTOT
  221 CONTINUE
      GO TO 26
   25 WRITE(6,260)
  260 FORMAT(1H0,5X,
     X '  NP1',5X,'NP2',5X,'FLUX1',8X,'FLUX2',8X,'LEAKAGES',46X,
     X  'POWER FRACTION',//)
```

```
      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(1H0,5X,'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(1H0,3X,I5,2X,3(2X,10E12.4,/))                               EDIT0063
      IF(ITHFB  .EQ.  0) RETURN                                          EDIT0064
      WRITE(6,140)                                                       EDIT0065
  140 FORMAT(1H0,5X,'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
      DO 170 NP2=1,ND2X
      DO 170 NP1=1,ND1X
      X=FLUX(1,NP1,NP2)*FISS(1,NP1,NP2)+FLUX(2,NP1,NP2)*FISS(2,NP1,NP2)
      X=X/XNU
      IF(X  .LE.  0.0) GO TO 175
      XTOT=XTOT+X
      HH=HX(NP1)*HY(NP2)
      HTOT=HTOT+HH
      X=X/HH
      S(NP1,NP2)=X*EPSIL
      GO TO 170
  175 S(NP1,NP2)=0.0
  170 CONTINUE
      PTOT=XTOT*EPSIL
      PM=PTOT/HTOT
      WRITE(6,28) ITS,TIME,PTOT,PM
   28 FORMAT(1H0,5X,'TIME STEP',I5,3X,'TIME IS ' ,E12.4,3X,
     X 'TOTAL POWER ',E14.6,/,10X,'MEAN POWER ',E14.6,//)
      IF(ITS  .EQ.  0) GO TO 25
      GO TO 27
      END
      SUBROUTINE TRANS
C
      COMMON / NAMES / WHX,WHY,WINTX,WINTY,WCORR,WNBOX,WFLUX,WMAT,
     X         WFISS,WRHS,WS,WB,WPREC,WOMGP,WOMGD,WALBX,WALBY,WEDX,WEDY
      COMMON / ORIGIN / KHX,KHY,KISTRX,KISTRY,KIENDX,KIENDY,KCORR,KNBOX,
     X         KFLUX,KMAT,KFISS,KRHS,KS1,KS2,KS3,KB,KPREC,KOMGP,KOMGD,
     X         KALBX,KALBY,KEDX,KEDY,KTEMP,KSA1
      COMMON / FIXED / NINNER,NOUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X         ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X         TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),RLAM(6),
     X         VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF
C
      COMMON A(1)
      LOGICAL JSW,JST,JEND
      LOGICAL JFIRST
      IF(ITRANS  .EQ.  0) STOP 1111
      CALL CPUTO
      WRITE(6,450)
  450 FORMAT(1H1,10X,'START TRANSIENT CALCULATIONS',//)
      JSW=.FALSE.
      JST=.FALSE.
      JEND=.FALSE.
      BTOT=0.0
      DO 10 ND=1,NDEL
   10 BTOT=BTOT+BETA(ND)
      IT=NPXY*2
      DO 20 NP=1,IT
   20 A(KOMGP+NP-1)=0.0
      IT=NPXY*NDEL
      DO 30 NP=1,IT
   30 A(KOMGD+NP-1)=0.0
      TIME=0.0
      ITS=0
      CALL PERTO(A(KCORR))
      CALL PRECO(A(KPREC),A(KFLUX),A(KISTRX),A(KIENDX),A(KFISS),
     X BETA,RLAM,NP1X,NP2X,NDEL,XKEFF)
      DO 100 NTTD=1,NTD
      NSTEP=NUMS(NTTD)
      DT=DELT(NTTD)
      DTI=1.0/DT
      DO 200 NST=1,NSTEP
      TIME=TIME+DT
      ITS=ITS+1
      CALL PERT(A(KCORR),JST,JEND)
      CALL TFER(A(KISTRX),A(KIENDX),NP1X,NP2X,A(KNBOX),A(KCORR),
     X A(KSA1),ND1X)
      CALL FDBK(A(KSA1),A(KTEMP),A(KS3),A(KISTRX),A(KIENDX),NP1X,NP2X,
     X A(KNBOX),A(KCORR),ND1X)
      CALL MATRX
      JFIRST=.FALSE.
      IF(NST  .GT.  1  .OR.  NTTD  .GT.  1) GO TO 50
      CALL DORPES
      JFIRST=.TRUE.
   50 CONTINUE
      CALL RHST(A(KRHS),A(KFLUX),A(KPREC),A(KS1),A(KS2),RLAM,A(KISTRX),
     X A(KIENDX),A(KHX),A(KHY),NP1X,NP2X,NDEL,VIN,DT)
      CALL FEXT(A(KFLUX),A(KB),A(KOMGP),A(KISTRX),A(KIENDX),NP1X,NP2X,
     X DT)
      CALL INNERS(JSW)
      CALL PREC1(A(KPREC),A(KFLUX),A(KS1),A(KS2),A(KOMGP),A(KOMGD),
     X A(KISTRX),A(KIENDX),A(KFISS),BETA,RLAM,NP1X,NP2X,NDEL,XKEFF,DTI)
      CALL TEMP(A(KFLUX),A(KFISS),A(KHX),A(KHY),A(KISTRX),
     X A(KIENDX),NP1X,NP2X,A(KTEMP),A(KS3),A(KS1),DT,JFIRST)
      I=(ITS/IEDTS)*IEDTS-ITS
      IF(I .EQ. 0) CALL EDIT(NP1X,NP2X,NEDX,NEDY,ITS,NPXY,A(KFLUX),
     X  A(KFISS),A(KS1),A(KEDX),A(KEDY),A(KS2),A(KB),TIME,
     X  A(KHX),A(KHY),A(KTEMP))
  200 CONTINUE
  100 CONTINUE
      CALL CPUT(CTIME)
      WRITE(6,1) CTIME
```

```
    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)
C
      COMMON / FIXED / NINNER,NCUT,N1,N2,N3,N4,DIFSS,DIFTD,RHO,ND1X,
     X        ND2X,NP1X,NP2X,NPXY,IBCL,IBCB,NCPX,XKEFF,ITS,IPERT,NCOMP,
     X        TST,TFIN,TIME,SIG1,SIG2,ITRANS,NTD,NDEL,BETA(6),BLAM(6),
     X        VIN(2),DELT(5),NUMS(5),NEDX,NEDY,NALB,BTOT,DT,DTI,IEDTS
C
      DIMENSION CORR(8,1)
      LOGICAL JST,JEND
      IF(IPERT  .EQ.  1) GO TO 10
      IF(JEND) RETURN
      IF(JST) GO TO 20
   12 IF(TIME  .LE.  TST) RETURN
      JST=.TRUE.
   20 IF(TIME  .GT.  TFIN) GO TO 21
      TOT=TFIN-TST
      FRAC=(TIME-TST)/TOT
      CORR(2,NCOMP)=SIGB1+SIG1*FRAC
      CORR(6,NCOMP)=SIGB2+SIG2*FRAC
      RETURN
   10 IF(JST) RETURN
      IF(TIME  .LE.  TST) RETURN
      JST=.TRUE.
      CORR(2,NCOMP)=CORR(2,NCOMP)+SIG1
      CORR(6,NCOMP)=CORR(6,NCOMP)+SIG2
      RETURN
   21 JEND=.TRUE.
      RETURN
      ENTRY PERT0(CORR)
      SIGB1=CORR(2,NCOMP)
      SIGB2=CORR(6,NCOMP)
      RETURN
      END
      SUBROUTINE FEXT(FLUX,BUCK,OMP,ISTR,IEND,NP1X,NP2X,DT)
      DIMENSION FLUX(2,1),BUCK(4,1),OMP(2,1),ISTR(1),IEND(1)
      DO 10 NP2=1,NP2X
      IS=ISTR(NP2+1)
      IE=IEND(NP2+1)
      DO 20 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      FAC1=EXP(OMP(1,NPP)*DT)
      FAC2=EXP(OMP(2,NPP)*DT)
      FLUX(1,NPP)=FLUX(1,NPP)*FAC1
      FLUX(2,NPP)=FLUX(2,NPP)*FAC2
      DO 30 ND=1,4
   30 BUCK(ND,NPP)=BUCK(ND,NPP)*FAC2
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE RHST(RHS,FLUX,PREC,SORC1,SORC2,RLAM,ISTR,IEND,
     X  HX,HY,NP1X,NP2X,NDEL,VIN,DT)
      DIMENSION RHS(4,1),FLUX(2,1),PREC(NDEL,1),RLAM(1),ISTR(1),
     X  IEND(1),HX(1),HY(1),VIN(1),SORC1(1),SORC2(1)
      DO 10 NP2=1,NP2X
      IS=ISTR(NP2+1)
      IE=IEND(NP2+1)
      DO 20 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      HH=HX(NP1)*HY(NP2)/DT
      SUM1=FLUX(1,NPP)*HH*VIN(1)
      SUM2=FLUX(2,NPP)*HH*VIN(2)
      DO 30 ND=1,NDEL
   30 SUM1=SUM1+(PREC(ND,NPP)*RLAM(ND))/(1.0+DT*RLAM(ND))
      RHS(1,NPP)=SUM1
      RHS(2,NPP)=SUM2
      SORC1(NPP)=FLUX(1,NPP)
      SORC2(NPP)=FLUX(2,NPP)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE FDBK(SA1,TEMP,SORC,ISTR,IEND,NP1X,NP2X,
     X  NBOX,CORR,ND1X)
C
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF
C
      DIMENSION SA1(1),TEMP(1),SORC(1),ISTR(1),IEND(1)
      DIMENSION NBOX(ND1X,1),CORR(3,1)
      IF(ITHFB  .EQ.  0) RETURN
      SR=SQRT(TREF)
      DO 10 NP2=1,NP2X
      ND2=NP2+1
      IS=ISTR(ND2)
      IE=IEND(ND2)
      DO 20 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      IF(SORC(NPP)  .LE.  0.0) GO TO 20
      K=NBOX(ND1,ND2)
      SR1=CORR(3,K)
      TOT=SA1(NPP)-SR1
      TOT=TOT*(1.0+GAMMA*(SQRT(TEMP(NPP))-SR))
      SA1(NPP)=TOT+SR1
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE PREC1(PREC,FLUX,SORC1,SORC2,OMP,OMD,ISTR,IEND,FISS,
     X  BETA,RLAM,NP1X,NP2X,NDEL,RL,DTI)
      DIMENSION PREC(NDEL,1),FLUX(2,1),SORC1(1),SORC2(1),OMP(2,1),
     X  OMD(NDEL,1),ISTR(1),IEND(1),FISS(2,1),BETA(1),RLAM(1)
      DO 10 NP2=1,NP2X
      IS=ISTR(NP2+1)
      IE=IEND(NP2+1)
      DO 20 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      S1=FLUX(1,NPP)
      S2=FLUX(2,NPP)
      F1=S1/SORC1(NPP)
      F2=S2/SORC2(NPP)
      IF(F1  .LE.  1.0  .OR.  F2  .LE.  1.0)  GO TO 15
      OMP(1,NPP)=DTI*ALOG(F1)
      OMP(2,NPP)=DTI*ALOG(F2)
      GO TO 50
   15 OMP(1,NPP)=0.0
      OMP(2,NPP)=0.0
   50 X1=(FISS(1,NPP)*FLUX(1,NPP)+FISS(2,NPP)*FLUX(2,NPP))/RL
      DO 30 ND=1,NDEL
      X=(X1*BETA(ND)+DTI*PREC(ND,NPP))/(DTI+RLAM(ND))
      IF(X  .LT.  1.0E-25) GO TO 31
      XOMG=(-RLAM(ND)*X+X1*BETA(ND))/X
      OMD(ND,NPP)=XOMG
   31 PREC(ND,NPP)=X
   30 CONTINUE
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE TEMP(FLUX,FISS,HX,HY,ISTR,IEND,NP1X,NP2X,TMPP,SORC1,
     X  SORC2,DT,JFIRST)
C
      COMMON / THFEED / ITHFB,XNU,WPCC,EPSIL,ALFA,GAMMA,TREF
C
      DIMENSION FLUX(2,1),FISS(2,1),HX(1),HY(1),ISTR(1),IEND(1),TMPP(1),
     X  SORC1(1),SORC2(1)
      LOGICAL JFIRST
      IF(ITHFB  .EQ.  0) RETURN
      DO 10 NP2=1,NP2X
      IS=ISTR(NP2+1)
      IE=IEND(NP2+1)
      DO 20 ND1=IS,IE
      NP1=ND1-1
      NPP=(NP2-1)*NP1X+NP1
      X=FLUX(1,NPP)*FISS(1,NPP)+FLUX(2,NPP)*FISS(2,NPP)
      HH=HX(NP1)*HY(NP2)*XNU
      SNEW=X/HH
      SOLD=SORC1(NPP)
      IF(JFIRST) SOLD=SNEW
      SHALF=(SNEW+SOLD)*0.5
      SORC1(NPP)=SNEW
      TNEW=TMPP(NPP)+ALFA*DT*SHALF
      TMPP(NPP)=TNEW
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
      SUBROUTINE CORE
C
C
C
C=========STORAGE
C
      COMMON DATA(1)
      REAL       BT(999),AT(99),A(1)
      INTEGER    DATA,AV,KOT(999),ITYP(1),KXT(999)
      INTEGER    NT(999),ITYPET(99)
C
C
C
C
C
C=========GETCOR
C
      ENTRY GETCOR(KMAX)
C
       CALL ZIGET(DATA(1),KMAX,4,KS,5000,&905)
      KFREE=KS
      KMX =KMAX+KS-1
      KS=KS+MOD(KS+1,2)
      KMAX=KMX-KS+1
      GO TO 900
C
C
C=========CLEAR
C
      ENTRY CLEAR
C
      IN=5
      IOUT=6
      IERFL=0
      IXX=999
      DO 110 K=KS,KMX
```

```
  110 DATA(K)=0
      DO 120 I=1,IXX
      BT(I)=0.0
      NT(I)=0
      KOT(I)=0
  120 KXT(I)=0
      IX=0
      KLST=0
      JX=0
      IP=0
      JXX=99
      DO 125 I=1,JXX
      AT(I)=0.0
  125 ITYPET(I)=0
      GO TO 900
C
C
C==========FREE DATA SPACE
C
      ENTRY FRECOR
      CALL ZIFREE (DATA(KFREE),&907)
      GO TO 900
C
C
C=========ALLOCATE
C
      ENTRY ALOC(B,N,KORG,KN)
      KY=KN+MOD(KN,2)
      IF(KY.LE.0) GO TO 900
      IF(N.LT.0) GO TO 900
      KAY=2
      ASSIGN 150 TO KAYP
      GO TO 300
C
  150 IF(IX.EQ.0) KORG=KS
      IF(IX.GT.0) KORG=KOT(IX)+KXT(IX)
      KXP=KORG+KY-1
      IF(KXP.GT.KMX) GO TO 901
      IF(KXP.GT.KLST) KLST=KXP
      IX=IX+1
      IF(IX.GT.IXX) GO TO 902
      BT(IX)=B
      NT(IX)=N
      KOT(IX)=KORG
      KXT(IX)=KY
      IF(KAY .EQ. 7) GO TO 840
      GO TO 900
C
C
C=========LENGTH
C
      ENTRY LENGTH(B,N,KXXX, *  )
C
C
C=========FIND
C
      ENTRY FIND(B,N,KORG, *  )
      KEY=1
      ASSIGN 230 TO KEYP
      ASSIGN 900 TO KOYP
200   IS=MAX0(IP,1)
      IF (IS.GT.IX) IS=1
      IE=IX
205   DO 210 I=IS,IE
      IF(B.NE.BT(I)) GO TO 210
      IF(N.NE.NT(I).AND.N.GE.0) GO TO 210
      IP=I
      GO TO 220
C
  210 CONTINUE
      IE=IP-1
      IP=0
      IF(IS.EQ.1) GO TO 220
      IS=1
      GO TO 205
C
  220 GO TO KEYP, (230,250,310,        560,620,670,     830,900)
C
  230 IF(IP.NE.0) GO TO 231
      GO TO 999
C
  231 KXXX=KXT(IP)
      KORG=KOT(IP)
      GO TO 900
C
C
C=========FILL
C
      ENTRY FILL(B,N,KORG,AV)
      KEY=2
      ASSIGN 250 TO KEYP
      ASSIGN 900 TO KOYP
      GO TO 200
  250 IF(IP.EQ.0) GO TO 900
      KORG=KOT(IP)
      KX=KXT(IP)
      DO 260 K=1,KX
  260 DATA(KORG+K-1)=AV
      GO TO 900
C
C
C=========DROP
C
      ENTRY DROP(B,N)
  290 KAY=1
      ASSIGN 900 TO KAYP
  300 KEY=3
      ASSIGN 310 TO KEYP
      ASSIGN 900 TO KOYP
      GO TO 200
C
  310 IF(IP.EQ.0) GO TO 380
      IF(KAY  .NE.  2) GO TO 315
      KK=KXT(IP)
      IF (KY .NE. KK)  GO TO 315
      AV=0
      GO TO 250
  315 CONTINUE
C
      IF(IP.EQ.IX) GO TO 350
      KL=KOT(IP+1)
      KH=KOT(IX)+KXT(IX)-1
      KX=KXT(IP)
      DO 320 K=KL,KH
  320 DAT A(K-KX)=DATA(K)
      KL=KH-KX+1
      DO 330 K=KL,KH
  330 DATA(K)=0
      IL=IP+1
      DO 340 I=IL,IX
      BT(I-1)=BT(I)
      KOT(I-1)=KOT(I)-KX
      NT(I-1)=NT(I)
      KXT(I-1)=KXT(I)
  340 CONTINUE
      GO TO 370
C
  350 KO=KOT(IP)
      KX=KXT(IP)
      DO 360 K=1,KX
  360 DATA(KO+K-1)=0
  370 BT(IX)=0.0
      NT(IX)=0
      KOT(IX)=0
      KXT(IX)=0
      IX=IX-1
C
  380 GO TO KAYP, (150,510,    900)
C
C
C=========LIST
C
      ENTRY LIST
  385 IF(IX.LE.0) GO TO 900
      WRITE(IOUT,390) IX,(BT(I),NT(I),KOT(I),KXT(I),I=1,IX)
  390 FORMAT(1H0,I3,7H BLOCKS/5H NAME,8X,13HORIGIN LENGTH/
     $       1X,10H----------,2(2X,6H------)/(1X,A4,I6,2I8))
      IF(IERFL.EQ.1) GO TO 386
      GO TO 900
C
C
C=========COUNT
C
      ENTRY COUNT
      WRITE(IOUT,500) KLST
  500 FORMAT(28H0LAST DATA LOCATION USED WAS,I6)
      GO TO 900
C
C
C=========OPEN
C
      ENTRY OPEN(B,N,KORG)
      IF(N.LT.0) GO TO 900
      KAY=3
      ASSIGN 510 TO KAYP
      GO TO 300
C
  510 IF(IX.EQ.0) KORG=KS
      IF(IX.GT.0) KORG=KOT(IX)+KXT(IX)
      IF(IX.GE.IXX) GO TO 902
      BT(IX+1)=B
      NT(IX+1)=N
      KOT(IX+1)=KORG
      GO TO 900
C
C
C=========CLOSE
C
      ENTRY CLOSE(B,N,KXN)
      IX=IX+1
      IF(IX.GT.IXX) GO TO 902
      IF(B.NE.BT(IX).OR.N.NE.NT(IX)) GO TO 903
      KXX=KXN+MOD(KXN,2)+KOT(IX)-1
      IF(KXX.GT.KMX) GO TO 901
      KXT(IX)=KXX-KOT(IX)+1
      IF(KXX.GT.KLST) KLST=KXX
      GO TO 900
C
C
C=========COPYB
C
      ENTRY COPYB(AA,M,BP,NP,KORG, *  )
      NCOPYB=1
      B=AA
      N=M
      KEY=21
      ASSIGN 830 TO KEYP
      ASSIGN 900 TO KOYP
      GO TO 200
C
 830  IF(IP.NE.0)   GO TO 831
      GO TO 999
C
  831 IF(NCOPYB.EQ.2) GO TO 832
      B=BP
      N=NP
      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
      DATA(K)=0
564   CONTINUE
      GO TO 900
C
  565 KL=KOT(IP+1)
      KH=KOT(IX)+KXT(IX)-1
      KX=KXT(IP)-KN-MOD(KN,2)
      IF(KX.LT.1) GO TO 904
      DO 570 K=KL,KH
  570 DATA(K-KX)=DATA(K)
      KXT(IP)=KN+MOD(KN,2)
      IL=IP+1
      DO 575 I=IL,IX
  575 KOT(I)=KOT(I)-KX
      KO=KOT(IX)+KXT(IX)-1
      DO 580 K=1,KX
  580 DATA(KC+K)=0
      GO TO 900
C
C
C=========SETNAM
C
      ENTRY SETNAM(A,ITYP,JXP)
      IF(JXP.LT.1) GO TO 900
      KEY=6
      ASSIGN 900 TO KEYP
      ASSIGN 600 TO KOYP
      DO 610 I=1,JXP
      AP=A(I)
      GO TO 620
C
600   IF(JP.GT.0) GO TO 610
      IF (JX+1.GT.JXX) GO TO 902
      JX=JX+1
      JP=JX
      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
C
C
C=========EXCHANGE NAME
C
      ENTRY EXNAME(AA,M,BP,NP, *  )
      KEY=14
      ASSIGN 620 TO KEYP
      ASSIGN 780 TO KOYP
      N=NP
      B=BP
      AP=BP
      GO TO 200
C
750   IF(IP.GT.0) GO TO 751
      GO TO 999
C
  751 I1P=IP
      J1P=JP
      AP=AA
      KEY=15
      ASSIGN 620 TO KEYP
      ASSIGN 760 TO KOYP
      B=AA
      N=M
      GO TO 200
C
760   IF (IP.GT.0) GO TO 761
      GO TO 999
C
  761 BT(IP)=BT(I1P)
      NT(IP)=NT(I1P)
      BT(I1P)=AA
      NT(I1P)=M
      IF(J1P.GT.0) AT(J1P)=AA
      IF(JP.GT.0) AT(JP)=BP
      GO TO 900
C
C
C=========RE-NAME
C
      ENTRY RENAME (AA,M,BP,NP, *  )
      KEY=16
      ASSIGN 620 TO KEYP
      ASSIGN 780 TO KOYP
      B=BP
      N=NP
      AP=B
      GO TO 200
C
 780  CONTINUE
      IF(IP.LE.0) GO TO 781
      GO TO 999
C
  781 IF(JP.NE.0) GO TO 782
      JX=JX+1
      IF(JX.GT.JXX)  GO TO 902
      JP=JX
      AT(JX)=AP
C
  782 KEY=17
      ASSIGN 620 TO KEYP
      ASSIGN 790 TO KOYP
      AP=AA
      N=M
      B=AA
      GO TO 200
C
 790  CONTINUE
      IF(IP.GE.1) GO TO 791
      GO TO 999
C
```

```
  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)
 9030 FORMAT(38H0*** CORE *** CLOSING BLOCK IMPROPERLY)
      KXA=KMX-(KOT(IX)+KXT(IX))+1
      WRITE(IOUT,162)  KXA
      STOP 1
  904 WRITE(IOUT,9040)
 9040 FORMAT(33H0*** CORE *** IMPROPER CLIP ENTRY)
      KXA=KMX-(KOT(IX)+KXT(IX))+1
      WRITE(IOUT,162)  KXA
      STOP 1
  905 WRITE(IOUT,9050)
 9050 FORMAT(41H0********** PROBLEM WITH ZIGET **********)
      STOP 1
  907 WRITE(IOUT,9070)
 9070 FORMAT(42H0********** PROBLEM WITH ZIFREE **********)
      STOP 1234
      END
      SUBROUTINE GPRNT(BL,NBL,LX,KORG  ,IDT,I1X)
C
      COMMON        DATA(1)
      INTEGER   I4DATA(1)
      REAL   R4DATA(1)
      DOUBLE PRECISION R8DATA(1)
      EQUIVALENCE (R8DATA(1),R4DATA(1),I4DATA(1),DATA(1))
C
C
       INTEGER   FMTI4(5),FMTR48( 7),INTGR(20)
      DATA  FMTI4 / 20H(1H+,12X,      I6/) /,
     $      FMTR48 / 28H(1H+,11X,       (1PE12.4)/)  /,
     $      INTGR /  80H   1   2   3   4   5   6   7   8   9  10  11  12
     $  13  14  15  16  17  18  19  20  /
C
C
      IN=5
      IOUT=6
C
C
      LO=0
      LOX=0
      NPL=10
       LIX=I1X
C
C
      IF((IDT.GE.1).AND.(IDT.LE.3)) GO TO 109
      WRITE(IOUT,9020) BL,NBL,IDT
      GO TO 900
C
C
 109  GO TO (120,110,120                      ),IDT
  110 NPL=20
C
C
  120 IF(LIX.GT.1) GO TO 130
```

```
      GO TO (121,121,122                            ), IDT
  121 LIX=LX
      GO TO 300
  122 LIX=LX/2
      GO TO 300
C
C
  130 GO TO (140,140,150                            ), IDT
  140 LOX=LX/LIX
      GO TO 160
  150 LX=LX/2
      LOX=LX/LIX
  160 CONTINUE
C
C
  300 WRITE(IOUT,6010) BL,NBL
      IF(LOX.EQ.0) GO TO 330
      WRITE(IOUT,7010)
  330 IPL=NPL/10
      GO TO (340,350),IPL
  340 WRITE(IOUT,7021)
      GO TO 360
  350 WRITE(IOUT,7022)
  360 IF(LOX.EQ.0) GO TO 370
      WRITE(IOUT,7030)
  370 WRITE(IOUT,7040)
C
C
  200 LO=LO+1
C
C
      GO TO (204,204,205                            ), IDT
  204 IR=(LO-1)*LIX+KORG-1
      GO TO 206
  205 IR=(LO-1)*LIX+(KORG-1)/2
C
```

```
C
  206 IF(LOX.EQ.0) GO TO 202
      IF(LO.EQ.1) GO TO 201
      IF(LIX.LE.NPL) GO TO 201
      WRITE(IOUT,7050)
  201 WRITE(IOUT,8030) LO
C
C
  202 L1=0
      L2=0
  207 L1=L2+1
      LR=L1-1
      L2=LR+NPL
      IF(L2.GT.LIX) L2=LIX
      IF(LOX.EQ.0) GO TO 203
      WRITE(IOUT,7060)
  203 WRITE(IOUT,8010) LR
C
      GO TO (210,230,220                                 ), IDT
  210 FMTR48(4)=INTGR(L2-L1+1)
      WRITE(IOUT,FMTR48)
     $                (R4DATA(IR+I),I=L1,L2)
      GO TO 240
  220 FMTR48(4)=INTGR(L2-L1+1)
      WRITE(IOUT,FMTR48)
     $                (R8DATA(IR+I),I=L1,L2)
      GO TO 240
  230 FMTI4(4)=INTGR(L2-L1+1)
      WRITE(IOUT,FMTI4)
     $               (I4DATA(IR+I),I=L1,L2)
C
  240 IF(L2.EQ.LIX) GO TO 250
      GO TO 207
C
  250 IF(LOX.EQ.0) GO TO 900
      IF(LO.LT.LOX) GO TO 200
```

```
C
C
  900 RETURN
C
C
 6010 FORMAT(1H0,60X,16H================/
     $       1X,60X,3H|| ,A4,1H.,I5,3H ||/
     $       1X,60X,16H================///)
 7010 FORMAT(1H+,1X,2HI2,1X,1H|)
 7021 FORMAT(1H+,7X,2HI1,1X,1H/,
     $       6X,2H+1,10X,2H+2,10X,2H+3,10X,2H+4,10X,2H+5,
     $       10X,2H+6,10X,2H+7,10X,2H+8,10X,2H+9,10X,3H+10/)
 7022 FORMAT(1H+,7X,2HI1,1X,1H/,
     $       4X,2H+1,4X,2H+2,4X,2H+3,4X,2H+4,4X,2H+5,
     $       4X,2H+6,4X,2H+7,4X,2H+8,4X,2H+9,3X,3H+10,
     $       3X,3H+11,3X,3H+12,3X,3H+13,3X,3H+14,3X,3H+15,
     $       3X,3H+16,3X,3H+17,3X,3H+18,3X,3H+19,3X,3H+20/)
 7030 FORMAT(1H+,5(1H-)/1H+,4X,1H|)
 7040 FORMAT(1H+,5X,127(1H-)/1H+,10X,1H|/)
 7050 FORMAT(1H+,132(1H-)/1H+,4X,1H|,5X,1H|/)
 7060 FORMAT(1H+,4X,1H|)
 8010 FORMAT(1H+,5X,I4,1X,1H|)
 8030 FORMAT(1H+,I3)
 9020 FORMAT(//1X,45HDATA TYPE INTEGER QUALIFIER OUT OF RANGE FOR ,A4,I5
     $      /1X,10HDATA TYPE ,I4//)
      END
```

```
*     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    FQU   *                                                       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
```