T*DAMEN, A Computer Code for Transient Radiation Damage Analysis

T.O. Hunter and G.L. Kulcinski

May 1978

UWFDM-247

Not widely distributed.

FUSION TECHNOLOGY INSTITUTE

UNIVERSITY OF WISCONSIN

MADISON WISCONSIN

T*DAMEN, A Computer Code for Transient Radiation Damage Analysis

T.O. Hunter and G.L. Kulcinski

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

http://fti.neep.wisc.edu

May 1978

UWFDM-247

Not widely distributed.

T-DAMEN

A Computer Code for Transient Radiation Damage Analysis

Thomas O. Hunter*

G. L. Kulcinski

April 1978

Fusion Research Program
University of Wisconsin
Madison, Wisconsin 53706

UWFDM-247

*Member Technical Staff, Sandia Laboratories, Albuquerque, NM 87115.
Work partially supported by the U.S. Department of Energy.

Table of Contents

I. INTRODUCTION

The T-DAMEN computer code is a general program for analysis of the transient radiation damage produced in materials from pulsed thermonuclear radiation. The models incorporated are discussed in UWFDM-196, 217, 232. The code was developed to provide a first order analysis of the energy deposition, temperature response, displacement production, and other subsequent effects produced in materials by transient pulses of photons or ions. The models used are approximate solutions to problems of ion and photon transport, radiation deposition, heat conduction, and primary defect production. These solutions are sufficiently efficient to allow simultaneous analysis of a wide range of ion and photon spectra which may be arbitrarily specified.

T-DAMEN is not meant to be used as a tool for precise analysis of any one specific phenomena, e.g., ion implantation distributions; but rather as a tool for assessment of combined effects and parametric analyses. It is for these reasons that the code has been used for applications such as the response of first walls in inertial confinement fusion reactors.

The code contains a complete data handling package including generation of spectra, intermediate data storage, and plotting. In addition, various independent routines for developing input data are included in the code. T-DAMEN is written in Fortran V for the UNIVAC-1110 at the University of Wisconsin, Madison. All routines with the exception of the file handling and plotting should be readily adaptable to any computer system.

The code is compatible with the Univac file system on mass storage devices. In general, a single file contains the entire program and is divided into many elements. Elements on the 1110 are of three types:

Symbolic - Fortran or run stream statements (similar to card images).

Relocatable - Compiled versions of FORTRAN symbolic elements.

Absolute - Executable machine language combinations of various relocatables.

Subsequent discussion will describe the structure of the symbolic elements associated with each absolute element. Input instructions will also be given for each program segment and finally examples will be given for each segment of the code. Also included in the final chapter are reference files which contain data values for various selected materials.

II. Description

The T-DAMEN code is divided into two major categories: Ion response and photon response. Each of these are in turn supported by routines which can develop input data, superimpose results, file output, and plot results.

The major sections and respective absolute elements of the code are shown in Figure 1 and are outlined as follows:

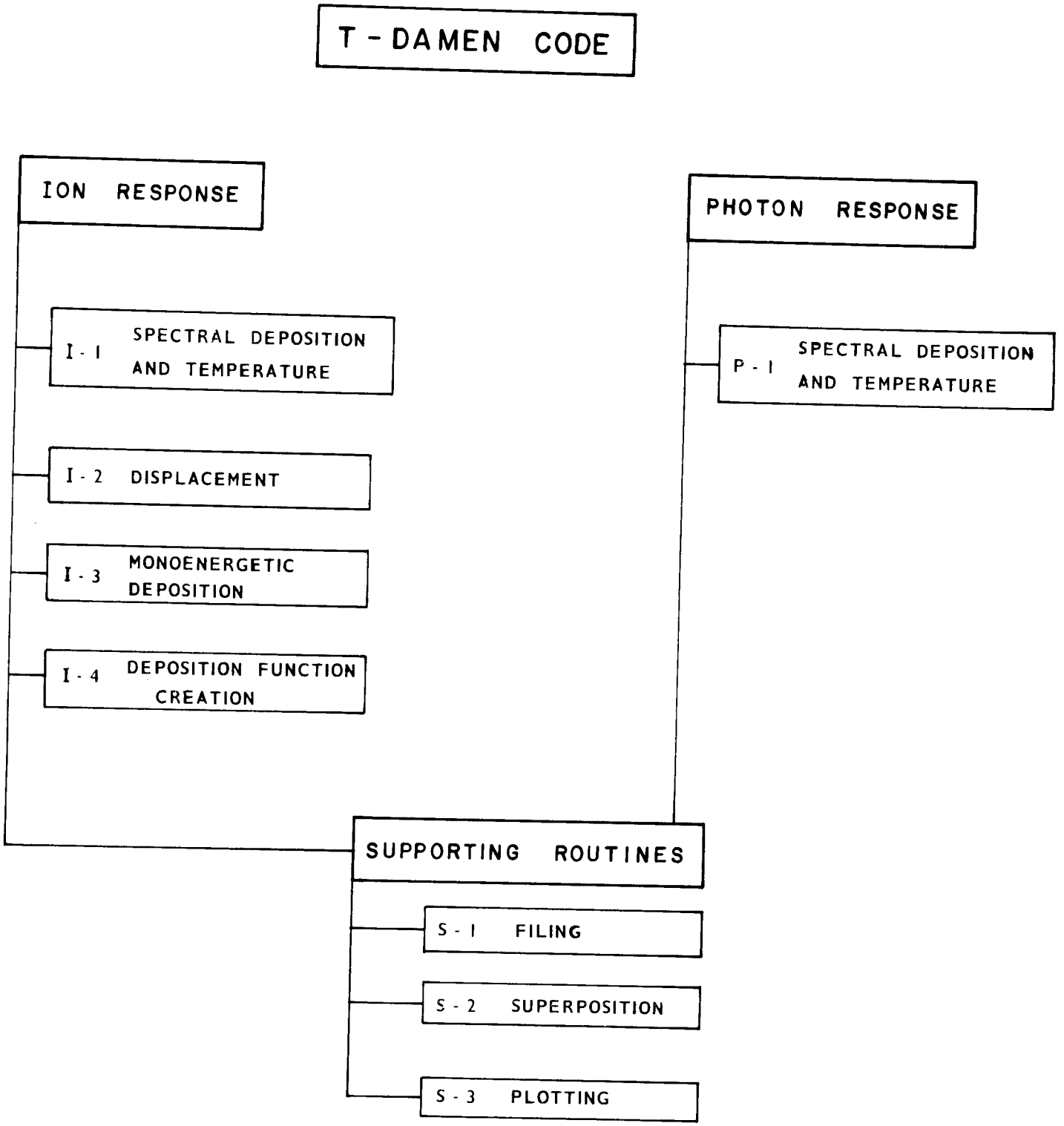
II.A. Photon Response

II.A.1. [*P-1*] *Spectral Deposition and Temperature* - This routine calculates the volumetric energy deposition for X-ray spectra or monoenergetic photons. Spectra may be specified as blackbodies or in histogram form. Deposition is based on a general library of photoelectric and incoherent cross sections which incudes all materials. Temperature calculations are done for the adiabatic case, an impulse solution, and a finite duration deposition. Gas protection is incorporated by allowing a total of 4 material layers through which the spectrum is modified. The absolute element is M/E.

II.B. Ion Response

II.B.1. [*I-1*] *Spectral Deposition and Temperature* - (IONCODE) This is the largest routine in T-DAMEN and contains numerous subroutines for the general responses of ions in materials. This program can generate ion spectra in the form of Maxwellians, Gaussians, or histograms. These spectra are normally of two types, either light ions ($Z \leq 2$) at high energy or other ions at any energy. The original spectra can be modified by a gaseous layer of specified pressure, temperature and type.

The flux which strikes the material is transformed into the time and energy dependent deposition profiles which are in the form of polynomials.


T - DAMEN CODE
ION RESPONSE
I - 1 SPECTRAL DEPOSITION AND TEMPERATURE
I - 2 DISPLACEMENT
I - 3 MONOENERGETIC DEPOSITION
I - 4 DEPOSITION FUNCTION CREATION
PHOTON RESPONSE
P - 1 SPECTRAL DEPOSITION AND TEMPERATURE
SUPPORTING ROUTINES
S - 1 FILING
S - 2 SUPERPOSITION
S - 3 PLOTTING

These polynomials are then used to reconstruct the volumetric energy deposition as a function of time.

The coefficients of the deposition polynomials are then supplied to the temperature routine to determine the temperature histories at various positions. Temperature may be determined on a time base which is determined by the ion arrival times or on a predetermined standard time base which allows comparison of response from various components.

The related thermal response solution for a heat flux on a semi-infinite slab is also contained as an independent routine. In addition, the eigenvalue solution for finite width materials after many pulses is also contained as a subroutine. Routines for assessment of temperature dependent sputtering, surface evaporation and ion implantation profiles are also included.

II.B.2. [*I-2*] *Displacement* - This routine provides the displacement production from arbitrary ion spectra. Spectra are generated and modified by gaseous layers as described above for spectral depositions and temperatures. Response is again performed for either light or heavy ions.

Light ion displacement calculations are performed by determining the mean local ion energy and an appropriate displacement cross section. Heavy ion calculations are based on the nuclear energy deposition functions described in section II.B.4.

Either method can develop the temporal and spatial dependent displacement production on a time base imposed by the ion arrivals or on a standard time base compatible with all the ion components.

II.B.3. [*I-3*] *Single Ion Deposition* - This routine allows determination of the deposition of a monoenergetic ion into a material. Either light or heavy

ions can be considered. Light ion calculations are based on the stopping power theory of Brice and the relations developed in FDM-217. Heavy ion calculations are based on the deposition functions which will be discussed below.

The light ion version contains an internal plotting package and is primarily used to evaluate input parameters for data supplied to the spectral deposition and temperature routine. The heavy ion version does require an external data file (see 4. below) from which deposition coefficients are obtained.

II.B.4. [*I-4*] *Deposition Function Creation* - This routine generates the deposition function which is used to determine the volumetric deposition of energy to both nuclear and electronic processes. This routine is used in conjunction with either the Brice implantation codes or tabulated deposition profiles available in the literature.

In the former case the Brice codes are used to obtain deposition profiles of a few energies for a specific ion target combination. These data are then transformed to proper format and placed in a data file. The file is read by the DEPFUN routine and the spatial profiles fit with polynomials by a least squares technique. The polynomials and their associated energies are stored in another data file for access by the routines above.

For tabulated data the procedure is identical except the data are first placed in a data element. After the deposition coefficients are created, deposition profiles may be reproduced and plotted for comparison with the original values.

II.C. Supporting Routines

II.C.1. [*S-1*] *Filing* - This routine is a general data handling system based on the IODR random access file system on the Univac-1110. Data which are generated by the photon as ion code may be placed in a data file according to

a prescribed format. In addition, a directory of file contents is simultaneously maintained so that the results from many executions may be identified. Any portion of the data may be subsequently addressed for superposition, printed output, or plotting.

II.C.2. [*S-2*] *Superposition* - Once data have been filed in a standard format it is possible with this routine to superimpose the results of several independent calculations; e.g., photons and ions. Selected "bins" within a data file are accessed, the results are combined, and the summations are refiled in another data bin with appropriate directory notation.

II.C.3. [*S-3*] *Plotting* - A plotting package for two-dimensional plots of selected variables from either ion or photon calculations is contained in this routine. Fifteen basic plots are available. Each may be obtained from any "bin" within the data file discussed above. The format is set-up for interactive graphics terminals and allows instantaneous selection of logarithmic or linear scales and choice of variable.

In the standard package the number of points for each plot is pre-determined by the filing code and the data are all drawn from the same data file. An alternate routine is available for a user specified number of points. Another routine is available for plotting of data from several data files.

III. Outline and Listings

The following sections contain information describing the construction of each routine discussed above. Each section includes:

a. A function Block diagram.

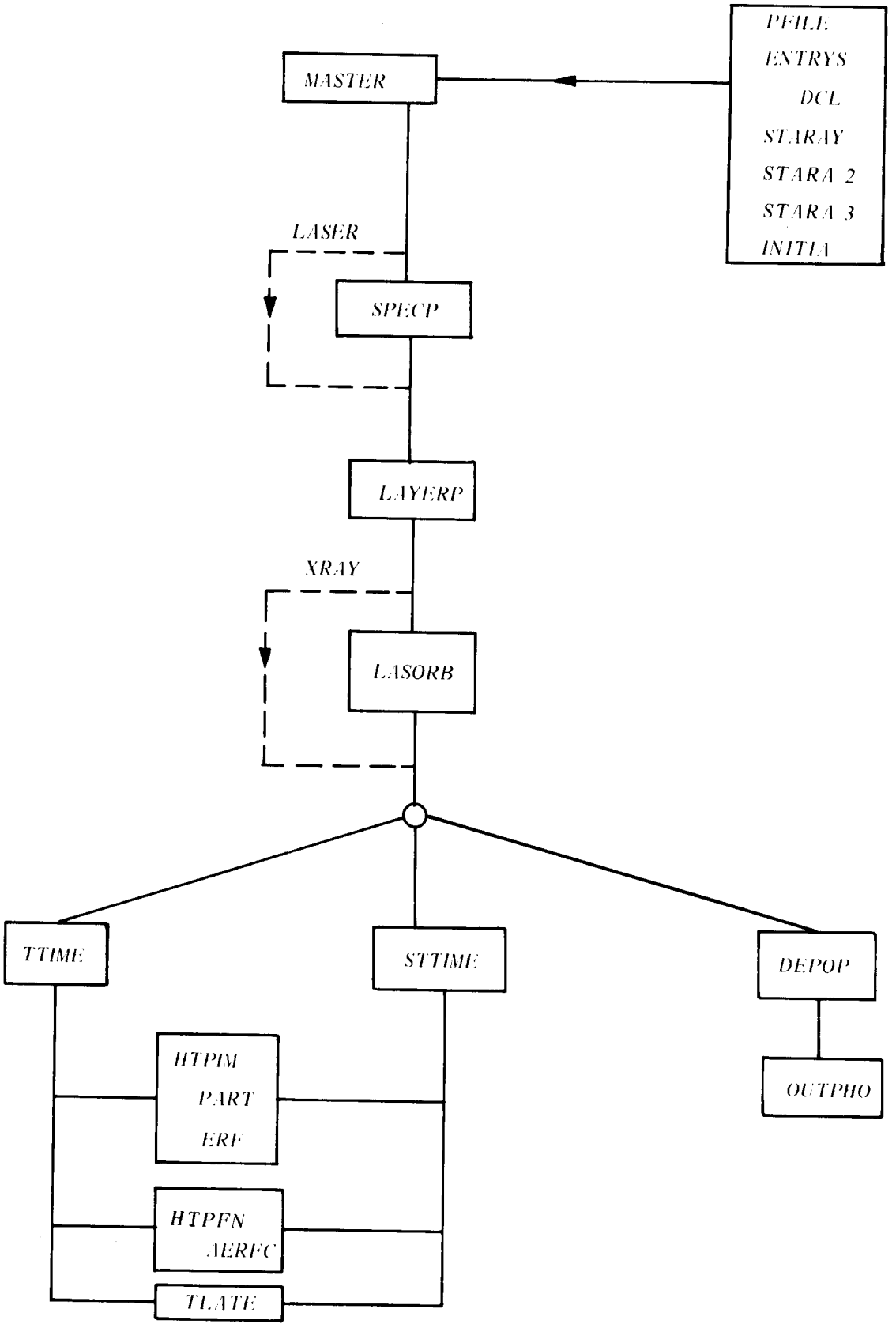
b. A descriptive outline of the elements within each routine.

c. A reproduction of the read statements in each routine.

d. A set of commands for "mapping" which shows which relocatable elements were collected into the absolute element.

e. A listing of all the elements in each routine.

PHOTON CODE


PFILE
ENTRYS
DCL
STARAY
STARA 2
STARA 3
INITIA
MASTER
LASER
SPECP
LAYERP
XRAY
LASORB
TTIME
STTIME
DEPOP
HTPIM
PART
ERF
OUTPHO
HTPFN
AERFC
TLATE

===== PHOTON =====

PHOTON RESPONSE
SECTION OF T*DAMEN CODE

ABSOLUTE
M/E

READER
READ/PHOTON

MAPPER
MAP/PHOTON

LISTER
LIST/PHOTON

SYMBOLICS
MASTER --MASTER ROUTINE FOR CALLING EACH SECTION
SPECP --CALCULATES A BLACK BODY SPECTRUM FOR A GIVEN TEMPERATURE AND TOTAL FLUX
LAYERP --SETS UP MATERIAL GRID UP TO 4 LAYERS WITH ANY NUMBER OR MATERIAL'S PER LAYER, DETERMINES COMPOSITE ABSORPTION COEF.,DENSITY,GAS PROP. ETC
INITIA --CALLS CROS TO GET CROSS SECTION DATA FROM ELT/ATOM
CROS --SELECTS FROM THE CROSS SECTION LIBRARY THE PHOTON ELECTRIC AND INCOHERENT VALVES FOR A GIVEN Z, READS IN ENTIRE LIBRARY
ELT/ATOM-ALL PHOTON CROSSECTIONS Z FROM 1-100, PLUS CALL TO INITIA WHICH READS THEM
GEN/XMU--CALCULATE PHOTON CROSSECTION FOR A GIVEN ENERGY FROM COEFFICIENTS IN CROS
DEPOP --CALCULATES PHOTON DEPOSITION, J/CUBIC CM, TRANSMITTED SPECTRA, ADIABATIC TEMPERATURE
TTIME --EVALUATES TEMPERATURE AT ANY TIME OR POSITION FOR ANY OF 3 MODELS
STTIME --SAME AS TTIME EXCEPT X AND T ARE STANDARD ARRAYS
HTPIM --EXPONENTIAL IN SPACE, IMPULSE IN TIME TEMPERATURE SOLUTION (MODEL 1)
HTPFN --EXPONENTIAL IN SPACE, FINITE DURATION IN TIME TEMPERATURE SOLUTION (MODEL 2)
PART --A SUBROUTINE NECESSARY IN HTPIM DERIVED FROM EXPRESSION FOR ERROR FUNCTION
AERFC --INTEGRAL COMPLEMENTARY ERROR FUNCTION
ERF --ERROR FUNCTION, SINGLE PRECISION
TLATE --A SOLUTION TO THE FINITE SLAB FOR EXPONENTIAL IN SPACE AND IMPULSE IN TIME (MODEL 3)

===== PHOTON =====

OUTPHO --A ROUTINE FOR OUTPUTTING THE RESULTS OF THE PHOTON CODE

LASORB --GENERATES A SINGLE MONO ENERGETIC ATTENUATION COEFFICIENT SIMILIAR TO LASER ATTENUATION

STARAY --GENERATES THE STANDARD X AND TIME ARRAYS

PFILE/IODR--FILES THE COMMON BLOCK DATA FROM THE PHOTON CODE INTO A PHOTON IODR FILE

ENTRYS/IODR--BASIC SUBROUTINE FOR IODR DATA BLOCKS

DCL/IODR-STRUCTURE OF DATA BLOCKS FOR IODR FILE SYSTEM

```
=====  READ/PHOTON  =====

                *****READ STATEMENTS FOR THE PHOTON RESPONSE CODE*****


           FOR THE MASTER SUBROUTINE:

      READ(5,10,END=99)INST
10    FORMAT(A6)
      IF(INST .EQ. 'SPECAR') CALL SPECAR
      IF (INST .EQ. 'INITIA') CALL INITIA
      IF(INST .EQ. 'STARAY') CALL STARAY
      IF(INST .EQ.'PSTEMP') CALL STTIME
      IF (INST .EQ. 'P-LAYE') CALL LAYERP
      IF (INST .EQ. 'P-SPEC') CALL SPECP
      IF (INST .EQ. 'P-DEPO') CALL DEPOP
      IF (INST .EQ. 'P-TIME' ) CALL TIMEP
      IF (INST .EQ. 'P-OUTP' ) CALL OUTPHO
      IF (INST .EQ. 'P-LASE' ) CALL LASORB
      IF (INST .EQ. 'OPEN') CALL OPEN
      IF (INST .EQ. 'REOPEN') CALL REOPEN
      IF (INST .EQ. 'LOGIN') CALL LOGIN
      IF (INST .EQ. 'P-FILE') CALL FILEP


           FOR THE SPECP SUBROUTINE:

      READ ,KEV,JK,FLUX,RADIUS
      IF(KEV .GT. 0.) GO TO 9
      READ ,NMHIST,(EHIST(I),AMP(I), I = 1,NMHIST)


           FOR THE LAYERP SUBROUTINE:

      READ(5,-,END=99)NUM,WIDTH(NUM),IWID(NUM)
      READ(5,-,END=98) IZ,RO,C(NUM),ALPHA,IG,AM,P,GTMP
      CALL  ONESET(IZ)
      IF (IG .EQ. 1) RO=AM*P/GTMP*1.721E-5


           FOR THE DEPOP SUBROUTINE:

      READ ,LAYLO,LAYHI


           FOR THE TTIME SUBROUTINE:

     READ    ,NLAYR,IMOD,SIZE
     READ,NUMY,(Y(I),I=1,NUMY),NUMT,(TM(I),I=1,NUMT),TD,W,NPLUS


           FOR THE STTIME SUBROUTINE:

      READ  ,NLAYR,IMOD,TD,w,NPLUS,SIZE
      READ,IXS,IXF,IXD


           FOR THE LASORB SUBROUTINE:
```

```
      =====  READ/PHOTON  =====

58.
59.          READ ,WAVLTH,FLUX,RADIUS
60.
```

```
      =====  MAP  =====

 1.     @MAP*MAP.MAP       ,T*DAMEN.M/E
 2.     IN T*DAMEN.MASTER,.LAYERP,.INITIA,.SPECP,.GEN/XMU,.DEPOP
 3.     IN   T*DAMEN.TTIME,.HTPIM,.HTPFN,.PART,.AERFC,.ERF
 4.     IN T*DAMEN.DERF
 5.     IN   T*DAMEN.TLATE,.OUTPHO
 6.     IN   T*DAMEN.LASORB
 7.     IN T*DAMEN.STARAY,.STTIME
 8.     IN   T*DAMEN.PFILE/IODR,.ENTRYS/IODR
 9.     NOT T*DAMEN.PFILE
10.     IN  T*DAMEN.CROS
11.     NOT T*DAMEN.XMU
12.     END
```

```
=====  LIST/PHOTON  =====

 1.     @PRT,S  T*DAMEN.LIST/PHOTON
 2.     @PRT,S  T*DAMEN.MASTER
 3.     @PRT,S  T*DAMEN.SPECP
 4.     @PRT,S  T*DAMEN.LAYERP
 5.     @PRT,S  T*DAMEN.INITIA
 6.     @PRT,S  T*DAMEN.GEN/XMU
 7.     @PRT,S  T*DAMEN.DEPOP
 8.     @PRT,S  T*DAMEN.TTIME
 9.     @PRT,S  T*DAMEN.STTIME
10.     @PRT,S  T*DAMEN.HTPIM
11.     @PRT,S  T*DAMEN.HTPFN
12.     @PRT,S  T*DAMEN.PART
13.     @PRT,S  T*DAMEN.AERFC
14.     @PRT,S  T*DAMEN.ERF
15.     @PRT,S  T*DAMEN.TLATE
16.     @PRT,S  T*DAMEN.OUTPHO
17.     @PRT,S  T*DAMEN.LASORB
18.     @PRT,S  T*DAMEN.STARAY
19.     @PRT,S  T*DAMEN.PFILE/IODR
20.     @PRT,S  T*DAMEN.ENTRYS/IODR
21.     @ .     SEE FILING
22.     @PRT,S  T*DAMEN.CROS
23.     @PRT,S  T*DAMEN.ELT/ATOM
24.     @PRT,S  T*DAMEN.DCL/IODR
25.     @ .     SEE FILING
```

```
=====   AERFC   =====

1.          FUNCTION AERFC(X)
2.           DOUBLE PRECISION X,DERF
3.          PI=3.14159
4.          AERFC   =-X*(1.-DERF(X))+1./SQRT(PI)*DEXP(-X*X)
5.          RETURN
6.          END
```

===== CROS =====

```
      4
            SUBROUTINE CROS
      C  APROGRAM TO READ THE X-RAY CROS SECTION LIBRARY
            DIMENSION A(3884),LOC(100),HEAD(10),B(971),ZOA(100)
            COMMON/XCOF/COEF(20,4),ELIM(20),NUM,ONEZOA
      100     FORMAT(A6)
      101     FORMAT(8E10.3)
      102     FORMAT(20I4)
      103     FORMAT(8F10.4)
      104     FORMAT(11F7.4)
            DO 2 I = 1,9
      2      READ 100, HEAD(I)
            READ 101, A
            READ 102,LOC
            READ 103,B
            READ 104, ZOA
             RETURN
            ENTRY ONESET(IZ)
            ONEZOA = ZOA(IZ)
            IPOS = LOC(IZ)
            IPOS1 = LOC(IZ +1)
            NUM = IPOS1 - IPOS
            ELIM(1) = .01
             DO 10 I = 1,NUM
            ELIM(I+ 1)  = B(IPOS - 1 + I)
      10     CONTINUE
            ICOS  = 4*(IPOS -1) +1
            DO 20 I = 1,NUM
            DO 20 J = 1,4
            IT = ICOS - 1 + J +4*(I-1)
            COEF(I,J) = A(IT)
      20     CONTINUE
            RETURN
            END
```

===== DEPOP =====

```
1.                      SUBROUTINE DEPOP
2.
3.      C          THIS SUB   CALCULATES THE DEPOSITION DUE TO PHOTONS
4.
5.      C        *OUTPUT OF THIS ROUTINE
6.               REAL*4  Y1(100,4),Y2(100,4)
7.               REAL*4   ADBTMP(100,4)
8.               COMMON  /PDEPO/ Y1,Y2,ADBTMP
9.
10.     C        *INPUT FROM LAYERP
11.              REAL*4  C(4),X2(4),RHO(4),U(100,4),X(100,4)
12.              INTEGER IX(4)
13.              COMMON /PLAY/ IX,X2,RHO,C,ALPHA,X,U
14.
15.     C        INPUT FROM  SPECP
16.              INTEGER JK
17.              REAL*4   DE,E(100),F(100,5)
18.              COMMON  /PSPEC/   JK,DE,E,F,KEV,FLUX,SUMFLU
19.
20.
21.     C        *START
22.              READ ,LAYLO,LAYHI
23.
24.     C        *DO THE GAS AND ONE LAYER OF WALL
25.              DO 11 NUMLAY=LAYLO,LAYHI
26.             CONST=RHO(NUMLAY)*C(NUMLAY)*4.186
27.
28.
29.              IXX=IX(NUMLAY)
30.             DO 22 I=1,IXX
31.                S1=0.
32.                S2=0.
33.
34.                DO 34 J=1,JK
35.                   F(J,NUMLAY+1)=EXP(-U(J,NUMLAY)*X(I,NUMLAY))*F(J,NUMLAY)
36.                   S1=S1+U(J,NUMLAY)*F(J,NUMLAY+1)*DE
37.                   S2=S2+F(J,NUMLAY+1)*DE
38.     34           CONTINUE
39.     33            CONTINUE
40.                Y1(I,NUMLAY)=S1
41.                Y2(I,NUMLAY)=S2
42.                ADBTMP(I,NUMLAY)=Y1(I,NUMLAY)/CONST
43.     22       CONTINUE
44.
45.     11        CONTINUE
46.
47.               RETURN
48.               END
```

```
      =====  ELT/ATOM  =====

INITIALIZE THE PHOTON CROSS SECTIONS

**********  XRAY COEFFICIENT DATA  **********
 COEFFICIENTS NOS. 1 THROUGH 100 AS OF REPORT SC-RR-710507
 REVISED AUGUST 1971
 THE LOWER ENERGY BOUND FOR THE DATA IN THIS FILE IS .01 KEV RATHER THAN THE
 .1 KEV VALUE USED IN THE PREVIOUS FILE.  TO USE THE ADDITIONAL DATA A MINOR
 CHANGE MUST BE MADE TO THE SUBROUTINE BIGGS.  THE CARD   IF(E.LT.0.1) E=0.1
 SHOULD BE REPLACED BY THE CARD     IF(E.LT.0.01) E=0.01     .
 3884  971  100
 0.          0.          0.          0.         -6.383E+01-6.446E+00 1.317E+01-5.045E-02
 3.051E+00-7.818E+00 1.144E+01 6.959E-02 7.636E-02-9.406E-01 6.144E+00 1.425E+00
 1.180E-03-8.236E-02 2.886E+00 5.534E+00 1.620E-05-5.610E-03 1.214E+00 1.761E+01
 1.034E-06-4.114E-04 6.287E-01 3.927E+01 4.200E-07 4.288E-04 2.492E-01 9.574E+01
 0.          0.          0.          0.         -7.150E+02 2.769E+02 4.455E+01-9.080E-01
 1.857E+01-4.079E+01 8.609E+01-2.618E+00 1.606E+00-1.726E+01 8.101E+01-4.928E+00
 3.663E-03-3.782E-01 2.288E+01 6.006E+01 1.080E-04-3.106E-02 1.325E+01 1.421E+02
 2.060E-05-1.074E-02 1.181E+01 1.704E+02 6.632E-06 9.453E-03 2.092E+00 1.728E+03
-2.418E+02 3.530E+01-2.699E-01 5.580E-04-1.888E+01 1.898E+01 2.381E+02-9.697E+00
 4.375E+00-5.094E+01 3.053E+02-3.064E+01 4.040E-02-2.930E+00 1.293E+02 1.828E+02
 1.414E-03-3.913E-01 7.450E+01 5.746E+02 3.769E-05 5.136E-03 3.650E+01 1.787E+03
 2.854E-05 4.028E-02 8.213E+00 8.286E+03-1.139E+02 9.342E+01-1.314E+00 5.228E-03
-6.254E+01 1.968E+02 5.011E+02-3.138E+01 1.037E+01-1.280E+02 8.807E+02-1.646E+02
 1.018E-01-8.264E+00 4.158E+02 4.366E+02 5.693E-03-1.595E+00 2.644E+02 1.566E+03
 4.261E-05 9.215E-02 9.639E+01 7.141E+03 9.101E-05 1.272E-01 2.504E+01 2.800E+04
-3.387E+02 1.923E+02-2.742E+00 1.204E-02-9.943E+02 1.816E+03 3.939E+02-1.755E+01
 3.689E+00-8.834E+01 1.525E+03-2.145E+02 6.447E-01-3.953E+01 1.281E+03 1.768E+02
 1.852E-03-1.356E+00 5.252E+02 5.160E+03 4.460E-04-6.960E-02 3.101E+02 1.521E+04
 2.273E-04 3.032E-01 1.013E+02 5.374E+04-6.531E+02 3.748E+02-5.790E+00 2.678E-02
-9.022E+02 1.760E+03 1.549E+03-2.280E+02-7.363E+00-1.537E+01 2.672E+03-4.482E+02
 1.640E+00-9.428E+01 2.872E+03-5.583E+02 5.621E-04-2.533E+00 1.169E+03 9.909E+03
 1.183E-03-2.891E-01 7.016E+02 3.359E+04 4.997E-04 6.323E-01 2.925E+02 9.317E+04
-8.291E+02 5.716E+02-1.010E+01 5.031E-02-2.338E+03 5.732E+03-2.062E+02 1.482E+02
-4.940E+00-8.442E+01 4.620E+03-1.186E+03 2.019E+00-1.249E+02 4.609E+03-9.421E+02
 1.709E-02-8.196E+00 2.345E+03 1.369E+04 1.719E-03-1.291E-01 1.192E+03 6.363E+04
 9.085E-04 1.142E+00 5.293E+02 1.787E+05-8.968E+02 8.086E+02-1.742E+01 1.025E-01
-7.181E+01 4.748E+02 5.542E+03-1.363E+03 2.745E+00-1.747E+02 7.159E+03-2.213E+03
 3.774E-02-1.559E+01 4.045E+03 1.810E+04 4.359E-03-1.372E+00 2.202E+03 9.357E+04
 1.521E-03 1.827E+00 1.132E+03 1.837E+05-6.041E+02 8.206E+02-1.627E+01 8.640E-02
-8.907E+01 6.850E+02 7.217E+03-2.022E+03 2.813E+00-2.020E+02 9.903E+03-4.454E+03
 6.097E-02-2.435E+01 6.099E+03 2.258E+04 5.064E-03-8.464E-01 3.076E+03 1.458E+05
 2.262E-03 2.631E+00 1.701E+03 3.144E+05 0.          0.          0.          0.
-1.183E+03 1.545E+03-5.393E+01 5.024E-01-8.680E+01 7.434E+02 1.077E+04-3.822E+03
 3.167E+00-2.491E+02 1.439E+04-8.187E+03 1.288E-01-4.471E+01 9.862E+03 2.497E+04
 5.202E-03 7.337E-02 4.614E+03 2.256E+05 3.536E-03 4.790E+00 1.147E+03 9.882E+05
 1.910E+01 1.575E+00-2.367E-02 8.190E-05-9.125E+02 1.627E+03-6.064E+01 4.588E-01
 1.732E+00-1.853E+02 1.712E+04-8.415E+03 2.726E-01-8.218E+01 1.475E+04 9.467E+03
 2.083E-03 4.548E+00 5.521E+03 3.358E+05 4.900E-03 6.789E+00 1.167E+03 1.600E+06
 3.150E+02-6.639E+00 3.825E-02-2.656E-05-8.778E+02 1.897E+03-2.962E+01-2.277E+00
-6.648E-01-1.084E+02 2.219E+04-1.241E+04 4.080E-01-1.205E+02 2.139E+04-1.143E+02
 8.338E-03 2.236E+00 8.819E+03 4.284E+05 7.016E-03 9.228E+00 2.818E+03 1.846E+06
 4.303E+02-8.931E+00 4.919E-02 6.325E-05-4.530E+02 1.420E+03 2.528E+02-2.366E+01
-3.674E+00-1.622E+01 2.732E+04-1.752E+04 4.158E-01-1.351E+02 2.716E+04 3.723E+02
 1.125E-02 2.747E+00 1.174E+04 5.695E+05 9.245E-03 1.188E+01 4.104E+03 2.352E+06
 8.816E+02-2.314E+01 1.090E-01 1.725E-03-3.946E+02 1.498E+03 4.883E+02-5.508E+01
-4.056E+00-1.089E+01 3.544E+04-2.595E+04 4.945E-01-1.682E+02 3.627E+04-1.605E+04
 2.716E-02-5.428E+00 1.775E+04 6.765E+05 1.261E-02 1.557E+01 7.662E+03 2.288E+06
```

===== ELT/ATOM =====

```
 9.405E+02-9.183E+00-7.379E-01 1.434E-02-2.678E+02 1.287E+03 9.558E+02-1.269E+02
-6.098E+00 7.712E+01 4.196E+04-3.437E+04 5.710E-01-2.001E+02 4.505E+04-3.853E+04
 3.613E-02-8.426E+00 2.277E+04 8.080E+05 1.582E-02 1.847E+01 1.110E+04 2.456E+06
 1.850E+03-7.648E+01 4.856E-01 1.655E-02-3.580E+02 1.751E+03 1.186E+03-1.847E+02
-1.021E+01 2.346E+02 5.164E+04-4.707E+04 7.467E-01-2.534E+02 5.800E+04-6.691E+04
 5.442E-02-1.719E+01 3.154E+04 9.102E+05 2.069E-02 2.375E+01 1.589E+04 2.713E+06
 1.558E+03-1.641E+01-1.325E+00 3.215E-02-2.174E+02 1.397E+03 1.833E+03-3.166E+02
-1.192E+01 3.253E+02 5.927E+04-5.980E+04 8.789E-01-2.920E+02 6.860E+04-1.018E+05
 6.556E-02-2.175E+01 3.895E+04 9.737E+05 2.483E-02 2.697E+01 2.078E+04 2.972E+06
 2.556E+03-1.624E+02 3.134E+00 2.626E-02-1.949E+02 1.368E+03 2.385E+03-4.739E+02
-1.446E+01 4.359E+02 6.578E+04-7.284E+04 8.786E-01-3.001E+02 7.682E+04-1.219E+05
 8.835E-02-3.637E+01 4.777E+04 9.352E+05 2.874E-02 3.165E+01 2.446E+04 3.037E+06
 5.292E+00 3.717E-01-4.740E-03 1.436E-05-1.733E+02 5.359E+02-3.738E+01 7.107E-01
-1.595E+02 1.345E+03 3.549E+03-7.756E+02-2.113E+01 7.161E+02 8.183E+04-9.859E+04
 1.048E+00-3.644E+02 9.844E+04-1.760E+05 1.040E-01-3.553E+01 6.096E+04 1.226E+06
 3.773E-02 4.007E+01 3.507E+04 3.555E+06 1.442E+02-3.204E+00 2.579E-02-8.084E-05
 1.440E+01 6.073E+02-4.745E+01 9.859E-01 7.171E+01 6.790E+01 6.646E+03-1.903E+03
-3.015E+01 1.141E+03 9.459E+04-1.195E+05 1.408E+00-4.553E+02 1.206E+05-2.531E+05
 1.154E-01-3.465E+01 7.523E+04 1.368E+06 4.664E-02 5.252E+01 3.960E+04 5.980E+06
 1.088E+03 5.062E+00-2.564E-01 1.298E-03 1.897E+02 4.984E+02-3.352E+01 6.726E-01
-1.507E+02 1.501E+03 5.139E+03-1.442E+03 4.615E-01-2.614E+02 1.217E+05-2.057E+05
 1.509E-01-6.913E+01 9.253E+04 1.096E+06 5.204E-02 6.443E+01 3.314E+04 9.765E+06
 1.581E+03 2.413E+01-7.791E-01 4.080E-03-4.626E+01 8.678E+02 7.308E+03-2.383E+03
 2.074E-01-2.292E+02 1.371E+05-2.489E+05 1.972E-01-9.550E+01 1.106E+05 1.067E+06
 6.044E-02 7.449E+01 4.320E+04 9.379E+06 1.887E+03 1.983E+01-7.427E-01 3.731E-03
-7.196E+01 1.132E+03 8.004E+03-2.803E+03 3.940E-01-2.733E+02 1.564E+05-3.184E+05
 2.007E-01-8.911E+01 1.254E+05 1.137E+06 6.962E-02 7.762E+01 5.692E+04 1.006E+07
 1.755E+03 9.204E+01-1.973E+00 9.457E-03 3.932E+01 1.537E+02 1.228E+04-4.984E+03
 2.740E-01-2.665E+02 1.818E+05-3.997E+05 2.727E-01-1.288E+02 1.543E+05 9.755E+05
 8.278E-02 9.017E+01 7.781E+04 8.227E+06 5.885E+03-1.396E+02 1.353E+00-4.829E-03
-5.021E+02 1.265E+03-8.805E+01 1.272E+00 5.200E+01 1.957E+01 1.447E+04-6.930E+03
 2.930E-01-2.703E+02 2.034E+05-4.886E+05 2.940E-01-1.342E+02 1.761E+05 8.747E+05
 9.427E-02 1.057E+02 8.606E+04 1.091E+07 5.852E+03-1.336E+02 1.336E+00-4.843E-03
-1.412E+02 1.086E+03-6.278E+01 2.043E-01-8.128E+00 6.550E+02 1.511E+04-7.079E+03
 2.151E-01-2.680E+02 2.341E+05-6.063E+05 3.511E-01-1.633E+02 2.090E+05 7.131E+05
 1.107E-01 1.252E+02 1.008E+05 1.273E+07 5.153E+03-3.891E+01-4.490E-01 4.045E-03
-1.957E+02 1.422E+03-1.321E+02 3.914E+00 2.092E+01 3.311E+02 1.820E+04-9.303E+03
 4.947E-02-2.478E+02 2.586E+05-7.185E+05 3.588E-01-1.492E+02 2.296E+05 8.856E+05
 1.244E-01 1.427E+02 1.134E+05 1.528E+07 6.887E+03-1.547E+02 1.332E+00-4.244E-03
 9.185E+01 1.073E+03-4.045E+01-1.313E+00 0.        6.600E+03 0.        0.
 0.        9.810E+03 0.        0.       -1.175E+02 2.171E+03 1.514E+04-6.042E+03
 1.685E-02-2.522E+02 3.005E+05-9.000E+05 4.681E-01-2.085E+02 2.782E+05 4.399E+05
 1.470E-01 1.687E+02 1.419E+05 1.446E+07 3.293E+03-2.208E+01-3.511E-01 2.914E-03
-3.217E+02 1.834E+03-2.631E+02 1.133E+01-5.089E+02 4.139E+03 7.468E+03-6.981E+02
-9.173E+01 1.862E+03 1.777E+04-8.202E+03 7.440E-03-2.371E+02 3.202E+05-1.028E+06
 4.729E-01-2.055E+02 3.000E+05 2.182E+05 1.589E-01 1.848E+02 1.452E+05 1.929E+07
 7.941E+03-2.347E+02 2.501E+00-8.979E-03-1.534E+02 1.829E+03-1.674E+02 4.439E+00
-2.807E+04 1.091E+05-1.511E+05 7.897E+04-3.057E+03 1.385E+04 3.780E+02-2.163E+02
-8.018E+01 1.794E+03 2.095E+04-1.070E+04-1.158E+00-3.871E+01 3.499E+05-1.154E+06
 4.820E-01-1.832E+02 3.296E+05 6.802E+05 1.796E-01 2.204E+02 1.529E+05 2.596E+07
 0.        0.        0.        3.400E-04 1.091E+04-5.719E+02 1.060E+01-6.716E-02
-6.776E+02 2.582E+03-2.726E+02 8.301E+00-1.457E+04 6.363E+04-1.011E+05 6.416E+04
 3.470E+03-8.674E+03 2.548E+04-8.411E+03-9.431E+01 2.078E+03 2.172E+04-1.101E+04
-1.253E+00-2.897E+01 3.783E+05-1.357E+06 5.591E-01-2.217E+02 3.625E+05 3.392E+05
 1.949E-01 2.371E+02 1.767E+05 2.400E+07 0.        0.        0.        1.800E-03
 1.942E+04-1.740E+03 5.357E+01-5.554E-01-9.082E+02 3.097E+03-3.039E+02 7.152E+00
-7.402E+03 3.636E+04-6.670E+04 5.346E+04-3.400E+02 3.232E+03 1.163E+04-1.089E+03
```

===== ELT/ATOM =====

```
-5.940E+01 1.646E+03 2.621E+04-1.507E+04-2.025E+00 1.283E+02 4.069E+05-1.510E+06
 6.478E-01-2.603E+02 4.045E+05-5.153E+04 2.155E-01 2.609E+02 2.074E+05 2.220E+07
 1.642E+02-2.624E+01 1.048E+00-5.116E-03 1.858E+04-2.363E+03 1.082E+02-1.737E+00
-8.391E+02 3.143E+03-4.102E+02 1.448E+01-4.525E+03 2.468E+04-5.133E+04 5.102E+04
-3.858E+02 1.013E+04 3.603E+03-7.657E+02-5.339E+01 1.649E+03 2.950E+04-1.800E+04
-1.655E+00 8.588E+01 4.504E+05-1.795E+06 7.880E-01-3.490E+02 4.641E+05-1.258E+06
 2.393E-01 2.906E+02 2.360E+05 2.145E+07 3.589E+02-4.858E+01 1.838E+00-8.229E-03
 4.647E+04-7.676E+03 4.318E+02-8.207E+00-9.469E+02 3.639E+03-4.086E+02 1.010E+01
-6.637E+02 4.194E+03-1.104E+04 2.806E+04-4.265E+02 9.660E+03 5.343E+03-1.174E+03
-3.968E+01 1.459E+03 3.327E+04-2.228E+04-3.461E+00 4.401E+02 4.690E+05-1.943E+06
 8.564E-01-3.723E+02 5.020E+05-1.430E+06 2.590E-01 3.237E+02 2.541E+05 2.321E+07
 7.741E+02-7.717E+01 2.608E+00-1.018E-02 3.275E+02 2.964E+03-4.214E+02 1.494E+01
 4.049E+03-1.310E+04 1.813E+04-6.454E+03 1.461E+02-1.050E+03 3.457E+03 2.000E+04
-4.318E+02 1.007E+04 6.548E+03-1.535E+03-4.033E+01 1.607E+03 3.681E+04-2.478E+04
-3.955E+00 5.482E+02 5.202E+05-2.262E+06 9.718E-01-4.209E+02 5.662E+05-2.099E+06
 2.907E-01 3.653E+02 2.908E+05 2.417E+07 5.868E+01 5.694E+01-3.164E+00 5.185E-02
-1.048E+03 4.558E+03-7.909E+02 3.526E+01-1.462E+03 6.037E+03-2.364E+03 5.585E+02
 3.611E+02-2.974E+03 1.308E+04 1.268E+04-4.163E+02 1.017E+04 7.535E+03-1.891E+03
-9.924E-01 8.455E+02 4.446E+04-3.490E+04-3.076E+00 4.257E+02 5.613E+05-2.587E+06
 1.050E+00-4.435E+02 6.114E+05-3.028E+06 3.136E-01 3.952E+02 3.252E+05 2.248E+07
 8.623E-01 7.942E-02-1.525E-03 6.932E-06 1.558E+02 9.161E+00-1.414E+00 7.049E-02
-1.446E+03 5.719E+03-1.213E+03 6.655E+01 3.524E+02-3.194E+03 1.657E+04 1.224E+04
-4.025E+02 1.067E+04 8.396E+03-2.275E+03 5.234E+00 7.494E+02 5.033E+04-4.276E+04
-2.207E+00 2.653E+02 6.269E+05-3.163E+06 1.101E+00-4.280E+02 6.664E+05-3.481E+06
 3.467E-01 4.515E+02 3.525E+05 2.783E+07 4.666E+01-1.282E+00 1.181E-02-3.649E-05
 7.687E+02-6.434E+01 1.634E+00 3.457E-02-2.899E+03 8.153E+03-1.741E+03 9.548E+01
 1.053E+03-2.725E+03 8.076E+03-2.820E+03 2.746E+02-2.830E+03 2.149E+04 7.655E+03
-4.181E+02 1.025E+04 1.159E+04-3.199E+03-4.367E+01 1.966E+03 4.835E+04-3.788E+04
-5.177E-01-2.546E+01 6.941E+05-3.733E+06 1.296E+00-5.192E+02 7.384E+05-5.043E+06
 3.800E-01 4.950E+02 4.111E+05 1.955E+07 1.646E+03-1.524E+01-3.033E-02 4.574E-04
 4.222E+01 7.685E+01-4.225E+00 8.261E-02-2.128E+03 8.026E+03-2.124E+03 1.548E+02
 3.585E+02-4.223E+02 6.620E+03-2.706E+03-8.813E+01 1.184E+03 2.198E+04-1.440E+03
-4.141E+02 1.043E+04 1.385E+04-4.020E+03-1.282E+01 1.204E+03 6.028E+04-5.752E+04
 4.265E-01-1.646E+02 7.684E+05-4.421E+06 1.334E+00-4.917E+02 8.066E+05-5.879E+06
 4.194E-01 5.651E+02 4.353E+05 2.981E+07 1.910E+02 3.638E+01-4.958E-01 1.598E-03
-8.578E+03 1.742E+04-5.382E+03 4.570E+02-3.681E+02 3.235E+03 1.544E+03-2.121E+02
-2.164E+02 3.512E+03 2.053E+04-3.382E+03-3.996E+02 1.064E+04 1.548E+04-4.776E+03
-3.011E+01 1.794E+03 6.153E+04-5.699E+04-5.900E-01 1.945E+02 8.031E+05-4.579E+06
 1.474E+00-5.530E+02 8.907E+05-7.927E+06 4.562E-01 6.248E+02 4.760E+05 3.216E+07
 5.478E+02-1.855E+01 1.190E+00-1.007E-02-7.034E+03 1.685E+04-5.527E+03 5.032E+02
 7.252E+02-1.862E+03 9.507E+03-3.524E+03-2.981E+02 6.188E+03 1.735E+04-4.420E+03
-3.923E+02 1.102E+04 1.751E+04-5.733E+03-1.658E+01 1.435E+03 7.050E+04-7.037E+04
-2.292E+00 5.516E+02 8.555E+05-5.018E+06 1.599E+00-5.764E+02 9.643E+05-8.517E+06
 4.986E-01 7.224E+02 4.910E+05 4.104E+07 5.429E+02 2.506E+00 5.446E-01-5.466E-03
-6.303E+03 1.614E+04-5.496E+03 5.143E+02-1.114E+02 1.895E+03 5.405E+03-2.261E+03
-3.027E+02 7.576E+03 1.570E+04-4.795E+03-3.784E+02 1.099E+04 1.983E+04-6.801E+03
-1.840E+01 1.552E+03 7.635E+04-8.181E+04-2.560E+00 7.317E+02 9.005E+05-5.349E+06
 1.540E+00-4.834E+02 1.021E+06-9.300E+06 5.359E-01 7.662E+02 5.243E+05 5.200E+07
 2.495E+02 6.721E+01-5.844E-01-2.482E-04-6.741E+03 1.690E+04-5.576E+03 5.102E+02
-3.012E+02 3.094E+03 4.154E+03-1.733E+03-2.871E+02 7.198E+03 1.867E+04-5.767E+03
-1.101E+02 1.635E+04 3.183E+03-1.491E+03-2.936E+01 1.924E+03 7.955E+04-8.669E+04
 5.721E-01 7.164E+01 1.004E+06-6.678E+06 1.769E+00-5.691E+02 1.097E+06-1.069E+07
 5.748E-01 8.119E+02 6.113E+05 3.605E+07 7.391E+02 3.482E+01 4.295E-01-6.811E-03
-8.171E+03 2.199E+04-8.951E+03 1.037E+03 3.087E+02-1.095E+02 1.012E+04-4.421E+03
-2.831E+02 7.880E+03 1.913E+04-6.518E+03-3.547E+02 1.093E+04 2.575E+04-9.586E+03
-1.311E+01 1.428E+03 9.237E+04-1.148E+05 1.007E-02 1.277E+02 1.086E+06-7.755E+06
 1.633E+00-4.030E+02 1.144E+06-9.817E+06 6.218E-01 9.117E+02 5.873E+05 6.614E+07
```

===== ELT/ATOM =====

```
 8.935E+02 2.878E+00 1.708E+00-9.542E-03-8.948E+03 2.146E+04-7.129E+03 4.790E+02
-5.992E+02 4.521E+03 3.942E+03-1.812E+03-2.665E+02 7.197E+03 2.390E+04-7.946E+03
-3.285E+02 9.342E+03 3.454E+04-1.203E+04-2.532E+01 2.009E+03 9.447E+04-1.115E+05
-3.043E+00 9.079E+02 1.126E+06-8.086E+06 1.870E+00-4.570E+02 1.251E+06-1.191E+07
 6.726E-01 9.644E+02 7.281E+05 4.409E+07 2.673E+02 8.162E+01-1.115E+00 3.804E-03
-1.793E+04 4.014E+04-1.899E+04 2.659E+03-8.864E+02 6.126E+03 2.706E+03-1.656E+03
-2.530E+02 7.129E+03 2.620E+04-9.068E+03-3.225E+02 1.001E+04 3.550E+04-1.354E+04
 1.642E+01 4.603E+02 1.181E+05-1.737E+05-2.888E+00 8.714E+02 1.203E+06-9.306E+06
 1.857E+00-3.662E+02 1.308E+06-1.219E+07 7.151E-01 1.066E+03 7.318E+05 6.061E+07
 1.221E+03 7.507E+01-1.363E+00 5.328E-03-4.940E+03 1.907E+04-7.444E+03 5.703E+02
 1.918E+02 5.874E+02 1.228E+04-6.006E+03-2.547E+02 7.851E+03 2.733E+04-1.038E+04
-3.170E+02 1.045E+04 3.850E+04-1.560E+04-1.817E+00 1.265E+03 1.192E+05-1.684E+05
-4.061E+00 1.339E+03 1.260E+06-9.752E+06 2.065E+00-4.444E+02 1.433E+06-1.533E+07
 7.734E-01 1.157E+03 8.006E+05 6.214E+07 5.246E+02 1.111E+02-1.429E+00 3.266E-03
-2.053E+03 1.198E+04-2.209E+03-5.686E+02-2.454E+02 8.162E+03 2.822E+04-1.156E+04
-2.677E+02 8.244E+03 4.935E+04-1.791E+04-2.306E+01 2.129E+03 1.161E+05-1.547E+05
-8.206E+00 2.269E+03 1.268E+06-9.746E+06 2.254E+00-4.904E+02 1.506E+06-1.642E+07
 8.123E-01 1.229E+03 8.686E+05 5.293E+07 0.        0.        1.850E-02 0.
 1.917E+03 1.793E+00-4.297E-02-5.864E-03-5.531E+02 5.425E+03 5.739E+03-2.983E+03
-2.324E+02 7.725E+03 3.333E+04-1.361E+04-2.675E+02 8.828E+03 5.249E+04-2.065E+04
-2.149E+01 2.143E+03 1.256E+05-1.775E+05-2.097E+00 1.045E+03 1.428E+06-1.216E+07
 2.233E+00-3.526E+02 1.578E+06-1.747E+07 8.685E-01 1.331E+03 9.171E+05 6.234E+07
 0.        0.        7.300E-02 0.        1.755E+03-3.700E+01 6.164E+00-1.436E-01
 3.184E+04-5.161E+04 3.430E+04-6.716E+03-8.411E+02 7.174E+03 3.429E+03-1.945E+03
-2.078E+02 6.795E+03 3.953E+04-1.546E+04-2.735E+02 1.006E+04 5.154E+04-2.316E+04
 1.409E+01 5.245E+02 1.554E+05-2.738E+05-6.053E+00 2.152E+03 1.430E+06-1.218E+07
 2.434E+00-4.441E+02 1.694E+06-2.116E+07 9.156E-01 1.438E+03 9.506E+05 6.990E+07
 6.617E+02-3.674E+01 7.073E-01-2.415E-03 2.657E+03-4.585E+02 4.392E+01-9.883E-01
 1.852E+04-3.091E+04 2.529E+04-5.381E+03-7.504E+02 7.011E+03 4.377E+03-1.656E+03
-2.154E+02 7.884E+03 3.884E+04-1.742E+04-2.413E+02 8.640E+03 6.267E+04-2.629E+04
-2.300E+01 2.351E+03 1.410E+05-2.130E+05-9.816E+00 3.157E+03 1.443E+06-1.193E+07
 2.470E+00-3.671E+02 1.779E+06-2.260E+07 9.713E-01 1.520E+03 1.016E+06 7.466E+07
 0.        0.        2.300E-01 0.        1.891E+03-3.992E+02 5.548E+01-1.669E+00
 4.769E+04-9.744E+04 7.551E+04-1.741E+04-4.397E+01 2.059E+03 1.571E+04-9.021E+03
-1.946E+02 7.129E+03 4.399E+04-1.939E+04-2.372E+02 9.166E+03 6.308E+04-2.911E+04
 1.086E+00 1.149E+03 1.640E+05-2.950E+05-3.401E+00 1.832E+03 1.571E+06-1.426E+07
 2.389E+00-1.733E+02 1.798E+06-2.273E+07 1.007E+00 1.586E+03 1.076E+06 7.154E+07
 4.617E+02-1.755E+01 2.146E-01 5.709E-03 2.438E+03-6.588E+02 1.076E+02-3.949E+00
-1.216E+03 2.425E+04-2.237E+04 8.492E+03-6.432E+02 7.074E+03 5.262E+03-2.211E+03
-1.833E+02 6.763E+03 5.201E+04-2.240E+04-2.150E+02 8.275E+03 7.525E+04-3.297E+04
-2.566E+01 2.615E+03 1.595E+05-2.557E+05-3.670E+00 1.872E+03 1.719E+06-1.646E+07
 2.225E+00 1.829E+02 1.881E+06-2.157E+07 1.098E+00 1.755E+03 1.154E+06 8.971E+07
 2.648E+02-1.704E+01 9.316E-01-7.699E-03 3.266E+03-1.120E+03 1.619E+02-6.243E+00
 7.789E+04-1.758E+05 1.423E+05-3.620E+04-2.422E+04 6.454E+04-3.914E+04 8.599E+03
-6.606E+02 7.269E+03 6.119E+03-3.383E+03-1.565E+02 5.749E+03 5.985E+04-2.396E+04
-2.236E+02 9.444E+03 7.463E+04-3.742E+04-2.010E+01 2.400E+03 1.720E+05-2.977E+05
-1.622E+00 1.830E+03 1.780E+06-1.721E+07 2.082E+00 3.625E+02 1.962E+06-2.448E+07
 1.148E+00 1.892E+03 1.133E+06 1.244E+08 5.253E+02-4.708E+01 1.701E+00-3.056E-03
-1.999E+03 3.398E+03-8.813E+02 6.592E+01-9.499E+03 4.412E+04-3.666E+04 1.130E+04
-3.460E+02 8.781E+03 1.848E+03-3.182E+02-2.807E+02 4.779E+03 1.240E+04-6.939E+03
-8.402E+01 2.872E+03 7.699E+04-1.853E+04-2.254E+02 1.015E+04 7.940E+04-4.284E+04
-1.987E+00 1.400E+03 2.040E+05-4.166E+05-4.238E+00 2.444E+03 1.873E+06-1.899E+07
 2.278E+00 4.044E+02 2.085E+06-2.635E+07 1.226E+00 2.030E+03 1.249E+06 1.169E+08
 6.159E+02-6.640E+01 2.701E+00-1.599E-02 1.191E+03 6.631E+02-9.405E+01 6.376E+00
-1.422E+04 4.480E+04-3.020E+04 8.387E+03-2.930E+02 8.866E+03 1.354E+03-2.310E+02
-2.751E+02 9.345E+03 1.542E+03-2.889E+02-6.491E+02 7.592E+03 8.802E+03-6.067E+03
-1.084E+02 4.043E+03 7.718E+04-2.532E+04-1.836E+02 7.952E+03 9.545E+04-4.532E+04
```

===== ELT/ATOM =====

```
 8.708E+00 7.581E+02 2.258E+05-5.011E+05-5.846E+00 3.162E+03 1.902E+06-1.948E+07
 2.590E+00 2.297E+02 2.236E+06-3.193E+07 1.281E+00 2.144E+03 1.303E+06 1.193E+08
 5.406E+02-5.009E+01 2.070E+00-1.381E-02 6.683E+02 9.673E+02-2.153E+02 1.381E+01
-2.289E+04 6.758E+04-4.780E+04 1.250E+04-4.024E+02 8.837E+03 2.398E+03-4.154E+02
-3.424E+02 9.463E+03 2.343E+03-4.528E+02-8.911E+02 1.007E+04 3.513E+03-2.142E+03
-9.085E+01 3.459E+03 8.639E+04-2.558E+04-1.485E+02 6.357E+03 1.121E+05-4.626E+04
-6.527E+00 1.901E+03 2.209E+05-4.602E+05-7.588E+00 3.685E+03 1.991E+06-2.054E+07
 2.727E+00 3.137E+02 2.356E+06-3.362E+07 1.365E+00 2.298E+03 1.393E+06 1.220E+08
 1.525E+03-1.451E+02 5.492E+00-4.193E-02 1.513E+03 5.125E+02-2.459E+02 2.320E+01
 5.551E+04-1.769E+05 2.024E+05-7.125E+04 4.656E+03 6.580E+03-3.256E+03 8.079E+02
-2.828E+02 1.019E+04 2.005E+03-4.180E+02-3.900E+02 5.970E+03 1.533E+04-9.604E+03
-7.263E+01 2.822E+03 9.606E+04-2.458E+04-1.695E+02 7.899E+03 1.139E+05-5.644E+04
 1.939E+01 3.799E+02 2.661E+05-6.583E+05-1.029E+01 4.409E+03 2.101E+06-2.280E+07
 2.723E+00 4.916E+02 2.494E+06-3.595E+07 1.464E+00 2.444E+03 1.486E+06 1.376E+08
 1.779E+03-1.728E+02 7.854E+00-6.296E-02 1.020E+03 1.017E+03-3.883E+02 3.566E+01
-6.489E+03 2.485E+04-1.060E+04 2.589E+03-5.349E+02 8.049E+03 5.499E+03-9.313E+02
-3.829E+02 1.003E+04 3.437E+03-7.202E+02-1.340E+02 4.371E+03 1.917E+04-1.118E+04
-4.523E+01 1.773E+03 1.075E+05-1.884E+04-1.401E+02 6.504E+03 1.315E+05-5.758E+04
 8.492E+00 8.664E+02 2.819E+05-7.256E+05-9.126E+00 4.223E+03 2.290E+06-2.709E+07
 3.207E+00 3.424E+02 2.697E+06-4.123E+07 1.557E+00 2.710E+03 1.566E+06 1.383E+08
-6.505E+02 2.083E+02-8.262E+00 1.263E-01 5.619E+02 1.452E+03-5.060E+02 4.545E+01
-5.304E+02 8.032E+03 3.889E+03-5.907E+02 2.239E+02 1.199E+04-1.181E+03 2.855E+02
-2.457E+02 5.622E+03 1.700E+04-9.398E+03-6.644E+01 2.796E+03 1.098E+05-2.924E+04
-1.114E+02 5.206E+03 1.457E+05-5.559E+04 8.287E+00 9.600E+02 2.935E+05-7.637E+05
-9.944E+00 4.886E+03 2.320E+06-2.702E+07 3.820E+00 8.492E+01 2.868E+06-4.750E+07
 1.633E+00 2.833E+03 1.759E+06 9.295E+07 1.690E+03 0.        0.        0.
 1.440E+03 6.486E+02-1.094E+02 5.512E+00-6.403E+00 7.436E+03 2.087E+01-3.609E+00
-5.546E+02 7.912E+03 4.958E+03-7.606E+02-5.157E+02 8.937E+03 6.066E+03-1.148E+03
-4.607E+02 9.760E+03 6.028E+03-1.298E+03-1.912E+02 5.138E+03 2.120E+04-1.345E+04
-7.165E+01 3.169E+03 1.166E+05-3.499E+04-2.589E+02 8.215E+03-1.102E+05 1.824E+06
-4.251E+01 4.740E+03 2.377E+05-4.279E+05 9.003E-02 2.739E+03 2.641E+06-3.336E+07
 2.857E+00 1.090E+03 2.888E+06-4.432E+07 1.742E+00 3.069E+03 1.744E+06 1.719E+08
 1.680E+03 0.        0.        0.       -3.980E+02 2.937E+03-5.059E+02 2.698E+01
 2.417E+03 6.924E+03-2.629E+03 5.790E+02-5.499E+02 7.881E+03 5.664E+03-8.952E+02
-4.675E+02 9.685E+03 5.381E+03-1.113E+03-4.771E+02 9.283E+03 7.693E+03-1.657E+03
-1.944E+02 5.237E+03 2.215E+04-1.434E+04-3.580E+01 1.576E+03 1.290E+05-2.201E+04
-6.400E+01 3.084E+03 1.751E+05-4.595E+04 7.096E+00 1.071E+03 3.267E+05-9.193E+05
-1.082E+01 5.919E+03 2.475E+06-2.997E+07 3.844E+00 4.077E+02 3.137E+06-5.576E+07
 1.799E+00 3.181E+03 1.898E+06 1.263E+08 1.660E+03 0.        0.        0.
-2.970E+02 2.959E+03-5.177E+02 2.881E+01 8.618E+02 8.414E+03-1.992E+03 4.240E+02
-5.507E+02 7.936E+03 6.518E+03-1.058E+03-4.589E+02 1.019E+04 5.669E+03-1.234E+03
-4.913E+02 8.740E+03 1.016E+04-2.142E+03-2.288E+02 5.934E+03 2.154E+04-1.326E+04
-2.466E+01 1.115E+03 1.400E+05-1.768E+04-6.266E+01 3.143E+03 1.878E+05-5.089E+04
-4.306E+00 2.026E+03 3.280E+05-9.012E+05-7.610E+00 5.488E+03 2.642E+06-3.287E+07
 3.783E+00 7.655E+02 3.245E+06-5.731E+07 1.904E+00 3.400E+03 2.020E+06 1.315E+08
 1.630E+03 0.        0.        0.        4.170E+01 2.626E+03-3.982E+02 2.107E+01
 1.902E+03 7.831E+03-2.992E+03 7.117E+02-5.394E+02 7.849E+03 7.295E+03-1.216E+03
-4.486E+02 1.029E+04 6.091E+03-1.380E+03-4.679E+02 9.424E+03 9.571E+03-2.201E+03
-3.017E+02 6.938E+03 1.944E+04-1.104E+04-5.768E+01 2.841E+03 1.366E+05-4.131E+04
-3.358E+01 1.701E+03 2.029E+05-3.312E+04-8.668E+00 2.379E+03 3.331E+05-9.285E+05
 3.209E+00 1.579E+03 3.144E+06-4.804E+07 1.966E+00 3.553E+03 2.103E+06 1.345E+08
 1.600E+03 0.        0.        0.       -1.794E+02 2.952E+03-4.363E+02 2.180E+01
 4.709E+02 8.920E+03-1.596E+03 3.598E+02-5.345E+02 7.632E+03 8.660E+03-1.449E+03
-4.988E+02 9.401E+03 9.226E+03-2.000E+03-4.700E+02 9.391E+03 1.117E+04-2.613E+03
-1.902E+02 5.724E+03 2.615E+04-1.782E+04-1.568E+02 1.116E+04 8.757E+04-7.647E+04
-4.533E+01 2.421E+03 2.126E+05-4.859E+04-4.721E+00 2.127E+03 3.644E+05-1.109E+06
 3.903E+00 1.250E+03 3.405E+06-5.637E+07 2.077E+00 3.763E+03 2.261E+06 1.155E+08
```

===== ELT/ATOM =====

```
 1.570E+03 0.        0.        0.       -2.930E+02 3.204E+03-4.744E+02 2.422E+01
 5.198E+02 9.201E+03-1.861E+03 4.412E+02-5.242E+02 7.751E+03 9.341E+03-1.624E+03
-4.635E+02 1.050E+04 8.066E+03-1.930E+03-4.608E+02 9.448E+03 1.216E+04-2.942E+03
-1.077E+02 4.675E+03 3.253E+04-2.519E+04-6.326E+01 3.407E+03 1.499E+05-5.526E+04
-7.262E+01 4.176E+03 2.130E+05-8.026E+04-1.800E+01 3.374E+03 3.526E+05-9.982E+05
 4.112E+00 1.344E+03 3.520E+06-5.867E+07 2.166E+00 3.928E+03 2.396E+06 1.007E+08
 1.530E+03 0.        0.        0.       -2.278E+02 3.277E+03-5.051E+02 2.412E+01
 1.858E+03 8.664E+03-3.790E+03 1.030E+03-5.221E+02 8.169E+03 9.852E+03-1.807E+03
-4.878E+02 9.589E+03 1.119E+04-2.580E+03-4.549E+02 9.349E+03 1.378E+04-3.393E+03
-1.379E+02 5.032E+03 3.428E+04-2.787E+04-4.939E+01 2.705E+03 1.628E+05-5.056E+04
-4.426E+01 2.555E+03 2.349E+05-5.948E+04-2.988E+00 2.094E+03 4.072E+05-1.335E+06
 5.131E+00 5.023E+02 4.000E+06-7.974E+07 2.274E+00 4.258E+03 2.387E+06 1.450E+08
 1.520E+03 0.        0.        0.       -4.193E+02 3.700E+03-5.990E+02 3.331E+01
 3.499E+03 6.300E+03-3.724E+03 1.119E+03-5.148E+02 8.464E+03 1.038E+04-1.998E+03
-4.781E+02 8.546E+03 1.463E+04-3.195E+03-4.470E+02 9.490E+03 1.489E+04-3.799E+03
-2.279E+02 6.501E+03 3.083E+04-2.363E+04-5.162E+01 2.947E+03 1.710E+05-5.828E+04
-5.360E+01 3.254E+03 2.449E+05-7.821E+04 6.456E+00 1.211E+03 4.583E+05-1.684E+06
 4.025E+00 2.031E+03 3.795E+06-6.651E+07 2.386E+00 4.497E+03 2.558E+06 1.403E+08
 1.500E+03 0.        0.        0.       -4.117E+02 3.892E+03-6.641E+02 4.384E+01
 3.309E+03 6.805E+03-4.124E+03 1.286E+03-5.074E+02 8.420E+03 1.161E+04-2.277E+03
-4.701E+02 8.553E+03 1.620E+04-3.609E+03-4.405E+02 9.321E+03 1.702E+04-4.382E+03
-2.641E+02 7.170E+03 3.109E+04-2.392E+04-5.189E+01 3.077E+03 1.808E+05-6.499E+04
-5.311E+01 3.348E+03 2.598E+05-8.657E+04-3.887E+00 2.132E+03 4.637E+05-1.723E+06
 4.124E+00 2.299E+03 3.956E+06-7.043E+07 2.512E+00 4.774E+03 2.690E+06 1.454E+08
 1.490E+03 0.        0.        0.       -5.724E+02 4.268E+03-7.115E+02 4.693E+01
-2.392E+02 2.574E+03 1.221E+04-1.237E+03-4.649E+02 1.046E+04 8.110E+03-1.884E+03
-4.593E+02 8.718E+03 1.696E+04-3.936E+03-4.300E+02 9.905E+03 1.674E+04-4.633E+03
-2.294E+02 6.908E+03 3.414E+04-2.810E+04-5.328E+01 3.291E+03 1.869E+05-7.332E+04
-7.311E+01 4.929E+03 2.602E+05-1.239E+05-2.434E+01 4.269E+03 4.243E+05-1.379E+06
 4.575E+00 2.312E+03 4.086E+06-7.562E+07 2.606E+00 4.980E+03 2.895E+06 9.923E+07
 1.490E+03 0.        0.        0.       -5.832E+02 4.426E+03-6.739E+02 3.958E+01
 7.177E+02 1.052E+04-3.382E+03 1.011E+03-4.936E+02 9.095E+03 1.255E+04-2.717E+03
-4.436E+02 8.391E+03 1.928E+04-4.456E+03-4.229E+02 1.051E+04 1.693E+04-4.987E+03
-1.794E+02 6.176E+03 4.132E+04-3.872E+04-5.213E+01 3.333E+03 1.974E+05-7.977E+04
-5.333E+01 3.627E+03 2.858E+05-1.071E+05 3.726E+00 1.486E+03 5.352E+05-2.222E+06
 5.266E+00 2.137E+03 4.329E+06-8.327E+07 2.736E+00 5.209E+03 3.153E+06 5.239E+07
 1.480E+03 0.        0.        0.       -5.793E+02 4.429E+03-4.123E+02-2.341E+01
 6.148E+02 1.077E+04-3.332E+03 1.032E+03-4.835E+02 9.549E+03 1.284E+04-2.957E+03
-4.488E+02 9.643E+03 1.803E+04-4.705E+03-4.146E+02 1.043E+04 1.883E+04-5.684E+03
-4.615E+02 1.113E+04 1.850E+04-2.974E+03-6.090E+01 4.112E+03 2.015E+05-9.934E+04
-7.386E+01 5.387E+03 2.859E+05-1.536E+05-1.873E+01 4.102E+03 4.698E+05-1.572E+06
 4.753E+00 2.673E+03 4.462E+06-8.960E+07 2.845E+00 5.554E+03 3.013E+06 1.535E+08
 1.470E+03 0.        0.        0.       -5.773E+02 4.832E+03-8.075E+02 4.906E+01
 2.954E+02 1.102E+04-2.147E+03 6.678E+02-4.771E+02 1.003E+04 1.338E+04-3.254E+03
-4.305E+02 8.881E+03 2.202E+04-5.544E+03-4.034E+02 9.726E+03 2.291E+04-6.762E+03
-8.593E+01 4.826E+03 5.410E+04-5.683E+04-5.521E+01 3.828E+03 2.145E+05-1.019E+05
-4.987E+01 3.641E+03 3.160E+05-1.250E+05-2.176E+00 2.172E+03 5.708E+05-2.482E+06
 5.194E+00 2.794E+03 4.645E+06-9.648E+07 2.973E+00 5.785E+03 3.320E+06 9.608E+07
 1.480E+03 0.        0.        0.       -6.028E+02 5.107E+03-8.969E+02 5.534E+01
 1.573E+02 1.107E+04-1.381E+03 4.379E+02-4.666E+02 1.057E+04 1.355E+04-3.503E+03
-4.159E+02 8.743E+03 2.421E+04-6.191E+03-3.984E+02 1.026E+04 2.349E+04-7.382E+03
-1.624E+02 6.121E+03 5.094E+04-5.128E+04-6.206E+01 4.525E+03 2.194E+05-1.223E+05
-1.211E+02 1.058E+04 2.701E+05-2.556E+05-2.105E+01 4.623E+03 5.109E+05-1.835E+06
 5.276E+00 2.999E+03 4.828E+06-1.030E+08 3.099E+00 6.112E+03 3.348E+06 1.310E+08
 1.570E+03 0.        0.        0.       -6.033E+02 5.404E+03-1.108E+03 9.354E+01
-1.992E+02 1.014E+04 2.835E+03-8.434E+02-4.438E+02 1.148E+04 1.253E+04-3.520E+03
-4.181E+02 9.456E+03 2.446E+04-6.751E+03-3.940E+02 1.126E+04 2.277E+04-7.800E+03
```

===== ELT/ATOM =====

```
-1.220E+02 5.627E+03 5.672E+04-5.943E+04-5.721E+01 4.284E+03 2.331E+05-1.270E+05
-1.192E+02 1.081E+04 2.828E+05-2.788E+05-8.548E+00 3.122E+03 5.972E+05-2.571E+06
 4.901E+00 3.621E+03 4.969E+06-1.085E+08 3.238E+00 6.600E+03 3.237E+06 2.204E+08
 1.720E+03 0.        0.        0.       -5.720E+02 5.552E+03-1.165E+03 9.523E+01
-2.871E+02 9.431E+03 5.398E+03-1.569E+03-4.312E+02 1.174E+04 1.298E+04-3.813E+03
-4.172E+02 1.036E+04 2.374E+04-7.166E+03-3.837E+02 1.147E+04 2.383E+04-8.523E+03
-1.463E+02 6.213E+03 5.661E+04-5.933E+04-5.946E+01 4.643E+03 2.401E+05-1.434E+05
-1.116E+02 1.035E+04 3.012E+05-2.961E+05-1.803E+01 4.548E+03 5.700E+05-2.246E+06
 5.735E+00 3.318E+03 5.232E+06-1.209E+08 3.351E+00 6.829E+03 3.509E+06 1.607E+08
 1.890E+03 0.        0.        0.       -6.457E+02 5.954E+03-1.292E+03 1.094E+02
-2.830E+02 9.680E+03 5.768E+03-1.746E+03-4.371E+02 1.150E+04 1.539E+04-4.560E+03
-4.125E+02 1.085E+04 2.464E+04-7.841E+03-3.774E+02 1.211E+04 2.431E+04-9.223E+03
-2.858E+01 3.946E+03 7.479E+04-9.186E+04-5.153E+01 4.097E+03 2.582E+05-1.419E+05
-1.366E+02 1.456E+04 2.707E+05-3.461E+05-1.995E+01 4.906E+03 6.064E+05-2.648E+06
 6.213E+00 3.458E+03 5.442E+06-1.286E+08 3.505E+00 7.126E+03 3.805E+06 1.114E+08
 2.060E+03 0.        0.        0.       -6.891E+02 6.259E+03-1.339E+03 1.009E+02
-3.204E+02 9.105E+03 8.268E+03-2.460E+03-4.333E+02 1.147E+04 1.710E+04-5.181E+03
-4.064E+02 1.162E+04 2.438E+04-8.327E+03-3.694E+02 1.237E+04 2.562E+04-1.013E+04
-1.003E+02 5.662E+03 6.740E+04-7.624E+04-6.719E+01 5.737E+03 2.565E+05-1.880E+05
-1.399E+02 1.605E+04 2.671E+05-3.742E+05 1.076E+01 9.617E+02 7.776E+05-4.207E+06
 6.088E+00 4.001E+03 5.552E+06-1.300E+08 3.646E+00 7.453E+03 3.931E+06 1.226E+08
 2.200E+03 0.        0.        0.       -7.040E+02 6.562E+03-1.438E+03 1.267E+02
-3.105E+02 9.446E+03 8.461E+03-2.642E+03-4.203E+02 1.212E+04 1.690E+04-5.448E+03
-3.968E+02 1.263E+04 2.339E+04-8.644E+03-3.646E+02 1.234E+04 2.840E+04-1.148E+04
-5.314E+00 3.877E+03 8.307E+04-1.051E+05-8.194E+01 7.574E+03 2.506E+05-2.318E+05
-1.134E+02 1.194E+04 3.355E+05-3.921E+05-3.392E+00 3.070E+03 7.247E+05-3.499E+06
 7.917E+00 2.918E+03 6.095E+06-1.596E+08 3.811E+00 7.846E+03 4.248E+06 4.571E+07
 2.210E+03 0.        0.        0.       -8.035E+02 7.031E+03-1.602E+03 1.385E+02
-3.336E+02 7.497E+03 1.461E+04-4.020E+03-3.961E+02 1.286E+04 1.582E+04-5.482E+03
-3.830E+02 1.343E+04 2.276E+04-8.999E+03-3.553E+02 1.248E+04 3.007E+04-1.263E+04
 1.179E+01 3.460E+03 9.044E+04-1.201E+05-7.533E+01 7.084E+03 2.675E+05-2.412E+05
-1.288E+02 1.524E+04 3.090E+05-4.397E+05 4.681E-01 2.405E+03 7.958E+05-4.232E+06
 6.983E+00 4.309E+03 5.980E+06-1.518E+08 3.946E+00 8.311E+03 4.256E+06 8.942E+07
 2.210E+03 0.        0.        0.       -9.684E+02 7.861E+03-2.292E+03 3.194E+02
-3.275E+02 7.729E+03 1.535E+04-4.426E+03-4.087E+02 1.156E+04 2.169E+04-7.188E+03
-3.790E+02 1.294E+04 2.638E+04-1.049E+04-3.455E+02 1.245E+04 3.232E+04-1.400E+04
 5.060E+01 2.722E+03 1.006E+05-1.441E+05-6.647E+01 6.324E+03 2.871E+05-2.439E+05
-1.405E+02 2.137E+04 2.340E+05-4.290E+05-1.053E+01 4.195E+03 7.543E+05-3.747E+06
 7.684E+00 4.048E+03 6.314E+06-1.714E+08 4.082E+00 8.543E+03 4.521E+06 5.178E+07
 2.200E+03 0.        0.        0.       -7.847E+02 7.343E+03-1.425E+03-3.703E-01
-3.238E+02 8.188E+03 1.567E+04-4.802E+03-4.016E+02 1.173E+04 2.316E+04-7.955E+03
-3.717E+02 1.215E+04 3.143E+04-1.232E+04-3.372E+02 1.292E+04 3.306E+04-1.511E+04
-1.004E+02 6.410E+03 7.864E+04-9.524E+04-7.125E+01 7.126E+03 2.931E+05-2.779E+05
-1.363E+02 1.992E+04 2.721E+05-4.887E+05-8.206E+00 4.058E+03 7.950E+05-4.115E+06
 7.595E+00 4.692E+03 6.429E+06-1.747E+08 4.240E+00 8.883E+03 4.754E+06 3.448E+07
 2.030E+03 0.        0.        0.       -1.030E+03 8.535E+03-2.431E+03 2.738E+02
-3.149E+02 7.642E+03 1.884E+04-5.632E+03-3.943E+02 1.129E+04 2.671E+04-9.164E+03
-3.641E+02 1.204E+04 3.463E+04-1.381E+04-3.300E+02 1.254E+04 3.747E+04-1.725E+04
 2.320E+01 3.479E+03 1.066E+05-1.595E+05-6.824E+01 7.005E+03 3.089E+05-2.951E+05
-1.337E+02 1.999E+04 2.903E+05-5.342E+05-1.411E+00 3.094E+03 8.901E+05-5.203E+06
 9.935E+00 3.289E+03 6.983E+06-2.020E+08 4.425E+00 9.396E+03 5.008E+06-5.280E+07
 1.830E+03 0.        0.        0.       -9.975E+02 8.677E+03-2.206E+03 1.699E+02
-3.167E+02 8.401E+03 1.878E+04-6.087E+03-3.899E+02 1.134E+04 2.963E+04-1.038E+04
-3.604E+02 1.242E+04 3.688E+04-1.531E+04-3.216E+02 1.192E+04 4.405E+04-2.004E+04
-4.664E+01 5.421E+03 9.815E+04-1.380E+05-7.471E+01 8.106E+03 3.143E+05-3.397E+05
-1.335E+02 2.199E+04 2.786E+05-5.569E+05 7.789E+00 1.767E+03 1.003E+06-6.443E+06
 8.525E+00 5.102E+03 6.990E+06-2.009E+08 4.633E+00 9.868E+03 5.142E+06 1.775E+07
```

===== ELT/ATOM =====

```
 1.610E+03 0.        0.        0.       -1.158E+03 9.785E+03-3.286E+03 4.977E+02
-2.939E+02 7.047E+03 2.494E+04-7.295E+03-3.889E+02 1.171E+04 3.195E+04-1.160E+04
-3.573E+02 1.264E+04 4.031E+04-1.720E+04-3.206E+02 1.263E+04 4.560E+04-2.199E+04
 1.227E+02 1.460E+03 1.347E+05-2.192E+05-6.880E+01 7.565E+03 3.411E+05-3.547E+05
-1.280E+02 1.941E+04 3.508E+05-6.556E+05-1.695E+00 3.621E+03 9.663E+05-5.746E+06
 9.913E+00 4.822E+03 7.370E+06-2.163E+08 4.871E+00 1.062E+04 5.352E+06-2.702E+07
 1.380E+03 0.        0.        0.       -9.559E+02 9.029E+03-2.571E+03 3.310E+02
-2.893E+02 7.615E+03 2.331E+04-7.562E+03-3.616E+02 1.088E+04 3.423E+04-1.248E+04
-2.652E+02 7.797E+03 5.879E+04-1.903E+04-2.980E+02 1.182E+04 4.838E+04-2.359E+04
-6.112E+01 5.945E+03 1.001E+05-1.407E+05-6.756E+01 7.739E+03 3.363E+05-3.748E+05
-1.220E+02 1.976E+04 3.335E+05-6.657E+05-5.899E-01 3.312E+03 9.833E+05-6.258E+06
 8.633E+00 5.888E+03 7.122E+06-2.120E+08 4.841E+00 1.057E+04 5.286E+06 1.022E+07
 1.250E+03 0.        0.        0.       -7.818E+02 8.542E+03-1.652E+03 4.668E+01
-2.906E+02 8.104E+03 2.416E+04-8.306E+03-3.615E+02 1.150E+04 3.541E+04-1.363E+04
-3.299E+02 1.230E+04 4.485E+04-2.020E+04-2.983E+02 1.294E+04 4.768E+04-2.521E+04
-2.362E+01 5.338E+03 1.099E+05-1.583E+05-6.236E+01 7.249E+03 3.612E+05-3.893E+05
-1.213E+02 2.052E+04 3.455E+05-7.195E+05-8.453E+00 4.660E+03 9.757E+05-6.010E+06
 1.104E+01 4.630E+03 7.706E+06-2.445E+08 5.063E+00 1.115E+04 5.666E+06-1.082E+08
 1.170E+03 0.        0.        0.       -9.110E+02 9.350E+03-2.300E+03 2.228E+02
-2.592E+02 6.680E+03 3.041E+04-9.387E+03-3.547E+02 1.164E+04 3.753E+04-1.492E+04
-3.193E+02 1.201E+04 4.919E+04-2.237E+04-2.920E+02 1.314E+04 5.024E+04-2.757E+04
-5.651E+01 6.065E+03 1.108E+05-1.601E+05-6.085E+01 7.287E+03 3.762E+05-4.161E+05
-1.192E+02 2.119E+04 3.500E+05-7.644E+05-1.024E+01 5.310E+03 9.756E+05-5.773E+06
 1.241E+01 3.727E+03 8.307E+06-2.833E+08 5.221E+00 1.168E+04 5.739E+06-9.046E+07
 1.110E+03 0.        0.        0.       -1.115E+03 1.053E+04-3.094E+03 2.300E+02
-2.814E+02 8.281E+03 2.798E+04-1.015E+04-3.474E+02 1.153E+04 4.113E+04-1.657E+04
-3.138E+02 1.208E+04 5.316E+04-2.471E+04-2.887E+02 1.389E+04 5.091E+04-2.962E+04
 7.657E+01 2.243E+03 1.528E+05-2.731E+05-6.339E+01 7.915E+03 3.884E+05-4.637E+05
-1.182E+02 2.194E+04 3.621E+05-8.244E+05 8.142E+00 2.287E+03 1.185E+06-8.493E+06
 1.003E+01 6.690E+03 7.961E+06-2.545E+08 5.479E+00 1.217E+04 5.897E+06-2.287E+07
 1.050E+03 0.        0.        0.       -9.331E+02 9.938E+03-2.564E+03 2.942E+02
-2.325E+02 6.130E+03 3.633E+04-1.104E+04-3.308E+02 1.094E+04 4.517E+04-1.812E+04
-3.073E+02 1.244E+04 5.401E+04-2.638E+04-2.797E+02 1.474E+04 4.897E+04-3.059E+04
 8.999E+00 4.311E+03 1.390E+05-2.331E+05-5.998E+01 7.669E+03 4.039E+05-4.843E+05
-1.152E+02 2.229E+04 3.694E+05-8.743E+05-1.965E+01 7.238E+03 9.916E+05-5.869E+06
 1.150E+01 6.085E+03 8.292E+06-2.758E+08 5.620E+00 1.245E+04 6.338E+06-1.549E+08
 1.000E+03 0.        0.        0.       -1.195E+03 1.151E+04-3.714E+03 5.109E+01
-2.349E+02 6.438E+03 3.849E+04-1.223E+04-3.240E+02 1.078E+04 5.022E+04-2.015E+04
-3.119E+02 1.369E+04 5.366E+04-2.824E+04-2.785E+02 1.523E+04 5.214E+04-3.377E+04
 1.482E+02 3.411E+02 1.839E+05-3.542E+05-6.436E+01 8.617E+03 4.180E+05-5.495E+05
-1.129E+02 2.138E+04 4.231E+05-9.875E+05 1.526E+01 1.356E+03 1.351E+06-1.082E+07
 1.254E+01 5.894E+03 8.877E+06-3.120E+08 5.918E+00 1.306E+04 6.683E+06-1.778E+08
 9.700E+02 0.        0.        0.       -1.023E+03 1.062E+04-1.978E+03-9.188E+02
-2.568E+02 7.987E+03 3.435E+04-1.308E+04-3.148E+02 1.094E+04 5.041E+04-2.121E+04
-2.917E+02 1.249E+04 5.991E+04-3.088E+04-2.653E+02 1.346E+04 6.280E+04-3.903E+04
 5.629E+00 4.645E+03 1.486E+05-2.622E+05-6.720E+01 9.489E+03 4.107E+05-6.000E+05
-1.033E+02 1.864E+04 4.878E+05-1.076E+06 6.281E-01 3.792E+03 1.281E+06-1.003E+07
 1.146E+01 7.568E+03 8.699E+06-3.106E+08 6.017E+00 1.327E+04 7.079E+06-2.456E+08
 9.500E+02 0.        0.        0.       -2.189E+03 1.746E+04-1.117E+04 1.713E+03
-2.579E+02 8.351E+03 3.599E+04-1.430E+04-3.232E+02 1.200E+04 5.096E+04-2.302E+04
-2.932E+02 1.303E+04 6.385E+04-3.410E+04-2.615E+02 1.325E+04 7.069E+04-4.406E+04
 6.006E+01 3.205E+03 1.724E+05-3.290E+05-6.355E+01 9.093E+03 4.439E+05-6.341E+05
-1.103E+02 2.501E+04 3.841E+05-1.044E+06 5.579E+00 3.489E+03 1.354E+06-1.078E+07
 1.272E+01 7.422E+03 9.191E+06-3.336E+08 6.330E+00 1.392E+04 7.484E+06-3.064E+08
 9.500E+02 0.        0.        0.       -1.043E+03 1.139E+04-2.517E+03-6.929E+02
-2.027E+02 5.876E+03 4.636E+04-1.482E+04-3.181E+02 1.245E+04 5.096E+04-2.427E+04
-2.888E+02 1.369E+04 6.306E+04-3.581E+04-2.404E+02 1.162E+04 8.310E+04-4.888E+04
```

```
     =====  ELT/ATOM  =====

 6.135E+01 3.329E+03 1.742E+05-3.276E+05-6.367E+01 9.444E+03 4.539E+05-6.824E+05
-1.065E+02 2.299E+04 4.461E+05-1.181E+06-1.248E+01 6.703E+03 1.250E+06-9.411E+06
 1.781E+01 3.158E+03 1.075E+07-4.693E+08 6.490E+00 1.422E+04 8.179E+06-5.340E+08
 9.600E+02 0.        0.        0.       -8.786E+02 1.063E+04-5.553E+02-1.750E+03
-2.044E+02 6.152E+03 4.844E+04-1.624E+04-3.188E+02 1.325E+04 5.127E+04-2.586E+04
-2.907E+02 1.505E+04 6.139E+04-3.761E+04-2.357E+02 1.168E+04 8.876E+04-5.342E+04
 1.398E+02 8.374E+02 2.097E+05-4.454E+05-6.201E+01 9.427E+03 4.778E+05-7.281E+05
-9.199E+01 1.716E+04 6.101E+05-1.350E+06-3.730E+00 5.425E+03 1.376E+06-1.109E+07
 1.966E+01 2.217E+03 1.154E+07-5.298E+08 6.763E+00 1.479E+04 8.647E+06-6.116E+08
 1.000E+03 0.        0.        0.       -8.139E+02 1.040E+04 4.547E+02-2.530E+03
-1.652E+02 4.824E+03 5.512E+04-1.585E+04-3.144E+02 1.378E+04 5.160E+04-2.733E+04
-2.793E+02 1.411E+04 6.926E+04-4.181E+04-2.092E+02 1.004E+04 1.017E+05-5.684E+04
 6.769E+01 3.243E+03 1.909E+05-3.844E+05-5.879E+01 9.126E+03 4.981E+05-7.597E+05
-1.010E+02 2.212E+04 5.193E+05-1.402E+06-1.147E+01 7.190E+03 1.327E+06-1.016E+07
 2.200E+01 1.083E+03 1.208E+07-5.684E+08 7.017E+00 1.493E+04 9.471E+06-8.527E+08
 1.100E+03 0.        0.        0.       -1.092E+03 1.257E+04-3.090E+03-6.861E+02
-2.062E+02 6.715E+03 5.159E+04-1.915E+04-3.108E+02 1.396E+04 5.452E+04-2.964E+04
-2.726E+02 1.392E+04 7.484E+04-4.567E+04-2.161E+02 1.101E+04 1.029E+05-6.243E+04
 1.590E+02-5.494E+01 2.383E+05-5.506E+05-6.035E+01 9.752E+03 5.095E+05-8.299E+05
-1.018E+02 2.538E+04 4.681E+05-1.412E+06-1.308E+01 7.916E+03 1.354E+06-1.049E+07
 2.374E+01 1.750E+02 1.285E+07-6.352E+08 7.219E+00 1.592E+04 9.491E+06-8.283E+08
 1.220E+03 0.        0.        0.       -9.355E+02 1.168E+04-4.663E+02-2.379E+03
-2.264E+02 8.092E+03 4.926E+04-2.117E+04-3.044E+02 1.361E+04 6.013E+04-3.273E+04
-2.641E+02 1.349E+04 8.233E+04-5.011E+04-2.018E+02 1.029E+04 1.132E+05-6.698E+04
 7.527E+01 3.285E+03 2.079E+05-4.385E+05-5.797E+01 9.571E+03 5.352E+05-8.755E+05
-9.148E+01 1.942E+04 6.490E+05-1.673E+06-5.416E+01 1.706E+04 8.630E+05-1.792E+06
 2.771E+01-2.903E+03 1.419E+07-7.456E+08 7.544E+00 1.589E+04 1.052E+07-1.089E+09
 1.330E+03 0.        0.        0.       -8.936E+02 1.145E+04 1.094E+03-3.692E+03
-2.152E+02 7.720E+03 5.372E+04-2.276E+04-2.914E+02 1.268E+04 6.828E+04-3.620E+04
-2.568E+02 1.341E+04 8.685E+04-5.387E+04-2.104E+02 1.155E+04 1.107E+05-7.279E+04
 1.088E+02 1.536E+03 2.458E+05-6.005E+05-5.691E+01 9.660E+03 5.556E+05-9.320E+05
-9.167E+01 2.050E+04 6.520E+05-1.784E+06 9.687E-01 5.331E+03 1.646E+06-1.518E+07
 3.315E+01-7.298E+03 1.587E+07-8.866E+08 7.728E+00 1.675E+04 1.090E+07-1.231E+09
 1.460E+03 0.        0.        0.       -1.082E+03 1.308E+04-9.100E+02-2.846E+03
-1.923E+02 6.709E+03 6.231E+04-2.410E+04-2.559E+02 1.024E+04 8.602E+04-3.977E+04
-2.466E+02 1.276E+04 9.751E+04-5.925E+04-2.304E+02 1.518E+04 9.220E+04-7.454E+04
 8.675E+01 2.132E+03 2.568E+05-6.538E+05-5.793E+01 1.018E+04 5.793E+05-1.018E+06
-8.637E+01 1.890E+04 7.445E+05-1.949E+06 5.556E-01 5.648E+03 1.738E+06-1.662E+07
 2.968E+01-2.026E+03 1.484E+07-7.825E+08 8.135E+00 1.726E+04 1.171E+07-1.346E+09
-0.       -0.       -0.       -0.
   1   9  17  24  31  38  45  52  58  64  71  77  83  89  95 101 107 113 119 126
 133 139 144 149 154 160 166 172 180 187 195 204 213 222 231 240 249 258 268 278
 287 296 305 314 323 332 341 350 358 367 377 387 397 407 418 429 441 453 465 477
 488 501 514 527 539 551 563 575 587 599 611 623 635 647 659 671 683 695 707 719
 731 743 755 767 779 791 803 815 827 839 851 863 875 887 899 911 923 935 947 959
     .0140     .1000     .8000    4.0000   20.0000  100.0000  500.000010000.0000
     .0250     .1000     .8000    4.0000   20.0000  100.0000  500.000010000.0000
     .0550     .8000    4.0000   20.0000  100.0000  500.000010000.0000     .1110
     .8000    4.0000   20.0000  100.0000  500.000010000.0000     .1880     .8000
    4.0000   20.0000  100.0000  500.000010000.0000     .2840     .8000    4.0000
   20.0000  100.0000  500.000010000.0000     .4000     .8000    4.0000   20.0000
  100.0000  500.000010000.0000     .5330    4.0000   20.0000  100.0000  500.0000
10000.0000     .6870    4.0000   20.0000  100.0000  500.000010000.0000     .0180
     .8670    4.0000   20.0000  100.0000  500.000010000.0000     .0320    1.0730
   20.0000  100.0000  500.000010000.0000     .0500    1.3050   20.0000  100.0000
  500.000010000.0000     .0730    1.5600   20.0000  100.0000  500.000010000.0000
     .1000    1.8390   20.0000  100.0000  500.000010000.0000     .1300    2.1440
```

===== ELT/ATOM =====

20.0000	100.0000	500.0000	10000.0000	.1650	2.4720	20.0000	100.0000	
500.0000	10000.0000	.2020	2.8240	20.0000	100.0000	500.0000	10000.0000	
.2450	3.2030	20.0000	100.0000	500.0000	10000.0000	.0180	.2940	
3.6070	20.0000	100.0000	500.0000	10000.0000	.0250	.3460	4.0370	
20.0000	100.0000	500.0000	10000.0000	.0320	.4010	4.4910	100.0000	
500.0000	10000.0000	.4560	4.9660	100.0000	500.0000	10000.0000	.5130	
5.4650	100.0000	500.0000	10000.0000	.5750	5.9890	100.0000	500.0000	
10000.0000	.1000	.6400	6.5390	100.0000	500.0000	10000.0000	.1000	
.7080	7.1120	100.0000	500.0000	10000.0000	.1000	.7790	7.7090	
100.0000	500.0000	10000.0000	.1000	.8540	.8710	1.0080	8.3320	
100.0000	500.0000	10000.0000	.1000	.9330	1.0960	8.9810	100.0000	
500.0000	10000.0000	.1000	1.0200	1.0430	1.1930	9.6590	100.0000	
500.0000	10000.0000	.0170	.1000	1.1150	1.1420	1.3000	10.3670	
100.0000	500.0000	10000.0000	.0280	.1000	1.2170	1.2480	1.4130	
11.1040	100.0000	500.0000	10000.0000	.0410	.1000	1.3230	1.3590	
1.5300	11.8670	100.0000	500.0000	10000.0000	.0550	.1000	1.4340	
1.4750	1.6520	12.6580	100.0000	500.0000	10000.0000	.0710	.8000	
1.5510	1.5970	1.7820	13.4740	100.0000	500.0000	10000.0000	.0900	
.8000	1.6750	1.7270	1.9210	14.3230	100.0000	500.0000	10000.0000	
.0310	.1100	1.8050	1.8630	2.0650	15.2000	100.0000	500.0000	
10000.0000	.0200	.1330	.8000	1.9400	2.0070	2.2160	16.1050	
100.0000	500.0000	10000.0000	.0250	.1560	.8000	2.0790	2.1560	
2.3730	17.0380	100.0000	500.0000	10000.0000	.1800	.8000	2.2230	
2.3070	2.5330	17.9980	100.0000	500.0000	10000.0000	.2040	.8000	
2.3700	2.4640	2.6980	18.9860	100.0000	500.0000	10000.0000	.2280	
.8000	2.5210	2.6250	2.8670	20.0000	100.0000	500.0000	10000.0000	
.2530	.8000	2.6770	2.7930	3.0430	21.0440	100.0000	500.0000	
10000.0000	.2800	.8000	2.8380	2.9670	3.2240	22.1170	100.0000	
500.0000	10000.0000	.3070	.8000	3.0040	3.1460	3.4120	23.2200	
100.0000	500.0000	10000.0000	.3350	.8000	3.1740	3.3300	3.6050	
24.3500	100.0000	500.0000	10000.0000	.3670	.8000	3.3510	3.5240	
3.8060	25.5140	100.0000	500.0000	10000.0000	.4040	3.5370	3.7270	
4.0180	26.7110	100.0000	500.0000	10000.0000	.0160	.4430	3.7300	
3.9380	4.2380	27.9400	100.0000	500.0000	10000.0000	.0240	.4850	
.7140	3.9290	4.1560	4.4650	29.2000	100.0000	500.0000	10000.0000	
.0310	.5280	.7660	4.1320	4.3810	4.6980	30.4910	100.0000	
500.0000	10000.0000	.0400	.5720	1.0060	4.3410	4.6120	4.9390	
31.8140	100.0000	500.0000	10000.0000	.0500	.6190	1.0720	4.5570	
4.8520	5.1880	33.1700	100.0000	500.0000	10000.0000	.0630	.6720	
.9360	1.1430	4.7820	5.1020	5.4450	34.5610	100.0000	500.0000	
10000.0000	.0790	.7260	1.0650	1.2170	5.0120	5.3600	5.7130	
35.9850	100.0000	500.0000	10000.0000	.0920	.7800	1.0610	1.1350	
1.2910	5.2470	5.6230	5.9870	37.4410	100.0000	500.0000	10000.0000	
.0990	.8320	1.1240	1.2040	1.3630	5.4840	5.8910	6.2660	
38.9250	100.0000	500.0000	10000.0000	.1050	.8830	1.1850	1.2730	
1.4350	5.7230	6.1640	6.5490	40.4430	100.0000	500.0000	10000.0000	
.1120	.9310	1.2420	1.3370	1.5050	5.9640	6.4400	6.8350	
41.9910	100.0000	500.0000	10000.0000	.1170	.9780	1.2980	1.5750	
6.2080	6.7220	7.1280	43.5690	100.0000	500.0000	10000.0000	.1220	
1.0270	1.0520	1.3570	1.4710	1.6480	6.4590	7.0130	7.4280	
45.1840	100.0000	500.0000	10000.0000	.1270	1.0780	1.1060	1.4190	
1.5410	1.7230	6.7160	7.3120	7.7360	46.8340	100.0000	500.0000	
10000.0000	.1340	1.1310	1.1610	1.4810	1.6140	1.8000	6.9770	
7.6180	8.0520	48.5190	100.0000	500.0000	10000.0000	.1400	1.1850	
1.2170	1.5440	1.6880	1.8810	7.2430	7.9300	8.3750	50.2390	
500.0000	10000.0000	.1470	1.2400	1.2740	1.6100	1.7650	1.9630	
7.5140	8.2520	8.7080	51.9960	500.0000	10000.0000	.1540	1.2950	

===== ELT/ATOM =====

1.3320	1.6760	1.8420	2.0460	7.7900	8.5800	9.0460	53.7880
500.000010000.0000		.1610	1.3510	1.3920	1.7430	1.9230	2.1300
8.0720	8.9180	9.3940	55.6180	500.000010000.0000		.1690	1.4090
1.4530	1.8120	2.0060	2.2170	8.3580	9.2640	9.7520	57.4860
500.000010000.0000		.1770	1.4680	1.5150	1.8810	2.0900	2.3060
8.6480	9.6170	10.1160	59.3900	500.000010000.0000		.1880	1.5280
1.5770	1.9500	2.1750	2.3980	8.9430	9.9780	10.4890	61.3320
500.000010000.0000		.1990	1.5910	1.6410	2.0240	2.2640	2.4940
9.2450	10.3490	10.8740	63.3160	500.000010000.0000		.2130	1.6620
1.7160	2.1080	2.3640	2.6000	9.5600	10.7390	11.2720	65.3450
500.000010000.0000		.2280	1.7350	1.7930	2.1940	2.4690	2.7090
9.8800	11.1360	11.6800	67.4160	500.000010000.0000		.2420	1.8090
1.8710	2.2810	2.5750	2.8200	10.2040	11.5410	12.0980	69.5250
500.000010000.0000		.2580	1.8830	1.9500	2.3680	2.6820	2.9340
10.5340	11.9570	12.5280	71.6760	500.000010000.0000		.2740	1.9600
2.0310	2.4570	2.7920	3.0520	10.8710	12.3850	12.9690	73.8710
500.000010000.0000		.2930	2.0400	2.1160	2.5510	2.9080	3.1730
11.2150	12.8240	13.4190	76.1110	500.000010000.0000		.3130	2.1220
2.2020	2.6450	3.0270	3.2970	11.5640	13.2730	13.8800	78.3950
500.000010000.0000		.3340	2.2060	2.2910	2.7430	3.1500	3.4250
11.9190	13.7340	14.3530	80.7250	500.000010000.0000		.3600	2.2950
2.3850	2.8470	3.2800	3.5620	12.2830	14.2090	14.8420	83.1020
500.000010000.0000		.3860	2.3890	2.4850	2.9560	3.4160	3.7040
12.6560	14.6970	15.3460	85.5300	500.000010000.0000		.4130	2.4840
2.5860	3.0660	3.5540	3.8510	13.0350	15.2000	15.8610	88.0040
500.000010000.0000		.4400	2.5810	2.6890	3.1770	3.6960	4.0000
13.4200	15.7140	16.3910	90.5260	500.000010000.0000		.4730	2.6830
2.7980	3.2950	3.8490	4.1560	13.8140	16.2440	16.9360	93.1050
500.000010000.0000		.5060	2.7870	2.9090	3.4160	4.0060	4.3170
14.2140	16.7850	17.4910	95.7300	500.000010000.0000		.5410	2.8920
3.0220	3.5380	4.1640	4.4820	14.6190	17.3370	18.0550	98.4040
500.000010000.0000		.5760	3.0000	3.1360	3.6640	4.3250	4.6520
15.0300	17.9040	18.6390	101.1370	500.000010000.0000		.6100	3.1090
3.2530	3.7910	4.4900	4.8240	15.4460	18.4840	19.2370	103.9220
500.000010000.0000		.6440	3.2190	3.3710	3.9180	4.6580	5.0020
15.8700	19.0830	19.8450	106.7590	500.000010000.0000		.6760	3.3320
3.4900	4.0460	4.8300	5.1820	16.3000	19.6930	20.4660	109.6510
500.000010000.0000		.7060	3.4420	3.6090	4.1740	5.0030	5.3640
16.7330	20.3140	21.1050	112.6010	500.000010000.0000		.7400	3.5520
3.7280	4.3040	5.1810	5.5480	17.1700	20.9480	21.7590	115.6060
500.000010000.0000		.7720	3.6640	3.8500	4.4350	5.3660	5.7350
17.6130	21.6000	22.4270	118.6700	500.000010000.0000		.8010	3.7780
3.9730	4.5680	5.5550	5.9270	18.0630	22.2700	23.1090	121.7970
500.000010000.0000		.8280	3.8940	4.1000	4.7030	5.7480	6.1220
18.5190	22.9580	23.8120	124.9900	500.000010000.0000		.8530	4.0120
4.2300	4.8390	5.9450	6.3220	18.9820	23.6630	24.5350	128.2530
500.000010000.0000		.8770	4.1320	4.3640	4.9770	6.1470	6.5260
19.4520	24.3850	25.2750	131.5900	500.000010000.0000		.9020	4.2540
4.5020	5.1170	6.3530	6.7350	19.9290	25.1250	26.0300	135.0050
500.000010000.0000		.9270	4.3780	4.6440	5.2590	6.5640	6.9490
20.4140	25.8830	26.8030	138.5020	500.000010000.0000		.9520	4.5040
4.7900	5.4030	6.7800	7.1680	20.9070	26.6590	27.5940	142.0850
500.000010000.0000		-0.0000					
.9921	.4997	.4323	.4438	.4625	.4995	.4998	.5000	.4737	.4955	.4785
.4936	.4818	.4985	.4843	.4990	.4795	.4506	.4859	.4990	.4671	.4593
.4515	.4616	.4551	.4656	.4581	.4769	.4564	.4589	.4446	.4408	.4405
.4306	.4380	.4296	.4329	.4337	.4387	.4385	.4413	.4378	.4343	.4353

===== ELT/ATOM =====

.4373	.4323	.4357	.4270	.4268	.4213	.4189	.4075	.4176	.4113	.4138
.4077	.4103	.4110	.4187	.4160	.4207	.4124	.4146	.4070	.4090	.4062
.4062	.4066	.4084	.4045	.4058	.4034	.4034	.4025	.4028	.3996	.4006
.3998	.4011	.3988	.3963	.3958	.3972	.4000	.4048	.3874	.3901	.3877
.3921	.3879	.3939	.3865	.3924	.3884	.3909	.3918	.3896	.3936	.3898
.3968										

```
      =====   ERF   =====

      FUNCTION ERF(Y)
      DATA PI/3.1415927  /,P/.3275911/
       YY = Y
       Y =  ABS(Y)
      T=1./(1.+P*Y)
      A1=.254829592  *T
      A2=-.284496736  *T**2
      A3=1.421413741  *T**3
      A4=-1.453152027  *T**4
      A5=1.061405427  *T**5
      ERF=1.0-(A1+A2+A3+A4+A5)* EXP(-Y**2)
      IF(Y.LT..1E-4)ERF=2.*(Y-Y**3+Y**5/10.)/ SQRT(PI)
       Y =YY
       IF (YY.LE.0.)ERF = -ERF
      RETURN
      END
```

```
=====  GEN/XMU  =====

      FUNCTION XMU(EM)
      COMMON/XCOF/COEF(20,4),ELIM(20),NUM,ONEZOA
      EM1 = EM
      EM2 = EM*EM
      EM3 = EM2*EM
      EM4 = EM3*EM
      DO 10 I = 1,NUM
      IF(EM .GE. ELIM(I) .AND. EM .LE. ELIM(I+1)) GO TO 11
10     CONTINUE
      PRINT 1,EM
1     FORMAT(' EM =',E8.2,' NOT FOUND')
      XMU=0.
      RETURN
11     CONTINUE
     XMU = COEF(I,1)/EM + COEF(I,2)/EM2 + COEF(I,3)/EM3 + COEF(I,4)/EM4
C
C    COMPUTE THE INCOHERENT TOTAL CROSSECTION
C
      X = EM/511.06
      TOP = 1. + 1.148*X + 0.06141*X**2
      BOTTOM = 1. + 3.171*X + 0.9328*X**2 + 0.0257*X**3
      AINC = 0.4006 * TOP/BOTTOM * ONEZOA
C
C    ADD TO PHOTOELECTRIC
C
      XMU = XMU + AINC
      RETURN
      END
```

```
      =====  HTPFN  =====

      FUNCTION HTPFN(U,F,DE,TM,DIST,TD)


C      *INPUT FROM LAYERP
       REAL*4  CP(4),X2(4),RHO(4),Q(100,4),X(100,4)
       INTEGER IX(4)
       COMMON /PLAY/ IX,X2,RHO,CP,ALPHA,X,Q




C      *INPUT FROM PTIME
      INTEGER NLAYR
      DIMENSION TEMP(50,250)
      COMMON  /PTEMP/     NLAYR,TEMP


       DOUBLE PRECISION TK,E,D,UTA,A,B,H1,H2,H3,ANSWER,H
      UTA=U**2*TD*ALPHA
       IF(UTA .LE. 1.E-3) GO TO 737
      TMM=TM
      TK=ALPHA*RHO(NLAYR)*CP(NLAYR)
      E=DEXP(-U*DIST)
      D=F*DE*U/(RHO(NLAYR)*CP(NLAYR)*4.186)
      N=0
10    A=SQRT(ALPHA*TM)
      N=N+1
      B=DIST/(2.*A)
      H1 = U  * 2. * A * AERFC(B)
      H2 =  E
      H3 = HTPIM(U,F,DE,TM,DIST)
       ANSWER = D * ( H1 - H2)/UTA + H3/UTA
       HTPFN = ANSWER
      IF(N .GE. 2) GO TO 20
      IF(TM .LE. TD) RETURN
      TM=TM-TD
      H=ANSWER
      GO TO 10
20    ANSWER=H-ANSWER
       HTPFN = ANSWER
      TM=TMM
      RETURN
737      CONTINUE
       TVAL = TM - TD/2.
       IF(TM .LE. TD ) TVAL = TM/2.
       HTPFN = HTPIM(U,F,DE,TVAL,DIST)
       RETURN
      END
```

```
      =====  HTPIM  =====

      FUNCTION HTPIM(U,F,DE,TM,DIST)
C      *INPUT FROM LAYERP
       REAL*4  CP(4),X2(4),RHO(4),Q(100,4),X(100,4)
       INTEGER IX(4)
       COMMON /PLAY/ IX,X2,RHO,CP,ALPHA,X,Q
C      *INPUT FROM PTIME
      INTEGER NLAYR
       DIMENSION TEMP(50,250)
      COMMON  /PTEMP/     NLAYR,TEMP

      A=SQRT(ALPHA*TM)
      B=U*A
      D=F*DE*U/(RHO(NLAYR)*CP(NLAYR)*4.186)
      IF(DIST .LE. 0.0) GO TO 10
      C=DIST/(2.*A)

      IF(B .LT. C) GO TO 20
      IF(B .EQ. C) GO TO 30
      HTPIM=D/2.*(PART(B-C)+PART(B+C))*EXP(-C*C)
      RETURN
20    HTPIM=D*EXP(B*B-U*DIST)+D/2.*(PART(C+B)-PART(C-B))*EXP(-C*C)
      RETURN
30    HTPIM=D/2.*(EXP(-B*B)+PART(B+C)*EXP(-C*C))
      RETURN
10    HTPIM=D*PART(B)
      RETURN
      END
```

```
      =====  INITIA  =====

 1.
 2.                          SUBROUTINE INITIA
 3.      C       THIS SUBROUTINE HANDLES ALL THE INITIALIZATION REQUIRED BY
 4.      C       THE WALL STUDY ROUTINES
 5.
 6.      C      COMMON/COEFF/A2(4,8),A6(4,5),A8(4,6),A10(4,7),A41(4,7),A82(4,12)
 7.      C     &,A29(4,7)
 8.
 9.
10.      C       READ IN ELEMENTS
11.              CALL CROS
12.      C      READ,A2,A6,A8,A10,A29,A41,A82
13.              RETURN
14.              END
```

===== LASORB =====

```
                        SUBROUTINE LASORB

      REAL*4 F(100,5),E(100),KEV,FLUX,SUMFLU
      REAL*4  C(4),WIDTH(4),RHO(4),U(100,4),X(100,4)
      INTEGER IWID(4)
      COMMON /PSPEC/ JK,DE,E,F,KEV,FLUX,SUMFLU
      REAL*4 PI/3.14159/ ,DE
      COMMON /PLAY/ IWID,WIDTH,RHO,C,ALPHA,X,U
      READ ,WAVLTH,FLUX,RADIUS
      TLENER = FLUX
      FLUX = FLUX/4./3.141593/(RADIUS*100.)**2
      JK = 1
      DE = 1.
      F(1,1) = FLUX
      F(1,2) = FLUX
      F(1,3) = FLUX
      UUU = 25.13274 * 1.E+4/WAVLTH
      U(1,1) = UUU
      U(1,2) = UUU
      PRINT 100, WAVLTH,FLUX,UUU
100     FORMAT(///,'LASER ABSORPTION   WAVELENGTH IS  '
     &,F10.3,//,'     FLUX IS ' ,F10.3,//,  'ABSORPTION
     &COEFFICIENT SI ' ,E12.3,//)
      RETURN
      END
```

```
=====  LAYERP  =====

                    SUBROUTINE LAYERP

C       THIS SUBROUTINE CALCULATES THE EFFECTS OF THE COMPOSITION
C      OF THE WALL


       REAL*4    XI
       INTEGER I,IZ


C      INPUT FROM SPECP
       REAL*4    DE,E(100),F(100,5)
       INTEGER JK
       COMMON /PSPEC/ JK,DE,E,F,KEV,FLUX,SUMFLU


C      OUTPUT OF THIS ROUTINE
       REAL*4  C(4),WIDTH(4),RHO(4),U(100,4),X(100,4)
       INTEGER IWID(4)
       COMMON /PLAY/ IWID,WIDTH,RHO,C,ALPHA,X,U




C      GET THE LAYER SIZE SPECS

5      CONTINUE
       READ(5,-,END=99)NUM,WIDTH(NUM),IWID(NUM)
C       WRITE(6,BUGS)

C      MAKE AN X-GRID (LOG)
       YYY = IWID(NUM)
       XI = (YYY - 1.)/5.
       IF (XI .EQ. 0. ) XI =1.

          III=IWID(NUM)
       DO 10 I=1,III
        FI=I
        X(I,NUM)=WIDTH(NUM)/1.0E5*(10.0**((FI - 1.0)/XI))
10     CONTINUE

C      NOW  HANDLE THE MATERIAL COMPOSITION
       DO 20 I=1,100
       U(I,NUM)=0.0
20     CONTINUE

       RHO(NUM)=0.

25     CONTINUE
       READ(5,-,END=98) IZ,RO,C(NUM),ALPHA,IG,AM,P,GTMP
       CALL  ONESET(IZ)
       IF (IG .EQ. 1) RO=AM*P/GTMP*1.721E-5

       DO 30 I=1,JK
       U(I,NUM)= U(I,NUM) + RO*XMU( E(I))
30     CONTINUE
       RHO(NUM)=RHO(NUM)+RO
```

```
=====  LAYERP  =====

      GO TO 25
98    CONTINUE
C      PRINT ,U
      IF(NRPT .EQ.5762) GO TO 5
      PRINT 100, NUM,IZ,RHO(NUM),C(NUM),ALPHA,IG,AM,P,GTMP
100    FORMAT(//,3X,'LAYER NUMBER',I5,'  HAS THE FOLLOWING PARAMETERS'
     &,//,5X,'     Z                =',I5,/,
     &5X,' COMPOSITE DENSITY   =    ',E12.5,/,
     &5X,' SPECIFIC HEAT  =         ',F12.5,/,
     &5X,' THERMAL DIFFUSIVITY  = ',F12.5,/,
     &5X,' GAS INDICATOR =            ',I5,/,
     &5X,'             A             = ',F12.5,/,
     &5X,' GAS PRESSURE               = ',F12.5,/,
     &5X,' GAS TEMPERATURE            = ',F12.5,//)
      GO TO 5
99     NRPT = 5762
      RETURN
      END
```

```
=====  MASTER  =====

 1.
 2.      C       ****************************************
 3.
 4.      C       THIS IS THE DRIVER ROUTINE FOR THE WALL LOADING STUDY
 5.
 6.      C       ********************************************************
 7.
 8.              INTEGER INST
 9.
10.      1       CONTINUE
11.      C       FIND OUT WHAT TO DO
12.
13.              READ(5,10,END=99)INST
14.              PRINT 1000,INST
15.      1000    FORMAT(' ',A6)
16.      10      FORMAT(A6)
17.
18.      C       GO DO IT
19.
20.              IF (INST .EQ. 'INITIA') CALL INITIA
21.              IF(INST .EQ. 'STARAY') CALL STARAY
22.              IF(INST .EQ.'PSTEMP') CALL STTIME
23.              IF (INST .EQ. 'P-LAYE') CALL LAYERP
24.              IF (INST .EQ. 'P-SPEC') CALL SPECP
25.              IF (INST .EQ. 'P-DEPO') CALL DEPOP
26.              IF (INST .EQ. 'P-TIME' ) CALL TIMEP
27.              IF (INST .EQ. 'P-OUTP' ) CALL OUTPHO
28.             IF (INST .EQ. 'P-LASE' ) CALL LASORB
29.              IF (INST .EQ. 'OPEN') CALL OPEN
30.              IF (INST .EQ. 'REOPEN') CALL REOPEN
31.              IF (INST .EQ. 'LOGIN') CALL LOGIN
32.              IF (INST .EQ. 'P-FILE') CALL FILEP
33.
34.              GO TO 1
35.
36.      99      STOP
37.              END
```

===== OUTPHO =====

```
                                    SUBROUTINE OUTPHO
C     THIS ROUTINE MAKES AN OUTPUT LISTING OF THE PHOTON
C   SITUATION


C      *INPUT FORM DEPOP
       REAL*4  Y1(100,4),Y2(100,4)
       REAL*4   ADBTMP(100,4)
       COMMON  /PDEPO/ Y1,Y2,ADBTMP

C      *INPUT FROM LAYERP

       REAL*4  C(4),X2(4),RHO(4),U(100,4),X(100,4)
       INTEGER IX(4)
       COMMON /PLAY/ IX,X2,RHO,C,ALPHA,X,U

C      INPUT FROM  SPECP
       INTEGER JK
       REAL*4   DE,E(100),F(100,5)
      COMMON /PSPEC/ JK,DE,E,F,KEV,FLUX,SUMFLU

C      *PRINT OUT THE SPECTRUM RELATED VARIABLES
       PRINT 101, KEV,FLUX,JK
       PRINT 102,SUMFLU
101    FORMAT('1',9X,'KT =',F5.2,5X,'FLUX =',F5.2,5X,'JK =',I5,/,10X,
     &        40('*'))
102    FORMAT(  10X,'INTEGAL OF FLUX =',F10.4)

C      *PRINT OUT  E , F ,& U ARRAYS
       PRINT 1
1      FORMAT(///,' ',
     .'   NUM        ENERGY           F1              MU1            F2   ',
     .'        MU2            F3')

         PRINT 13,(I,E(I),F(I,1),U(I,1),F(I,2),U(I,2),F(I,3),I=1,JK,10)
13     FORMAT(' ',I7,6E14.4)


C      *NOW OUTPUT THE X-RELATED VARIABLES
       PRINT 10
10     FORMAT('1',//,
     .'    NUM          X(1)          Y1(1)          Y2(1)      ADBTEMP(1)'
     .,'         X(2)          Y1(2)          Y2(2)      ADBTEMP(2)')

       IXX=MAX(IX(1),IX(2))
       PRINT    15,(I,(X(I,J),Y1(I,J),Y2(I,J),ADBTMP(I,J),J=1,2),
     .          I=1,IXX)
       IF(IX(3) .LE.0)  GO TO 20
       PRINT 11
11     FORMAT('1',//,
     .'    NUM          X(3)          Y3(3)          Y2(3)      ADBTEMP(3)'
     .,'         X(4)          Y1(4)          Y4(4)      ADBTEMP(4)')
       IXX = MAX(IX(3),IX(4))
       PRINT    15,(I,(X(I,J),Y1(I,J),Y2(I,J),ADBTMP(I,J),J=3,4),
     .          I=1,IXX)
20       CONTINUE

```

```
      =====  OUTPHO  =====

      15      FORMAT(' ',I7,8E14.4)
              RETURN
              END
```

```
      =====  PART  =====

      FUNCTION PART(X)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
       REAL*4   PART,X
      P=.3275911D0
      XX=X
      X=DABS(X)
      T=1./(1.+P*X)
      A1=.254829592D0*T
      A2=-.284496736D0*T**2
      A3=1.421413741D0*T**3
      A4=-1.453152027D0*T**4
      A5=1.061405427D0*T**5
      PART=A1+A2+A3+A4+A5
      X=XX
      IF(XX .LE. 0.) PART=-PART
      RETURN
      END
```

```
=====  PFILE/IODR  =====

                    SUBROUTINE FILEP


C           PHOTON DATA FROM THE COMMON BLOCKS

C      *OUTPUT OF P-TIME
       INTEGER NLAYR
        DIMENSION  TEMP(50,250)
       REAL*4      TM(250),Y(50)
       COMMON  /PTEMP/     NLAYR,TEMP,TM,Y,NUMT,NUMY

C      *INPUT FORM DEPOP
        REAL*4  Y1(100,4),Y2(100,4)
        REAL*4   ADBTMP(100,4)
        COMMON  /PDEPO/ Y1,Y2,ADBTMP

C       *INPUT FROM LAYERP

        REAL*4  C(4),X2(4),RHO(4),U(100,4),X(100,4)
        INTEGER IX(4)
        COMMON /PLAY/ IX,X2,RHO,C,ALPHA,X,U

C       INPUT FROM  SPECP
        INTEGER JK
        REAL*4   DE,E(100),F(100,5)
       COMMON /PSPEC/ JK,DE,E,F,KEV,FLUX,SUMFLU

        INTEGER NPTS(50)

C      *************  END OF DATA SECTION  ********************

        CALL OPN BIN('PHOTON')
C       __  FILE THE DATA
        NPTS(1)=JK
        NPTS(2)=NUMY
        NPTS(3)=NUMT
        NPTS(4)=IX(1)
        NPTS(5)=IX(2)
        NPTS(6)=IX(3)
        NPTS(7)=IX(4)

        CALL FILE(NPTS,1)
       CALL FILE(TEMP,250)
       CALL FILE(TM,5)
       CALL FILE(Y,1)
       CALL FILE(F,10)
       CALL FILE(E,2)
       CALL FILE(U,8)
       CALL FILE(Y1,8)
       CALL FILE(X,8)
       CALL FILE(ADBTMP,8)

C       ** CLEAR THE BUFFER
       CALL CLEAR
       RETURN
        END
```

===== SPECP =====

```
1.                          SUBROUTINE SPECP
2.
3.      C        *OUTPUT OF THIS ROUTINE
4.               REAL*4 F(100,5),E(100),KEV,FLUX,SUMFLU
5.               COMMON /PSPEC/ JK,DE,E,F,KEV,FLUX,SUMFLU
6.               REAL*4 PI/3.14159/,B,DE,EM
7.               DIMENSION EHIST(20),AMP(20)
8.
9.
10.              READ ,KEV,JK,FLUX,RADIUS
11.              TLENER = FLUX
12.              FLUX = FLUX/4./3.141593/(RADIUS*100.)**2
13.              IF(KEV .GT. 0.) GO TO 9
14.     C
15.     C   ---------HISTOGRAM SPECTRUM
16.     C
17.              READ ,NMHIST,(EHIST(I),AMP(I), I = 1,NMHIST)
18.              DO 76 I = 1,NMHIST
19.     76         AMP(I) = AMP(I)*FLUX/TLENER
20.              EMAX = EHIST(NMHIST)
21.              EMIN = EHIST(1)
22.             FJK=JK
23.             DE=(EMAX-EMIN)/FJK
24.             EM=EMIN
25.             SUMFLU=0.
26.              DO 8 J = 1,JK
27.              DO 7 I = 1,NMHIST
28.     7         IF(EM .LE. EHIST(I)) GO TO 77
29.     77        F(J,1) = AMP(I-1)
30.              IF(I .EQ.1)F(J,1) = AMP(1)
31.              E(J) = EM
32.              EM = EM + DE
33.                SUMFLU=SUMFLU+F(J,1)*DE
34.     8        CONTINUE
35.               RETURN
36.     9         CONTINUE
37.     C
38.     C----------BLACK BODY SPECTRUM
39.     C
40.            EMAX=KEV*10.0
41.            EMIN=KEV*0.1
42.            FJK=JK
43.            DE=(EMAX-EMIN)/FJK
44.            B=FLUX*15.0/(PI**4)/KEV
45.            EM=EMIN+DE/2.
46.            SUMFLU=0.
47.            DO 11 J=1,JK
48.               V=EM/KEV
49.               F(J,1)=B*(V**3)/(EXP(V)-1.0)
50.                E(J)=EM
51.               SUMFLU=SUMFLU+F(J,1)*DE
52.               EM=EM+DE
53.     11     CONTINUE
54.
55.
56.
57.     C        DO SOME OUTPUT TO COMPARE W/ KNOWN RESULTS
```

```
                =====   SPECP   =====

58.
59.       C       PRINT 101, KEV,FLUX,JK
60.       C       PRINT 103,SUMFLU
61.       C       PRINT 102,(J,E(J),F(J,1),J=1,JK)
62.       C
63.       C101    FORMAT('1',9X,'KT =',F5.2,5X,'FLUX =',F5.2,5X,'JK =',I5,/,10X,
64.       C      &        40('*'))
65.       C102    FORMAT(//,4(4X,'J',5X,'E',7X,'F'),/,25(/,4(2X,I3,2F8.4,)))
66.       C103    FORMAT(  10X,'INTEGAL OF FLUX =',F10.4)
67.               RETURN
68.               END
```

===== STARAY =====

```
      SUBROUTINE STARAY
C A PROGRAM TO CALCULATE A STANDARD TIME AND LOCATION ARRAY FOR
C ION AND PHOTON RESPONSE MODELS
      COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
      NST = 115
      NSX = 15
      NTP = 22
      DO 10 J = 1,19
      X = J
      ST(J) = 10**((X-1.)/6. - 10.)
10     CONTINUE
      DO 20 J = 20,109
      X = J
      ST(J) = 10**((X-19.)/30. - 7.)
20     CONTINUE
      DO 30 J = 110,115
      X = J
      ST(J) = 10**((X -109)/6. -4.)
30     CONTINUE
      SX(1) = 0.
      SX(2) = 1.E-8
      SX(1) = 1.E-7
      SX(2) = 1.E-6
      SX(3) = 1.E-5
      SX(4) = 2.E-5
      SX(5) = 5.E-5
      SX(6) = 7.E-5
      SX(7) = 1.E-4
      SX(8) = 1.5E-4
      SX(9) = 2.E-4
      SX(10) = 3.E-4
      SX(11) = 5.E-4
      SX(12) = 7.E-4
      SX(13) = 1.E-3
      SX(14) = 3.E-3
      SX(15) = 1.E-2
      DO 50 J = 1,22
      X = J
      STP(J) = 10**((X-1.)/3. -10.)
50     CONTINUE
      RETURN
      END
```

```
      =====  STTIME  =====

 1.                        SUBROUTINE STTIME
 2.     C----------CALCULATE TEMPERATURE AT SELECTED TIME
 3.     C----------IF IMP=1 CALL HTPIM,IFN=1 CALL HTPFN,ILATE=1 CALL TLATE
 4.
 5.           INTEGER IMP/0/,IFN/0/,ILATE/1/ ,IBDUMP/1/
 6.
 7.     C     *OUTPUT OF THIS ROUTINE
 8.           INTEGER NLAYR
 9.            DIMENSION  TEMP(50,250)
10.           REAL*4      TM(250),Y(50)
11.           COMMON  /PTEMP/      NLAYR,TEMP,TM,Y,NUMT,NUMY
12.
13.
14.     C      *INPUT FROM SPECP
15.            REAL*4   DE,E(100),F(100,5)
16.            INTEGER JK
17.            COMMON /PSPEC/ JK,DE,E,F,KEV,FLUX,SUMFLU
18.
19.
20.     C      *INPUT FROM LAYERP
21.            REAL*4  C(4),WIDTH(4),RHO(4),U(100,4),X(100,4)
22.            INTEGER IWID(4)
23.            COMMON /PLAY/ IWID,WIDTH,RHO,C,ALPHA,X,U
24.            COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
25.
26.     C     READ IN SOME PARAMETERS
27.           READ    ,NLAYR,IMOD,TD,w,NPLUS,SIZE
28.
29.            NUMY = NSX
30.            READ,IXS,IXF,IXD
31.            DO 50 I = 1,NSX
32.     50      Y(I) = SX(I)
33.            NUMT = NTP
34.            DO 60 I = 1,NUMT
35.            TM(I) = STP(I)
36.            IF(IMOD .EQ. 2 ) TM(I) = STP(I) + TD
37.     60     CONTINUE
38.            PRINT 120
39.     120     FORMAT(///,15X,'TEMPERATURES WILL BE EVALUATED AT STANDARD'
40.          &,'TIMES',//)
41.
42.             GO TO (1,2,3),IMOD
43.     1       CONTINUE
44.           DO 11 K=IXS,IXF,IXD
45.              DO 11 J=1,NUMT
46.              TEMP(K,J)=0.0
47.                 DO 11 I=1,JK
48.     11       TEMP(K,J)=TEMP(K,J)+HTPIM(U(I,NLAYR),F(I,NLAYR),DE,TM(J),Y(K))
49.             PRINT 104
50.     104      FORMAT(//,15X,' IMPULSE PHOTON MODEL RESULTS',//)
51.             GO TO 77
52.     2      CONTINUE
53.           DO 22 K=IXS,IXF,IXD
54.              DO 22 J=1,NUMT
55.              TEMP(K,J)=0.0
56.                 DO 22 I=1,JK
57.     22     TEMP(K,J)=TEMP(K,J)+HTPFN(U(I,NLAYR),F(I,NLAYR),DE,TM(J),Y(K),TD)
```

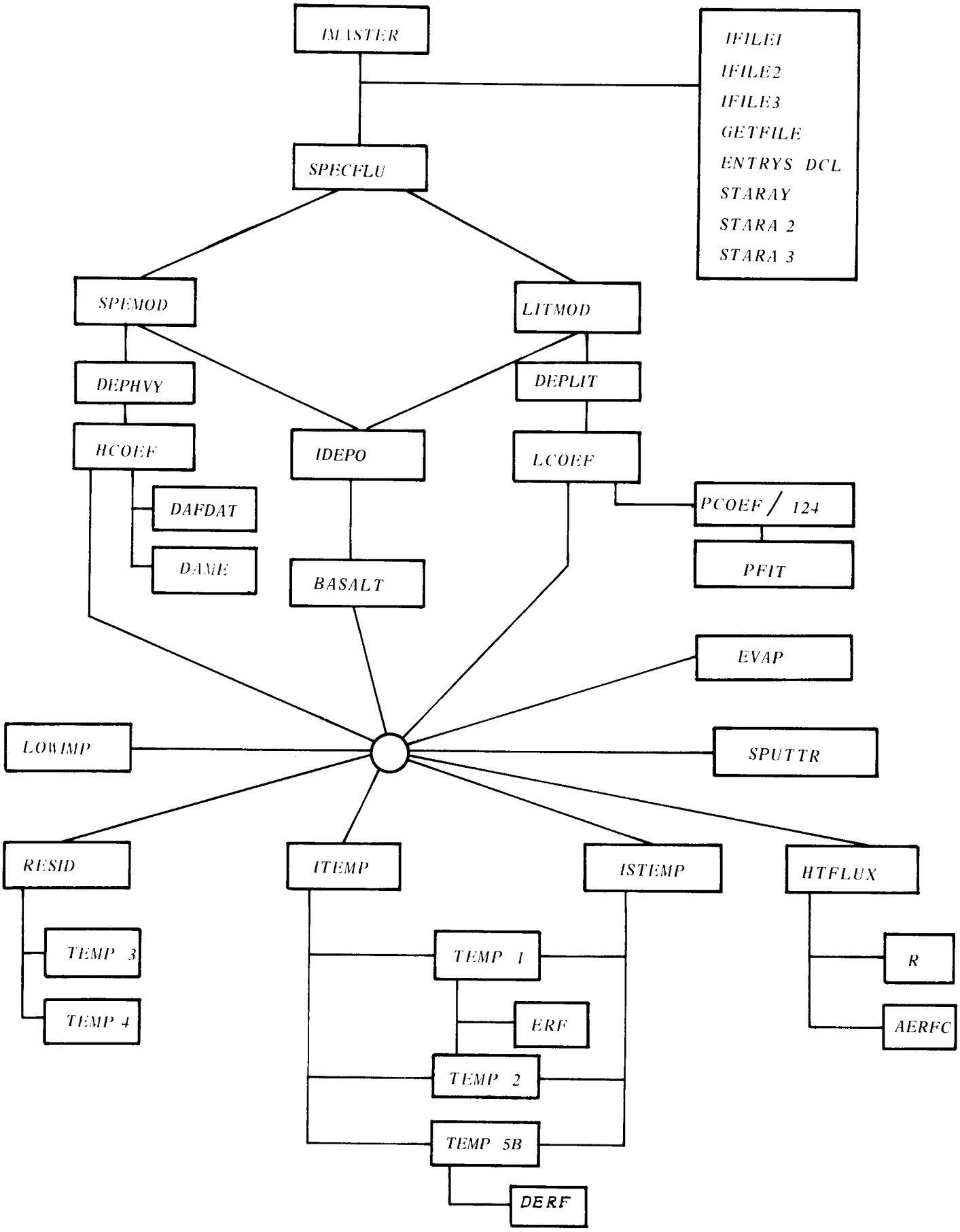
```
      =====  STTIME  =====

      DO 23 J=1,NUMT
23     TM(J) = STP(J)
      PRINT 109, TD
109    FORMAT(//,15X,'FINITE DURATION PHOTON RESPONSE MODEL',//
     &,20X,'PULSE DURATION OF ',5X,E9.3,//
     &  ,'       TIMES ARE REFERENCED TO THE END OF THE PULSE  ',//)
      GO TO 77
3     CONTINUE

      DO 33 K=IXS,IXF,IXD
         DO 33 J=1,NUMT
         TEMP(K,J)=0.0
       HL = 0.0
            DO 33 I=1,JK
      HL =HL +TLATE(U(I,NLAYR),F(I,NLAYR),DE,TM(J),Y(K),W,NPLUS,SIZE)
33     TEMP(K,J) = HL
       PRINT 110,W,NPLUS
110      FORMAT(//,15X,'RESIDUAL TEMPERATURE MODEL',//,10X,'PULSE',
     &'SPACING IS',5X,E9.3,'  SECONDS',5X,   I8,'   PULSES',//)
77     CONTINUE
       NUMY1 = IXF
       IF(IXD .LE.1 .AND. IXF .GT. 10) NUMY1 = 10
       PRINT 105,(Y(I),I = IXS,NUMY1,IXD)
       PRINT 108
      DO 400 I = 1,NUMT
       PRINT 106, TM(I),(TEMP(J,I),J = IXS,NUMY1,IXD)
400      CONTINUE
       IF(IXD.GT.1 .OR. IXF .LT. 10)GO TO 450
      PRINT 107
       PRINT 105,(Y(I),I = 11,NUMY)
      PRINT 108
       DO 410 I = 1,NUMT
       PRINT 106,TM(I),(TEMP(J,I),J = 11,NUMY)
410    CONTINUE
       PRINT 107
450      CONTINUE

105    FORMAT(14X,' X = ',10(2X,E9.3))
106    FORMAT(3X,11(2X,E9.3))
107    FORMAT(//)
108    FORMAT(//,3X,'TIME', 40X,'TEMPERATURE',//)
      RETURN
      END
```

```
=====  TLATE  =====

                      FUNCTION TLATE(U,F,DE,T,X,W,NPLUS,SIZE)

C       *INPUT FROM LAYERP
        REAL*4  CP(4),X2(4),RHO(4),Q(100,4),XXX(100,4)
        INTEGER IX(4)
        COMMON /PLAY/ IX,X2,RHO,CP,ALPHA,XXX,Q



C       *INPUT FROM PTIME
      INTEGER NLAYR
        DIMENSION TEMP(50,250)
      COMMON  /PTEMP/     NLAYR,TEMP


        REAL PI/3.1415926/

      BB=U*SIZE
      Z=X/SIZE
      THETA=ALPHA*(T+W)/SIZE**2
      GAMMA=ALPHA*W/SIZE**2
      CONST=NPLUS
      DELTO=0.
      DO 11 NN=1,20
         FN=NN
         ARG=(2.*FN-1.)*PI/2.
         A1=COS(ARG*Z)
         A2=EXP(-ARG**2*THETA)
         A3=BB**2/(BB**2+ARG**2)
         A4=1.-EXP(-BB)*(COS(ARG)-ARG/BB*SIN(ARG))
         FI=ARG**2*GAMMA
         A5=(1.-EXP(-FI*CONST))/(1.-EXP(-FI))
         XX=A1*A2*A3*A4*A5
         DELTO=DELTO+XX
C         CONV=XX/DELTO
C         IF(CONV .LE. 0.001) GO TO 12
11    CONTINUE
12    CONTINUE
      TLATE=DELTO*F*DE*2./4.186/RHO(NLAYR)/CP(NLAYR)
C     PRINT 100,NN,CONV
100   FORMAT(I5,3E10.3)
      RETURN
      END
```

===== TTIME =====

```
1.                          SUBROUTINE TIMEP
2.      C----------CALCULATE TEMPERATURE AT SELECTED TIME
3.      C----------IF IMP=1 CALL HTPIM,IFN=1 CALL HTPFN,ILATE=1 CALL TLATE
4.
5.            INTEGER IMP/0/,IFN/0/,ILATE/1/ ,IBDUMP/1/
6.
7.      C     *OUTPUT OF THIS ROUTINE
8.            INTEGER NLAYR
9.             DIMENSION  TEMP(50,250)
10.           REAL*4      TM(250),Y(50)
11.           COMMON  /PTEMP/     NLAYR,TEMP,TM,Y,NUMT,NUMY
12.
13.
14.     C      *INPUT FROM SPECP
15.            REAL*4   DE,E(100),F(100,5)
16.            INTEGER JK
17.            COMMON /PSPEC/ JK,DE,E,F,KEV,FLUX,SUMFLU
18.
19.
20.     C      *INPUT FROM LAYERP
21.            REAL*4  C(4),WIDTH(4),RHO(4),U(100,4),X(100,4)
22.            INTEGER IWID(4)
23.            COMMON /PLAY/ IWID,WIDTH,RHO,C,ALPHA,X,U
24.
25.     C      READ IN SOME PARAMETERS
26.            READ    ,NLAYR,IMOD,SIZE
27.            READ,NUMY,(Y(I),I=1,NUMY),NUMT,(TM(I),I=1,NUMT),TD,W,NPLUS
28.
29.
30.             GO TO (1,2,3),IMOD
31.     1       CONTINUE
32.            DO 11 K=1,NUMY
33.               DO 11 J=1,NUMT
34.               TEMP(K,J)=0.0
35.                  DO 11 I=1,JK
36.     11       TEMP(K,J)=TEMP(K,J)+HTPIM(U(I,NLAYR),F(I,NLAYR),DE,TM(J),Y(K))
37.             PRINT 104
38.     104      FORMAT(//,15X,' IMPULSE PHOTON MODEL RESULTS',//)
39.             GO TO 77
40.     2       CONTINUE
41.            DO 22 K=1,NUMY
42.               DO 22 J=1,NUMT
43.               TEMP(K,J)=0.0
44.                  DO 22 I=1,JK
45.     22     TEMP(K,J)=TEMP(K,J)+HTPFN(U(I,NLAYR),F(I,NLAYR),DE,TM(J),Y(K),TD)
46.            PRINT 109, TD
47.     109     FORMAT(//,15X,'FINITE DURATION PHOTON RESPONSE MODEL',//
48.           &,20X,'PULSE DURATION OF ',5X,E9.3,//)
49.             GO TO 77
50.     3       CONTINUE
51.
52.            DO 33 K=1,NUMY
53.               DO 33 J=1,NUMT
54.               TEMP(K,J)=0.0
55.             HL = 0.0
56.                  DO 33 I=1,JK
57.            HL =HL +TLATE(U(I,NLAYR),F(I,NLAYR),DE,TM(J),Y(K),W,NPLUS,SIZE)
```

```
=====  TTIME  =====

C       IF(I .EQ.1 .OR.I .EQ. 30 .OR. I .EQ. 70) PRINT 106,HL
33      TEMP(K,J) = HL
        PRINT 110,W,NPLUS
110       FORMAT(//,15X,'RESIDUAL TEMPERATURE MODEL',//,10X,'PULSE',
     &'SPACING IS',5X,E9.3,'  SECONDS',5X,   I8,'   PULSES',//)
77      CONTINUE
        NUMY1 = NUMY
        IF(NUMY .GE.10) NUMY1 = 10
        PRINT 105,(Y(I),I = 1,NUMY1)
        PRINT 108
       DO 400 I = 1,NUMT
        PRINT 106, TM(I),(TEMP(J,I),J = 1,NUMY1)
400       CONTINUE
        IF(NUMY.LE.10)GO TO 450
       PRINT 107
        PRINT 105,(Y(I),I = 11,NUMY)
       PRINT 108
        DO 410 I = 1,NUMT
        PRINT 106,TM(I),(TEMP(J,I),J = 11,NUMY)
410     CONTINUE
        PRINT 107
450       CONTINUE

105     FORMAT(14X,' X = ',10(2X,E9.3))
106     FORMAT(3X,11(2X,E9.3))
107     FORMAT(//)
108     FORMAT(//,3X,'TIME', 40X,'TEMPERATURE',//)
       RETURN
       END
```


IMASTER
IFILE1
IFILE2
IFILE3
GETFILE
ENTRYS DCL
STARAY
STARA 2
STARA 3
SPECFLU
SPEMOD
LITMOD
DEPHVY
DEPLIT
HCOEF
IDEPO
LCOEF
DAFDAT
DAME
PCOEF / 124
PFIT
BASALT
EVAP
LOWIMP
SPUTTR
RESID
ITEMP
ISTEMP
HTFLUX
TEMP 3
TEMP 4
TEMP 1
ERF
TEMP 2
TEMP 5B
DERF
R
AERFC

===== ION =====

ION RESPONSE
SECTION OF T*DAMEN CODE

ABSOLUTE
IONCODE

READER
READ/ION

MAPPER
MAP/ION

LISTER
ION/LIST

SYMBOLICS
IMASTER--MASTER ROUTINE FOR CALLING EACH SECTION
SPECFLU--CREATES THE PARTICLE FLUXES FROM THE SPECTRA
SPECTR --GENERATES MAXWELLIAN OR GAUSSIAN SPECTRUM
SPEMOD --MODIFIES A HEAVY ION SPECTRUM VIA DIFFUSION APPROXIMATION
LITMOD --MODIFIES A LIGHT ION SPECTRUM USING SLOWING DOWN APPROXIMATION
HCOEF --GENERATES COEFFICIENTS MATRIX FOR HEAVY ION DEPOSITION
DEPLIT --CALCULATES DEPOSITION RATE IN BUFFER GAS FOR LIGHT IONS
DEPHVY --CALCULATES DEPOSITION RATE IN BUFFER GAS FOR HEAVY IONS
LCOEF --GENERATES COEFFICIENTS MATRIX FOR LIGHT ION DEPOSITION
PCOEF/124--GENERATES POLYNOMIAL COEFFICIENTS FOR SET OF INCIDENT ENERGIES
DAME --CALCULATES COEFFICIENTS OF DAMAGE FUNCTIONS A GIVEN VALUE OF ENERGY
DAFDAT --READS DAMAGE FUNCTIONS FROM FILE 11 BY A NUMBER WHICH INDICATES A CERTAIN ION-TARGET COMBINATEION
PFIT --A GENERAL 1-4TH ORDER POLYNOMIAL INTERPOLATION ROUTINE
IDEPO --DOES ENERGY DEPOSITION FOR THE IONS
STARAY --GENERATES 15 VALUES OF X (10-7--10-2) AND 115 VALUES OF TIMES (10-10--10-2)
STARA2 --STANDARD ARRAY FOR 15 VALUES OF X (10-7--10-2) AND 115 VALUES OF TIMES (10-7--10-4)

```
58.
59.
60.
61.
62.
63.
64.              STARA3 --STANDARD ARRAY FOR 50 VALUES OF X (LIMIT IS
65.                       READ IN) AND 115 VALUES OF TIMES
66.              STARAY/PRINT--SYMBOLIC AND ABSOLUTE ELEMENT FOR
67.                           PRINTING STANDARD TIMES AND LOCATIONS
68.                           FROM STARAY
69.              STARA2/PRINT--SYMBOLIC AND ABSOLUTE ELEMENT FOR
70.                           PRINTING STANDARD TIMES AND LOCATIONS
71.                           FROM STARA2
72.              BASALT --TRANSFORMS ARBITRARY TIME BASE TO STANDARD TIME BASE
73.              ITEMP  --EVALUATES TEMPERATURE (X,T) FOR ARBITRARY VALUES
74.              ISTEMP --EVALUATES TEMPERATURE (X,T) FOR THE STANDARD TIMES
75.                       AND LOCATIONS (FROM STARAY)
76.              TEMP1  --TEMPERATURE RESPONSE AT ONE TIME FOR LINEARLY
77.                       DECREASING DEPOSITION
78.              TEMP2  --TEMPERATURE RESPONSE AT ONE TIME FOR UNIFORM
79.                       RESPONSE
80.              TEMP3  --TEMPERATURE FOR FINITE SLAB WITH LINEARLY
81.                       DECREASING DEPOSITION
82.              TEMP4  --TEMPERATURE FOR FINITE SLAB WITH UNIFORM DEPOSITION
83.              TEMP5B --GENERAL TEMPERATURE RESPONSE MODEL FOR DEPOSITION
84.                       IN FORM OF POLYNOMIAL COEFFICIENTS
85.              DERF   --ERROR FUNCTION, DOUBLE PRECISION
86.              ERF    --ERROR FUNCTION, SINGLE PRECISION
87.              IFILE/IODR--FILING OF COMMON BLOCK  IN IODR FILE
88.              ENTRYS/IODR--ROUTINES FOR FILING MECHANICS
89.              DCL/IODR-BASIC FILE STRUCTURE FOR IODR DATA BLOCKS.
90.              LOWIMP --DETERMINES IMPLANTATION DISTRIBUTION FOR LIGHT IONS
91.              HTFLUX --CALCULATES T(X,T) FOR A FLUX OF FINITE
92.                       DURATION USING HEAT FLUX BOUNDARY CONDITION
93.              R      --SUPPORTING FUNCTION FOR HTFLUX
94.              AERFC  --INTERGRAL COMPLEMENTARY ERROR FUNCTION
95.              RESID  --CALCULATES RESIDUAL TEMPERATURE FROM N PREVIOUS
96.                       PULSES
97.              SPUTTR --CALCULATES SPUTTERING YIELD FOR HEAVY IONS
98.                       INCLUDING TEMPERATURE DEPENDENCE
99.              EVAP   --CALCULATES EVAPORATION RATE FROM SURFACE
100.                      TEMPERATURE AND VAPOR PRESSURE DATA
```

```
=====  READ/ION  =====

                *****READ STATEMENTS FOR THE ION CODE*****


           FOR THE IMASTER SUBROUTINE:

      READ (5,44,END=99)INST
44    FORMAT(A6)


       IF ( INST .EQ. 'DEPLIT') CALL GSDPLT
       IF ( INST .EQ. 'DEPHVY') CALL DEDXGS
       IF ( INST .EQ. 'LOWIMP') CALL LOWIMP
       IF ( INST .EQ. 'HCOEF ') CALL HCOEF
       IF ( INST .EQ. 'HCOEF ') CALL COMCOF
       IF ( INST .EQ. 'LCOEF ') CALL LCOEF
       IF ( INST .EQ. 'LITMOD') CALL LITMOD
       IF ( INST .EQ. 'SPEMOD') CALL SPEMOD
       IF ( INST .EQ. 'I-RESI') CALL RESID
      IF( INST .EQ. 'I-SPEC') CALL SPECFL
       IF( INST .EQ. 'I-DEPO' ) CALL IDEPO
      IF ( INST .EQ. 'I-TEMP' ) CALL ITEMP
       IF( INST .EQ. 'STARAY') CALL STARAY
       IF( INST .EQ. 'STARA2') CALL STARA2
       IF( INST .EQ. 'STARA3') CALL STARA3

       IF( INST .EQ.'I-STEM') CALL ISTEMP
       IF (INST .EQ. 'I-OPEN') CALL OPEN
       IF(INST .EQ. 'I-REOP') CALL REOPEN
       IF (INST .EQ. 'I-1FIL') CALL IFILE1
       IF (INST .EQ. 'I-2FIL') CALL IFILE2
       IF (INST .EQ. 'I-3FIL') CALL IFILE3
       IF (INST .EQ. 'LOGIN') CALL LOGIN
       IF (INST .EQ. 'HTFLUX') CALL HTFLUX

           FOR THE SPECFLU SUBROUTINE:

       READ(5,-,END = 11)ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS,IPRI
       IF(ISPEC .NE.3) GO TO 76
       READ,NMHIST,(EHIST(I),AMP(I), I = 1,NMHIST)
       READ,IMODEL,ALPHA,RHO,CP,X2,NX,C,E0,A2,A3,
       IF (IMODEL .NE.5) GO TO 77
       READ,E1,E2,S1,S2,EM


           FOR THE SPEMOD SUBROUTINE:

       READ 377, INST1
       READ 377,INST2
377      FORMAT(A4)
       READ ,E1,R1,SIG1,E2,R2,SIG2,REFP,REFT
       READ , RBUF,GASP,GAST,IPRI
       IF(INST2 .EQ. 'RANG') CALL ERANGE
       IF(INST1 .EQ. 'FILE')CALL IFILE1


           FOR THE LITMOD SUBROUTINE:
```

```
            =====  READ/ION  =====

 58.
 59.             READ 729,INST1
 60.             READ 729,INST2
 61.     729      FORMAT(A4)
 62.             READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
 63.             READ , REFP,REFT
 64.             READ ,RBUF,GASP,GAST,IPRI
 65.             IF(INST2 .NE. 'RANG') GO TO 621
 66.             IF(INST2 .NE. 'RANG') GO TO 92
 67.             IF( IPRI    .EQ. 1) GO TO 4126
 68.             IF(INST1 .EQ. 'FILE') CALL IFILE1
 69.
 70.
 71.                FOR THE HCOEF SUBROUTINE:
 72.
 73.             READ ,LOCDFN,LOCDFE
 74.
 75.
 76.                FOR THE LCOEF SUBROUTINE:
 77.
 78.             READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
 79.
 80.
 81.                FOR THE DEPLIT SUBROUTINE:
 82.
 83.             READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
 84.             READ , REFP,REFT
 85.             READ ,RBUF,GASP,GAST,IPRI
 86.             IF(RBUF .LT. XM(I)) GO TO 30
 87.             IF(RBUF .LT. XH(I)) GO TO 20
 88.             IF(RBUF .LT. XL(I)) GO TO 10
 89.
 90.
 91.                FOR THE DEPHVY SUBROUTINE:
 92.
 93.             READ ,E1,R1,SIG1,E2,R2,SIG2,REFP,REFT
 94.             READ , GASP,GAST
 95.             READ , LAYOUT,RINNER,ROUTER,NSPOTS
 96.             IF(LAYOUT .EQ. 0) X(MJ) = X(1) * (X(NSPOTS)/X(1))**ZZZ
 97.             IF(LAYOUT .EQ. 1) X(MJ) = X(MJ-1) + DELXD
 98.     392     READ (5,144,END=9908)INST
 99.     144      FORMAT(A6)
100.             IF( INST .EQ. 'I-1FIL') CALL IFILE1
101.             IF( INST .EQ. 'I-2FIL') CALL IFILE2
102.     C
103.     C
104.                FOR THE DAFDAT/DEPFUN SUBROUTINE:
105.
106.     5        READ (11,END = 10)
107.     777     READ(11,END = 100) LOC,NE,IDENT,CG
108.
109.
110.                FOR THE IDEPO SUBROUTINE:
111.
112.             READ , IXMATX
113.             IF(IXMATX .NE. 50) GO TO 79
114.             IF(IXMATX .NE. 50) GO TO 89
```

```
      =====  READ/ION  =====
115.
116.
117.              FOR THE STARA3 SUBROUTINE:
118.
119.           READ , BIGX
120.
121.
122.              FOR THE ITEMP SUBROUTINE:
123.
124.           READ ,NY
125.           READ,( Y(I),I = 1,NY)
126.           READ,NT
127.           IF (NT .EQ. 0) GO TO 5
128.           READ,(TA(I),I=1,NT)
129.
130.
131.              FOR THE ISTEMP SUBROUTINE:
132.
133.           READ,IXS,IXF,IXD
134.           IF(IXS .NE. 777) GO TO 48
135.           IMODEL = IXF
136.           READ,IXS,IXF,IXD
137.
138.
139.
140.              FOR THE HTFLUX SUBROUTINE:
141.
142.           READ ,F,TD,ALPHA,TK,M,W
143.
144.
145.              FOR THE RESID SUBROUTINE:
146.
147.           READ ,NY
148.           READ,( Y(I),I = 1,NY)
149.           READ,NT
150.           IF (NT .EQ. 0) GO TO 5
151.           READ,(TA(I),I=1,NT)
152.           READ,NCONV
153.
154.
155.              FOR THE EVAP SUBROUTINE:
156.
157.           READ , BINT
158.           READ , STIPR,AMASS,TAMB
159.           READ , PZERO,HSUB
160.           READ 234,INSTF
161.     234   FORMAT(A6)
162.           IF(INSTF .NE.'I-1FIL') GO TO 266
```

```
=====  MAP/ION  =====

 1      @MAP*MAP.MAP    ,T*DAMEN.IONCODE
 2      IN   T*DAMEN.IMASTER,.SPECFLU,.SPECTR,.IDEPO,.ITEMP,.TEMP1,.ERF
 3      IN     T*DAMEN.TEMP2,.TEMP3,.TEMP4,.TEMP5B
 4      IN T*DAMEN.ISTEMP,.STARAY,.STARA2,.STARA3
 5      NOT  T*DAMEN.STARAY/PRINT
 6      NOT  T*DAMEN.STARA2/PRINT
 7      IN T*DAMEN.DERF
 8      IN T*DAMEN.HCOEF,.LCOEF,.BASALT,.LOWIMP
 9      IN T*DAMEN.DAME/DEPFUN,.DAFDAT/DEPFUN
10      IN T*DAMEN.PCOEF/124,.PFIT
11      IN   T*DAMEN.IFILE/IODR,.ENTRYS/IODR
12      IN T*DAMEN.HTFLUX,.R,.AERFC,.RESID
13      IN T*DAMEN.SPEMOD,.LITMOD
14      IN T*DAMEN.DEPLIT,.DEPHVY
15      END
```

```
=====  MAP/SPUTTER  =====

1.    @MAP*MAP.MAP    ,T*DAMEN.SPUTTER
2.    IN T*DAMEN.SPUTTR,.GETFIL/IODR,.ENTRYS/IODR,.IFILE/IODR
3.    END
```

```
      =====  MAP/EVAP  =====

1.    @MAP*MAP.MAP    ,T*DAMEN.EVAP
2.    IN T*DAMEN.GETFIL/IODR,.ENTRYS/IODR,.IFILE/IODR
3.    IN T*DAMEN.EVAP
4.    END
```

```
=====  LIST/ION  =====

 1.     @PRT,S  T*DAMEN.LIST/ION
 2.     @PRT,S  T*DAMEN.MAP/ION
 3.     @PRT,S  T*DAMEN.MAP/SPUTTER
 4.     @PRT,S  T*DAMEN.IMASTER
 5.     @PRT,S  T*DAMEN.SPECFLU
 6.     @PRT,S  T*DAMEN.SPEMOD
 7.     @PRT,S  T*DAMEN.LITMOD
 8.     @PRT,S  T*DAMEN.HCOEF
 9.     @PRT,S  T*DAMEN.LCOEF
10.     @PRT,S  T*DAMEN.DEPLIT
11.     @PRT,S  T*DAMEN.DEPHVY
12.     @PRT,S  T*DAMEN.LOWIMP
13.     @PRT,S  T*DAMEN.PCOEF/124
14.     @PRT,S  T*DAMEN.DAME/DEPFUN
15.     @ .     SEE DEPOSITION FUNCTION CREATION
16.     @PRT,S  T*DAMEN.DAFDAT/DEPFUN
17.     @ .     SEE DEPOSITION FUNCTION CREATION
18.     @PRT,S  T*DAMEN.PFIT
19.     @PRT,S  T*DAMEN.IDEPO
20.     @PRT,S  T*DAMEN.SPECTR
21.     @PRT,S  T*DAMEN.STARAY
22.     @ .     SEE PHOTON
23.     @PRT,S  T*DAMEN.STARA2
24.     @PRT,S  T*DAMEN.STARA3
25.     @PRT,S  T*DAMEN.BASALT
26.     @PRT,S  T*DAMEN.ITEMP
27.     @PRT,S  T*DAMEN.ISTEMP
28.     @PRT,S  T*DAMEN.TEMP1
29.     @PRT,S  T*DAMEN.TEMP2
30.     @PRT,S  T*DAMEN.TEMP3
31.     @PRT,S  T*DAMEN.TEMP4
32.     @PRT,S  T*DAMEN.TEMP5B
33.     @PRT,S  T*DAMEN.DERF
34.     @PRT,S  T*DAMEN.ERF
35.     @ .     SEE PHOTON
36.     @PRT,S  T*DAMEN.IFILE/IODR
37.     @PRT,S  T*DAMEN.ENTRYS/IODR
38.     @ .     SEE FILING
39.     @PRT,S  T*DAMEN.DCL/IODR
40.     @ .     SEE FILING
41.     @PRT,S  T*DAMEN.HTFLUX
42.     @PRT,S  T*DAMEN.R
43.     @PRT,S  T*DAMEN.AERFC
44.     @ .     SEE PHOTON
45.     @PRT,S  T*DAMEN.RESID
46.
47.     @PRT,S  T*DAMEN.SPUTTR
```

```
            =====  BASALT  =====

 1.             SUBROUTINE BASALT
 2.      C
 3.      C   A PROGRAM TO MODIFY A TIME BASE FROM ION TIME TO STANDARD TIMES
 4.      C
 5.             COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 6.            &          DEPX(50,200),DEL(200),X(200),FF(200)
 7.             COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
 8.             COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
 9.             DIMENSION BIGST(50)
10.             DO 666 I = 1,NST
11.             IF(T(1) .LE. ST(I)) GO TO 667
12.      666    CONTINUE
13.      667     ISTART = I
14.             DO 668 I = 1,NST
15.             IF(T(NE) .LE.ST(I)) GO TO 669
16.      668      CONTINUE
17.      669      ISTOP = I-1
18.             DO 555 I = 1,50
19.      555      BIGST(I) = 0.0
20.             DO 100 I = 1,NSX
21.             DO 90 K = 1,NE
22.             DEL(K) = DEPX(I,K)
23.      90     PR(K) = TEMP(I,K)
24.             DO 50 J = ISTART,ISTOP
25.             DO 10 K = 1,NE
26.             IF(ST(J) .LT. T(K)) GO TO 11
27.      10      CONTINUE
28.      11      K = K-1
29.             FRAC = (ST(J) - T(K))/(T(K+1) - T(K))
30.             DEPX(I,J) = DEL(K) + FRAC * (DEL(K+1) - DEL(K))
31.             TEMP(I,J) = PR(K) + FRAC * (PR(K+1) - PR(K))
32.             BIGST(I) = AMAX1(BIGST(I),TEMP(I,J))
33.             IF(TEMP(I,J) .LT. BIGST(I) ) TEMP(I,J) = BIGST(I)
34.      50      CONTINUE
35.      100     CONTINUE
36.             PRINT 200,NE,NST,NSX
37.      200      FORMAT(///,30X,' TIME BASE MODIFICATION WAS PERFORMED ',//,
38.           15X,I5,' VALUES OF TIME WERE PUT INTO ',I5,' STANDARD VALUES',
39.           2/,5X,'   FOR  ',I5,'  LOCATIONS',//)
40.      C
41.      C    PRINT OUT SOME VALUES
42.      C
43.             PRINT 502,(SX(I),I = 1,12)
44.             DO 500 J = ISTART,ISTOP
45.      500     PRINT 501,(ST(J),(DEPX(I,J),I = 1,12))
46.      501     FORMAT(1X,13E10.3)
47.      502     FORMAT(//,31X,'FUNCTION AT RESPECTIVE LOCATIONS',//,
48.           1    '       X =  ',1X,12E10.3,//'  TIME  ',/)
49.      C
50.      C   PUT ZEROS IN NONSTANDARD LOCATIONS
51.      C
52.             DO 700 I = 1,200
53.      700      T(I) = ST(I)
54.             INX = ISTART-1
55.             NE = NST
56.             NX = NSX
57.             DO 600 I = 1,INX
```

```
=====  BASALT  =====

58.               DO 600 J = 1,NSX
59.               DEPX(J,I) = 0.0
60.      600      TEMP(J,I) = 0.0
61.               INW = ISTOP + 1
62.               DO 800 I = INW,NST
63.               DO 800 J = 1,NSX
64.               DEPX(J,I) = 0.0
65.               TEMP(J,I) = BIGST(J)
66.      800       CONTINUE
67.               RETURN
68.                END
```

```
=====  DEPHVY  =====

      SUBROUTINE DEDXGS
      COMMON/MAT/IMODEL ,ALPHA,RHO,CP,X2,C,EO,A2,A3
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
       COMMON/HIEN/E1,E2,S1,S2,EM,BO,AO
       COMMON/NWSPEC/NMHIST,EHIST(20),AMP(20)
       COMMON/DEPTIM/STANFT(200),STANPW(200),STANE(200),SDEL(200),
     1 ISTART,ISTOP
       COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
       DIMENSION  EREF(200),W(200)
       COMMON/NECESS/VSTAR,COEF,FACT,REND,DLEND,TEND,GAMMA,RLIM,
     1RINS,CONINS,SMSIG,SIGINS,FLXINS,VSPINS,EINS,ESPINS
       INTEGER  FINALJ
       EQUIVALENCE (C,AO),(EO,A1)
C
C     DEFINE  FUNCTIONS
C
C    STANDARD DEVIATION AND RANGE AT END OF RANGE
C
         RANGE(V)   = CONST * V **(1./(1.-AK))
         DELVR(V) = CONDEL* EXP(-V/BDEL) * RANGE(V)
C
C    DIFFUSION COEFFICIENT FOR TIME AT STOP
C   AND STANDARD DEVIATION AT ANY OTHER TIME
C
         SIGLOC(T,G) =  G * SQRT(2.*T)
          DCOF(DP,T) = DP/SQRT(2.*T)
C
C    VELOCITY FROM ENERGY AND INVERSE  KEV AND CM/SEC
C
         ENER(V) = 1.036E-15* A * V**2/2.
         VEL(EN)  = 4.39E+7 * SQRT(EN /A)
C
C     TIME AND VELOCITY AT A GIVEN INTERMEDIATE RANGE
C
          TIME(Y) = COEF*(1. - (1. - Y/CONST * FACT)**AK)
         VELOC(Z) = (VSTAR**(1./(1.-AK)) - Z/CONST)**(1.- AK)
C
C  FUNCTIONS FOR TIME SCAN
C
         VELT(TM) = (VSTAR**P1 - AK*TM/CONST)**PRP
         RV(VL) = CONST * (VSTAR**PN - VL**PN)
         GAUS(A,B,C) = TOT/SQRT(6.2832)/B * EXP( -.500 * ((A-C)/B)**2)
C
C    DE/DX AS A FUNCTION OF VELOCITY UNITS ARE KEV/CM
C
         DEDX(VION) = A/9.636E+14*(1.-AK)/CONST*VION**((2.*AK-1.)/(AK-1.))
C
C********************* END OF FUNCTIONS ***********************************
C
C
C     DETERMINE VELOCTIY - RANGE CONSTANTS FROM ENERGY RANGE DATA
C
C
C   INPUT
```

```
            =====  DEPHVY  =====

58.      C
59.      377       FORMAT(A4)
60.               READ ,E1,R1,SIG1,E2,R2,SIG2,REFP,REFT
61.               V1 = VEL(E1)
62.               V2 = VEL(E2)
63.               AK = 1. - (ALOG(V1/V2)/ALOG(R1/R2))
64.               CONST = R1/V1**(1./(1.-AK))
65.                PDEL1 = SIG1/R1
66.                PDEL2 = SIG2/R2
67.                BDEL = ALOG(PDEL1/PDEL2)/(V2 - V1)
68.               BDEL = 1./BDEL
69.                CONDEL = PDEL1 * EXP(V1/BDEL)
70.               P0 = 1./(AK-1.)
71.               P1 = AK/(1.-AK)
72.               PRP = 1./P1
73.               PN = -P0
74.               PDTV = (2.*AK - 1.)/(1. - AK)
75.               PRINT 50 ,A,E1,V1,R1,SIG1,E2,V2,R2,SIG2,AK,CONST,BDEL,CONDEL
76.      50        FORMAT(5X,'ION MASS OF   ',F6.2,'  AMU',/
77.            1 ,3X,'    ENERGY       VELOCITY       RANGE      SIGMA ',//
78.            2 ,3X,4E10.3,/,3X,4E10.3,/, '    AK =  ',F7.4,6X,'C = ',E10.4,//,
79.            3 ' BDEL = ',E10.4,' CONDEL = ',E10.4,//)
80.      C
81.      C    READ BUFFER CONDITIONS
82.      C
83.               READ , GASP,GAST
84.      C
85.      C    SET UP RADIUS ARRAY
86.      C
87.               READ , LAYOUT,RINNER,ROUTER,NSPOTS
88.               X(1) = RINNER
89.                X(NSPOTS) = ROUTER
90.               DELXD = (ROUTER-RINNER)/(NSPOTS - 1)
91.               DO 496 MJ = 2,NSPOTS
92.               EEE = MJ
93.               DDD = NSPOTS
94.               ZZZ = EEE/DDD
95.               IF(LAYOUT .EQ. 0) X(MJ) = X(1) * (X(NSPOTS)/X(1))**ZZZ
96.               IF(LAYOUT .EQ. 1) X(MJ) = X(MJ-1) + DELXD
97.      496       CONTINUE
98.               CONST = CONST *REFP/GASP*GAST/REFT
99.      C
100.     C     SET UP LIMITS FOR TIME AND ENERGY
101.     C
102.              DO 9643 IMDEP = 1,NSPOTS
103.              RBUF = X(IMDEP)
104.              CALL SETUP(EMAX)
105.              PRINT 442,REND,DLEND,TEND
106.     442       FORMAT(//,10X,'RANGE OF MAXIMUM  ENERGY IS  ',E12.3,'  CM',/
107.           1,10X,'SIGMA AT THAT RANGE IS  ',E12.3,'  CM',/
108.           2,10X,'TIME TO STOP IS  ',E12.3,'  SEC',//)
109.              IF(RBUF .GE. REND ) GO TO 100
110.              SS2 = TIME(RBUF)
111.              VELMAX = VELOC(RBUF)
112.              ESCMAX = ENER(VELMAX)
113.              E(1) = EMAX
114.              DE = (EMAX -EMIN)/(NE-1)
```

```
      =====  DEPHVY  =====

115.            DO 438 I = 1,NE
116.            FF(I) = SPECTR(E(I))
117.            E(I+1) = E(I) - DE
118.     438      CONTINUE
119.            ELIM = EMIN
120.     C
121.     C   LOWER LIMITS
122.     C
123.            FINALJ= NE
124.            CALL SETUP(EMIN)
125.            PRINT 443,REND,DLEND,TEND
126.     443     FORMAT(//,10X,'RANGE OF MINIMUM  ENERGY IS  ',E12.3,'  CM',/
127.          1,10X,'SIGMA AT THAT RANGE IS  ',E12.3,'  CM',/
128.          2,10X,'TIME TO STOP IS  ',E12.3,'  SEC',//)
129.            IF(RBUF .GE. REND) GO TO 77
130.            GO TO 99
131.     77       DO 8 I = 1,NE
132.            CALL SETUP(E(NE +1 -I))
133.            IF( REND .GT. RBUF) GO TO 88
134.     8      CONTINUE
135.     88      ELIM = E(NE +1 -I)
136.            FINALJ = NE-I + 1
137.     99        CONTINUE
138.            SS1 = TIME(RBUF)
139.            VELMIN = VELOC(RBUF)
140.            ESCMIN = ENER(VELMIN)
141.            TLION = FL * 4. * 3.14159 * (R*100.)**2
142.     C
143.     C
144.     C    PRINT OUT INCIDENT VALUES AND TRANSMITTED LIMITS
145.     C
146.            IF(IMDEP .GE.2) GO TO 741
147.            PRINT 19, ISPEC,EMN,SIG,EMIN,EMAX,NE,TLION,FL,R,A
148.     19      FORMAT(///'              INPUT TO THE  SPEMOD CALCULATION ',
149.          1 ///,5X,'ISPEC =   ',I5,/,
150.          25X,'CHAR ENERGY = ',E10.3,/,
151.          35X,'SIGMA = ',E10.3,/,5X,'EMIN = ',E10.3,/,
152.          55X,'EMAX = ',E10.3,/,5X,'NE = ',I5,/,
153.          75X,'TOTAL IONS = ',E10.3,/,5X,'WALL FLUENCE = ',E10.3,
154.          8 /,5X,'WALL RADIUS = ',F10.3,/,5X,'ION MASS =',F10.3,///)
155.     741     CONTINUE
156.            PRINT 747,ELIM,EMAX,RBUF,ESCMAX,ESCMIN,SS2,SS1
157.     747     FORMAT('  THAT PORTION OF INCIDENT SPECTRUM TRANSMITTED IS FROM',
158.          1//,16X,F12.2,'  KEV   TO  ',F12.2,'  KEV',//,
159.          2'   MAXIMUM ENERGY AT ',F12.3,'  CM  IS',E12.3,'  KEV
160.          3  MINIMUN IS  ',E12.3,'   KEV',//,'   TIMES OF ARRIVAL ARE FROM',
161.          4 E12.3,'  SECONDS   TO',E12.3,'  SECONDS',//)
162.     C
163.            TLION = FL * 4. * 3.14159 * (R*100.)**2
164.     C
165.     C
166.     C   PRINT DEFINING PARAMETERS
167.             PRINT 707, RBUF,GASP,GAST,SS2,SS1,TLION,ELIM
168.     707     FORMAT(//,'  RADIUS     GASP      GAST   START-TIME STOP-TIME
169.          1 TOTAL IONS  E-LIMIT ',//,7E10.3,//)
170.     C
171.     C
```

```
      C     ESTIMATE FIRST TIME STEP
      C
              CALL SETUP(E(1))
              TTT1 = TIME(RBUF)
              VVV1 = VELOC(RBUF)
              EEE1 = ENER(VVV1)
              CALL SETUP(E(2))
              TTT2 = TIME(RBUF)
              VVV2 = VELOC(RBUF)
              EEE2 = ENER(VVV2)
              DT = TTT2 - TTT1
              DES = EEE1 - EEE2
      C
      C
      C
      C    GET CONTRIBUTION A TIME J FOR EACH VALUE OF INCIDENT SPECTRUM
      C
              DO 40 I = 1,NE
      40       TEMP(IMDEP,I) = 0.0
              DO 20 J = 1,FINALJ
              CALL SETUP(E(J))
              TOT = FF(J) * DE
      C
      C
      C
      C     SET UP FLUX AS A FUNCTION OF TIME  J IS THE TIME INDEX
      C
              VINS = VELOC(RBUF)
              EREF(J) = ENER(VINS)
               T(J) = TIME(RBUF)
              IF ( J .EQ. 1) GO TO 6543
               DT = T(J) - T(J-1)
              DES = EREF(J-1) - EREF(J)
      6543       CONTINUE
              FT(J) = FF(J) * DE /DT
              FF(J) = FF(J) * DE/DES
              DEPX(IMDEP,J) =  FT(J) * (R*100./RBUF)**2 * 1.602E-16
            1   * DEDX(VINS)
              IF ( J .EQ. 1) GO TO 6555
              TEMP(IMDEP,J) = TEMP(IMDEP,J-1) + DEPX(IMDEP,J) * DT
      6555       CONTINUE
      C
      C  UNITS HAVE BEEN CONVERTED TO JOULES/CM3
      C
      20       CONTINUE
      30        CONTINUE
      C
      42       CONTINUE
      C
      C    PRINT THE OUTPUT
      C
              PRINT 7, X(IMDEP)
      7       FORMAT(///,20X,'ENERGY DEPOSITION IN BUFFER LAYER',//
            1,'    X = ',E10.3,//,'          TIME      J/CM3/SEC    J/CM3',
            2'  ENERGY IN      ENERGY OUT      FLUX    SPECTRUM',//)
              DO 899 I = 1,FINALJ
      C         PRINT 9,I, T(I),DEPX(IMDEP,I),TEMP(IMDEP,I)
```

```
      =====  DEPHVY  =====

C     1   ,E(I),EREF(I),FT(I),FF(I)
899     CONTINUE
9       FORMAT(I4,E12.3,6E10.3)
392     READ (5,144,END=9908)INST
144     FORMAT(A6)
        IF( INST .EQ. 'I-1FIL') CALL IFILE1
        IF( INST .EQ. 'I-2FIL') CALL IFILE2
        GO TO 392
9908    CONTINUE
9643    CONTINUE
C C
C   CALCULATE TOTAL DEPOSITED ENEERGY
C
        PRINT 2233
2233      FORMAT(//,'  I    NE   LOCATION    DEP(-1)    DEP   AVDEP   VOL
     1UME    ENERGY  ',//)
        TOTABS = 0.0
        NECC = NE
        DO 412 I = 2,NSPOTS
        DVOL = 4. * 3.141592/3.  * (X(I) **3 - X(I-1) **3)
       IF(TEMP(I-1,NE) .GT. 1.E-36 .AND. TEMP(I,NE) .GT. 1.E-36)GO TO 291
        NECC = NE
293     NE = NE-1
       IF(TEMP(I-1,NE) .GT. 1.E-36 .AND. TEMP(I,NE) .GT. 1.E-36)GO TO 291
        GO TO 293
291     AVDEP = (TEMP(I-1,NE) + TEMP(I,NE))/2.
        TOTABS = TOTABS + AVDEP * DVOL
        PRINT 3355 , I,NE,X(I),TEMP(I-1,NE), TEMP(I,NE),AVDEP,DVOL,TOTABS
          NE = NECC
3355     FORMAT(2I5,6E10.4)
412      CONTINUE
102     RETURN
100       CONTINUE
        PRINT 737
737       FORMAT('  RADIUS EXCEEDED THE MAXIMUM RANGE OF ION',//)
        STOP DONE
C
C
C
C
        SUBROUTINE SETUP(E)
        VSTAR = VEL(E)
238       FORMAT( ' GOT TO HERE IN SETUP')
        COEF = CONST/AK * VSTAR**P1
        FACT = VSTAR**P0
        REND = RANGE(VSTAR)
        DLEND = DELVR(VSTAR)
        TEND = TIME(REND)
        GAMMA = DCOF(DLEND,TEND)
        RLIM = REND + 2.*DLEND
        RETURN
C
C
C
C
        END
```

```
            =====  DEPLIT  =====

 1.           SUBROUTINE GSDPLT
 2.           COMMON/MAT/IMODEL ,ALPHA,RHO,CP,X2,C,E0,A2,A3
 3.           COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 4.          &          DEPX(50,200),DEL(200),X(200),FF(200)
 5.           COMMON/VAR/DT,DX,BB,NX
 6.           COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
 7.            COMMON/HIEN/ZZ1,ZZ2,ZZ3,ZZ4,ZZ5,ZZ6,ZZ7
 8.            COMMON/NWSPEC/NMHIST,EHIST(20),AMP(20)
 9.            EQUIVALENCE (C,A0),(E0,A1)
10.            COMMON/STOP/E0,S0,E1,S1,E2,S2,E3,S3,E4,S4,SMAX,B0,A0,A00,B00
11.          1  ,EINT,SINT
12.            COMMON/PLYCOF/XM(200),XH(200),XL(200),COMATX(200,5,3),QQ1(200)
13.          &  ,INCID(200)
14.            COMMON/INTEQ/EFINAL,NEWXL,DI(100)
15.            VEL(EN)  = 4.39E+7 * SQRT(EN /A)
16.      C       **   INPUT
17.      729     FORMAT(A4)
18.            READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
19.            READ , REFP,REFT
20.            READ ,RBUF,GASP,GAST,IPRI
21.      C  NEW  PARAMETERS FOR REGION 1
22.      C
23.      C
24.            A00 = S0**2/(2.*S0 - S1)
25.           B00 = -ALOG(1. - S0/A00)/E0
26.      C
27.      C
28.
29.
30.      C       ** CONSTANTS ARE CALCULATED
31.            B0=(E3-E2)/ALOG(S2/S3)
32.            A0=S2*EXP(E2/B0)
33.      C
34.      C     FIND THE INTERSECTION OF CURVE 2 AND CURVE 3
35.      C
36.            SMDIF = 1.E+6
37.            ETR = E1
38.            DTE = (E2-E1)/100.
39.            DO 427  I = 1,100
40.            FORM2 = A00 *(1. - EXP(-B00 *ETR))
41.            FORM3 = A0 * EXP(-ETR/B0)
42.            DIFT = ABS(FORM3 - FORM2)
43.            CSM = SMDIF
44.           SMDIF = AMIN1(SMDIF,DIFT)
45.            CDIF = CSM - SMDIF
46.            IF(CDIF .LE. 0.) GO TO 428
47.            ETR = ETR + DTE
48.      427     CONTINUE
49.      428     CONTINUE
50.            EINT = ETR
51.            SINT = A0*EXP(-EINT/B0)
52.      C
53.      C      PRINT THE IN PUT AND THE INTERSECTION POINTS
54.      C
55.            PRINT 111,E0,S0,E1,S1,E2,S2,E3,S3,EINT,SINT,SMAX
56.      111     FORMAT(///,8X,'E0 = ',E10.3,5X,'S0 = ',E10.3,/,
57.           1             8X,'E1 = ',E10.3,5X,'S1 = ',E10.3,/,
```

```
     2              8X,'E2 = ',E10.3,5X,'S2 = ',E10.3,/,
     3              8X,'E3 = ',E10.3,5X,'S3 = ',E10.3,/,
     3              6X,'EINT = ',E10.3,3X,'SINT = ',E10.3,/,
     48X,'SMAX = ',E10.3,///)
C
      TLION = 4. * 3.14159 * (R*100.)**2 * FL
C
C     CONVERT FROM KEV/MICRON TO KEV/CM
C
      S0 = S0 * GASP/REFP *REFT/GAST*1.E+4
      S1 = S1 * GASP/REFP *REFT/GAST*1.E+4
      S2 = S2 * GASP/REFP *REFT/GAST*1.E+4
      S3 = S3 * GASP/REFP *REFT/GAST*1.E+4
      SINT = SINT * GASP/REFP *REFT/GAST*1.E+4
      A0 = A0 * GASP/REFP *REFT/GAST*1.E+4
      A00 = A00 * GASP/REFP *REFT/GAST*1.E+4
      PRINT 510,REFP,REFT,GASP,GAST
510    FORMAT(//,8X,'REFERENCE VALUES: PRESSURE',E10.3,'  TEMP',E10.3,
     1      //,8X,'  INPUT   VALUES: PRESSURE',E10.3,'  TEMP',E10.3,//)
C
C    SET UP RADIUS ARRAY
C
      READ , LAYOUT,RINNER,ROUTER,NSPOTS
      X(1) = RINNER
       X(NSPOTS) = ROUTER
      DELXD = (ROUTER-RINNER)/(NSPOTS - 1)
      DO 496 MJ = 2,NSPOTS
      EEE = MJ
      DDD = NSPOTS
      ZZZ = EEE/DDD
      IF(LAYOUT .EQ. 0) X(MJ) = X(1) * (X(NSPOTS)/X(1))**ZZZ
      IF(LAYOUT .EQ. 1) X(MJ) = X(MJ-1) + DELXD
496    CONTINUE
      NX = NSPOTS
      E(1) = EMAX
      DE = ( EMAX - EMIN)/(NE-1)
C
C    ESTABLISH AND NORMALIZE THE INCIDENT SPECTRUM
C
      AF = .02*FL
      TONORM = 0.0
      DO 600 I = 1,NE
      E(I+1) = E(I) - DE
      FF(I) = SPECTR(E(I))
      PR(I) = FF(I)
      TONORM = TONORM + FF(I) * DE
600     CONTINUE
      DIFNOM = ABS(TONORM - FL)
      IF( DIFNOM .LT. AF ) GO TO 620
      RAT = FL/TONORM
      TONORM = 0.0
      DO 610 I = 1,NE
      FF(I) = FF(I) * RAT
      TONORM = FF(I) * DE + TONORM
610     CONTINUE
620    CONTINUE
      PRINT 605, R, TONORM,FL,TLION
```

```
115.      605      FORMAT(5X,'INTEGRATED NORMALIZED SPECTRUM AT',E10.3,'  METERS',
116.           1/,5X, 'CONTAINS A TOTAL OF ',E10.3,'  PARTICLES/CM2',
117.           2/,5X,'INPUT FLUENCE WAS  ',E10.3,'   FOR A TOTAL OF  ',E10.3,
118.           3'  IONS',//)
119.      C
120.      C    SET UP COEFFICIENTS
121.      C
122.             DO 7333 I = 1,NE
123.             CALL PCOEF(E(I),I)
124.      7333    CONTINUE
125.      C
126.      C
127.      C
128.             DO 9643 IMDEP = 1,NSPOTS
129.             RBUF = X(IMDEP)
130.             ILAST = NE
131.             IF(INST2 .NE. 'RANG') GO TO 621
132.             PRINT 622
133.      622      FORMAT(4X,'  E(IN)     E(OUT)     XM         XH         XL         RBUF
134.           1       TIME      INCID ',//)
135.      621      CONTINUE
136.             DO 100 I = 1,NE
137.             TINT1 = 0.0
138.             TINT2 = 0.0
139.             TINT3 = 0.0
140.             IF(RBUF .LT. XM(I)) GO TO 30
141.             IF(RBUF .LT. XH(I)) GO TO 20
142.             IF(RBUF .LT. XL(I)) GO TO 10
143.             GO TO 101
144.      C
145.      C    RBUF IS IN REGION 3
146.      C
147.      30      TINT3 = TIME(0.,RBUF,E(I),3)
148.             GO TO 34
149.      C
150.      C    RBUF IS IN REGION 2
151.      C
152.      20      IF(INCID(I) .EQ. 2) GO TO 22
153.             TINT3 = TIME(0.,XM(I),E(I),3)
154.             TINT2 = TIME(XM(I),RBUF,EINT,2)
155.               GO TO 24
156.      22       TINT2 = TIME(0.,RBUF,E(I),2)
157.      24       GO TO 34
158.      C
159.      C     RBUF IS REGION 1
160.      10       CONTINUE
161.             IMPET = INCID(I)
162.             GO TO (11,12,13),IMPET
163.      11       TINT1 = TIME(0.,RBUF,E(I),1)
164.             GO TO 34
165.      12      TINT2 = TIME(0.,XH(I),E(I),2)
166.             TINT1 = TIME(XH(I),RBUF,E0,1)
167.             GO TO 34
168.      13      TINT3 = TIME(0.,XM(I),E(I),3)
169.             TINT2 = TIME(XM(I),XH(I),EINT,2)
170.             TINT1 = TIME(XH(I),RBUF,E0,1)
171.      34       CONTINUE
```

```
172.              IF(NEWXL .EQ. 1 ) GO TO 101
173.              DEL(I) = EFINAL
174.              T(I) = TINT3 + TINT2 + TINT1
175.       215       FORMAT(I4,7E9.3,4X,I4)
176.       100       CONTINUE
177.              GO TO 102
178.       101      ILAST = I -1
179.       102      CONTINUE
180.              IF(ILAST .LE. 0 ) GO TO 1009
181.       C
182.              SPHFAC = (R*100./RBUF) ** 2
183.              SUMEFT = 0.0
184.              DO 200 I = 1,ILAST
185.              IF(I.EQ.1) GO TO 201
186.              IF(I .EQ. ILAST) GO TO 296
187.              DT = (T(I+1) - T(I-1))/2.
188.              DEPS = ( DEL(I-1) - DEL(I+1))/2.
189.              GO TO 297
190.       201      DT = T(2) - T(1)
191.              DEPS = DEL(1) - DEL(2)
192.              GO TO 297
193.       296     DT = T(I) - T(I-1)
194.              DEPS = DEL(I-1) - DEL(I)
195.       297      CONTINUE
196.              FT(I) = FF(I) * DE/DT * SPHFAC
197.              PR(I) = FF(I) * DE/DEPS  * SPHFAC
198.              SUMEFT = SUMEFT + FT(I) * DT
199.              FRACFL = SUMEFT/FL/SPHFAC
200.       200      CONTINUE
201.              PRINT 216,RBUF,R,SUMEFT,FRACFL
202.       216     FORMAT(///,10X,'POSITION IS',E12.3,'  WALL RADIUS IS',E12.3,
203.            1  //,10X,'TOTAL FLUX IS',E12.3,'  FRACTION TRANS IS',E12.3, //)
204.       C
205.       C
206.       C    CALCULATE THE DEPOSITION
207.       C
208.              DO 110 I = 1,ILAST
209.              J = IMDEP
210.              IF(X(J) .LE. 1.E-8) GO TO 170
211.              IF(X(J) .LT. XM(I)) GO TO 140
212.              IF(X(J) .LT. XH(I)) GO TO 150
213.              IF(X(J) .LT. XL(I))   GO TO 160
214.              DEPX(J,I) = 0.0
215.              GO TO 180
216.       140    N = 3
217.              DEPX(J,I) = EVAL(N)
218.              GO TO 180
219.       150     N = 2
220.              DEPX(J,I) = EVAL(N)
221.              GO TO 180
222.       160     N = 1
223.              DEPX(J,I) = EVAL(N)
224.              GO TO 180
225.       170    MM = INCID(I)
226.              DEPX(J,I) = COMATX(I,1,MM)
227.       180     DEPX(J,I) = DEPX(J,I) * FT(I) * 1.6E-16
228.              IF(DEPX(J,I) .LE. 0.0) DEPX(J,I) = 0.0
```

```
      =====  DEPLIT  =====

229.      C
230.      C
231.            DI(J) = DI(J) + DEPX(J,I) * DT
232.            TEMP(J,I) = DI(J)
233.      110    CONTINUE
234.      CC
235.      C
236.      C
237.      C    PRINT THE OUTPUT
238.      C
239.            PRINT 7, X(IMDEP)
240.      7     FORMAT(///,20X,'ENERGY DEPOSITION IN BUFFER LAYER',//
241.           1,'    X = ',E10.3,//,'  TIME J/CM3/SEC J/CM3',
242.           2'ENER IN ENER OUT  FLUX  SPECTRUM',
243.           3'    XM       XH       XL',//)
244.            DO 899 I = 1,ILAST
245.             PRINT 9, T(I),DEPX(IMDEP,I),TEMP(IMDEP,I)
246.           1   ,E(I),DEL(I),FT(I),PR(I),XM(I),XH(I),XL(I)
247.      899    CONTINUE
248.      9      FORMAT(10E8.3)
249.      392    READ (5,144,END=9908)INST
250.      144    FORMAT(A6)
251.            IF( INST .EQ. 'I-1FIL') CALL IFILE1
252.            IF( INST .EQ. 'I-2FIL') CALL IFILE2
253.            GO TO 392
254.      9908   CONTINUE
255.            GO TO 9643
256.      1009     CONTINUE
257.            PRINT 737
258.      737     FORMAT('  RADIUS EXCEEDED THE MAXIMUM RANGE OF ION',//)
259.      9643   CONTINUE
260.      C C
261.      C   CALCULATE TOTAL DEPOSITED ENEERGY
262.      C
263.            PRINT 2233
264.      2233    FORMAT(//,'  I    NE   LOCATION    DEP(-1)    DEP   AVDEP   VOL
265.           1UME    ENERGY  ',//)
266.            TOTABS = 0.0
267.            NECC = NE
268.            DO 412 I = 2,NSPOTS
269.            DVOL = 4. * 3.141592/3.  * (X(I) **3 - X(I-1) **3)
270.           IF(TEMP(I-1,NE) .GT. 1.E-36 .AND. TEMP(I,NE) .GT. 1.E-36)GO TO 291
271.            NECC = NE
272.      293   NE = NE-1
273.           IF(TEMP(I-1,NE) .GT. 1.E-36 .AND. TEMP(I,NE) .GT. 1.E-36)GO TO 291
274.            GO TO 293
275.      291   AVDEP = (TEMP(I-1,NE) + TEMP(I,NE))/2.
276.            TOTABS = TOTABS + AVDEP * DVOL
277.            PRINT 3355 , I,NE,X(I),TEMP(I-1,NE), TEMP(I,NE),AVDEP,DVOL,TOTABS
278.              NE = NECC
279.      3355   FORMAT(2I5,6E10.4)
280.      412    CONTINUE
281.      1029   RETURN
282.      C
283.      C
284.      C
285.      C
```

```
=====  DEPLIT  =====

      FUNCTION TIME(X1,X2,E,I)
      NEWXL = 0
      IF(I .EQ. 1) GO TO 10
      V1 = VEL(E)
      XINTR = X2 - X1
      EFINAL = ENERGY(E,XINTR,I)
      V2 = VEL(EFINAL)
      XMID = (X1 + X2)/2.
      XINMID = XMID-X1
      EMID = ENERGY(E,XINMID,I)
      VMID = VEL(EMID)
       TIME = AVTIME(X1,XMID,X2,V1,VMID,V2)
       RETURN
C
10      CONTINUE
      IQ = 0
      XINTR = X2 - X1
      EL = ENERGY(E,XINTR,I)
      ELOW = 2.*A
      IF(EL .GT. ELOW) GO TO 4
      ELA = EL
      EL = ELOW
       IQ = 1
4     TIME = SQRT(A *E0)/S0  *ALOG(E/EL)/4.39E+7
      EFINAL = EL
      IF(IQ .NE. 1) RETURN
      XE2 = 2. * SQRT(E0)/S0 * ( SQRT(E) - SQRT(2.)) + X1
      RNEW = 2. / S0 /SQRT(2./E0)
      NEWXL = 0
      IF(X2 .LT. RNEW )  GO TO 423
      NEWXL = 1
       RETURN
423     CONTINUE
      EFINAL = 2. - S0*SQRT(2./E0) * (X2 - XE2)
      AV2 = (VEL(EL) + VEL(EFINAL))/2.
      ADTIM = (X2 - XE2)/AV2
      TIME = TIME + ADTIM
      RETURN
C
C
      FUNCTION AVTIME(Y1,Y2,Y3,W1,W2,W3)
      BFIRST = (W2 -W1)/(Y2-Y1)
      BLAST = (W3 -W2)/(Y3-Y2)
      TFIRST =  ALOG(W2/W1)/BFIRST
      TLAST = ALOG(W3/W2)/BLAST
      AVTIME = TFIRST + TLAST
      RETURN
C
C
      FUNCTION ENERGY(ESTAR,SSS,M)
      GO TO (1,2,3),M
1     ENERGY = ESTAR - S0*SSS*SQRT(ESTAR/E0) + (S0*SSS)**2/4./E0
      GO TO 5
2     ENERGY = ALOG(1. + EXP(B00*(QQ1(I) - A00*SSS)))/B00
      GO TO 5
3     ENERGY = B0*ALOG(EXP(ESTAR/B0)- A0*SSS/B0)
5     CONTINUE
```

```
=====  DEPLIT  =====

343.              RETURN
344.        C
345.        C
346.              FUNCTION EVAL(N)
347.              EVAL = 0.0
348.               DO 766 IJK = 1,5
349.        766   EVAL = EVAL + COMATX(I,IJK,N) * X(J) **(IJK-1)
350.              RETURN
351.              END
```

```
=====  DERF  =====

      REAL*8 FUNCTION DERF(Y)
         REAL*8 Y,YY, T,A1,A2,A3,A4,A5
      DATA PI/3.1415927  /,P/.3275911/
       YY = Y
       Y =  DABS(Y)
      T=1./(1.+P*Y)
      A1=.254829592  *T
      A2=-.284496736  *T**2
      A3=1.421413741  *T**3
      A4=-1.453152027  *T**4
      A5=1.061405427  *T**5
      DERF=1.D0-(A1+A2+A3+A4+A5)* DEXP(-Y**2)
      IF(Y.LT..1D-4)DERF=2.*(Y-Y**3+Y**5/10.)/ SQRT(PI)
       Y =YY
       IF (YY.LE.0.D0)DERF = -DERF
      RETURN
      END
```

```
            =====  EVAP  =====

 1.           COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 2.          &          DEPX(50,200),DEL(200),X(200),FF(200)
 3.           COMMON/VAR/DT,DX,BB,NX
 4.           COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
 5.            COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
 6.           COMMON/OUT/NT,TA(250),NY,Y(250)
 7.            INTEGER BINT
 8.     C
 9.     C   OPEN THE IODR FILE
10.     C
11.           CALL REOPEN
12.           READ , BINT
13.     C
14.     C   GET SURFACE TEMPERATURE
15.     C
16.           CALL GETFIL(BINT,'RDFIL3')
17.     C
18.           PRINT 71
19.     71     FORMAT(//,'        TIME  SURFACE TEMPERATURE',//)
20.           DO 95 I = 1,NST
21.            X(I) = TEMP(1,I)
22.     95     PRINT 1000,I,TA(I),X(I)
23.     C
24.     C      READ EVAPORATION PARAMETERS
25.     C   STIPR = STICKING PROB  AMASS = WALL MASS TAMB = TEMP AMBIENT
26.     C
27.           READ , STIPR,AMASS,TAMB
28.     C
29.     C    READ VAPOR PRESSURE DATA
30.     C     PZERO = TORR    HSUB = SUBLIMATION ENERGY EV/ATOM
31.     C
32.           READ , PZERO,HSUB
33.     C
34.     C   ZERO OUT STORAGE
35.     C
36.           DO 100 I = 1,NST
37.           DEPX(1,I) = 0.0
38.     100    DEPX(2,I) = 0.0
39.     C
40.     C   EVAP RATES ARE IN DEPX(1,I)
41.     C   EVAP AMOUNTS ARE IN DEPX(2,I)  ATOMS/CM2
42.     C
43.           DO 200 I = 1,NST
44.           ACTEM = TAMB + TEMP(1,I)
45.           DEPX(1,I) = STIPR * 3.5E+22/SQRT(AMASS)*VAPRES(ACTEM)/SQRT(ACTEM)
46.     200    CONTINUE
47.     C
48.     C   INTEGRATE
49.     C
50.           DO 300 I = 2,NST
51.           DELTIM = TA(I) - TA(I-1)
52.           DEPX(2,I) = DEPX(2,I-1) + DEPX(1,I-1) * DELTIM
53.     300    CONTINUE
54.     C
55.     C   PRINT THE OUTPUT
56.     C
57.           PRINT 400, STIPR,AMASS,TAMB,PZERO,HSUB
```

```
            =====  EVAP  =====

400      FORMAT(///,10X,'EVAPORATION  RATE DATA',///,5X,
     1'INPUT PATAMETERS',//'   STICKING COEF    WALL MASS     AMB TEMP
     2      PZERO     HSUB',//,5(3X,E9.3,3X),// 5X,
     3'   TIME     EVAP RATE  TOTAL (A/CM2)  TEMP   VAPOR PRES',//)
        NE = NST
        NX = NSX
        DO 500 I = 1,NST
        T(I) = TA(I)
        ACTEM = TEMP(1,I) + TAMB
        VPRS = VAPRES(ACTEM)
        PRINT 1000, I,TA(I),DEPX(1,I),DEPX(2,I),ACTEM,VPRS
        DEPX(3,I) = ACTEM
        DEPX(4,I) = VPRS
500      CONTINUE
1000       FORMAT(I5,5E10.4)
C
C  READ INSTRUCTIONS FOR FILING
C
        READ 234,INSTF
234    FORMAT(A6)
        IF(INSTF .NE.'I-1FIL') GO TO 266
        CALL REOPEN
        CALL IFILE1
266      CONTINUE
        STOP DONE
C
C
C
        FUNCTION VAPRES(TTTT)
        VAPRES = PZERO * EXP(-HSUB/8.617E-5/TTTT)
        RETURN
C
C
        END
```

```
=====  HCOEF  =====

      SUBROUTINE HCOEF
      COMMON/PLYCOF/XM(200),XH(200),XL(200),COMATX(200,5,3),QQ1(200)
     &  ,INCID(200)
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
      COMMON/XFL/ C(20),NC
C
C    THIS SUBROUTINE ESTABLISHES THE COEFFICIENT MATRIX FOR
C    THE HEAVY ION SPECTRUM FOR BOTH NUCLEAR AND ELECTRONIC LOSS
C
      READ ,LOCDFN,LOCDFE
C
C  NUCLEAR GOES IN TO REGION 1 AND 2 ELEC GOES INTO 3
C
C ***************NUCLEAR  *********************************
C
      CALL DAFDAT(LOCDFN)
       DO 100 I = 1,NE
      INCID(I) = 3
      CALL DAME(E(I))
      XH(I) = C(1)
      XL(I) = C(2)
      PRINT ,I,XH(I),XL(I) ,E(I)
      DO 50 IJK = 1,5
50     COMATX(I,IJK,2) = C(2 + IJK)
C
C  CONVERT OVER TO ORIGIN BASIS
C
      V1 = C(8)
      V2 = C(9)
      V3 = C(10)
      V4 = C(11)
      V5 = C(12)
      A9 = C(1)
      COMATX(I,1,1) = V1 - V2 * A9 + V3 * A9**2 - V4 * A9**3 +V5 * A9**4
      COMATX(I,2,1) = V2 - 2.*V3*A9 + 3.*V4*A9**2 - 4.*V5*A9**3
      COMATX(I,3,1) = V3 - 3.*V4*A9 + 6.*V5*A9**2
      COMATX(I,4,1) = V4 - 4.*V5*A9
      COMATX(I,5,1) = V5
100    CONTINUE
C
C  **********************ELECTRONIC ***************************
C
      CALL DAFDAT(LOCDFE)
      DO 200 I = 1,NE
      CALL DAME(E(I))
      XM(I) = C(2)
      DO 6 IJK = 1,5
6     COMATX(I,IJK,3) = C(2 + IJK)
200    CONTINUE
      RETURN
C
C
      ENTRY COMCOF
C
C  A ROUTINE TO ADD TOGETHER THE ELECTRONIC AND NUCLEAR
```

```
      =====  HCOEF  =====

58.      C  COEFFICIENTS
59.      C
60.              DO 300 I = 1,NE
61.              IF(XM(I) .LT. XH(I)) GO TO 320
62.              IF(XM(I) .LT. XL(I)) GO TO 310
63.              XMM = XM(I)
64.              XM(I) = XH(I)
65.              XH(I) = XL(I)
66.              XL(I) = XMM
67.              DO 305 IJK = 1,5
68.              HOLD = COMATX(I,IJK,3)
69.              COMATX(I,IJK,3) = HOLD + COMATX(I,IJK,2)
70.              COMATX(I,IJK,2) = HOLD + COMATX(I,IJK,1)
71.      305     COMATX(I,IJK,1) = HOLD
72.              GO TO 330
73.      310     XMM = XM(I)
74.              XM(I) = XH(I)
75.              XH(I) = XMM
76.              DO 315 IJK = 1,5
77.              HOLD = COMATX(I,IJK,3)
78.              COMATX(I,IJK,3) = HOLD + COMATX(I,IJK,2)
79.      315     COMATX(I,IJK,2) = HOLD + COMATX(I,IJK,1)
80.              GO TO 330
81.      320     DO 325 IJK = 1,5
82.              HOLD = COMATX(I,IJK,3)
83.      325     COMATX(I,IJK,3) = HOLD + COMATX(I,IJK,2)
84.      330       CONTINUE
85.      300       CONTINUE
86.      C       CONVERT FROM KEV/MICRON TO KEV/CM
87.      C
88.              DO 123 K = 1,NE
89.              DO 129 IF = 1,3
90.              DO 12 IJK = 1,5
91.      12       COMATX(K,IJK,IF) = COMATX(K,IJK,IF) * 1.E+4
92.      C       PRINT 479,K,(COMATX(K,IJK,IF),IJK = 1,5),XM(K),XH(K),XL(K)
93.      479      FORMAT(I3,8E9.3)
94.      129      CONTINUE
95.      123       CONTINUE
96.              RETURN
97.              END
```

```
      =====  HTFLUX  =====

 1.            SUBROUTINE HTFLUX
 2.           COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 3.          &          DEPX(50,200),DEL(200),X(200),FF(200)
 4.            COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
 5.           COMMON/OUT/NT,TA(250),NY,Y(250)
 6.            COMMON/PRMTR/F,ALPHA,TK,TD
 7.            NY = NSX
 8.            NT = NTP
 9.            READ ,F,TD,ALPHA,TK,M,W
10.            N = M-1
11.            DO 33 K = 1,NSX
12.            Y(K) = SX(K)
13.            DO 22 J = 1,NTP
14.            TA(J) = STP(J)
15.            DM = TA(J) - TD
16.            SUM = 0.0
17.            TEMP(K,J) = 0.0
18.            IF(N .EQ. 0) GO TO 10
19.            DO 11 I = 1,N
20.            FI = I
21.            TI = FI*W + TA(J)
22.            TJ = TI-TD
23.            SUM = SUM + R(TI,Y(K)) - R(TJ,Y(K))
24.     11      CONTINUE
25.     10      TEMP(K,J) = R(TA(J),Y(K)) + SUM
26.            IF(TA(J) .GT. TD) TEMP(K,J) = TEMP(K,J) - R(DM,Y(K))
27.     22       CONTINUE
28.     33      CONTINUE
29.            PRINT 101,F,ALPHA,TK,TD,M,W
30.            PRINT 102,(SX(I),I = 1,NSX)
31.            DO 77 I = 1,NTP
32.     77      PRINT 103,I,TA(I),(TEMP(J,I),J = 1,NSX)
33.     101     FORMAT(//,'       FINITE DURATION HEAT FLUX MODEL ',//,
34.          1' FLUX IS                  ',F10.4,3X,'J/CM2',/,
35.          2' THERMAL DIFFUSIVITY IS ',F10.4,3X,'CM2/SEC',/,
36.          3' THERMAL CONDUCTIVITY IS',F10.4,3X,'CAL/C/CM/SEC',/,
37.          B'  PULSE DURATION IS',E10.4,3X,'SECONDS',/,
38.          4' NUMBER OF PULSES IS    ',I5,3X,/,
39.          5' PULSE SPACING IS       ',F10.4,3X,'SEC',//,
40.          6 '                  TEMPERATURE DISTRIBUTION',//)
41.     102     FORMAT(8X,' X =',15(1X,E7.3),/,' TIME')
42.     103     FORMAT(1X,I3,16(1X,E7.3))
43.            RETURN
44.            END
```

```
=====  IDEPO  =====

      SUBROUTINE IDEPO
      COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,EO,A2,A3
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200)
      COMMON/VAR/DT,DX,BB,NX
       COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
       COMMON/HIEN/E1,E2,S1,S2,EM,BO,AO
       COMMON/PLYCOF/XM(200),XH(200),XL(200),COMATX(200,5,3),QQ1(200)
     &  ,INCID(200)
       COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
       DIMENSION DI(100)

C       **  ZERO THE DEP AND TEMP ARRAYS
       DO 10001  J=1,50
       DO 10002  I=1,250
10002  TEMP(J,I)=0.

       DO 10003 I=1,200
10003  DEPX(J,I)=0.0
10001  CONTINUE

C
C    THIS ROUTINE CALCULATES THE VOLUMETRIC DEPOSITION FOR
C    BOTH HEAVY AND LIGHT IONS FROM THE GENERAL COEFFICIENT
C    MATRIX DETERMINED IN EITHER HCOEF OR LCOEF
C
       READ , IXMATX
       IF(IMODEL .NE. 2) GO TO 100
C
C****************** HEAVY IONS  ***********************************
C
       CALL HCOEF
       DX = XL(1)/(NX-1)
       X(1) = 0.0
       DO 77 K = 1,50
       DI(K) = 0.0
77     X(K+1) = X(K) + DX
       IF(IXMATX .NE. 50) GO TO 79
       DO 78 K = 1,NSX
78     X(K) = SX(K)
       NX = NSX
79      CONTINUE
       DO 210 I = 1,NE
       DO 220 J = 1,NX
       IF(X(J) .LE. 1.E-8) GO TO 270
       IF(X(J) .LT. XH(I)) GO TO 250
       IF(X(J) .LT. XL(I)) GO TO 260
       DEPX(J,I) = 0.0
       GO TO 280
250     N = 2
       DEPX(J,I) = EVAL(N)
       GO TO 280
260     N = 1
       DEPX(J,I) = EVAL(N)
       GO TO 280
270     DEPX(J,I) = COMATX(I,1,2)
280     DEPX(J,I) = DEPX(J,I) * FT(I) * 1.6E-16
```

```
      IF(DEPX(J,I) .LE. 0.0) DEPX(J,I) = 0.0
C
C
      DI(J) = DI(J) + DEPX(J,I) * DT
      TEMP(J,I) = DI(J)
220     CONTINUE
210      CONTINUE
      PRINT 2000
2000   FORMAT('1',//,30X,'HEAVY IONS - NUCLEAR ENERGY DEPOSTION',//)
      PRINT 2001,(X(I) , I = 1,12)
2001   FORMAT(30X,'LOCATIONS ',//,'    TIME',//,12X,12E10.4,//)
      DO 285 I = 1,NE
285   PRINT 2002, I, T(I),(DEPX(J,I),J = 1,12)
2002   FORMAT(I3,E9.3,12E10.4)
      PRINT 2004
2004   FORMAT('1',//,30X,'HEAVY IONS - INT NUC ENERGY DEPOSTION',//)
      PRINT 2001,(X(I) , I = 1,12)
      DO 286 I = 1,NE
286   PRINT 2002, I, T(I),(TEMP(J,I),J = 1,12)
C
C*************************ELECTRONIC*************************
C
      DO 37 I = 1,50
37    DI(I) = 0.0
      DO 310 I = 1,NE
      DO 320 J = 1,NX
      IF(X(J) .LE. 1.E-8) GO TO 370
      IF(X(J) .LT. XM(I)) GO TO 360
      DEPX(J,I) = 0.0
      GO TO 380
360    N = 3
      DEPX(J,I) = EVAL(N)
      GO TO 380
370    DEPX(J,I) = COMATX(I,1,3)
380    DEPX(J,I) = DEPX(J,I) * FT(I) * 1.6E-16
      IF(DEPX(J,I) .LE. 0.0) DEPX(J,I) = 0.0
C
C
      DI(J) = DI(J) + DEPX(J,I) * DT
      TEMP(J,I) = DI(J)
320     CONTINUE
310      CONTINUE
      PRINT 3000
3000   FORMAT('1',//,30X,'HEAVY IONS - ELECTRONIC ENERGY DEPOSTION',//)
      PRINT 2001,(X(I) , I = 1,12)
      DO 385 I = 1,NE
385   PRINT 2002, I, T(I),(DEPX(J,I),J = 1,12)
      PRINT 2005
2005   FORMAT('1',//,30X,'HEAVY IONS - INT ELEC ENERGY DEPOSTION',//)
      PRINT 2001,(X(I) , I = 1,12)
      DO 386 I = 1,NE
386   PRINT 2002, I, T(I),(TEMP(J,I),J = 1,12)
C
C************************ COMBINATION  ************************
C
      IF(IMODEL .EQ. 2) CALL COMCOF
100    CONTINUE
```

```
=====  IDEPO  =====

115.            IF(IMODEL .NE. 2) CALL LCOEF
116.            DX = XL(1)/(NX-1)
117.            X(1) = 0.0
118.            DO 87 K = 1,50
119.            DI(K) = 0.0
120.    87      X(K+1) = X(K) + DX
121.            IF(IXMATX .NE. 50) GO TO 89
122.            DO 88 K = 1,NSX
123.    88      X(K) = SX(K)
124.            NX = NSX
125.    89       CONTINUE
126.            DO 110 I = 1,NE
127.            DO 120 J = 1,NX
128.            IF(X(J) .LE. 1.E-8) GO TO 170
129.            IF(X(J) .LT. XM(I)) GO TO 140
130.            IF(X(J) .LT. XH(I)) GO TO 150
131.            IF(X(J) .LT. XL(I))   GO TO 160
132.            DEPX(J,I) = 0.0
133.            GO TO 180
134.    140    N = 3
135.            DEPX(J,I) = EVAL(N)
136.            GO TO 180
137.    150      N = 2
138.            DEPX(J,I) = EVAL(N)
139.            GO TO 180
140.    160      N = 1
141.            DEPX(J,I) = EVAL(N)
142.            GO TO 180
143.    170     MM = INCID(I)
144.             DEPX(J,I) = COMATX(I,1,MM)
145.    180      DEPX(J,I) = DEPX(J,I) * FT(I) * 1.6E-16
146.            IF(DEPX(J,I) .LE. 0.0) DEPX(J,I) = 0.0
147.    C
148.    C
149.            DI(J) = DI(J) + DEPX(J,I) * DT
150.            TEMP(J,I) = DI(J)
151.    120      CONTINUE
152.    110      CONTINUE
153.            PRINT 1000
154.    1000     FORMAT('1',//,30X,'TOTAL ENERGY DEPOSTION',//)
155.            PRINT 2001,(X(I) , I = 1,12)
156.            DO 185 I = 1,NE
157.    185     PRINT 2002, I, T(I),(DEPX(J,I),J = 1,12)
158.            PRINT 2006
159.    2006     FORMAT('1',//,30X,'TOTAL - INTEGRATED ENERGY DEPOSTION',//)
160.            PRINT 2001,(X(I) , I = 1,12)
161.            DO 186 I = 1,NE
162.    186     PRINT 2002, I, T(I),(TEMP(J,I),J = 1,12)
163.            RETURN
164.    C
165.    C
166.            FUNCTION EVAL(N)
167.            EVAL = 0.0
168.             DO 766 IJK = 1,5
169.    766     EVAL = EVAL + COMATX(I,IJK,N) * X(J) **(IJK-1)
170.            RETURN
171.            END
```

```
=====  IFILE/IODR  =====

                    SUBROUTINE FILEI
C
C           ION DATA FROM THE COMMON BLOCKS
      COMMON/MAT/IMODEL ,ALPHA,RHO,CP,X2,C,E0,A2,A3
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
       COMMON/HIEN/E1,E2,S1,S2,EM,B0,A0
       EQUIVALENCE (C,A0),(E0,A1)
       COMMON/OUT/NT,TA(250),NY,Y(250)

       INTEGER NPTS(50)

C      ************  END OF DATA SECTION  ******************

       RETURN

C      ------------------------------------------------------------
       ENTRY IFILE3
       CALL OPN BIN('ION-3')
C      __  FILE THE DATA
       CALL NP FILL
       CALL FILE(NPTS,1)
      CALL FILE(TEMP,250)
       CALL FILE(FT,4)
      CALL FILE(FF,4)
       CALL FILE(E,4)
       CALL FILE(T,4)
       CALL FILE(TA,4)
      CALL FILE(Y,1)
       CALL CLEAR

      RETURN
C      ------------------------------------------------------




C      ------------------------------------------------
       ENTRY IFILE2

       CALL OPN BIN('ION-2')
C      __  FILE THE DATA
       CALL NP FILL
       CALL FILE(NPTS,1)
      CALL FILE(TEMP,250)
       CALL FILE(FT,4)
      CALL FILE(FF,4)
       CALL FILE(E,4)
       CALL FILE(T,4)
       CALL FILE(X,4)
       CALL CLEAR

      RETURN
```

```
      =====  IFILE/IODR  =====

C      ------------------------------------------------------
C      ------------------------------------------------------
       ENTRY IFILE1

       CALL OPN BIN('ION-1')
C      __  FILE THE DATA
       CALL NP FILL
       CALL FILE(NPTS,1)
       CALL FILE(DEPX,200)
      CALL FILE(FT,4)
       CALL FILE(FF,4)
      CALL FILE(E,4)
       CALL FILE(T,4)
       CALL FILE(X,4)
       CALL CLEAR
      RETURN
C      ------------------------------------------------------------
C      ----------------------------------------
           SUBROUTINE NP FILL
       NPTS(1)=NX
       NPTS(2)=NE
       NPTS(3)=NY
       NPTS(4)=NT
       RETURN
C      ----------------------
      END
```

```
      =====  IMASTER  =====

 1.          COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,E0,A2,A3
 2.          COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 3.         &          DEPX(50,200),DEL(200),X(200)
 4.          COMMON/VAR/DT,DX,BB,NX
 5.          COMMON/OUT/NT,TA(250),NY,Y(250)
 6.          COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
 7.           COMMON/HIEN/E1,E2,S1,S2,EM,B0,A0
 8.           COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
 9.     C     EQUIVALENCE (C,A0),(E0,A1)
10.
11.
12.     5     CONTINUE
13.           READ (5,44,END=99)INST
14.     44    FORMAT(A6)
15.
16.
17.            IF ( INST .EQ. 'DEPLIT') CALL GSDPLT
18.            IF ( INST .EQ. 'DEPHVY') CALL DEDXGS
19.            IF ( INST .EQ. 'LOWIMP') CALL LOWIMP
20.            IF ( INST .EQ. 'HCOEF ') CALL HCOEF
21.            IF ( INST .EQ. 'HCOEF ') CALL COMCOF
22.            IF ( INST .EQ. 'LCOEF ') CALL LCOEF
23.            IF ( INST .EQ. 'LITMOD') CALL LITMOD
24.            IF ( INST .EQ. 'SPEMOD') CALL SPEMOD
25.            IF ( INST .EQ. 'I-RESI') CALL RESID
26.          IF( INST .EQ. 'I-SPEC') CALL SPECFL
27.           IF( INST .EQ. 'I-DEPO' ) CALL IDEPO
28.          IF ( INST .EQ. 'I-TEMP' ) CALL ITEMP
29.           IF( INST .EQ. 'STARAY') CALL STARAY
30.           IF( INST .EQ. 'STARA2') CALL STARA2
31.           IF( INST .EQ. 'STARA3') CALL STARA3
32.           IF( INST .EQ.'I-STEM') CALL ISTEMP
33.           IF (INST .EQ. 'I-OPEN') CALL OPEN
34.           IF(INST .EQ. 'I-REOP') CALL REOPEN
35.           IF (INST .EQ. 'I-1FIL') CALL IFILE1
36.           IF (INST .EQ. 'I-2FIL') CALL IFILE2
37.           IF (INST .EQ. 'I-3FIL') CALL IFILE3
38.           IF (INST .EQ. 'LOGIN') CALL LOGIN
39.           IF (INST .EQ. 'HTFLUX') CALL HTFLUX
40.     C     DO 100 I=1,NT
41.     C     DO 100 J=1,NY
42.     C00   DEPX(J,I)=DEPX(J,I)+TEMP(J,I)
43.           GO TO 5
44.     99    STOP DONE
45.           END
```

===== ISTEMP =====

```
      SUBROUTINE ISTEMP
      COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,E0,A2,A3
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/OUT/NT,TA(250),NY,Y(250)
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
      COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP


C       *** ZERO OUT THE TEMP ARRAY  *****
      DO 1287 I=1,50
      DO 1287 J=1,250
1287  TEMP(I,J)=0.0


C         *** FILL THE Y ARRAY ***
      DO 1387 I=1,50
      Y(I)=SX(I)
1387  CONTINUE
      IMODEL = 5

      READ,IXS,IXF,IXD
      IF(IXS .NE. 777) GO TO 48
      IMODEL = IXF
      READ,IXS,IXF,IXD
48     CONTINUE
      PRINT 52
      PRINT 49,A,R,FL,RHO,CP,ALPHA
49    FORMAT(//,'1 ION MASS IS ',F5.1,/,' WALL RADIUS IS',F5.1,/,
     &' ION FLUENCE IS',E9.3,'  PARTICLES/CM2',/,'  DENSITY IS',
     &F9.3,/,'SPECIFIC HEAT IS  ',F9.3,/,' ALPHA IS   ',F9.3,//)
      PRINT 50,(SX(I),I = IXS,IXF,IXD)
50    FORMAT(1H1,' TEMPERATURE AS A FUNCTION OF TIME AT VARIOUS
     &VALUES OF X',///,  ' X = ',8X,10(3X,E9.3),//)
      PRINT 53
53     FORMAT(//)
      NT = NST
      NY = NSX
C
C   FIND VALUES FOR STANDARD TIMES
C
      INC = 1
      I = 1
C************************************  TIME LOOP   **********************
774       CONTINUE
      TA(I) = ST(I)
      NET = NE
      IF(TA(I) .LE. T(1)) GO TO 203
      IF(TA(I) .GE. T(NE)) GO TO  9
      DO 6 JJJ = 1,NE
6     IF(T(JJJ) .GT. TA(I)) GO TO 7
7     NET = JJJ-1
      DIF = TA(I) - T(NET)
      TIME = T(NET) + 1.E-10
      ISM = I + 1
      GO TO 10
```

```
C************************************************ T GT T(NE) **********
9        CONTINUE
         JRM = NT - ISM
         IEV = MOD(JRM,3)
         IF(IEV .NE. 0) GO TO 95
         INC = 3
         GO TO 96
95       ISM = I+1
96       TIME = TA(I) + 1.E-10
C************************************************ X LOOP***********
10       DO 202 J = IXS,IXF,IXD
         Y(J) = SX(J)
        TT=0.
        S=Y(J)
         TT2 = 0.0
         ALTIM = TIME
        DO 200 K=1,NET
         GO TO (61,62,63,64,65),IMODEL
61        ADDER  = TEMP1(TIME,S,K)
63       GO TO 66
62        ADDER = TEMP2(TIME,S,K)
64       GO TO 66
65         ADDER = TEMP5(TIME,S,K)
66         CONTINUE
         TT = TT + ADDER
200       CONTINUE
         IF(NET .EQ. NE) GO TO 2001
         TIME = T(JJJ) + 1.E-10
         DO 2000 K = 1,JJJ
         GO TO (71,72,73,74,75),IMODEL
71        ADDER  = TEMP1(TIME,S,K)
73       GO TO 77
72        ADDER = TEMP2(TIME,S,K)
74       GO TO 77
75         ADDER = TEMP5(TIME,S,K)
77         CONTINUE
         TT2 = TT2 + ADDER
2000       CONTINUE
         TT = TT + (TT2-TT)*DIF/DT
2001       CONTINUE
         TIME = ALTIM
        TEMP(J,I)=TT/4.186/RHO/CP
202      CONTINUE
C************************************** X LOOP  ***************
         GO TO 204
203      IFIRST = I+1
204        CONTINUE
         I = I + INC
         IF(I .LE. NT) GO TO 774
C    ************************************ TIME LOOP  ****************
         NTL = NT- 3
         DO 300 I = ISM,NTL,3
         DO 300 J = IXS,IXF,IXD
         TA(I+1) = ST(I+1)
         TA(I+2) = ST(I+2)
         SLOPE = (TEMP(J,I+3) - TEMP(J,I))/ALOG(TA(I+3) / TA(I))
         TEMP(J,I+1) = TEMP(J,I) + SLOPE * ALOG(TA(I+1) / TA(I))
```

```
      =====  ISTEMP  =====

115.            TEMP(J,I+2) = TEMP(J,I) + SLOPE * ALOG(TA(I+2) / TA(I))
116.    300       CONTINUE
117.            DO 400 I = IFIRST,NT
118.    400     PRINT 51,TA(I),(TEMP(J,I),J= IXS,IXF,IXD)
119.    51      FORMAT(10(3X,E9.3))
120.    52      FORMAT(//,3X,                'YOU HAVE CHOSEN TO EVALUATE THE ION
121.           &TEMPERATURE RESPONSE AT THE STANDARD TIMES')
122.
123.
124.            RETURN
125.            END
```

```
=====  ITEMP  =====

      SUBROUTINE ITEMP
      COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,E0,A2,A3
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/OUT/NT,TA(250),NY,Y(250)
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS


C       **   ZERO OUT THE TEMP ARRAY  *******
        DO 1287 I=1,50
        DO 1287 J=1,250
1287    TEMP(I,J)=0.0

        READ ,NY
        READ,( Y(I),I = 1,NY)
        PRINT 49,A,R,FL,RHO,CP,ALPHA
49      FORMAT(//,' ION MASS IS ',F5.1,/,' WALL RADIUS IS',F5.1,/,
     &' ION FLUENCE IS',E9.3,'  PARTICLES/CM2',/,'  DENSITY IS',
     &F9.3,/,'SPECIFIC HEAT IS  ',F9.3,/,' ALPHA IS   ',F9.3,//)
        PRINT 50,(Y(I),I = 1,NY)
50      FORMAT(1H1,' TEMPERATURE AS A FUNCTION OF TIME AT VARIOUS
     &VALUES OF X',///,  ' X = ',10(3X,E9.3),//)
        READ,NT
        IF (NT .EQ. 0) GO TO 5
        READ,(TA(I),I=1,NT)
5       DO 70 I = 1,NT
70      TA(NE+I) = T(NE) + TA(I)
        DO 72 I = 1,NE
72      TA(I) = T(I)
        NT = NT + NE
C NOTICE THE INNER LINIT OF THE LOOP
        DO 203 I = 1,NT
        NET = I
        IF (I .GT. NE) NET = NE
        TIME = TA(I) + 1.E-10
C PREVIOUS ROUTINE FOR SETTING UP TIME ARRAY
C-------------------------------------------------
C      NNN=1
C      IF(NT .EQ. 0) NNN=0
C      IF(NNN .EQ.0) NT=10
C      DO 203 I=1,NT
C      NET=NE
C      IF(NNN .EQ.0) GO TO 8
C      TIME=TA(I)+1.E-10
C      IF(TA(I) .GT. T(NE)) GO TO 10
C      DO 6 J=1,NE
C6      IF(T(J) .GT.TA(I)) GO TO 7
C7      NET=J
C      TIME=T(J)+1.E-10
C      GO TO 10
C8      NET=NE*I/10.
C      TIME=T(NET)+1.E-10
10      DO 202 J=1,NY
        TT=0.
        S=Y(J)
        DO 200 K=1,NET
```

```
      =====  ITEMP  =====

58.            IF(IMODEL .EQ. 1) TT=TT+TEMP1(TIME,S,K)
59.            IF(IMODEL .EQ. 2) TT=TT+TEMP2(TIME,S,K)
60.             IF(IMODEL .EQ. 3) TT = TT +TEMP3(TIME,S,K )
61.             IF(IMODEL  .EQ. 4 ) TT = TT + TEMP4(TIME,S,K )
62.             IF(IMODEL .EQ. 5) TT = TT + TEMP5(TIME,S,K)
63.      200   CONTINUE
64.            TEMP(J,I)=TT/4.186/RHO/CP
65.      202    CONTINUE
66.             PRINT 51,TIME,(TEMP(J,I),J= 1,NY)
67.      51     FORMAT(10(3X,E9.3))
68.      203    CONTINUE
69.
70.
71.            RETURN
72.             END
```

```
            =====  LCOEF  =====

 1.            SUBROUTINE LCOEF
 2.            COMMON/PLYCOF/XM(200),XH(200),XL(200),COMATX(200,5,3),QQ1(200)
 3.           &  ,INCID(200)
 4.            COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 5.           &          DEPX(50,200),DEL(200),X(200),FF(200)
 6.            COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
 7.            COMMON/STOP/E0,S0,E1,S1,E2,S2,E3,S3,E4,S4,SMAX,B0,A0,A00,B00
 8.           1  ,EINT,SINT
 9.     C
10.     C   THIS SUBROUTINE FILLS THE COEFFICIENT MATRIX FOR A LIGHT ION
11.     C      SPECTRUM
12.     C
13.     C        **   INPUT
14.            READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
15.     C   NEW  PARAMETERS FOR REGION 1
16.     C
17.     C
18.            A00 = S0**2/(2.*S0 - S1)
19.           B00 = -ALOG(1. - S0/A00)/E0
20.     C
21.     C
22.
23.
24.     C        ** CONSTANTS ARE CALCULATED
25.            B0=(E3-E2)/ALOG(S2/S3)
26.            A0=S2*EXP(E2/B0)
27.     C
28.     C     FIND THE INTERSECTION OF CURVE 2 AND CURVE 3
29.     C
30.            SMDIF = 1.E+6
31.            ETR = E1
32.            DTE = (E2-E1)/100.
33.            DO 427  I = 1,100
34.            FORM2 = A00 *(1. - EXP(-B00 *ETR))
35.            FORM3 = A0 * EXP(-ETR/B0)
36.            DIFT = ABS(FORM3 - FORM2)
37.            CSM = SMDIF
38.           SMDIF = AMIN1(SMDIF,DIFT)
39.            CDIF = CSM - SMDIF
40.            IF(CDIF .LE. 0.) GO TO 428
41.            ETR = ETR + DTE
42.     427      CONTINUE
43.     428      CONTINUE
44.            EINT = ETR
45.            SINT = A0*EXP(-EINT/B0)
46.     C
47.     C      PRINT THE IN PUT AND THE INTERSECTION POINTS
48.     C
49.            PRINT 111,E0,S0,E1,S1,E2,S2,E3,S3,EINT,SINT,SMAX
50.     111      FORMAT(///,8X,'E0 = ',F9.2,5X,'S0 = ',F9.2,/,
51.          1               8X,'E1 = ',F9.2,5X,'S1 = ',F9.2,/,
52.          2               8X,'E2 = ',F9.2,5X,'S2 = ',F9.2,/,
53.          3               8X,'E3 = ',F9.2,5X,'S3 = ',F9.2,/,
54.          3               6X,'EINT = ',F9.2,3X,'SINT = ',F9.2,/,
55.          48X,'SMAX = ',F9.2,///)
56.            DO 10 I = 1,200
57.            DO 10 J = 1,5
```

```
      =====  LCOEF  =====

58.             DO 10 K = 1,3
59.     10       COMATX(I,J,K) = 0.0
60.             DO 100 I = 1,NE
61.             CALL PCOEF(E(I),I)
62.     100        CONTINUE
63.     C   CONVERT THE COEFFIEIENTS BASED ON MICRONS TO BASED ON CM
64.     C     AND CONVERT FROM KEV/MICRON TO KEV/CM
65.     C    FACTORS ARE 10**IJK-1 TIMES 10**4
66.     C
67.             DO 123 K = 1,NE
68.             DO 12 IF = 1,3
69.             DO 12 IJK = 1,5
70.     12       COMATX(K,IJK,IF) = COMATX(K,IJK,IF) * 10.**(4*IJK)
71.             XM(K) = XM(K) * 1.E-4
72.             XH(K) = XH(K) * 1.E-4
73.             XL(K) = XL(K) * 1.E-4
74.     123      CONTINUE
75.             RETURN
76.             END
```

```
            =====  LITMOD  =====

 1.           SUBROUTINE LITMOD
 2.           COMMON/MAT/IMODEL ,ALPHA,RHO,CP,X2,C,E0,A2,A3
 3.           COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 4.          &          DEPX(50,200),DEL(200),X(200),FF(200)
 5.           COMMON/VAR/DT,DX,BB,NX
 6.           COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
 7.            COMMON/HIEN/ZZ1,ZZ2,ZZ3,ZZ4,ZZ5,ZZ6,ZZ7
 8.            COMMON/NWSPEC/NMHIST,EHIST(20),AMP(20)
 9.            EQUIVALENCE (C,A0),(E0,A1)
10.            COMMON/STOP/E0,S0,E1,S1,E2,S2,E3,S3,E4,S4,SMAX,B0,A0,A00,B00
11.          1  ,EINT,SINT
12.            COMMON/INTREQ/EFINAL,QQ1(200),XL(200),XH(200),XM(200),INCID(200)
13.          1 ,NEWXL
14.            VEL(EN)  = 4.39E+7 * SQRT(EN /A)
15.     C       **   INPUT
16.            READ 729,INST1
17.            READ 729,INST2
18.     729     FORMAT(A4)
19.            READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
20.            READ , REFP,REFT
21.            READ ,RBUF,GASP,GAST,IPRI
22.     C  NEW  PARAMETERS FOR REGION 1
23.     C
24.     C
25.            A00 = S0**2/(2.*S0 - S1)
26.           B00 = -ALOG(1. - S0/A00)/E0
27.     C
28.     C
29.
30.
31.     C       ** CONSTANTS ARE CALCULATED
32.            B0=(E3-E2)/ALOG(S2/S3)
33.            A0=S2*EXP(E2/B0)
34.     C
35.     C     FIND THE INTERSECTION OF CURVE 2 AND CURVE 3
36.     C
37.            SMDIF = 1.E+6
38.            ETR = E1
39.            DTE = (E2-E1)/100.
40.            DO 427  I = 1,100
41.            FORM2 = A00 *(1. - EXP(-B00 *ETR))
42.            FORM3 = A0 * EXP(-ETR/B0)
43.            DIFT = ABS(FORM3 - FORM2)
44.            CSM = SMDIF
45.           SMDIF = AMIN1(SMDIF,DIFT)
46.            CDIF = CSM - SMDIF
47.            IF(CDIF .LE. 0.) GO TO 428
48.            ETR = ETR + DTE
49.     427      CONTINUE
50.     428      CONTINUE
51.            EINT = ETR
52.            SINT = A0*EXP(-EINT/B0)
53.     C
54.     C      PRINT THE IN PUT AND THE INTERSECTION POINTS
55.     C
56.            PRINT 111,E0,S0,E1,S1,E2,S2,E3,S3,EINT,SINT,SMAX
57.     111      FORMAT(///,8X,'E0 = ',E10.3,5X,'S0 = ',E10.3,/,
```

```
     1               8X,'E1 = ',E10.3,5X,'S1 = ',E10.3,/,
     2               8X,'E2 = ',E10.3,5X,'S2 = ',E10.3,/,
     3               8X,'E3 = ',E10.3,5X,'S3 = ',E10.3,/,
     3               6X,'EINT = ',E10.3,3X,'SINT = ',E10.3,/,
    48X,'SMAX = ',E10.3,///)
C
      TLION = 4. * 3.14159 * (R*100.)**2 * FL
C
C     CONVERT FROM KEV/MICRON TO KEV/CM
C
      S0 = S0 * GASP/REFP *REFT/GAST*1.E+4
      S1 = S1 * GASP/REFP *REFT/GAST*1.E+4
      S2 = S2 * GASP/REFP *REFT/GAST*1.E+4
      S3 = S3 * GASP/REFP *REFT/GAST*1.E+4
      SINT = SINT * GASP/REFP *REFT/GAST*1.E+4
      A0 = A0 * GASP/REFP *REFT/GAST*1.E+4
      A00 = A00 * GASP/REFP *REFT/GAST*1.E+4
      PRINT 510,REFP,REFT,GASP,GAST
510    FORMAT(//,8X,'REFERENCE VALUES: PRESSURE',E10.3,'  TEMP',E10.3,
     1      //,8X,'  INPUT   VALUES: PRESSURE',E10.3,'  TEMP',E10.3,//)
      E(1) = EMAX
      DE = ( EMAX - EMIN)/(NE-1)
C
C   ESTABLISH AND NORMALIZE THE INCIDENT SPECTRUM
C
      AF = .02*FL
      TONORM = 0.0
      DO 600 I = 1,NE
      E(I+1) = E(I) - DE
      FF(I) = SPECTR(E(I))
      TONORM = TONORM + FF(I) * DE
600     CONTINUE
      DIFNOM = ABS(TONORM - FL)
      IF( DIFNOM .LT. AF ) GO TO 620
      RAT = FL/TONORM
      TONORM = 0.0
      DO 610 I = 1,NE
      FF(I) = FF(I) * RAT
      TONORM = FF(I) * DE + TONORM
610     CONTINUE
620    CONTINUE
      PRINT 605, R, TONORM,FL,TLION
605    FORMAT(5X,'INTEGRATED NORMALIZED SPECTRUM AT',E10.3,'  METERS',
     1/,5X, 'CONTAINS A TOTAL OF ',E10.3,'  PARTICLES/CM2',
     2/,5X,'INPUT FLUENCE WAS  ',E10.3,'   FOR A TOTAL OF  ',E10.3,
     3'  IONS',//)
      ILAST = NE
      IF(INST2 .NE. 'RANG') GO TO 621
      PRINT 622
622     FORMAT(4X,'  E(IN)    E(OUT)    XM        XH        XL        RBUF
     1      TIME     INCID ',//)
621     CONTINUE
      DO 100 I = 1,NE
      CALL DIVENG(E(I),I)
      TINT1 = 0.0
      TINT2 = 0.0
      TINT3 = 0.0
```

```
115.            IF(RBUF .LT. XM(I)) GO TO 30
116.            IF(RBUF .LT. XH(I)) GO TO 20
117.            IF(RBUF .LT. XL(I)) GO TO 10
118.            GO TO 101
119.      C
120.      C    RBUF IS IN REGION 3
121.      C
122.      30     TINT3 = TIME(0.,RBUF,E(I),3)
123.            GO TO 34
124.      C
125.      C    RBUF IS IN REGION 2
126.      C
127.      20     IF(INCID(I) .EQ. 2) GO TO 22
128.            TINT3 = TIME(0.,XM(I),E(I),3)
129.            TINT2 = TIME(XM(I),RBUF,EINT,2)
130.              GO TO 24
131.      22      TINT2 = TIME(0.,RBUF,E(I),2)
132.      24      GO TO 34
133.      C
134.      C     RBUF IS REGION 1
135.      10     CONTINUE
136.            IMPET = INCID(I)
137.            GO TO (11,12,13),IMPET
138.      11      TINT1 = TIME(0.,RBUF,E(I),1)
139.            GO TO 34
140.      12     TINT2 = TIME(0.,XH(I),E(I),2)
141.            TINT1 = TIME(XH(I),RBUF,E0,1)
142.            GO TO 34
143.      13     TINT3 = TIME(0.,XM(I),E(I),3)
144.            TINT2 = TIME(XM(I),XH(I),EINT,2)
145.            TINT1 = TIME(XH(I),RBUF,E0,1)
146.      34      CONTINUE
147.            IF(NEWXL .EQ. 1 ) GO TO 101
148.            DEL(I) = EFINAL
149.            T(I) = TINT3 + TINT2 + TINT1
150.            IF(INST2 .NE. 'RANG') GO TO 92
151.            PRINT 215, I,E(I),DEL(I),XM(I),XH(I),XL(I),RBUF,T(I),INCID(I)
152.      92      CONTINUE
153.      215      FORMAT(I4,7E9.3,4X,I4)
154.      100      CONTINUE
155.            GO TO 102
156.      101     ILAST = I -1
157.      102     CONTINUE
158.      C
159.      C
160.            PRINT 747,E(ILAST),E(1),RBUF,DEL(1),DEL(ILAST),T(1),T(ILAST)
161.      747    FORMAT('  THAT PORTION OF INCIDENT SPECTRUM TRANSMITTED IS FROM',
162.           1//,16X,F12.2,'  KEV   TO  ',F12.2,'  KEV',//,
163.           2'    MAXIMUM ENERGY AT ',F12.3,'  CM  IS',E12.3,'  KEV
164.           3  MINIMUN IS  ',E12.3,'   KEV',//,'   TIMES OF ARRIVAL ARE FROM',
165.           4 E12.3,'  SECONDS   TO',E12.3,'  SECONDS',//)
166.      C
167.      C  CALCULATE FLUXES
168.      C
169.            IF(ILAST .LE. 0)   GO TO 900
170.            PRINT 214
171.      214   FORMAT(6X,'FLUX AND SPECTRUM AT TIMES FOR EVEN ENERGY INTERVALS',
```

```
     1//,5X,'  TIME      E(IN)     E(OUT)     FLUX    SPECTRUM INTEGRAL FRAC
     2TION',//)
       SUMEFT = 0.0
       DO 200 I = 1,ILAST
       IF(I.EQ.1) GO TO 201
       IF(I .EQ. ILAST) GO TO 296
       DT = (T(I+1) - T(I-1))/2.
       DEPS = ( DEL(I-1) - DEL(I+1))/2.
       GO TO 297
201      DT = T(2) - T(1)
       DEPS = DEL(1) - DEL(2)
       GO TO 297
296     DT = T(I) - T(I-1)
       DEPS = DEL(I-1) - DEL(I)
297      CONTINUE
       FT(I) = FF(I) * DE/DT
       FF(I) = FF(I) * DE/DEPS
       SUMEFT = SUMEFT + FT(I) * DT
       FRACFL = SUMEFT/FL
       PRINT  216,I,T(I),E(I),DEL(I),FT(I),FF(I),SUMEFT,FRACFL
216     FORMAT(I5,7E9.3)
200      CONTINUE
C
C    CALCULATE FLUXES AT EVEN TIME STEPS
C
       DT = ( T(ILAST) - T(1))/(NE - 1)
       X(1) = T(1)
       PR(1) = FT(1)
       E(1) = DEL(1)
       POW(1) = FF(1)
       NEM1 = NE -1
       DO 300 I = 2,NEM1
       X(I) = X(I-1) + DT
       DO 350 J = 1,ILAST
350      IF(T(J) .GT. X(I)) GO TO 355
355      CONTINUE
       J = J-1
       FRAC = (X(I) - T(J))/(T(J) - T(J+1))
       PR(I) = FT(J) + FRAC * (FT(J) - FT(J+1))
       E(I) = DEL(J) + FRAC * (DEL(J) - DEL(J+1))
       POW(I) = FF(J) + FRAC * (FF(J) - FF(J+1))
300      CONTINUE
       X(NE) = T(ILAST)
       PR(NE) = FT(ILAST)
       E(NE) = DEL(ILAST)
       POW(NE) = FF(ILAST)
      TOTET=0.
      TOTPT=0.
       PRINT 19, ISPEC,EMN,SIG,EMIN,EMAX,NE,TLION,FL,R,A
19      FORMAT(///'                OUTPUT OF THE  SPECFLU CALCULATION ',
     1 ///,5X,'ISPEC =   ',I5,/,
     25X,'CHAR ENERGY = ',E10.3,/,
     35X,'SIGMA = ',E10.3,/,5X,'EMIN = ',E10.3,/,
     55X,'EMAX = ',E10.3,/,5X,'NE = ',I5,/,
     75X,'TOTAL IONS = ',E10.3,/,5X,'WALL FLUENCE = ',E10.3,
     8 /,5X,'WALL RADIUS = ',E10.3,/,'ION MASS =',E10.3,///)
       DO 400 I = 1,NE
```

```
229.              T(I) = X(I)
230.              FT(I) = PR(I)
231.              FF(I) = POW(I)
232.             POW(I)=FT(I)*E(I)*1.6E-16
233.             PR(I)=FT(I)*7.29E-17*SQRT(A*E(I))
234.               C = C
235.              ZZ6 = ZZ6
236.              ZZ7 = ZZ7
237.              IF (IMODEL .EQ.1) DEL(I) = 2.*SQRT(E(I)*E0)/C*1.E-4
238.             IF(IMODEL.EQ.2)DEL(I)=A0+A1*E(I)+A2*E(I)**2+A3*E(I)**3
239.             IF(IMODEL.EQ.3)DEL(I)=A0+A1*E(I)+A2*E(I)**2+A3*E(I)**3
240.              IF(IMODEL .EQ. 5.AND. E(I) .GT. ZZ5) DEL(I) =
241.            1 (ZZ6/ZZ7*(EXP(E(I)/ZZ6)-EXP(ZZ5/ZZ6)) + 2*SQRT(ZZ5*E0)/C)*1.E-4
242.              IF(IMODEL .EQ.5 .AND.  E(I) .LE. ZZ5) DEL(I) =
243.            1     2*SQRT(E(I)*E0)/C*1.E-4
244.             TOTPT=TOTPT+FT(I)*DT
245.             TOTET=TOTET+POW(I) * DT
246.              TIMPT = TIMPT + PR(I)*DT
247.     400        CONTINUE
248.              IF( IPRI   .EQ. 1) GO TO 1107
249.              PRINT 7
250.     7         FORMAT(1H1,3X,'TIME',7X,'FLUX',6X,'ENERGY',6X,'POWER',
251.           &6X,'DEPTH',6X,'PRESSURE',6X,'FF'//)
252.              DO 9 I = 1,NE
253.     9        PRINT 74,T(I),FT(I),E(I),POW(I),DEL(I),PR(I),FF(I)
254.              PRINT 18, TOTET,TOTPT,FL,TIMPT
255.     18        FORMAT(5X,'TOTAL ENERGY BY TIME INTEGRATION = ',E9.3,
256.           &/,'TOTAL PARTICLES BY TIME INTEGRATION = ',E9.3,/,
257.           &'FLUENCE WAS ',E9.3,'PARTICLES/CM2',/, ' TOTAL IMPULSE IS',
258.           &F9.3,'  DYNES SEC /CM2',//)
259.     74       FORMAT(7(1X,E10.3))
260.     1107        CONTINUE
261.              IF(INST1 .EQ. 'FILE') CALL IFILE1
262.              RETURN
263.     900        PRINT 373
264.     373        FORMAT(//,12X,'LOCATION EXCEEDED THE MAXIMU RANGE OF THE MOST E
265.           1NERGETIC ION',//)
266.              RETURN
267.     C
268.     C
269.     C
270.              FUNCTION TIME(X1,X2,E,I)
271.              NEWXL = 0
272.              IF(I .EQ. 1) GO TO 10
273.              V1 = VEL(E)
274.              XINTR = X2 - X1
275.              EFINAL = ENERGY(E,XINTR,I)
276.              V2 = VEL(EFINAL)
277.              XMID = (X1 + X2)/2.
278.              XINMID = XMID-X1
279.              EMID = ENERGY(E,XINMID,I)
280.              VMID = VEL(EMID)
281.               TIME = AVTIME(X1,XMID,X2,V1,VMID,V2)
282.               RETURN
283.     C
284.     10         CONTINUE
285.              IQ = 0
```

```
286.              XINTR = X2 - X1
287.              EL = ENERGY(E,XINTR,I)
288.              ELOW = 2.*A
289.              IF(EL .GT. ELOW) GO TO 4
290.              ELA = EL
291.              EL = ELOW
292.               IQ = 1
293.      4       TIME = SQRT(A *E0)/S0  *ALOG(E/EL)/4.39E+7
294.              EFINAL = EL
295.              IF(IQ .NE. 1) RETURN
296.              XE2 = 2. * SQRT(E0)/S0 * ( SQRT(E) - SQRT(2.)) + X1
297.              RNEW = 2. / S0 /SQRT(2./E0)
298.              NEWXL = 0
299.              IF(X2 .LT. RNEW )  GO TO 423
300.              NEWXL = 1
301.               RETURN
302.      423        CONTINUE
303.              EFINAL = 2. - S0*SQRT(2./E0) * (X2 - XE2)
304.              AV2 = (VEL(EL) + VEL(EFINAL))/2.
305.              ADTIM = (X2 - XE2)/AV2
306.              TIME = TIME + ADTIM
307.              RETURN
308.      C
309.      C
310.              FUNCTION AVTIME(Y1,Y2,Y3,W1,W2,W3)
311.              BFIRST = (W2 -W1)/(Y2-Y1)
312.              BLAST = (W3 -W2)/(Y3-Y2)
313.              TFIRST =  ALOG(W2/W1)/BFIRST
314.              TLAST = ALOG(W3/W2)/BLAST
315.              AVTIME = TFIRST + TLAST
316.              RETURN
317.      C
318.      C
319.              FUNCTION ENERGY(ESTAR,SSS,M)
320.              GO TO (1,2,3),M
321.      1       ENERGY = ESTAR - S0*SSS*SQRT(ESTAR/E0) + (S0*SSS)**2/4./E0
322.              GO TO 5
323.      2       ENERGY = ALOG(1. + EXP(B00*(QQ1(I) - A00*SSS)))/B00
324.              GO TO 5
325.      3       ENERGY = B0*ALOG(EXP(ESTAR/B0)- A0*SSS/B0)
326.      5       CONTINUE
327.              RETURN
328.      C
329.      C
330.               SUBROUTINE DIVENG(ENER,I)
331.      C
332.      C    UNITS ARE KEV,  MICRON, AND KEV/MICRON
333.      C
334.              IF(ENER .GE. EINT)GO TO 1001
335.              IF(ENER .LT. EINT .AND. ENER .GT. E0) GO TO 1002
336.              IF(ENER .LE. E0) GO TO 1004
337.      C
338.      C******************** E GREATER THAN EINT*******************       *
339.      1001     CONTINUE
340.      C
341.      C
342.      C      ION OF INCIDENT ENERGY IN REGION 3
```

```
343.              INCID(I) = 3
344.              XM(I) = B0/A0 *(EXP(ENER/B0) - EXP(EINT/B0))
345.              PAREN = 1. - EXP( -B00*EINT)
346.              QQ1(I) = EINT + ALOG(PAREN)/B00
347.              XEND2 = (QQ1(I) - E0 -ALOG(1.- EXP(-E0*B00))/B00)/A00
348.              XH(I) = XM(I) + XEND2
349.              XL(I)  = XH(I) + 2.*E0/S0
350.              GO TO 1006
351.      C
352.      C***************************** E BETWEEN E0 AND EINT********
353.      C
354.      1002     CONTINUE
355.      C
356.      C     ION OF INCIDENT ENERGY IN REGION 2
357.      C
358.              INCID(I) = 2
359.              XM(I) = 0.
360.      C
361.              PAREN = 1. - EXP( -B00*ENER)
362.              QQ1(I) = ENER + ALOG(PAREN)/B00
363.              XEND2 = (QQ1(I) - E0 -ALOG(1.- EXP(-E0*B00))/B00)/A00
364.              XH(I) = XM(I) + XEND2
365.      C
366.              XL(I)  = XH(I) + 2.*E0/S0
367.              GO TO 1006
368.      C
369.      C*********************  E LESS THAN E0***************
370.      1004      CONTINUE
371.      C
372.      C
373.      C      IOF ON INCIDENT ENERGY IN REGION 1
374.      C
375.              INCID(I) = 1
376.              XM(I) = 0.
377.              XH(I) = 0.
378.              XL(I) = XH(I) + 2. * SQRT(ENER*E0)/S0
379.              GO TO 1006
380.      1006      CONTINUE
381.              RETURN
382.              END
```

```
=====  LOWIMP  =====

      SUBROUTINE LOWIMP
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/PLYCOF/XM(200),XH(200),XL(200),COMATX(200,5,3),QQ1(200)
     &  ,INCID(200)
      COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
      COMMON/VAR/DT,DX,BB,NX
      COMMON/OUT/NT,TA(250),NY,Y(250)
C
C
C
      CALL LCOEF
C
      PRINT 100
100    FORMAT(5X,' ENERGY      TOA       RANGE  DISTRIBUTION',/,
     1          5X,'  (KEV)    (SEC)       (CM)    P/CM3/CM',//)
C
C
      NT = NE
      NY = 15
      NLST = NE-1
      DINT = 0.0
      DO 200 I = 1,NLST
      TA(I) = XL(I)
      DELX = XL(I) - XL(I+1)
      TEMP(1,I) = FT(I) * DT / DELX
      DINT = DINT + TEMP(1,I) * DELX
200    CONTINUE
      TA(NE) = XL(NE)
      TEMP(1,NE) = FT(NE) * DT / DELX
      DINT = DINT + TEMP(1,NE) * DELX
      DO 400 I = 1,NE
400   PRINT 300 , I,E(I),T(I),TA(I),TEMP(1,I)
300     FORMAT(I5,7E10.4)
      PRINT 500,DINT
500     FORMAT(//,5X,' TOTAL PARTICLES IN DISTRIBUTION = ',E12.4,//)
C
C
C
C   DISTRIBUTION IS STORED IN TEMP ARRAY AS A TIME VARIBLE
C  RANGES ARE STORED IN TA
C   MUST BE STORED AS ION-3
C   PLOTTED AS BIN-7-1
C   RANGE UNITS = CM DISTRIBUTION = P/CM3
C
         RETURN
         END
```

```
=====  PCOEF/124  =====

      SUBROUTINE PCOEF(ENER,I)
      COMMON/STOP/E0,S0,E1,S1,E2,S2,E3,S3,E4,S4,SMAX,B0,A0,A00,B00
     1 ,EINT,SINT
      COMMON/PLYCOF/XM(200),XH(200),XL(200),COMATX(200,5,3),QQ1(200)
     & ,INCID(200)
      DIMENSION W(10),FF(10),COEF(10)
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
C
C    UNITS ARE KEV,  MICRON, AND KEV/MICRON
C
C   SET THE ENERGY FOR CHANGE OF STOPPING POWER
C
      EBOTH = 10. * A
C
C
      IF(ENER .GE. EINT)GO TO 1001
      IF(ENER .LT. EINT .AND. ENER .GT. E0) GO TO 1002
      IF(ENER .LE. E0) GO TO 1004
C
C*********************************************************************
C******************************** E GREATER THAN EINT*****************
C*********************************************************************
1001    CONTINUE
C
C      ION OF INCIDENT ENERGY IN REGION 3
C
C   COEFFICIENTS FOR REGION 3
C
C    X VALUES ARE REFERENCED TO X = 0 IN ALL REGIONS
C
      INCID(I) = 3
      CC = A0 * EXP(-ENER/B0)
      CCC = CC/B0
      COMATX(I,1,3) = CC
      COMATX(I,2,3) = CCC*CC
      COMATX(I,3,3) = CCC*CC*CCC
      COMATX(I,4,3) = CCC*CC*CCC**2
      COMATX(I,5,3) = CCC*CC*CCC**3
C
C     COEFFICIENT IN REGION 2
C
      XM(I) = B0/A0 *(EXP(ENER/B0) - EXP(EINT/B0))
      PAREN = 1. - EXP( -B00*EINT)
      QQ1(I) = EINT + ALOG(PAREN)/B00
      XEND2 = (QQ1(I) - E0 -ALOG(1.- EXP(-E0*B00))/B00)/A00
      XH(I) = XM(I) + XEND2
C
C
      W(1) = XM(I)
      W(2) = XM(I) + XEND2/3.
      W(3) = XM(I) + XEND2 * 2./3.
       W(4) = XH(I)
      EVL2 = W(2) - XM(I)
      EVL3 = W(3) - XM(I)
      FF(1) = SINT
      FF(2) = A00/(1. + EXP(B00*(A00*EVL2 - QQ1(I))))
      FF(3) = A00/(1. + EXP(B00*(A00*EVL3 - QQ1(I))))
```

```
 58.              FF(4) = S0
 59.              CALL PFIT(3,W,FF,COEF)
 60.              COMATX(I,1,2) = COEF(1)
 61.              COMATX(I,2,2) = COEF(2)
 62.              COMATX(I,3,2) = COEF(3)
 63.              COMATX(I,4,2) = COEF(4)
 64.              COMATX(I,5,2) = 0.
 65.      C
 66.      C
 67.      C DETERMINE VALUES FOR  REGION 1
 68.      C
 69.              XL(I)  = XH(I) + 2.*E0/S0 - SQRT(E0*EBOTH)/S0
 70.              W(1) = XH(I)
 71.              W(3) = XL(I) - SQRT(EBOTH*E0)/S0
 72.              W(2) = (W(3) + W(1))/2.0
 73.              W(4) = XL(I)
 74.              FF(1) = S0
 75.              FF(3) = S0 * SQRT(EBOTH/E0)
 76.              FF(2) = ( FF(3) + FF(1) ) /2.0
 77.               FF(4) = FF(3)
 78.              CALL PFIT(3,W,FF,COEF)
 79.              COMATX(I,1,1) = COEF(1)
 80.              COMATX(I,2,1) = COEF(2)
 81.              COMATX(I,3,1) = COEF(3)
 82.              COMATX(I,4,1) = COEF(4)
 83.              COMATX(I,5,1) = 0.
 84.              GO TO 1006
 85.      C
 86.      C*************************************************************************
 87.      C************************************ E BETWEEN E0 AND EINT********
 88.      C*************************************************************************
 89.      1002     CONTINUE
 90.      C
 91.      C     ION OF INCIDENT ENERGY IN REGION 2
 92.      C
 93.      C
 94.      C    COEFFICIENTS FOR REGION 3
 95.      C
 96.              INCID(I) = 2
 97.              DO 43 IJK = 1,5
 98.      43      COMATX(I,IJK,3) = 0.0
 99.              XM(I) = 0.
100.      C
101.      C      COEFFICIENTS FOR REGION 2
102.      C
103.              PAREN = 1. - EXP( -B00*ENER)
104.              QQ1(I) = ENER + ALOG(PAREN)/B00
105.              XEND2 = (QQ1(I) - E0 -ALOG(1.- EXP(-E0*B00))/B00)/A00
106.              XH(I) = XM(I) + XEND2
107.      C
108.      C
109.              W(1) = XM(I)
110.              W(2) = XM(I) + XEND2/3.
111.              W(3) = XM(I) + XEND2 * 2./3.
112.               W(4) = XH(I)
113.              EVL2 = W(2) - XM(I)
114.              EVL3 = W(3) - XM(I)
```

```
115.              FF(1) = A00*PAREN
116.              FF(2) = A00/(1. + EXP(B00*(A00*EVL2 - QQ1(I))))
117.              FF(3) = A00/(1. + EXP(B00*(A00*EVL3 - QQ1(I))))
118.              FF(4) = S0
119.              CALL PFIT(3,W,FF,COEF)
120.              COMATX(I,1,2) = COEF(1)
121.              COMATX(I,2,2) = COEF(2)
122.              COMATX(I,3,2) = COEF(3)
123.              COMATX(I,4,2) = COEF(4)
124.              COMATX(I,5,2) = 0.
125.        C
126.        C
127.        C DETERMINE VALUES FOR  REGION 1
128.        C
129.              XL(I)  = XH(I) + 2.*E0/S0 - SQRT(E0*EBOTH)/S0
130.              W(1) = XH(I)
131.              W(3) = XL(I) - SQRT(EBOTH*E0)/S0
132.              W(2) = (W(3) + W(1))/2.0
133.              W(4) = XL(I)
134.              FF(1) = S0
135.              FF(3) = S0 * SQRT(EBOTH/E0)
136.              FF(2) = ( FF(3) + FF(1) ) /2.0
137.               FF(4) = FF(3)
138.              CALL PFIT(3,W,FF,COEF)
139.              COMATX(I,1,1) = COEF(1)
140.              COMATX(I,2,1) = COEF(2)
141.              COMATX(I,3,1) = COEF(3)
142.              COMATX(I,4,1) = COEF(4)
143.              COMATX(I,5,1) = 0.
144.              GO TO 1006
145.        C
146.        C*************************************************************************
147.        C**********************************  E LESS THAN E0***************
148.        C*************************************************************************
149.        1004      CONTINUE
150.        C
151.        C      IOF ON INCIDENT ENERGY IN REGION 1
152.        C
153.        C
154.        C    COEFFICIENTS FOR REGION 3 AND REGION 2
155.        C
156.              INCID(I) = 1
157.              DO 44 IJK = 1,5
158.              COMATX(I,IJK,3) = 0.0
159.        44     COMATX(I,IJK,2) = 0.0
160.        C
161.              XM(I) = 0.
162.              XH(I) = 0.
163.        C
164.        C DETERMINE THE VALUES FOR  REGION 1
165.        C
166.        C
167.              XL(I) = XH(I) + 2. * SQRT(ENER*E0)/S0 - SQRT(E0*EBOTH)/S0
168.              IF(ENER .LT. EBOTH ) GO TO 473
169.              W(1) = XH(I)
170.              W(3) = XL(I) - SQRT(EBOTH*E0)/S0
171.              W(2) = (W(3) + W(1))/2.0
```

```
=====  PCOEF/124  =====

172.              W(4) = XL(I)
173.              FF(1) = S0 * SQRT(ENER/E0)
174.              FF(3) = S0 * SQRT(EBOTH/E0)
175.              FF(2) = ( FF(3) + FF(1) ) /2.0
176.               FF(4) = FF(3)
177               VASL = FF(1) - FF(3)
178               TENCNT  = .02* FF(3)
179               IF(VASL .LT. TENCNT) GO TO 473
180               CALL PFIT(3,W,FF,COEF)
181               COMATX(I,1,1) = COEF(1)
182               COMATX(I,2,1) = COEF(2)
183               COMATX(I,3,1) = COEF(3)
184               COMATX(I,4,1) = COEF(4)
185               COMATX(I,5,1) = 0.
186               GO TO 1006
187       473      XL(I) = XH(I) + SQRT(EBOTH*E0)/S0  *ENER/EBOTH
188               COMATX(I,1,1) = S0 * SQRT(EBOTH/E0)
189               COMATX(I,2,1) = 0.0
190               COMATX(I,3,1) = 0.
191               COMATX(I,4,1) = 0.
192               COMATX(I,5,1) = 0.
193               GO TO 1006
194       1006     CONTINUE
195               END
```

```
=====  PFIT  =====

      SUBROUTINE PFIT(I,X,F,C)
      DIMENSION X(10),F(10),C(10)
      A0 = F(1)
      A1 = (F(2) - A0)/(X(2)-X(1))
      C(1) = A0 - X(1)*A1
      C(2) = A1
      IF(I.EQ. 1) RETURN
      A2 = (F(3) - A0 - A1*(X(3) - X(1)))/(X(3)-X(2))/(X(3)-X(1))
      C(1) = C(1) + A2*X(2)*X(1)
      C(2) = C(2) -A2*(X(2)+X(1))
      C(3) = A2
      IF(I .EQ. 2) RETURN
      A3 = ( F(4) - A0 - A1*(X(4)-X(1)) - A2*(X(4)-X(2))*(X(4)-X(1)))/
     1   (X(4)-X(3))/(X(4)-X(2))/(X(4)-X(1))
      C(1) = C(1) - A3*X(3)*X(2)*X(1)
      C(2) = C(2) + A3*(X(1)*X(3) + X(1)*X(2) + X(2)*X(3))
      C(3) = C(3) - A3*(X(1) + X(2) + X(3))
      C(4) = A3
      IF (I .EQ. 3 ) RETURN
      RETURN
      END
```

```
=====   R   =====

1.              FUNCTION R(TM,X)
2.              COMMON/PRMTR/F,ALPHA,TK,TD
3.              A = SQRT(ALPHA*TM)
4.              B = X/(2.*A)
5.              R = 2./4.186*F/(TD*TK) * A* AERFC(B)
6.              RETURN
7.              END
```

```
=====  RESID  =====

      SUBROUTINE RESID
      COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,E0,A2,A3
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/OUT/NT,TA(250),NY,Y(250)
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
       COMMON/CONVER/NCONV,CONV


C       **   ZERO OUT THE TEMP ARRAY  *******
      DO 1287 I=1,50
      DO 1287 J=1,250
1287  TEMP(I,J)=0.0

      READ ,NY
      READ,( Y(I),I = 1,NY)
      PRINT 49,A,R,FL,RHO,CP,ALPHA
49    FORMAT(//,' ION MASS IS ',F5.1,/,' WALL RADIUS IS',F5.1,/,
     &' ION FLUENCE IS',E9.3,'  PARTICLES/CM2',/,'  DENSITY IS',
     &F9.3,/,'SPECIFIC HEAT IS  ',F9.3,/,' ALPHA IS   ',F9.3,//)
      PRINT 50,(Y(I),I = 1,NY)
50    FORMAT(1H1,' TEMPERATURE AS A FUNCTION OF TIME AT VARIOUS
     &VALUES OF X',///,  ' X = ',10(3X,E9.3),//)
      READ,NT
      IF (NT .EQ. 0) GO TO 5
      READ,(TA(I),I=1,NT)
      GO TO 6
5     NT = 1
6       CONTINUE
      READ,NCONV
      DO 203 I = 1,NT
      NET = NE
10    DO 202 J=1,NY
      TT=0.
      S=Y(J)
       TIME = TA(I)
      DO 200 K=1,NET
       IF(IMODEL .EQ. 3) TT = TT +TEMP3(TIME,S,K )
       IF(IMODEL  .EQ. 4 ) TT = TT + TEMP4(TIME,S,K )
       DEPX(J,K) = CONV
C       PRINT,K,CONV,TT,DEL(K),POW(K),FT(K),W,NPLUS
200   CONTINUE
      TEMP(J,I)=TT/4.186/RHO/CP
202    CONTINUE
       PRINT 51,TIME,(TEMP(J,I),J= 1,NY)
51     FORMAT(10(3X,E9.3))
203    CONTINUE
       DO 204 I = 1,NE
        PRINT 52,I,T(I),(DEPX(J,I),J = 1,NY)
204      CONTINUE
52     FORMAT(I5,10(3X,E9.3))
      RETURN
       END
```

```
=====  SPECFLU  =====

      SUBROUTINE SPECFL
      COMMON/MAT/IMODEL ,ALPHA,RHO,CP,X2,C,EO,A2,A3
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
       COMMON/HIEN/E1,E2,S1,S2,EM,BO,AO
       COMMON/NWSPEC/NMHIST,EHIST(20),AMP(20)
       EQUIVALENCE (C,AO),(EO,A1)
       TIME(X) = BB/SQRT(X)
       ENER(X) = BB**2/X**2
       READ(5,-,END = 11)ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS,IPRI
        TLION = FL
        FL = FL/4./3.141593/(R*100.)**2
       IF(ISPEC .NE.3) GO TO 76
       READ,NMHIST,(EHIST(I),AMP(I), I = 1,NMHIST)
       DO 70 I = 1,NMHIST
70      AMP(I) = AMP(I)*FL/TLION
76      CONTINUE
       READ,IMODEL,ALPHA,RHO,CP,X2,NX,C,EO,A2,A3,
       IF (IMODEL .NE.5) GO TO 77
       READ,E1,E2,S1,S2,EM
       BO = (E2 - E1)/LOG(S1/S2)
       AO =  S1 * EXP(E1/BO)
77       CONTINUE
       BB = 2284.E-9*R*SQRT(A)
       GO TO (1,2,333),ISPEC
C
C----------MAXWELLIAN SPECTRUM SET DEFAULT LIMITS.
C
  1    IF(EMIN .LE. 1.E-9) EMIN=EMN/8.
       IF(EMAX .LE. 1.E-9) EMAX=EMN*4.
       GO TO 3
C
C----------GAUSSIAN SPECTRUM SET DEFAILT LIMITS.
C
  2    IF(EMIN .LE. 1.E-9) EMIN=EMN-2*SIG
       IF(EMAX .LE. 1.E-9) EMAX=EMN+2*SIG
       IF(EMIN .GT. 0.)  GO TO 3
       PRINT 500
       GO TO 11
C
C     ARBITRARY HISTOGRAM SPECTRUM
C
333      EMAX = EHIST(NMHIST)
        EMIN = EHIST(1)
3      CONTINUE
       S1=TIME(EMIN )
       S2=TIME(EMAX )
       DT=(S1-S2)/(NE-1)
       TOTET=0.
       TOTPT=0.
       T(1)=S2
        PRINT 19, ISPEC,EMN,SIG,EMIN,EMAX,NE,TLION,FL,R,A
19       FORMAT(///'                OUTPUT OF THE  SPECFLU CALCULATION ',
     1 ///,5X,'ISPEC =    ',I5,/,
     25X,'CHAR ENERGY = ',E10.3,/,
```

```
      =====  SPECFLU  =====

     35X,'SIGMA = ',E10.3,/,5X,'EMIN = ',E10.3,/,
     55X,'EMAX = ',E10.3,/,5X,'NE = ',I5,/,
     75X,'TOTAL IONS = ',E10.3,/,5X,'WALL FLUENCE = ',E10.3,
     8 /,5X,'WALL RADIUS = ',E10.3,/,'ION MASS =',E10.3,///)
5     DO 10 I=1,NE
       AF  = .98 * FL
      E(I)=ENER(T(I) )
      E(1)=EMAX
      T(I+1)=T(I)+DT
      FF(I)=SPECTR( E(I))
      FT(I)=FF(I)*2.*E(I)**1.50/BB
      POW(I)=FT(I)*E(I)*1.6E-16
      PR(I)=FT(I)*7.29E-17*SQRT(A*E(I))
       IF (IMODEL .EQ.1) DEL(I) = 2.*SQRT(E(I)*E0)/C*1.E-4
      IF(IMODEL.EQ.2)DEL(I)=A0+A1*E(I)+A2*E(I)**2+A3*E(I)**3
      IF(IMODEL.EQ.3)DEL(I)=A0+A1*E(I)+A2*E(I)**2+A3*E(I)**3
       IF(IMODEL .EQ. 5.AND. E(I) .GT. EM) DEL(I) =
     1 (B0/A0*(EXP(E(I)/B0)-EXP(EM/B0)) + 2*SQRT(EM*E0)/C)*1.E-4
       IF(IMODEL .EQ.5 .AND.  E(I) .LE. EM) DEL(I) =
     1     2*SQRT(E(I)*E0)/C*1.E-4
      TOTPT=TOTPT+FT(I)*DT
      TOTET=TOTET+POW(I) * DT
       TIMPT = TIMPT + PR(I)*DT
10    CONTINUE
      IF(TOTPT .GE. AF) GO TO 81
      RAT=FL/TOTPT
      TOTPT=0.
      TOTET=0.
      DO 8 I=1,NE
      FT(I)=FT(I)*RAT
       FF(I) = FF(I) * RAT
      POW(I)=POW(I)*RAT
      PR(I)=PR(I)*RAT
      TOTPT=TOTPT+FT(I)*DT
      TOTET=TOTET+POW(I)*DT
       TIMPT = TIMPT + PR(I)*DT
8      CONTINUE


81     CONTINUE
       IF( IPRI   .EQ. 1) GO TO 11
       PRINT 7
7      FORMAT(1H1,3X,'TIME',7X,'FLUX',6X,'ENERGY',6X,'POWER',
     &6X,'DEPTH',6X,'PRESSURE',6X,'FF'//)
       DO 9 I = 1,NE
9      PRINT 12,T(I),FT(I),E(I),POW(I),DEL(I),PR(I),FF(I)
       PRINT 18, TOTET,TOTPT,FL,TIMPT
18     FORMAT(5X,'TOTAL ENERGY BY TIME INTEGRATION = ',E9.3,
     &/,'TOTAL PARTICLES BY TIME INTEGRATION = ',E9.3,/,
     &'FLUENCE WAS ',E9.3,'PARTICLES/CM2',/, ' TOTAL IMPULSE IS',
     &F9.3,'  DYNES SEC /CM2',//)
12     FORMAT(7(1X,E10.3))
500    FORMAT('EMIN IS LESS  THAN ZERO',//)

11     CONTINUE

       RETURN
```

```
      =====   SPECFLU   =====
115.          END
```

```
      =====  SPECTR  =====

      FUNCTION SPECTR (E)
       COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
       COMMON/NWSPEC/NMHIST,EHIST(20),AMP(20)
       GO TO (1,2,3),ISPEC
C  GAUSSIAN
2       CONTINUE
      AA = FL/SQRT( 6.283185)/SIG
      U = E/SIG
      UM = EMN/SIG
      SPECTR      = EXP(-.50 * (U - UM) **2) * AA
      RETURN
C  MAXWELLIAN
1       CONTINUE
      AA = 2*FL/SQRT(3.14159)/EMN
      A1 = SQRT(E/EMN)
      A2 = EXP(-E/EMN)
      SPECTR      = AA* A1 *A2
       RETURN
C
C    ARBITRARY HISTOGRAM
C
3       CONTINUE
       DO 333 I = 1,NMHIST
C  FIND WHERE E IS
333     IF(E.LE. EHIST(I) ) GO TO 334
334       SPECTR =AMP(I-1)
      IF(E .GT. EHIST(NMHIST)) SPECTR = 0.
      IF(E .LT. EHIST(1)) SPECTR = 0.
       RETURN
      END
```

```
=====  SPEMOD  =====

      SUBROUTINE SPEMOD
      COMMON/MAT/IMODEL ,ALPHA,RHO,CP,X2,C,EO,A2,A3
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
       COMMON/HIEN/E1,E2,S1,S2,EM,BO,AO
       COMMON/NWSPEC/NMHIST,EHIST(20),AMP(20)
       COMMON/DEPTIM/STANFT(200),STANPW(200),STANE(200),SDEL(200),
     1 ISTART,ISTOP
       COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
       DIMENSION  EREF(200),W(200),SUMSPE(200)
       COMMON/NECESS/VSTAR,COEF,FACT,REND,DLEND,TEND,GAMMA,RLIM,
     1RINS,CONINS,SMSIG,SIGINS,FLXINS,VSPINS,EINS,ESPINS
       EQUIVALENCE (C,AO),(EO,A1)
C
C      DEFINE  FUNCTIONS
C
C     STANDARD DEVIATION AND RANGE AT END OF RANGE
C
        RANGE(V)  = CONST * V **(1./(1.-AK))
        DELVR(V) = CONDEL* EXP(-V/BDEL) * RANGE(V)
C
C     DIFFUSION COEFFICIENT FOR TIME AT STOP
C    AND STANDARD DEVIATION AT ANY OTHER TIME
C
        SIGLOC(T,G) =  G * SQRT(2.*T)
         DCOF(DP,T) = DP/SQRT(2.*T)
C
C     VELOCITY FROM ENERGY AND INVERSE  KEV AND CM/SEC
C
        ENER(V) = 1.036E-15* A * V**2/2.
        VEL(EN)  = 4.39E+7 * SQRT(EN /A)
C
C      TIME AND VELOCITY AT A GIVEN INTERMEDIATE RANGE
C
         TIME(Y) = COEF*(1. - (1. - Y/CONST * FACT)**AK)
        VELOC(Z) = (VSTAR**(1./(1.-AK)) - Z/CONST)**(1.- AK)
C
C  FUNCTIONS FOR TIME SCAN
C
        VELT(TM) = (VSTAR**P1 - AK*TM/CONST)**PRP
        RV(VL) = CONST * (VSTAR**PN - VL**PN)
        GAUS(A,B,C) = TOT/SQRT(6.2832)/B * EXP( -.500 * ((A-C)/B)**2)
C
C********************* END OF FUNCTIONS ***********************************
C
C
C      DETERMINE VELOCTIY - RANGE CONSTANTS FROM ENERGY RANGE DATA
C
C
C   INPUT
C
        READ 377, INST1
        READ 377,INST2
377       FORMAT(A4)
        READ ,E1,R1,SIG1,E2,R2,SIG2,REFP,REFT
```

```
58.             V1 = VEL(E1)
59.             V2 = VEL(E2)
60.             AK = 1. - (ALOG(V1/V2)/ALOG(R1/R2))
61.             CONST = R1/V1**(1./(1.-AK))
62.              PDEL1 = SIG1/R1
63.              PDEL2 = SIG2/R2
64.              BDEL = ALOG(PDEL1/PDEL2)/(V2 - V1)
65.             BDEL = 1./BDEL
66.              CONDEL = PDEL1 * EXP(V1/BDEL)
67.             P0 = 1./(AK-1.)
68.             P1 = AK/(1.-AK)
69.             PRP = 1./P1
70.             PN = -P0
71.             PDTV = (2.*AK - 1.)/(1. - AK)
72.             PRINT 50 ,A,E1,V1,R1,SIG1,E2,V2,R2,SIG2,AK,CONST,BDEL,CONDEL
73.     50       FORMAT(5X,'ION MASS OF   ',F6.2,'  AMU',/
74.          1 ,3X,'     ENERGY       VELOCITY       RANGE      SIGMA ',//
75.          2 ,3X,4E10.3,/,3X,4E10.3,/, '    AK =  ',F7.4,6X,'C = ',E10.4,//,
76.          3 ' BDEL = ',E10.4,' CONDEL = ',E10.4,//)
77.     C
78.     C    READ BUFFER CONDITIONS
79.     C
80.             READ , RBUF,GASP,GAST,IPRI
81.             CONST = CONST *REFP/GASP*GAST/REFT
82.     C
83.     C     SET UP LIMITS FOR TIME AND ENERGY
84.     C
85.             CALL SETUP(EMAX)
86.             PRINT 442,REND,DLEND,TEND
87.     442      FORMAT(//,10X,'RANGE OF MAXIMUM  ENERGY IS  ',E12.3,'  CM',/
88.          1,10X,'SIGMA AT THAT RANGE IS  ',E12.3,'  CM',/
89.          2,10X,'TIME TO STOP IS  ',E12.3,'  SEC',//)
90.             IF( RBUF .GE.RLIM) GO TO 100
91.             RFIN = REND*.9999
92.             RFIN = AMIN1(RBUF,RFIN)
93.             TIMER = TIME(RFIN)
94.             SIGRB = SIGLOC(TIMER,GAMMA)
95.             DIFRB = RBUF - 2.*SIGRB
96.             TIMER = TIME(DIFRB)
97.             SIGRB = SIGLOC(TIMER,GAMMA)
98.             DIFRB = RBUF - 2.*SIGRB
99.             SS2 = TIME(DIFRB)
100.            PRINT , SS2,TIMER,SIGRB,DIFRB
101.            SS2 = AMAX1(SS2,T(1))
102.            VELMAX = VELOC(DIFRB)
103.            ESCMAX = ENER(VELMAX)
104.            E(1) = EMIN
105.            DE = (EMAX -EMIN)/(NE-1)
106.            ELIM = EMIN
107.    C
108.    C    LOWER LIMITS
109.    C
110.            IFIRST= 1
111.            CALL SETUP(EMIN)
112.            PRINT 443,REND,DLEND,TEND
113.    443      FORMAT(//,10X,'RANGE OF MINIMUM  ENERGY IS  ',E12.3,'  CM',/
114.         1,10X,'SIGMA AT THAT RANGE IS  ',E12.3,'  CM',/
```

```
     2,10X,'TIME TO STOP IS  ',E12.3,'  SEC',//)
       IF(RBUF .GE. RLIM) GO TO 77
       GO TO 99
77       DO 8 I = 1,NE
       CALL SETUP(E(I))
       IF( RLIM .GT. RBUF) GO TO 88
       E(I+1) = E(I) + DE
8      CONTINUE
88      ELIM = E(I)
       IFIRST = I
99        CONTINUE
       RLAST = REND - 2.* DLEND
       IF(RLAST .GT. RBUF) GO TO 113
       SMSTVL = VEL(2.0)
       DIFRB = RV(SMSTVL)
        GO TO 114
113      CONTINUE
       TIMER = TIME(RBUF)
       SIGRB = SIGLOC(TIMER,GAMMA)
       DIFRB = RBUF + 2.*SIGRB
       TIMER = TIME(DIFRB)
       SIGRB = SIGLOC(TIMER,GAMMA)
       DIFRB = RBUF + 2.*SIGRB
114       CONTINUE
       SS1 = TIME(DIFRB)
       VELMIN = VELOC(DIFRB)
       ESCMIN = ENER(VELMIN)
       DES = (ESCMAX - ESCMIN)/(NE - 1)
       DT = (SS1-SS2)/(NE-1)
       TLION = FL * 4. * 3.14159 * (R*100.)**2
C
C
C    PRINT OUT INCIDENT VALUES AND TRANSMITTED LIMITS
C
         PRINT 19, ISPEC,EMN,SIG,EMIN,EMAX,NE,TLION,FL,R,A
19        FORMAT(///'                INPUT TO THE  SPEMOD CALCULATION ',
     1 ///,5X,'ISPEC =   ',I5,/,
     25X,'CHAR ENERGY = ',E10.3,/,
     35X,'SIGMA = ',E10.3,/,5X,'EMIN = ',E10.3,/,
     55X,'EMAX = ',E10.3,/,5X,'NE = ',I5,/,
     75X,'TOTAL IONS = ',E10.3,/,5X,'WALL FLUENCE = ',E10.3,
     8 /,5X,'WALL RADIUS = ',F10.3,/,5X,'ION MASS =',F10.3,///)
         PRINT 747,ELIM,EMAX,RBUF,ESCMAX,ESCMIN,SS2,SS1
747      FORMAT('  THAT PORTION OF INCIDENT SPECTRUM TRANSMITTED IS FROM',
     1//,16X,F12.2,'  KEV   TO  ',F12.2,'  KEV',//,
     2'    MAXIMUM ENERGY AT ',F12.3,'  CM  IS',E12.3,'  KEV
     3  MINIMUN IS  ',E12.3,'   KEV',//,'   TIMES OF ARRIVAL ARE FROM',
     4 E12.3,'  SECONDS   TO',E12.3,'  SECONDS',//)
C
C   ESTABLISH INCIDENT SPECTRUM AND ENERGY DOMAIN FOR
C   TRANSMITTED SPECTRUM
C
         EREF(1) = ESCMIN
         T(1) = SS2
         AF = .03*FL
         TOTPE = 0.0
         DO 10 K = 1,NE
```

```
172.              T(K+1) = T(K) + DT
173.              E(K+1) = E(K) + DE
174.              FF(K) = SPECTR(E(K))
175.              EREF (K+1) = EREF(K) + DES
176.              W(K) = 0.
177.              FT(K) = 0.
178.              SUMSPE(K) = 0.0
179.              TOTPE = TOTPE + FF(K) *DE
180.      10       CONTINUE
181.              FLDIF = ABS(TOTPE - FL)
182.              IF(FLDIF .LE. AF ) GO TO 122
183.              TLION = 0.0
184.              DO 11 K = 1,NE
185.              FF(K) = FF(K) * FL/TOTPE
186.      11        CONTINUE
187.      122        CONTINUE
188.              TLION = FL * 4. * 3.14159 * (R*100.)**2
189.      C
190.      C
191.      C    PRINT DEFINING PARAMETERS
192.               PRINT 707, RBUF,GASP,GAST,SS2,SS1,TLION,ELIM
193.      707       FORMAT(//,'  RADIUS     GASP      GAST   START-TIME STOP-TIME
194.           1 TOTAL IONS  E-LIMIT ',//,7E10.3,//)
195.      C
196.      C
197.      C    GET CONTRIBUTION A TIME I FOR EACH VALUE OF INCIDENT SPECTRUM
198.      C
199.      C     GO DO A RANGE CALCULATION FOR THE INCIDENT SPECTRUM
200.      C
201.              IF(INST2 .EQ. 'RANG') CALL ERANGE
202.              DO 30 J = IFIRST,NE
203.              CALL SETUP(E(J))
204.              TOT = FF(J) * DE
205.      C
206.      C   CALCULATE TRANSMITTED SPECTRUM
207.      C
208.              DO 62 K = 1,NE
209.              IF (EREF(K) .GE. E(J) ) GO TO 63
210.              VINS = VEL(EREF(K))
211.              CALL SPECV(VINS)
212.              SUMSPE(K) = SUMSPE(K) + ESPINS
213.      62        CONTINUE
214.      63        CONTINUE
215.      C
216.      C   FINISH TRANSMITTED SPECTRUM
217.      C
218.      C
219.      C     SET UP FLUX AS A FUNCTION OF TIME  I IS THE TIME INDEX
220.      C
221.              DO 20 I = 1,NE
222.              VINS = VELT(T(I))
223.              CALL SPECV(VINS)
224.              FT(I)  = FT(I) + FLXINS
225.              W(I) = W(I) + FLXINS * EINS
226.      20       CONTINUE
227.      30        CONTINUE
228.      C
```

```
                =====  SPEMOD  =====

229.      C     FILE THE MODIFIED SPECTRUM AND THE FLUX FOR PLOTTING
230.      C
231.              CALL FILMOD
232.      C
233.      C       FILL OUT VALUES WITH NEW FLUX AND SPECTRA
234.      C
235.              TIMPT = 0.0
236.              TOTET = 0.0
237.              TOTPT = 0.0
238.              DO 40 I = 1,NE
239.              AO = AO
240.              BO = BO
241.              C = C
242.              E(I) = W(I)/FT(I)
243.              FF(I) = SUMSPE(NE - I + 1)
244.             POW(I)=FT(I)*E(I)*1.6E-16
245.             PR(I)=FT(I)*7.29E-17*SQRT(A*E(I))
246.              IF (IMODEL .EQ.1) DEL(I) = 2.*SQRT(E(I)*EO)/C*1.E-4
247.             IF(IMODEL.EQ.2)DEL(I)=AO+A1*E(I)+A2*E(I)**2+A3*E(I)**3
248.             IF(IMODEL.EQ.3)DEL(I)=AO+A1*E(I)+A2*E(I)**2+A3*E(I)**3
249.              IF(IMODEL .EQ. 5.AND. E(I) .GT. EM) DEL(I) =
250.            1 (BO/AO*(EXP(E(I)/BO)-EXP(EM/BO)) + 2*SQRT(EM*EO)/C)*1.E-4
251.              IF(IMODEL .EQ.5 .AND.  E(I) .LE. EM) DEL(I) =
252.            1      2*SQRT(E(I)*EO)/C*1.E-4
253.             TOTPT=TOTPT+FT(I)*DT
254.             TOTET=TOTET+POW(I) * DT
255.              TIMPT = TIMPT + PR(I)*DT
256.      40        CONTINUE
257.              IF( IPRI    .EQ. 1) GO TO 102
258.              PRINT 7
259.      7        FORMAT(1H1,3X,'TIME',7X,'FLUX',6X,'ENERGY',6X,'POWER',
260.             &6X,'DEPTH',6X,'PRESSURE',6X,'FF'//)
261.              DO 9 I = 1,NE
262.      9       PRINT 12,T(I),FT(I),E(I),POW(I),DEL(I),PR(I),FF(I)
263.              PRINT 18, TOTET,TOTPT,FL,TIMPT
264.      18       FORMAT(5X,'TOTAL ENERGY BY TIME INTEGRATION = ',E9.3,
265.             &/,'TOTAL PARTICLES BY TIME INTEGRATION = ',E9.3,/,
266.             &'FLUENCE WAS ',E9.3,'PARTICLES/CM2',/, ' TOTAL IMPULSE IS',
267.             &F9.3,'  DYNES SEC /CM2',//)
268.      12      FORMAT(7(1X,E10.3))
269.      102     RETURN
270.      100        CONTINUE
271.              PRINT 737
272.      737       FORMAT('  RADIUS EXCEEDED THE MAXIMUM RANGE OF ION',//)
273.              STOP DONE
274.      C
275.      C
276.      C
277.      C
278.              SUBROUTINE SETUP(E)
279.              VSTAR = VEL(E)
280.      238        FORMAT( ' GOT TO HERE IN SETUP')
281.              COEF = CONST/AK * VSTAR**P1
282.              FACT = VSTAR**PO
283.              REND = RANGE(VSTAR)
284.              DLEND = DELVR(VSTAR)
285.              TEND = TIME(REND)
```

```
286.            GAMMA = DCOF(DLEND,TEND)
287.            RLIM = REND + 2.*DLEND
288.            RETURN
289.      C
290.      C
291.      C
292.            SUBROUTINE SPECV(V)
293.      238    FORMAT(' GOT TO HERE')
294.            RINS = RV(V)
295.            VTIME = TIME(RINS)
296.            CONINS = 0.
297.             SMSIG = REND/1.E+7
298.            SIGINS = SIGLOC(VTIME,GAMMA)
299.            IF( SIGINS .LE. SMSIG) GO TO 66
300.            CONINS = GAUS(RINS,SIGINS,RBUF)
301.      66     CONTINUE
302.            FLXINS = CONINS * V
303.            VSPINS = FLXINS * CONST * PN * V **PDTV
304.            EINS = ENER(V)
305.            ESPINS = VSPINS * 2.20E+7/SQRT(A*EINS)
306.            RETURN
307.      C
308.      C       ROUTINE FOR FILING THE TRANSMITTED SPECTRUM AND FLUX
309.      C
310.            SUBROUTINE FILMOD
311.            PRINT 142,RBUF
312.      142    FORMAT(//,10X,'SPECTRUM TRANSMITTED TO  ',F12.3,'  CM',
313.           1 //,5X,'INDEX       ENERGY       SPECTRUM       INTEGRAL       FRAC
314.           2TION   ',/,13X,'    (KEV)      PART/KEV/CM2   PART/CM2',//)
315.            CUMSPE = 0.0
316.            DO 175 I = 1,NE
317.            E(I) = EREF(I)
318.            FF(I) = SUMSPE(I)
319.            CUMSPE = CUMSPE + FF(I) * DES
320.            PERTOT = CUMSPE/FL
321.            PRINT 143,I,E(I),FF(I),CUMSPE,PERTOT
322.      143    FORMAT(5X,I4,5E14.4)
323.      175    CONTINUE
324.            IF(INST1 .EQ. 'FILE')CALL IFILE1
325.            RETURN
326.      C
327.      C  SUBROUTINE FOR RANGE CALCULATION
328.      C
329.            SUBROUTINE ERANGE
330.            PRINT 150
331.            IDN = 1
332.            IF(NE .GT.50 ) IDN = 2
333.            IF(NE .GT. 100 ) IDN = 3
334.            IF(NE .GT. 150) IDN = 4
335.            NRE = NE/IDN
336.            NX = NRE
337.            DO 100 J = 1,NRE
338.            JER = J*IDN
339.            CALL SETUP(E(JER))
340.            X(J) = E(JER)
341.            DEPX(J,1) = REND
342.            DEPX(J,2) = DLEND
```

```
      =====  SPEMOD  =====

343.            DEPX(J,3) = TEND
344.            PRINT 200, JER,E(JER),VSTAR,REND,DLEND,TEND
345.      100     CONTINUE
346.      200     FORMAT(10X,I5,6E12.4)
347.      150     FORMAT(20X,'RANGE TABLE FOR INCIDENT SPECTRUM',//,
348.           115X,'    ENERGY      VELOCITY     RANGE         END SIGMA      END
349.           2 TIME   ',//)
350.            RETURN
351.            END
```

```
      =====  SPUTTR  =====

      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
      COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
      COMMON/OUT/NT,TA(250),NY,Y(250)
      INTEGER BINT,BINF
C
C   OPEN THE IODR FILE
C
      CALL REOPEN
      READ , BINT
C
C   GET SURFACE TEMPERATURE
C
      CALL GETFIL(BINT,'RDFIL3')
C
      PRINT 71
71     FORMAT(//,'        TIME  SURFACE TEMPERATURE',//)
      DO 95 I = 1,NST
       X(I) = TEMP(1,I)
95     PRINT 1000,I,TA(I),X(I)
C
C    GET FLUX
C
4     READ(5,-,END = 7000)  BINF
C
      CALL GETFIL(BINF,'RDFIL1')
C
C     **    READ IN SPUTTERING PARAMETERS    **
       READ, SPTEMC, SPTEME, TREF
C      READ, SPENEO, SPENSO, SPENB, SPENC
       READ , CONST,FIFTY,Z1,Z2,AM1,AM2
C
C    INTEGRATE ION FLUX
C
      DT = T(2) -T(1)
      DEL(1) = FT(1) * DT
      DO 200 I = 2,NE
200     DEL(I) = DEL(I-1) + FT(I) * DT
      PRINT 72
72     FORMAT(//,'        TIME      FLUX     ENERGY  INT FLUX',//)
      DO 110 I = 1,NE
      AWW = SPENR(E(I))
110    PRINT 1000,I,T(I),FT(I),E(I),DEL(I),AWW
C
C   FIND FLUX AND ENERGY AT STANDARD TIMES
C
      DO 666 I = 1,NST
      IF(T(1) .LE. TA(I)) GO TO 667
666   CONTINUE
667    ISTART = I
      DO 668 I = 1,NST
      IF(T(NE) .LE.TA(I)) GO TO 669
668     CONTINUE
669     ISTOP = I-1
      DO 50 J = ISTART,ISTOP
```

```
      =====  SPUTTR  =====

      DO 10 K = 1,NE
      IF(TA(J) .LT. T(K)) GO TO 11
10     CONTINUE
11      K = K-1
      FRAC = (TA(J) - T(K))/(T(K+1) - T(K))
      POW(J) = DEL(K) + FRAC * (DEL(K+1) - DEL(K))
      PR(J) = E(K) + FRAC * (E(K+1) - E(K))
50     CONTINUE
100    CONTINUE
      DO 300 J = ISTART,ISTOP
300     DEL(J) = (POW(J) - POW(J-1))/(TA(J)-TA(J-1))
C
C  DEL IS FLUX AT STANDARD TIMES
C  PR IS ENERGY AT STANDARD TIMES
C
C     **    ZERO OUT DEPX ARRAY
       DO 53 I=1,50
       DO 53 J=1,200
53     DEPX(I,J)=0.0
C     **    MAXE TOTAL # OF POINTS NST INSTEAD OF NE FOR PLOTS
       IONE=NE
       IOXX=NX
       NE=NST
       NX=NSX
C     **    PRINT HEADINGS
       PRINT 2000
2000    FORMAT(///,'         TIME    ENERGY    FLUX  IONS SPUT  TOTAL  IO
     1NS SPUT  TOTAL  RATIO',/
     2,35X,'W/O TEMP',9X,'WITH TEMP',//)
       SUMSPT = 0.
       SUMSPE = 0.
       DO 700 J=ISTART,ISTOP
       DELTIM    = TA(J) - TA(J-1)
       DEPX(2,J) = SPENR(PR(J)) * DEL(J)
       DEPX(1,J) = DEPX(2,J) * SPTEM(TREF + X(J))
       SUMSPT = SUMSPT + DEPX(1,J) * DELTIM
       SUMSPE = SUMSPE + DEPX(2,J) * DELTIM
       DEPX(3,J) = SUMSPT
       DEPX(4,J) = SUMSPE
        DEPX(5,J) = SUMSPT/SUMSPE
C     **    PRINT OUTPUT    ***
       PRINT 1000, J, TA(J), PR(J), DEL(J), DEPX(2,J),
     &            DEPX(4,J), DEPX(1,J), DEPX(3,J),DEPX(5,J)
700     CONTINUE
1000    FORMAT(I5,8E9.3)
       DO 800 I = 1,NST
       T(I) = TA(I)
       E(I) = PR(I)
       FT(I) = DEL(I)
800     CONTINUE
       READ 234,INSTF
234    FORMAT(A6)
       IF(INSTF .NE.'I-1FIL') GO TO 266
       CALL REOPEN
       CALL IFILE1
266     CONTINUE
       GO TO 4
```

```
                =====  SPUTTR  =====

115.      7000     CONTINUE
116.      C
117.               STOP DONE
118.      C
119.      C
120.      C
121.      C
122.               FUNCTION SPTEM(ZIP)
123.               SPTEM = 1. + SPTEMC * EXP(-SPTEME/8.617E-5/ZIP)
124.               RETURN
125.      C
126.      C
127.      C
128.      C
129.               FUNCTION SPENR(ZAP)
130.                AAA = CONST * (Z1 * Z2) **2 *AM1/AM2
131.                EEV = ZAP*1.E+3
132.                BBB = EEV/(EEV + FIFTY* Z1*Z2)**2
133.                SPENR = AAA*BBB
134.      C         DIV    = ZAP/SPENO
135.      C         STOP   = SPENSO * (ALOG10(DIV)) ** (1. +SPENC/DIV)
136.      C         BOT    = DIV ** (1. + SPENB/DIV)
137.      C         SPENR = TOP/BOT
138.               RETURN
139.               END
```

===== STARA2 =====

```
      SUBROUTINE STARA2
C A PROGRAM TO CALCULATE A STANDARD TIME AND LOCATION ARRAY FOR
C ION AND PHOTON RESPONSE MODELS
      COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
      NST = 115
      NSX = 15
      NTP = 22
      DO 10 J = 1,115
      X = J
      ST(J) = 10**((X-1.)/38. - 7.)
10     CONTINUE
      SX(1) = 1.E-7
      SX(2) = 1.E-6
      SX(3) = 1.E-5
      SX(4) = 2.E-5
      SX(5) = 5.E-5
      SX(6) = 7.E-5
      SX(7) = 1.E-4
      SX(8) = 1.5E-4
      SX(9) = 2.E-4
      SX(10) = 3.E-4
      SX(11) = 5.E-4
      SX(12) = 7.E-4
      SX(13) = 1.E-3
      SX(14) = 3.E-3
      SX(15) = 1.E-2
      DO 50 J = 1,22
      X = J
      STP(J) = 10**((X-1.)/3. -10.)
50     CONTINUE
      RETURN
      END
```

```
=====  STARA3  =====

 1.              SUBROUTINE STARA3
 2.      C A PROGRAM TO CALCULATE A STANDARD TIME AND LOCATION ARRAY FOR
 3.      C ION AND PHOTON RESPONSE MODELS
 4.              COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
 5.              READ , BIGX
 6.              NST = 115
 7.              NSX = 50
 8.              NTP = 22
 9.              DO 10 J = 1,19
10.              X = J
11.              ST(J) = 10**((X-1.)/6. - 10.)
12.      10       CONTINUE
13.              DO 20 J = 20,109
14.              X = J
15.              ST(J) = 10**((X-19.)/30. - 7.)
16.      20       CONTINUE
17.              DO 30 J = 110,115
18.              X = J
19.              ST(J) = 10**((X -109)/6. -4.)
20.      30       CONTINUE
21.              NMI = NSX-1
22.              DX = BIGX/NMI
23.              SX(1) = 0.0
24.              DO 40 J = 1,NMI
25.      40       SX(J+1) = SX(J) + DX
26.              DO 50 J = 1,22
27.              X = J
28.              STP(J) = 10**((X-1.)/3. -10.)
29.      50       CONTINUE
30.              RETURN
31.              END
```

```
=====  TEMP1  =====

      FUNCTION TEMP1(TIME,S,K)
       COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,E0,A2,A3
       COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &DEPX(50,200),DEL(200),X(200)
       COMMON /VAR/DT,DX,BB,NX
       COMMON/OUT/NT,TA(250),NY,Y(250)
      IF(K .GE.2) GO TO 1
      B1=C*BB*1.E+4/SQRT(E0)
      B2=(C*1.E+4)**2/2./E0
    1  CONTINUE
      RELT=ABS(T(K)-TIME)
      SEND=2.*BB*SQRT(E0)/T(K)/(C*1.E+4)
      IF(RELT .LE. 1.E-13) GO TO 43
      RT=SQRT(ALPHA*RELT)
       F0 = S/2./RT
      F1=(SEND-S)/2./RT
      F2=(SEND+S)/2./RT
      TF1=B1/T(K)*(ERF(F1)+ERF(F2))/2.
      TF2=B2*S/2.*(ERF(F1)-ERF(F2)+2.*ERF(F0))
      TF3=B2*RT/SQRT(3.141592)*(-EXP(-F1**2)-EXP(-F2**2)+2.*EXP(-F0**2))
      GO TO 44
   43 CONTINUE
      IF(S .GT. SEND) GO TO 45
      TF1=B1/T(K)
      TF2=B2*S
      TF3=0.
      GO TO 44
   45 TF1=0.
      TF2=0.
      TF3=0.
   44 TEMP1=FT(K)*(TF1-TF2-TF3)*DT*1.6E-16
       RETURN
       END
```

```
      =====  TEMP2  =====

      FUNCTION TEMP2(TIME,S,K)

       COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,EO,A2,A3
       COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &DEPX(50,200),DEL(200),X(200)
       COMMON /VAR/DT,DX,BB,NX
       COMMON/OUT/NT,TA(250),NY,Y(250)


      AA1=DEL(K)-S
      AA2=DEL(K)+S
      B1=2.*SQRT(ALPHA*(TIME-T(K)))
      TEMP2=POW(K)/DEL(K)*.5*(ERF(AA1/B1)+ERF(AA2/B1))*DT
      RETURN
      END
```

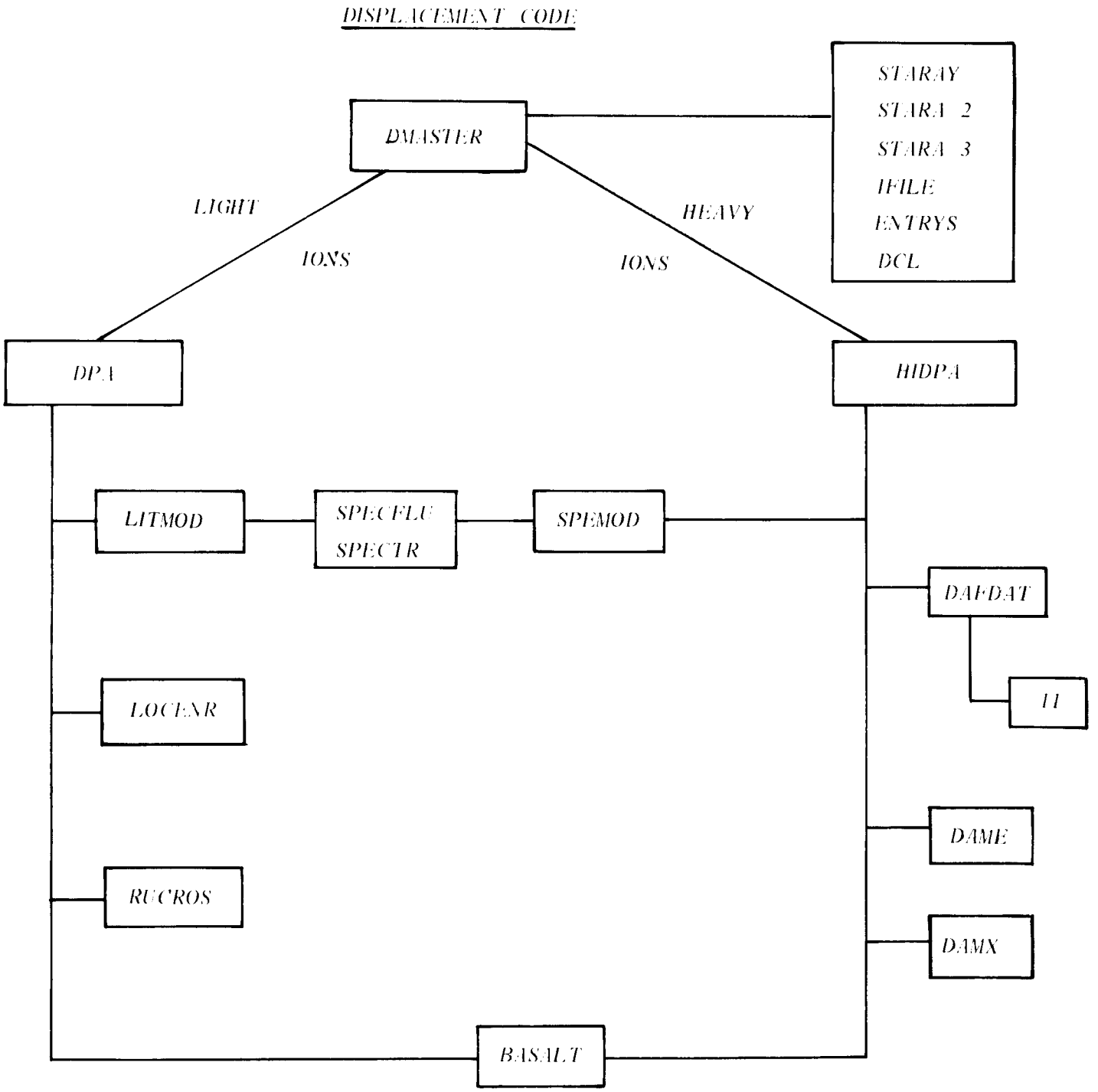
```
=====  TEMP3  =====

                FUNCTION TEMP3(TIME,S,K )

  REAL*4 PI/3.1415926/
  COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,EO,A2,A3
  COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
&DEPX(50,200),DEL(200),X(200)
  COMMON /VAR/DT,DX,BB,NX
  COMMON/OUT/NT,TA(250),NY,Y(250)
 COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
  COMMON/CONVER/NCONV,CONV



 Z=S/X2
 THETA=ALPHA*(TIME +W-T(K))/X2**2
 GAMMA=ALPHA*w/X2**2
 CONST=NPLUS
  CONV = 1.
 DELTO=0.
 DO 11 NN=1,NCONV
    FN=NN
    ARG=(2.*FN-1.)*PI/2.
    A1=COS(ARG*Z)
    A2=EXP(-ARG**2*THETA)
  A3=1./ARG
  A4=SIN(DEL(K)*ARG/X2)
    FI=ARG**2*GAMMA
    A5=(1.-EXP(-FI*CONST))/(1.-EXP(-FI))
    XX=A1*A2*A3*A4*A5
  DELTO = DELTO + XX
    CONV=XX/DELTO
11  CONTINUE
12   CONTINUE
 TEMP3=2.*DELTO*DT*POW(K) /DEL(K)
 RETURN
 END
```

```
      =====  TEMP4  =====

 1.                    FUNCTION TEMP4(TIME,S,K )
 2.
 3.
 4.           REAL*4 PI/3.1415926/
 5.           COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,EO,A2,A3
 6.           COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 7.          &DEPX(50,200),DEL(200),X(200)
 8.           COMMON /VAR/DT,DX,BB,NX
 9.           COMMON/OUT/NT,TA(250),NY,Y(250)
10.           COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
11.
12.
13.           B1=(C*1.E+4)*BB/T(K)/SQRT(EO)
14.           B2=(C*1.E+4)**2/2./EO
15.          Z=S/X2
16.          THETA=ALPHA*(TIME + W-T(K))/X2**2
17.          GAMMA=ALPHA*W/X2**2
18.          CONST=NPLUS
19.          DELTO=0.
20.          DO 11 NN=1,20
21.             FN=NN
22.             ARG=(2.*FN-1.)*PI
23.             A1=COS(ARG*Z)
24.             A2=EXP(-ARG**2*THETA)
25.           A3=B2*X2/ARG**2
26.           A4=1.-COS(ARG*B1/B2/X2)
27.             FI=ARG**2*GAMMA
28.             A5=(1.-EXP(-FI*CONST))/(1.-EXP(-FI))
29.             XX=A1*A2*A3*A4*A5
30.           IF(NN.GT.1) CONV=XX/DELTO
31.           DELTO = DELTO + XX
32.           IF (CONV .LE.  0.001) GO TO 12
33.     11    CONTINUE
34.     12     CONTINUE
35.           WRITE(-,-)NN
36.          TEMP4=DELTO*DT*FT(K)
37.          RETURN
38.          END
```

```
=====  TEMP5B  =====

      FUNCTION TEMP5(TIME,S,K)
      COMMON/MAT/IMODEL,ALPHA,RHO,CP,X2,C,EO,A2,A3
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &DEPX(50,200),DEL(200),X(200)
      COMMON /VAR/DT,DX,BB,NX
      COMMON/OUT/NT,TA(250),NY,Y(250)
      COMMON/PLYCOF/XM(200),XH(200),XL(200),COMATX(200,5,3),QQ1(200)
     &  ,INCID(200)
       REAL*8 F1,F2,F3,I1,IO,DERF,IOALT
      DOUBLE PRECISION Q00,Q10,Q11,Q20,Q21,Q22,Q30,Q31,Q32,Q33,Q40,
     &  Q41,Q42,Q43,Q44,M00,M01,M02,M03,M04,M11,M12,M13,M14,M22,M23,M24,
     &  M33,M34,M44,WN,WP,AWN
      REAL*8  SOL(5),ION(4),IOP(4),I1N(4),I1P(4),
     &       F1N(4),F1P(4),F2N(4),F2P(4),F3N(4),F3P(4)
       IO(X) = RTPI/2.D0 * DERF(X)
      I1(X)= -DEXP(-X**2)/2.
      F1(X)=X*DEXP(-X**2)
      F2(X)=X*F1(X)
      DATA RTPI/1.772454/
      F3(X)=X*F2(X)
       IOALT(X) = X - X**3/3. + X**5/10. - X**7/42.


C
C
C    DEFINE Q MATRIX FOR THIS TIME AND POSITION
C
      RELT=ABS(T(K)-TIME)
      A=2.*SQRT(RELT*ALPHA)
      Q00=1.
      Q10=S
      Q11=A
      Q20=A**2/2. + S**2
      Q21=A*S*2.
      Q22=-A**2/2.
      Q30=3.*A**2*S/2. +S**3
      Q31=A**3 + 3.*A*S**2
      Q32=-3.*A**2*S/2.
      Q33=-A**3/2.
      Q40=3.*A**4/4. + 6.*A**2*S**2/2. + S**4
      Q41=4.*A**3*S + 4.*A*S**3.
      Q42=-3.*A**4/4.  - 6.*A**2*S**2/2.
      Q43=-4.*A**3*S/2.
      Q44=-A**4/2.
      TTHIRD = 0.0
      TSECND = 0.0
      TFIRST = 0.0
      KKK = INCID(K)
      GO TO (10,20,30),KKK
10      CALL FUNLIM(0.0,2)
      CALL FUNLIM(XL(K),1)
      GO TO 15
20     CALL FUNLIM(0.0,3)
      CALL FUNLIM(XH(K),2)
      CALL FUNLIM(XL(K),1)
      GO TO 25
```

```
=====  TEMP5B  =====

      30      CALL FUNLIM(0.0,4)
              CALL FUNLIM(XM(K),3)
              CALL FUNLIM(XH(K),2)
              CALL FUNLIM(XL(K),1)
              GO TO 35
      C
      C     EVALUATE RESPONSE FOR EACH SECTION
      C
      35      CALL MATRIX(3,4)
              CALL MULMTX
      C       PRINT 479, K,(SOL(NM),NM = 1,5),(COMATX(K,IO,3),IO = 1,3)
              DO 37 IJK = 1,5
      37      TTHIRD = TTHIRD + COMATX(K,IJK,3) * SOL(IJK)
      C
      C
      25      CALL MATRIX(2,3)
              CALL MULMTX
      C       PRINT 479, K,(SOL(NM),NM = 1,5),(COMATX(K,IO,2),IO = 1,3)
              DO 27 IJK = 1,5
      27      TSECND = TSECND + COMATX(K,IJK,2) * SOL(IJK)
      C
      C
      15      CALL MATRIX(1,2)
              CALL MULMTX
      C       PRINT 479, K,(SOL(NM),NM = 1,5),(COMATX(K,IO,1),IO = 1,3)
              DO 17 IJK = 1,5
      17      TFIRST = TFIRST + COMATX(K,IJK,1) * SOL(IJK)
      C
      C
             TEMP5 = 1./RTPI * FT(K) * 1.6E-16 * DT * (TTHIRD + TSECND +TFIRST)
      C        PRINT 479,K,TTHIRD,TSECND,TFIRST,XM(K),XH(K),XL(K),
      479     FORMAT(I3,8E9.3)
              RETURN
      C
      C
              SUBROUTINE MATRIX(M,L)
              M00 = I0N(M) - I0N(L) + I0P(M) - I0P(L)
              M01 = I0N(M) - I0N(L) - I0P(M) + I0P(L)
              M02 = M00
              M03 = M01
              M04 = M00
              M11 = I1N(M) - I1N(L) + I1P(M) - I1P(L)
              M12 = I1N(M) - I1N(L) - I1P(M) + I1P(L)
              M13 = M11
              M14 = M12
              M22 = F1N(M) - F1N(L) + F1P(M) - F1P(L)
              M23 = F1N(M) - F1N(L) - F1P(M) + F1P(L)
              M24 = M22
              M33 = F2N(M) - F2N(L) + F2P(M) - F2P(L)
              M34 = F2N(M) - F2N(L) - F2P(M) + F2P(L)
              M44 = F3N(M) - F3N(L) + F3P(M) - F3P(L)
              RETURN
      C
      C
              SUBROUTINE FUNLIM(W,LIM)
              WN = (W-S)/A
              WP = (W+S)/A
```

```
=====  TEMP5B  =====

115.              AWN = DABS(WN)
116.              IF(AWN .GT. .3D0) GO TO 42
117.              ION(LIM) = IOALT(WN)
118.              GO TO 43
119.        42    ION(LIM) = IO(WN)
120.        43     IF(WP .GT. .3D0) GO TO 44
121.              IOP(LIM) = IOALT(WP)
122.              GO TO 45
123.        44    IOP(LIM) = IO(WP)
124.        45    CONTINUE
125.              I1N(LIM) = I1(WN)
126.              I1P(LIM) = I1(WP)
127.              F1N(LIM) = F1(WN)
128.              F1P(LIM) = F1(WP)
129.              F2N(LIM) = F2(WN)
130.              F2P(LIM) = F2(WP)
131.              F3N(LIM) = F3(WN)
132.              F3P(LIM) = F3(WP)
133.              RETURN
134.        C
135.        C
136.              SUBROUTINE MULMTX
137.              SOL(1) = Q00*M00
138.              SOL(2) = Q10*M01 + Q11*M11
139.              SOL(3) = Q20*M02 + Q21*M12 + Q22*M22
140.              SOL(4) = Q30*M03 + Q31*M13 + Q32*M23 + Q33*M33
141.              SOL(5) = Q40*M04 + Q41*M14 + Q42*M24 + Q43*M34 + Q44*M44
142.              RETURN
143.        C
144.        C
145.              END
```


DISPLACEMENT CODE
DMASTER
STARAY
STARA 2
STARA 3
IFILE
ENTRYS
DCL
LIGHT
IONS
HEAVY
IONS
DPA
HIDPA
LITMOD
SPECFLU
SPECTR
SPEMOD
DAFDAT
II
LOCENR
DAME
RUCROS
DAMX
BASALT

```
=====  DISPLACEMENT  =====

 1.
 2.
 3.
 4.
 5.
 6.                    DISPLACEMENT
 7.                SECTION OF T*DAMEN CODE
 8.
 9.
10.          ABSOLUTE
11.              DISPLACEMENT
12.
13.
14.          READER
15.              READ/DPA
16.
17.
18.          MAPPER
19.              MAP/DISPLACEMENT
20.
21.
22.          LISTER
23.              LIST/DISPLACEMENT
24.
25.
26.          SYMBOLICS
27.              DMASTER--A MASTER ROUTINE FOR CALLING EACH SUBROUTINE
28.              DPA     --CALCULATES LOW Z ION DPA RATES FOR STANDARD ARRAY
29.              RUCROS  --EVALUATES THE RUTHERFORD CROSSECTION OR LINDHARD
30.                        NUCLEAR DERIVED CROSS SECTION FOR IONS OF
31.                        ENERGY E
32.              SPECFLU--CREATES THE PARTICLE FLUXES FROM THE SPECTRA
33.              SPECTR  --GENERATES MAXWELLIAN OR GAUSSIAN SPECTRUM
34.              SPEMOD  --MODIFIES A HEAVY ION SPECTRUM VIA DIFFUSION
35.                        APPROXIMATION
36.              LITMOD  --MODIFIES A LIGHT ION SPECTRUM USING SLOWING
37.                        DOWN APPROXIMATION
38.              LOCENR  --GENERATES THE ENERGY AS A FUNCTION OF X FOR
39.                        AN ION OF INCIDENT ENERGY E*
40.              STARAY  --GENERATES THE STANDARD X AND T ARRAYS
41.              STARA2  --STANDARD ARRAY FOR 15 VALUES OF X(LIMIT IS READ IN)
42.                        AND 115 VALUES OF TIME (10-7--10-4)
43.              STARA3  --STANDARD ARRAY FOR 50 VALUES OF X(LIMIT IS READ IN)
44.                        AND 115 VALUES OF TIME (10-7--10-4)
45.              HIDPA   --CALCULATES DPA-RATES FOR HIGH Z IONS OR LOW ENERGY
46.                        LOW Z IONS
47.              BASALT  --TRANSFORMS ARBITRARY TIME BASE TO STANDARD TIME BASE
48.              DAFDAT  --READS DAMAGE FUNCTIONS FROM FILE 11 BY
49.                        A NUMBER WHICH INDICATES A CERTAIN ION-TARGET
50.                        COMBINATION
51.              DAME    --CALCULATES COEFFICIENTS OF DAMAGE FUNCTIONS
52.                        FOR A GIVEN VALUE OF ENERGY
53.              DAMX    --CALCULATES THE DAMAGE FUNCTION AT A POSITION
54.                        FOR THE COEFFICIENTS FOUND IN PREVIOUS CALL TO
55.                        DAME
56.              IFILE/IODR--FILING OF COMMON BLOCK  IN IODR FILE
57.              ENTRYS/IODR--ROUTINES FOR FILING MECHANICS
```

```
=====  READ/DISPLACEMENT  =====

                *****READ STATEMENT FOR THE DISPLACEMENT CODE*****


           FOR THE DMASTER SUBROUTINE:

      READ(5,10,END=99)INST
10    FORMAT(A6)

      IF(INST .EQ. 'I-SPEC') CALL SPECFL
      IF(INST .EQ. 'LITMOD') CALL LITMOD
      IF(INST .EQ. 'SPEMOD') CALL SPEMOD
      IF(INST .EQ. 'BASALT') CALL BASALT
      IF(INST .EQ. 'DISPLA') CALL DPA
      IF(INST .EQ. 'HIDPA') CALL HIDPA
      IF(INST .EQ. 'STARAY') CALL STARAY
      IF(INST .EQ. 'STARA2') CALL STARA2
      IF(INST .EQ. 'STARA3') CALL STARA3
      IF (INST .EQ. 'OPEN') CALL OPEN
      IF (INST .EQ. 'REOPEN') CALL REOPEN
      IF (INST .EQ. 'LOGIN') CALL LOGIN
      IF (INST .EQ. 'I-1FIL') CALL IFILE1
      IF (INST .EQ. 'I-2FIL') CALL IFILE2
      IF (INST .EQ. 'I-3FIL') CALL IFILE3


           FOR THE DPA SUBROUTINE:

      READ,M1,Z1,M2,Z2,ED


           FOR THE SPECFLU SUBROUTINE:

      READ(5,-,END = 11)ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS,IPRI
      READ,NMHIST,(EHIST(I),AMP(I), I = 1,NMHIST)
      READ,IMODEL,ALPHA,RHO,CP,X2,NX,C,EO,A2,A3,
      READ,E1,E2,S1,S2,EM


           FOR THE SPEMOD SUBROUTINE:

      READ 377, INST1
      READ 377,INST2
377     FORMAT(A4)
      READ ,E1,R1,SIG1,E2,R2,SIG2,REFP,REFT
      READ , RBUF,GASP,GAST,IPRI


           FOR THE LITMOD SUBROUTINE:

      READ 729,INST1
      READ 729,INST2
729    FORMAT(A4)
      IF(INST2 .NE. 'RANG') GO TO 621
      IF(INST2 .NE. 'RANG') GO TO 92
      READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
      READ , REFP,REFT
```

```
                =====  READ/DISPLACEMENT  =====

58.              READ ,RBUF,GASP,GAST,IPRI
59.
60.
61.                 FOR THE LOCENR SUBROUTINE:
62.
63.              READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
64.
65.
66.                 FOR THE HIDPA SUBROUTINE:
67.
68.              READ , LOCDFN,CONVER
69.              CALL DAFDAT(LOCDFN)
70.
71.
72.                 FOR THE DAFDAT/DEPFUN SUBROUTINE:
73.
74.               REWIND  11
75.       5       READ (11,END = 10)
76.       777    READ(11,END = 100) LOC,IDENT,CG
```

```
      =====  MAP/DISPLACEMENT  =====

 1.   @MAP*MAP.MAP     ,T*DAMEN.DISPLACEMENT
 2.   IN T*DAMEN.SPECFLU,.RUCROS,.SPECTR
 3.   IN T*DAMEN.DPA,.STARAY,.STARA2,.STARA3,.BASALT
 4.   NOT T*DAMEN.STARA2/PRINT,.STARAY/PRINT
 5.   IN T*DAMEN.IFILE/IODR,.ENTRYS/IODR
 6.   IN T*DAMEN.DMASTER
 7.   IN T*DAMEN.LOCENR,.SPEMOD,.LITMOD
 8.   IN T*DAMEN.HIDPA,.DAFDAT/DEPFUN
 9.   IN T*DAMEN.DAME/DEPFUN,.DAMX/DEPFUN
10.   END
11.   @EOF
```

===== LIST/DISPLACEMENT =====

```
 1.     @PRT,S  T*DAMEN.LIST/DISPLACEMENT
 2.     @PRT,S  T*DAMEN.MAP/DISPLACEMENT
 3.     @PRT,S  T*DAMEN.DMASTER
 4.     @PRT,S  T*DAMEN.DPA
 5.     @PRT,S  T*DAMEN.RUCROS
 6.     @PRT,S  T*DAMEN.SPECFLU
 7.     @ .     SEE ION
 8.     @PRT,S  T*DAMEN.SPECTR
 9.     @ .     SEE ION
10.     @PRT,S  T*DAMEN.SPEMOD
11.     @ .     SEE ION
12.     @PRT,S  T*DAMEN.LITMOD
13.     @ .     SEE ION
14.     @PRT,S  T*DAMEN.LOCENR
15.     @PRT,S  T*DAMEN.STARAY
16.     @ .     SEE PHOTON
17.     @PRT,S  T*DAMEN.STARA2
18.     @ .     SEE ION
19.     @PRT,S  T*DAMEN.STARA3
20.     @ .     SEE ION
21.     @PRT,S  T*DAMEN.HIDPA
22.     @PRT,S  T*DAMEN.BASALT
23.     @ .     SEE ION
24.     @PRT,S  T*DAMEN.DAFDAT/DEPFUN
25.     @ .     SEE DEPOSITION FUNCTION CREATION
26.     @PRT,S  T*DAMEN.DAME/DEPFUN
27.     @ .     SEE DEPOSITION FUNCTION CREATION
28.     @PRT,S  T*DAMEN.DAMX/DEPFUN
29.     @ .     SEE DEPOSITION FUNCTION CREATION
30.     @PRT,S  T*DAMEN.IFILE/IODR,.ENTRYS/IODR
31.     @ .             SEE ION,    SEE FILING
```

```
=====  DMASTER  =====

 1.
 2.      C       ****************************************
 3.
 4.      C       THIS IS THE DRIVER ROUTINE FOR THE WALL LOADING STUDY
 5.
 6.      C       ********************************************************
 7.
 8.              INTEGER INST
 9.
10.      1       CONTINUE
11.      C       FIND OUT WHAT TO DO
12.
13.              READ(5,10,END=99)INST
14.              PRINT 1000,INST
15.      1000    FORMAT(' ',A6)
16.      10      FORMAT(A6)
17.
18.      C       GO DO IT
19.              IF(INST .EQ. 'I-SPEC') CALL SPECFL
20.              IF(INST .EQ. 'LITMOD') CALL LITMOD
21.              IF(INST .EQ. 'SPEMOD') CALL SPEMOD
22.              IF(INST .EQ. 'BASALT') CALL BASALT
23.              IF(INST .EQ. 'DISPLA') CALL DPA
24.              IF(INST .EQ. 'HIDPA') CALL HIDPA
25.              IF(INST .EQ. 'STARAY') CALL STARAY
26.              IF(INST .EQ. 'STARA2') CALL STARA2
27.              IF(INST .EQ. 'STARA3') CALL STARA3
28.              IF (INST .EQ. 'OPEN') CALL OPEN
29.              IF (INST .EQ. 'REOPEN') CALL REOPEN
30.              IF (INST .EQ. 'LOGIN') CALL LOGIN
31.              IF (INST .EQ. 'I-1FIL') CALL IFILE1
32.              IF (INST .EQ. 'I-2FIL') CALL IFILE2
33.              IF (INST .EQ. 'I-3FIL') CALL IFILE3
34.
35.              GO TO 1
36.
37.      99      STOP
38.              END
```

```
            =====  DPA  =====

 1.            SUBROUTINE DPA
 2.           COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 3.          &          DEPX(50,200),DEL(200),X(200),FF(200)
 4.            COMMON/STOP/E0,S0,E1,S1,E2,S2,E3,S3,E4,S4,SMAX,B0,A0,A00,B00
 5.          1 ,EINT,SINT
 6.            COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
 7.           COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
 8.           COMMON/VAR/DT,DX,BB,NX
 9.           DIMENSION ENER(50,250),DISP(50,200),TDISP(50,250),TOTAL(50)
10.            REAL M1,M2
11.            EQUIVALENCE (TEMP,ENER),(DEPX,DISP),(TEMP,TDISP)
12.      C
13.      C    INITIALIZE THE DEPX AND TEMP ARRAYS
14.      C
15.              NX = NSX
16.              DO 278 I = 1,50
17.              X(I) = SX(I)
18.              DO 277 J = 1,200
19.              DISP(I,J) = 0.
20.      277        TDISP(I,J ) = 0.
21.              DO 278 J = 201,250
22.      278      TDISP(I,J) = 0.
23.      C
24.              READ,M1,Z1,M2,Z2,ED
25.      C
26.      C
27.      C    COMPUTE LOCAL ENERGY
28.      C
29.              CALL LOCENR
30.      C
31.      C    COMPUTE DISPLACEMENTS AT EACH LOCATION FOR EACH ION
32.      C
33.              DO 30 I = 1,NSX
34.      30       TOTAL(I) = 0.
35.              DO 100 J = 1,NE
36.              DO 50 I = 1,NSX
37.              IF(ENER(I,J) .LE. 0.)GO TO 41
38.              CALL RUCROS(ENER(I,J),M1,Z1,M2,Z2,ED,SIGDIS,SIG2)
39.              DISP(I,J) = SIG2*FT(J) * 1.E-24
40.              TOTAL(I) = TOTAL(I) + DISP(I,J)* DT
41.      41      TDISP(I,J) = TOTAL(I)
42.      50      CONTINUE
43.      100      CONTINUE
44.      C
45.      C
46.      C       PRINT SOME OUTPUT
47.      C
48.              PRINT 498,(SX(K),K = 1,12)
49.      498      FORMAT(//,'   DISPLACEMENT RATE AT RESPECTIVE LOCATIONS',
50.           1  //,'     X =   ',12E10.3,//,'  TIME',/)
51.              DO 200 J = 1,NE
52.              PRINT  500,T(J),(DISP(I,J),I = 1,12)
53.      500      FORMAT(1X,13E10.3)
54.      200     CONTINUE
55.              PRINT 499,(SX(K),K = 1,12)
56.      499      FORMAT(//,'   DISPLACEMENT  AT RESPECTIVE LOCATIONS',
57.           1  //,'     X =   ',12E10.3,//,'  TIME',/)
```

```
=====  DPA  =====

58.              DO 300 J = 1,NE
59.              PRINT  500,T(J),(TDISP(I,J),I = 1,12)
60.      300      CONTINUE
61.              END
```

```
=====  HIDPA  =====

      SUBROUTINE HIDPA
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/STOP/E0,S0,E1,S1,E2,S2,E3,S3,E4,S4,SMAX,B0,A0,A00,B00,B,PP
      COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
      COMMON/VAR/DT,DX,BB,NX
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
      DIMENSION ENER(50,250),DISP(50,200),TDISP(50,250),TOTAL(50)
      EQUIVALENCE (TEMP,ENER),(DEPX,DISP),(TEMP,TDISP)
C
C    INITIALIZE THE DEPX AND TEMP ARRAYS
C
         NX = NSX
         DO 278 I = 1,50
         X(I) = SX(I)
         DO 277 J = 1,200
         DISP(I,J) = 0.
277         TDISP(I,J ) = 0.
         DO 278 J = 201,250
278       TDISP(I,J) = 0.
C
         READ , LOCDFN,CONVER
         CALL DAFDAT(LOCDFN)
C
C
C    COMPUTE DISPLACEMENTS AT EACH LOCATION FOR EACH ION
C
         DO 30 I = 1,NSX
30        TOTAL(I) = 0.
         DO 100 J = 1,NE
         CALL DAME(E(J))
         DO 50 I = 1,NSX
         SIG2 = DAMX(X(I)) * CONVER
         DISP(I,J) = SIG2*FT(J) * 1.E-24
         IF(DISP(I,J) .LE. 0.0) DISP(I,J) = 0.0
         TOTAL(I) = TOTAL(I) + DISP(I,J)* DT
         TDISP(I,J) = TOTAL(I)
50       CONTINUE
100       CONTINUE
C
C
C        PRINT SOME OUTPUT
C
         PRINT 498,(SX(K),K = 1,12)
498       FORMAT(//,'   DISPLACEMENT RATE AT RESPECTIVE LOCATIONS',
     1   //,'     X =   ',12E10.3,//,'  TIME',/)
         DO 200 J = 1,NE
         PRINT  500,T(J),(DISP(I,J),I = 1,12)
500       FORMAT(1X,13E10.3)
200      CONTINUE
         PRINT 499,(SX(K),K = 1,12)
499       FORMAT(//,'   DISPLACEMENT  AT RESPECTIVE LOCATIONS',
     1   //,'     X =   ',12E10.3,//,'  TIME',/)
         DO 300 J = 1,NE
         PRINT  500,T(J),(TDISP(I,J),I = 1,12)
300       CONTINUE
         END
```

```
            =====  LOCENR  =====

 1.             SUBROUTINE LOCENR
 2.            COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
 3.           &          DEPX(50,200),DEL(200),X(200),FF(200)
 4.             COMMON/STOP/E0,S0,E1,S1,E2,S2,E3,S3,E4,S4,SMAX,B0,A0,A00,B00
 5.           1,EINT,SINT
 6.             COMMON/INTREQ/EFINAL,QQ1(200),XL(200),XH(200),XM(200),INCID(200)
 7.             COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
 8.             COMMON/DEPTIM/STANFT(200),STANPW(200),STANE(200),SDEL(200),
 9.           1 ISTART,ISTOP
10.            COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
11.             DIMENSION ENER(50,250)
12.             EQUIVALENCE (TEMP,ENER)
13.
14.      C       **   INPUT
15.             READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
16.      C   NEW  PARAMETERS FOR REGION 1
17.      C
18.      C
19.             A00 = S0**2/(2.*S0 - S1)
20.            B00 = -ALOG(1. - S0/A00)/E0
21.      C
22.      C
23.
24.
25.      C       ** CONSTANTS ARE CALCULATED
26.             B0=(E3-E2)/ALOG(S2/S3)
27.             A0=S2*EXP(E2/B0)
28.      C
29.      C     FIND THE INTERSECTION OF CURVE 2 AND CURVE 3
30.      C
31.             SMDIF = 1.E+6
32.             ETR = E1
33.             DTE = (E2-E1)/100.
34.             DO 427  I = 1,100
35.             FORM2 = A00 *(1. - EXP(-B00 *ETR))
36.             FORM3 = A0 * EXP(-ETR/B0)
37.             DIFT = ABS(FORM3 - FORM2)
38.             CSM = SMDIF
39.            SMDIF = AMIN1(SMDIF,DIFT)
40.             CDIF = CSM - SMDIF
41.             IF(CDIF .LE. 0.) GO TO 428
42.             ETR = ETR + DTE
43.      427      CONTINUE
44.      428      CONTINUE
45.             EINT = ETR
46.             SINT = A0*EXP(-EINT/B0)
47.      C
48.      C      PRINT THE IN PUT AND THE INTERSECTION POINTS
49.      C
50.             PRINT 111,E0,S0,E1,S1,E2,S2,E3,S3,EINT,SINT,SMAX
51.      111      FORMAT(///,8X,'E0 = ',F9.2,5X,'S0 = ',F9.2,/,
52.           1               8X,'E1 = ',F9.2,5X,'S1 = ',F9.2,/,
53.           2               8X,'E2 = ',F9.2,5X,'S2 = ',F9.2,/,
54.           3               8X,'E3 = ',F9.2,5X,'S3 = ',F9.2,/,
55.           3               6X,'EINT = ',F9.2,3X,'SINT = ',F9.2,/,
56.           48X,'SMAX = ',F9.2,///)
57.      C
```

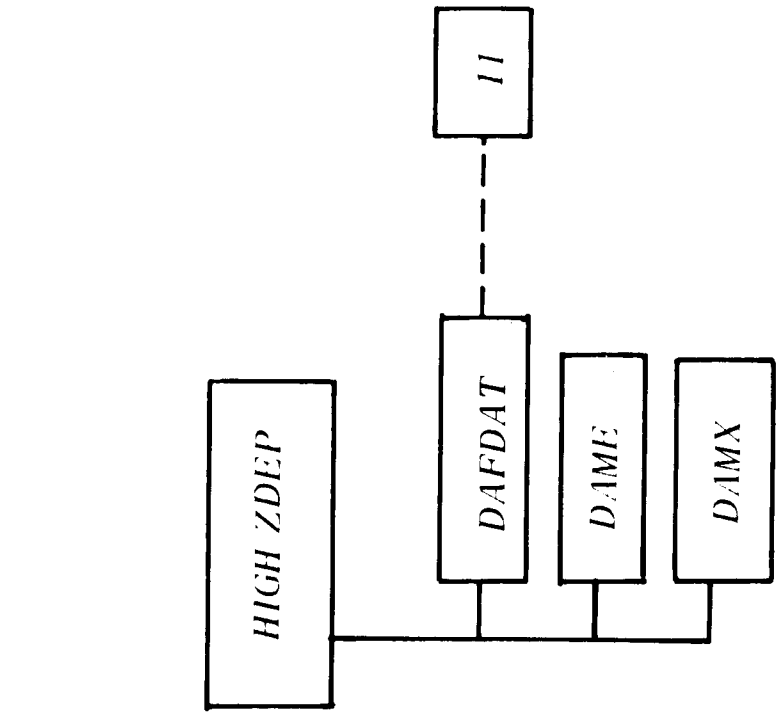
```
      =====  LOCENR  =====

58.      C    DEFINE THE REGION WHERE STOPPING POWER CHANGES
59.      C
60.              EBOTH = 10. * A
61.      C
62.      C     FIND LOCAL ENERGY FOR EACH VALUE OF IMPACT ENERGY
63.      C
64.              DO 25 J = 1,NE
65.              ENER(1,J) = E(J)
66.      C
67.      C   SET UP XM,XH,XL AND PARAMETERS FOR EACH INCIDENT ENERGY
68.      C
69.              CALL DIVENG( E(J),J)
70.      C
71.      C     FIND ENERGY AT EACH STANDARD VLAUE OF X
72.      C
73.              PRINT , J,E(J),XM(J),XH(J),XL(J),INCID(J)
74.              DO  20 I = 1,NSX
75.              H = SX(I) * 1.E+4
76.               IF(H .LT. XM(J) ) GO TO 3
77.               IF(H .LT. XH(J) ) GO TO 4
78.               IF(H .LT. XL(J) ) GO TO 5
79.               ENER(I,J) = 0.0
80.              GO TO 7
81.      3       EBEGIN = E(J)
82.              IREG = 3
83.              GO TO 6
84.      4       EBEGIN  = AMIN1(E(J),EINT)
85.              IREG = 2
86.              H = H - XM(J)
87.              GO TO 6
88.      5       EBEGIN = AMIN1(E(J),EO)
89.              IREG = 1
90.              H = H - XH(J)
91.      6       CONTINUE
92.              ENER(I,J) = ENERGY(EBEGIN,H,IREG)
93.      7       CONTINUE
94.      20      CONTINUE
95.      25      CONTINUE
96.              PRINT 502,(SX(I),I = 1,12)
97.              DO 500 J = 1,NE
98.      500      PRINT 501,(T(J),(ENER(I,J),I = 1,12))
99.      501      FORMAT(1X,13E10.3)
100.     502      FORMAT(//,31X,'ENERGY(KEV) AT RESPECTIVE LOCATIONS',//,
101.          1    '      X =  ',1X,12E10.3,//'  TIME  ',/)
102.             RETURN
103.     C
104.             FUNCTION ENERGY(ESTAR,SSS,M)
105.             GO TO (1,2,3),M
106.     1       ENERGY = ESTAR - SO*SSS*SQRT(ESTAR/EO) + (SO*SSS)**2/4./EO
107.             IF(ESTAR .LT. EBOTH) GO TO 2846
108.             ZBOTH = 2.*SQRT(EO)/SO  *(SQRT(ESTAR) - SQRT(EBOTH))
109.             IF(SSS .GT. ZBOTH) ENERGY = EBOTH - SO*SQRT(EBOTH/EO)*(SSS-ZBOTH)
110.             GO TO 5
111.     2846     ENERGY = ESTAR - SO*SQRT(EBOTH/EO)*SSS
112.             GO TO 5
113.     2       ENERGY = ALOG(1. + EXP(BOO*(QQ1(J) - AOO*SSS)))/BOO
114.             GO TO 5
```

```
            =====  LOCENR  =====

115.     3       ENERGY = BO*ALOG(EXP(ESTAR/BO)- AO*SSS/BO)
116.     5       CONTINUE
117.             RETURN
118.     C
119.     C
120.              SUBROUTINE DIVENG(ENER,I)
121.     C
122.     C    UNITS ARE KEV,  MICRON, AND KEV/MICRON
123.     C
124.             IF(ENER .GE. EINT)GO TO 1001
125.             IF(ENER .LT. EINT .AND. ENER .GT. E0) GO TO 1002
126.             IF(ENER .LE. E0) GO TO 1004
127.     C
128.     C******************** E GREATER THAN EINT********************        *
129.     1001     CONTINUE
130.     C
131.     C
132.     C      ION OF INCIDENT ENERGY IN REGION 3
133.              INCID(I) = 3
134.              XM(I) = BO/AO *(EXP(ENER/BO) - EXP(EINT/BO))
135.              PAREN = 1. - EXP( -BOO*EINT)
136.              QQ1(I) = EINT + ALOG(PAREN)/BOO
137.              XEND2 = (QQ1(I) - E0 -ALOG(1.- EXP(-E0*BOO))/BOO)/AOO
138.              XH(I) = XM(I) + XEND2
139.              XL(I)  = XH(I) + 2.*E0/S0 - SQRT(E0*EBOTH)/S0
140.              GO TO 1006
141.     C
142.     C***************************** E BETWEEN E0 AND EINT********
143.     C
144.     1002     CONTINUE
145.     C
146.     C     ION OF INCIDENT ENERGY IN REGION 2
147.     C
148.              INCID(I) = 2
149.              XM(I) = 0.
150.     C
151.              PAREN = 1. - EXP( -BOO*ENER)
152.              QQ1(I) = ENER + ALOG(PAREN)/BOO
153.              XEND2 = (QQ1(I) - E0 -ALOG(1.- EXP(-E0*BOO))/BOO)/AOO
154.              XH(I) = XM(I) + XEND2
155.     C
156.              XL(I)  = XH(I) + 2.*E0/S0 - SQRT(E0*EBOTH)/S0
157.              GO TO 1006
158.     C
159.     C********************  E LESS THAN E0***************
160.     1004      CONTINUE
161.     C
162.     C
163.     C      IOF ON INCIDENT ENERGY IN REGION 1
164.     C
165.              INCID(I) = 1
166.              XM(I) = 0.
167.              XH(I) = 0.
168.              XL(I) = XH(I) + 2. * SQRT(ENER*E0)/S0 - SQRT(E0*EBOTH)/S0
169.              GO TO 1006
170.     1006      CONTINUE
171.              RETURN
```

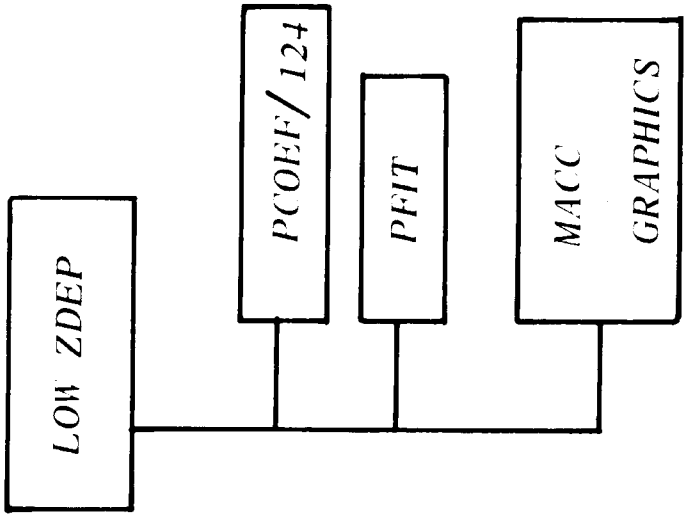
```
=====  LOCENR  =====

172.          END
```

```
=====  RUCROS  =====

      SUBROUTINE RUCROS(E,M1,Z1,M2,Z2,ED,SIGDIS,SIG2)
      REAL M1,M2,K
      PART(T)=1.+K*T/EL+0.40244*K*(T/EL)**.750+3.4008*K*(T/EL)**.166667
      ER = .0136
       C = 3.E+10
      EL = 0.08693*Z2**2.3333333
      K = 0.1337*Z2**.666667/SQRT(M2)
1     V = 4.38E+7*SQRT(E/M1)
      CON = 1.
      B = V/C
      Y = 100.*B/Z1**.6667
      G = 1. - EXP(-1.316*Y + .112*Y**2 -.065*Y**3)
      Z11 = Z1 *G
      EMAX = E*4.*M1*M2/(M1+M2)**2
      EAV = ED * LOG(EMAX/ED)/(1-ED/EMAX)
      BBB= 4*3.14159*(.53E-8)**2 *M1 * Z1**2* Z2**2*ER**2 /M2
      FAC = (E/EL)**2*CON
      EAV2 = 2.*ED * LOG(EMAX/2./ED)/(1-2.*ED/EMAX)
      SIGDIS = BBB*G**2/E * (.4*ALOG(EMAX/2./ED)/PART(CON*EAV2)+.5)/ED
      TED = 2.*ED
     IF(EMAX .LT.TED .AND. EMAX.GT.ED)SIGDIS = BBB*G**2/E*(1/ED-1/EMAX)
      IF(EMAX .LE. ED) SIGDIS = 0.
C    CHANGE TO BARNS
      SIGDIS = SIGDIS*1.E+24
      GG0 = PART(2.*ED)
      GG1 = PART(EAV)
      GG2 = PART(EMAX)
C   COMPUTE THE GENERAL MODEL FOR ALLENERGIES
C
      P = 2./3.
      A0 = 0.4683/SQRT(Z1**P + Z2**P)*1.E-8
      EE = 4.803E-10
      A = M2/M1
      ELIND = (1+A)/A * Z1*Z2 *EE**2/A0/1.602E-9
      RN = (1+A)**2/(4.*A*3.14159*A0**2)
C
C   UNITS ARE NOW KEV AND CM
      EP = E/ELIND
      F9 = 4./9.
      U = 1.37823 * EP**F9
      PAREN = SQRT( 1. + U**2)
      TERM1 = ALOG(U + PAREN)
      TERM2 = U /PAREN
      GM = EP*SQRT(2.*ED/EMAX)
      U = 1.37823 * GM**F9
      PAREN = SQRT( 1. + U**2)
      TERM3 = ALOG(U + PAREN)
      TERM4 = U /PAREN
      DEDRHO = 9. * (TERM1 -TERM2 - TERM3 + TERM4)/8./EP
      IF(EMAX .LT. TED) DEDRHO = 0.
      T2 = 3./2.
      F3 = 4./3.
      TT = EP**2*1.5*ED/EMAX
      DT = TT/1.5
      IF(EMAX .LT.TED .AND.EMAX .GT. ED)TT = EP**2*(ED+EMAX)/2./EMAX
      IF(EMAX .LT. TED .AND. EMAX .GT. ED) DT = EP**2/EMAX*(EMAX-ED)
     ADD = 3.14159*A0**2/2.* 1.309/TT**F3/(1. + 1.8995*TT**F9)**T2*DT
```

```
=====  RUCROS  =====

      IF(EMAX .LE. ED) ADD = 0.
      ECROS = DEDRHO*ELIND/RN
      SIG2 = ECROS* .8/2./ED *1.E+24
      SIG2 = SIG2 + ADD * 1.E+24
C
C
C   DPA CROSS SECTION FORM THE GENERAL MODEL
      RETURN
      END
```


11
HIGH ZDEP
DAFDAT
DAME
DAMX

DEPOSITION


LOW ZDEP
PCOEF/124
PFIT
MACC
GRAPHICS

```
=====  DEPOSITION  =====

 1.
 2.
 3.
 4.
 5.
 6.
 7.                    DEPOSITION
 8.                SECTION OF T*DAMEN CODE
 9.
10.
11.          ABSOLUTE
12.              LOWZDEP--LOW Z IONS CODE, MODEL 1 (LINEAR), MODEL 2 (CUBIC),
13.                       MODEL 4 (QUART), PLOTS CALCULATES ELECTRONIC
14.                       STOPPING POWER, SPATIAL DEPOSITION AND LOCAL
15.                       MEAN ENERGY FOR LIGHT IONS
16.              HIGHZDEP-HIGH Z IONS CORE, BASED ON DEPOSITION FUNCTIONS
17.
18.
19.          READER
20.              READ/DEPOSITION
21.
22.
23.          MAPPER
24.              MAP/LOWZDEP
25.              MAP/HIGHZDEP
26.
27.
28.          LISTER
29.              LIST/DEPOSITION
30.
31.
32.          SYMBOLICS
33.              PCOEF/124--GENERATES POLYNOMIAL COEFFICIENTS FOR SET
34.                         OF INCIDENT ENERGIES
35.              LOWZDEP--MAIN ROUTINE CALCULATES NUCLEAR STOPPING POWER,
36.                       AND DEPOSITION FROM SET OF COEFFICIENT
37.              PFIT    --A GENERAL 1-4TH ORDER POLYNOMIAL INTERPOLATION
38.                       ROUTINE
39.              HIGHZDEP-MAIN ROUTINE FOR EVALUATING ENERGY DEPO-
40.                       SITION FROM COEFFICIENTS IN FILE 11
41.              DAFDAT  --READS DAMAGE FUNCTIONS FROM FILE 11 BY A
42.                       NUMBER WHICH INDICATES A CERTAIN ION-TARGET
43.                       COMBINATION
44.              DAME    --CALCULATES COEFFICIENTS OF DAMAGE FUNCTIONS
45.                       A GIVEN VALUE OF ENERGY
46.              DAMX    --CALCULATES THE DAMAGE FUNCTION AT A POSITION
47.                       X FOR THE COEFFICIENT FOUND IN PREVIOUS CALL
48.                       TO DAME
49.
50.
51.          RUNSTREAMS
52.
53.
```

```
=====  READ/DEPOSITION  =====

                    *****READ STATEMENTS FOR THE DEPOSITION CODE*****


                FOR THE LOWZDEP SUBROUTINE:

              READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
              READ,  Z,AP,AN,M1,RHO,M2,Z1,Z2
              READ, IND1,IND2,IND3,IPLOT,IPRINT
              READ,  FIRSTE,FINE
              READ,NUM,(X3(J),J = 1,NUM)


                FOR THE HIGHZDEP SUBROUTINE:

              READ , LOCDFN,LOCDFE,CONV
              READ, NUM,(E(I),I = 1,NUM)
              READ , IPRINT,IPLOT


                FOR THE DAFDAT SUBROUTINE:

      5        READ (11,END = 10)
      777     READ(11,END = 100) LOC,IDENT,CG
```

```
=====  MAP/LOWZDEP  =====

1.      @MAP*MAP.MAP    ,T*DAMEN.LOWZDEP
2.      IN T*DAMEN.LOWZDEP,.PCOEF/124
3.      IN T*DAMEN.PFIT
4.      END
```

```
=====  MAP/HIGHZDEP  =====

1.    @MAP*MAP.MAP    ,T*DAMEN.HIGHZDEP
2.    IN T*DAMEN.HIGHZDEP
3.    IN T*DAMEN.DAFDAT
4.    IN T*DAMEN.DAME/DEPFUN,.DAMX/DEPFUN
5.    END
```

```
=====  LIST/DEPOSITION  =====

 1.     @PRT,S  T*DAMEN.LIST/DEPOSITION
 2.     @PRT,S  T*DAMEN.MAP/DEPOSITION
 3.     @PRT,S  T*DAMEN.PCOEF/124
 4.     @ .     SEE ION
 5.     @PRT,S  T*DAMEN.LOWZDEP
 6.     @PRT,S  T*DAMEN.PFIT
 7.     @ .     SEE ION
 8.     @PRT,S  T*DAMEN.HIGHZDEP
 9.     @PRT,S  T*DAMEN.DAFDAT/DEPFUN
10.     @ .     SEE DEPOSITION FUNCTION CREATION
11.     @PRT,S  T*DAMEN.DAME/DEPFUN
12.     @ .     SEE DEPOSITION FUNCTION CREATION
13.     @PRT,S  T*DAMEN.DAMX/DEPFUN
14.     @ .     SEE DEPOSITION FUNCTION CREATION
```

```
=====  HIGHZDEP  =====

 1.            DIMENSION E(100),X(100),DEP(100),DEPE(100),DEPSUM(100)
 2.            COMMON /XFL/C(20),NC
 3.            READ , LOCDFN,LOCDFE,CONV
 4.            READ, NUM,(E(I),I = 1,NUM)
 5.            READ , IPRINT,IPLOT
 6.      C
 7.      C    GET COEFFICIENTS FROM FILE 11
 8.      C
 9.            CALL DAFDAT(LOCDFN)
10.      C
11.      C
12.      C     CALCULATE THE DEPOSITION PROFILES FOR EACH INCIDENT ENERGY
13.      C
14.            DO 25 J = 1,NUM
15.            CALL DAME(E(J))
16.            DX  = C(2)/50.
17.            X(1) = 0.
18.            DO 50 I = 1,51
19.            DEP(I) = DAMX(X(I)) * CONV
20.            X(I+1)  = X(I) + DX
21.      50     CONTINUE
22.      25     CONTINUE
23.      C
24.      C    GET COEFFICIENTS FROM FILE 11
25.      C   FOR ELECTRONIC STOPING
26.      C
27.            CALL DAFDAT(LOCDFE)
28.      C
29.      C
30.      C     CALCULATE THE DEPOSITION PROFILES FOR EACH INCIDENT ENERGY
31.      C
32.            DO 125 J = 1,NUM
33.            CALL DAME(E(J))
34.            DO 150 I = 1,51
35.            DEPE(I) = DAMX(X(I)) * CONV
36.            DEPSUM(I) = DEPE(I) + DEP(I)
37.      150     CONTINUE
38.      125     CONTINUE
39.      C
40.      C    PRINT THE OUTPUT
41.      C
42.            IF(IPRINT.NE. 1) GO TO 81
43.            DO 225 J = 1,NUM
44.            PRINT 55, E(J)
45.            DO 58 I = 1,51
46.            PRINT 56, I,X(I),DEP(I),DEPE(I),DEPSUM(I)
47.      58      CONTINUE
48.      55     FORMAT(//,15X,' ENERGY DEPOSITION',//,
49.           1  5X,'ION DEPOSITION',5X,F10.2,//,5X,
50.           2 'LOCATION',7X,'NUCLR(KEV/MICRON)','   ELEC          TOTAL  '   ,//)
51.      56     FORMAT(1X,I2,3X,4E15.3)
52.      225     CONTINUE
53.      81     CONTINUE
54.            END
```

```
            =====  LOWZDEP  =====

 1.            REAL*4 F1(100),F2(100)
 2.           DIMENSION EP(100),DEDX(100), X2(100),X3(100),F3(100),YY(100)
 3.           DIMENSION X(100,10),F4(100),F5(100),ENER(100,10),PLTE(100)
 4.          1  ,XPT(100)
 5.           REAL M1,M2
 6.            COMMON/STOP/E0,S0,E1,S1,E2,S2,E3,S3,E4,S4,SMAX,B0,A0,A00,B00
 7.          1  ,EINT,SINT
 8.            COMMON/PLYCOF/XM(200),XH(200),XL(200),COMATX(200,5,3),QQ1(200)
 9.          &  ,INCID(200)
10.
11.     C       **   INPUT
12.            READ,E0,S0,E1,S1,E2,S2,E3,S3,SMAX
13.           READ,  Z,AP,AN,M1,RHO,M2,Z1,Z2
14.            READ, IND1,IND2,IND3,IPLOT,IPRINT
15.            READ,  FIRSTE,FINE
16.            CALL DEVSET('PLTTR')
17.            READ,NUM,(X3(J),J = 1,NUM)
18.     C   NEW  PARAMETERS FOR REGION 1
19.     C
20.     C
21.            A00 = S0**2/(2.*S0 - S1)
22.           B00 = -ALOG(1. - S0/A00)/E0
23.     C
24.     C
25.
26.
27.     C       ** CONSTANTS ARE CALCULATED
28.            B0=(E3-E2)/ALOG(S2/S3)
29.            A0=S2*EXP(E2/B0)
30.     C
31.     C     FIND THE INTERSECTION OF CURVE 2 AND CURVE 3
32.     C
33.            SMDIF = 1.E+6
34.            ETR = E1
35.            DTE = (E2-E1)/100.
36.            DO 427  I = 1,100
37.            FORM2 = A00 *(1. - EXP(-B00 *ETR))
38.            FORM3 = A0 * EXP(-ETR/B0)
39.            DIFT = ABS(FORM3 - FORM2)
40.            CSM = SMDIF
41.           SMDIF = AMIN1(SMDIF,DIFT)
42.            CDIF = CSM - SMDIF
43.            IF(CDIF .LE. 0.) GO TO 428
44.            ETR = ETR + DTE
45.     427      CONTINUE
46.     428      CONTINUE
47.           EINT = ETR
48.            SINT = A0*EXP(-EINT/B0)
49.     C
50.     C      PRINT THE IN PUT AND THE INTERSECTION POINTS
51.     C
52.            PRINT 111,E0,S0,E1,S1,E2,S2,E3,S3,EINT,SINT,SMAX
53.     111      FORMAT(///,8X,'E0 = ',F9.2,5X,'S0 = ',F9.2,/,
54.          1              8X,'E1 = ',F9.2,5X,'S1 = ',F9.2,/,
55.          2              8X,'E2 = ',F9.2,5X,'S2 = ',F9.2,/,
56.          3              8X,'E3 = ',F9.2,5X,'S3 = ',F9.2,/,
57.          3              6X,'EINT = ',F9.2,3X,'SINT = ',F9.2,/,
```

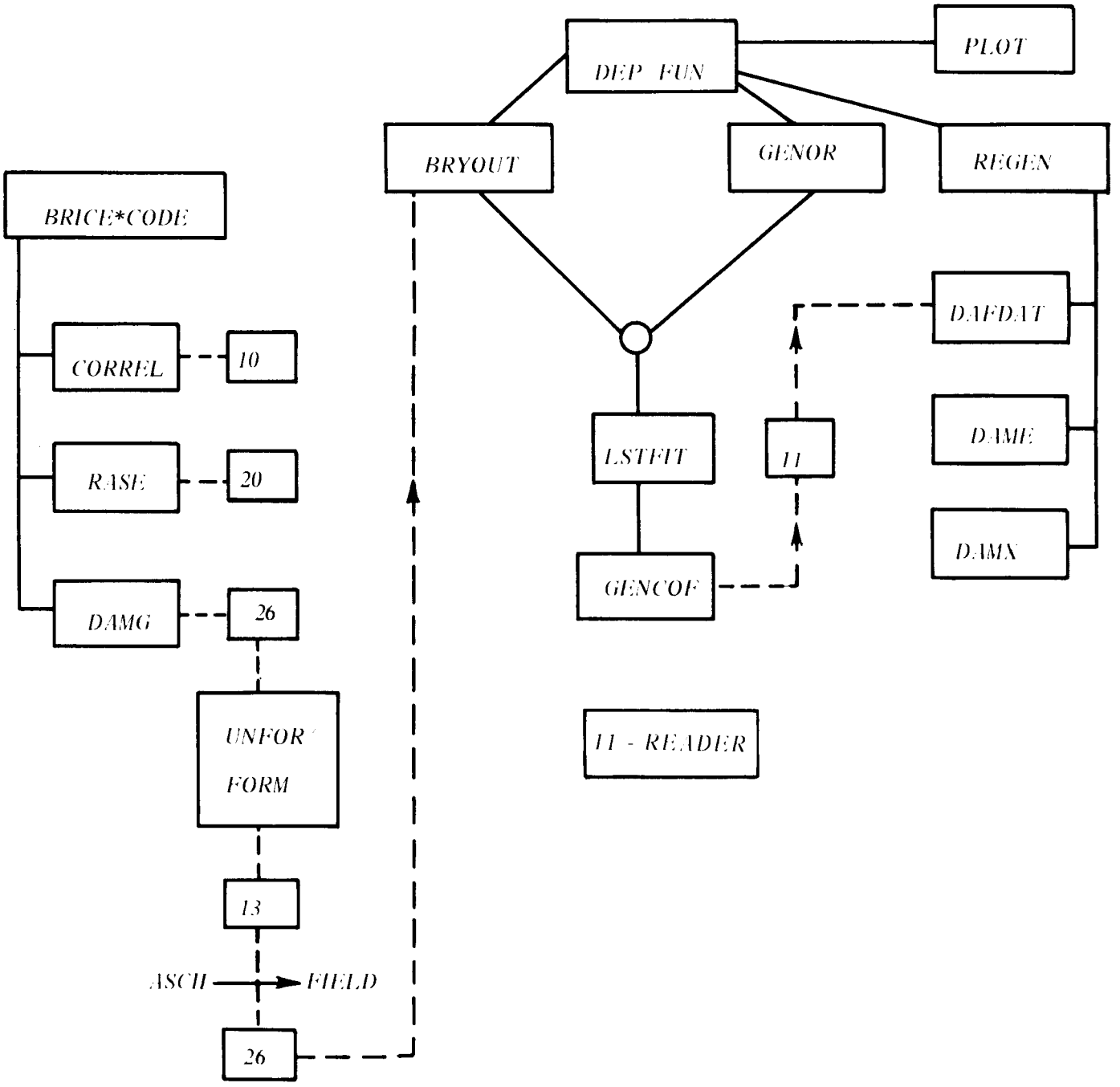
```
            =====  LOWZDEP  =====

     48X,'SMAX = ',F9.2,///)
       PRINT 112, Z,AP, AN, M1,RHO,M2,Z1,Z2
112     FORMAT(//,'      Z       AP  AN           M1          RHO          M2
&       Z1            Z2          ',//,8F10.4,//,'          BRICE       ',
&'   F1            F3            F4           F5   ',///)

       DO 9 I = 1,100
       F1(I) = 0.0
9       F2(I) = 0.0

C
C
C     *****************   STOPPING POWER CURVE   ********

C
C
       IF(IND1 .NE.  1) GO TO 15
      E =FIRSTE
      DELE = (FINE -FIRSTE)/100
       ONEDEV = 0.
      DO 10 I = 1,100
         F1(I)=S0*SQRT(E/E0)
         F5(I) = A00*(1. - EXP(-B00*E))
         F3(I)=A0*EXP(-E/B0)
       IF(E .LT.  E0)  F4(I)= F1(I)
       IF(E .GE.  E0 .AND. E .LT.  EINT) F4(I) = F5(I)
       IF(E .GE.  EINT)  F4(I)= F3(I)
       U = E/(Z**2 * M1* 100.)
      Y = 1. + (4. * Z**2 * AP**2  * U ) **(AN/2)
      F = 1./Y
      ZZ = SQRT(U)
      SA1=ZZ*(30.*U**2+53.*U+21.)/(3.*(1.+U)**2)
      SA2 = ( 10. * U + 1. ) * DATAN(ZZ)
      SEA =(SA1 + SA2) * 0.60961E-15
      XN = RHO * 6.023E+23/M2
      SE = XN * (Z1 + Z2) *SEA * F / 1000. * 1.E-4
      YY(I)= 1./SE
      EP(I) = E
      DEDX(I) = SE

C       **  PRIMARY OUTPUT VARIABLES  **
C          TRY TO PLT DEDX,F1,F2 VS EP  'LIN VS LIN'T
       IF(IPRINT .NE. 1) GO TO 999
      PRINT 1,EP(I),  DEDX(I),F1(I),F3(I),F4(I),F5(I)
1     FORMAT(7 (4X,E10.3))
999       CONTINUE
      E = E+ DELE
  10  CONTINUE
        IF(IPLOT .NE. 1) GO TO 15
      CALL GRAPH(EP,'LINEAR',DEDX,'LINEAR',100,'NONE','SOLID','ENERGY$'
&,'STOPPING POWER$$','NORMAL','COMPARISON OF STOPPING POWER$$',
& 'FULL','NORMAL')
       NUM1 =  100
       CALL GRAPHM(EP,'SCAL1',F4,'SCAL1',NUM1,'NONE',1)
C      CALL GRAPHM(X2,'SCAL1',YY,'SCAL1',NUM2,'NONE',1)

15     IF(IND2 .NE. 1 ) GO TO 30
C
```

```
      =====  LOWZDEP  =====

115.      C
116.      C         ********************* CALCULATE THE  SPATIAL DEPOSITION *****
117.      C
118.      C
119.      C   CONSTANTS FOR REGION 2
120.      C
121.      C
122.              DO 25 J = 1,NUM
123.              ENER(1,J) = X3(J)
124.              PRINT  103,ENER(1,J)
125.      C
126.      C    CALCULATE FOR VARIOUS VALUES OF X
127.      C
128.              CALL PCOEF(ENER(1,J),J)
129.              K = INCID(J)
130.              TOTE = 0.
131.              DELX = XL(J)/50.
132.              X(1,J) = 0.
133.              DO 20 I = 1,51
134.              H = X(I,J)
135.              IF(X(I,J) .GE. XM(J) ) GO TO 16
136.              IF(X(I,J) .GE. XL(J) ) GO TO 18
137.      C
138.      C
139.      C     ************REGION 3 ************
140.      C
141.                F4(I) = A0/(EXP(ENER(1,J)/B0) - A0*X(I,J)/B0)
142.               F5(I) = COMATX(J,1,3)  +  COMATX(J,2,3) *H  +  COMATX(J,3,3) *H**2
143.              &   +    COMATX(J,4,3) *H**3  + COMATX(J,5,3) *H**4
144.                ENER(I,J) = B0*ALOG(EXP(ENER(1,J)/B0)- A0*X(I,J)/B0)
145.                DEV = (F4(I) - F5(I))/F4(I)
146.      C
147.      C           **************END OF REGION3**************
148.      C
149.      C
150.      C
151.              GO TO 18
152.      16       CONTINUE
153.              IF(X(I,J) .GE. XH(J)) GO TO 161
154.      C
155.      C
156.      C     **********************REGION2*******************
157.      C
158.      C
159.                SSS =  X(I,J) - XM(J)
160.               F5(I) = COMATX(J,1,2)  +  COMATX(J,2,2) *H  +  COMATX(J,3,2) *H**2
161.              &   +    COMATX(J,4,2) *H**3  + COMATX(J,5,2) *H**4
162.                F4(I) = A00/(1. + EXP(B00*(A00*SSS - QQ1(J))))
163.                ENER(I,J) = ALOG(1. + EXP(B00*(QQ1(J) - A00*SSS)))/B00
164.      C
165.      C
166.      C     **********  END OF REGION2
167.      C
168.      C
169.              GO TO 18
170.      161       CONTINUE
171.      C
```

```
      =====  LOWZDEP  =====

172.      C
173.      C       ***************** REGION1  ***************
174.      C
175.      C
176.      C
177.            SSS = X(I,J) - XH(J)
178.            ER1 = ENER(1,J)
179.            IF( K .GE. 2 ) ER1 = E0
180.            F4(I) = S0*SQRT(ER1/E0) - S0**2*SSS/2./E0
181.           F5(I) = COMATX(J,1,1)  +  COMATX(J,2,1) *H  +  COMATX(J,3,1) *H**2
182.          &  +   COMATX(J,4,1) *H**3  + COMATX(J,5,1) *H**4
183.            ENER(I,J) = ER1 - S0*SSS*SQRT(ER1/E0) + (S0*SSS)**2/4./E0
184.      18      CONTINUE
185.      C
186.      C    PRINT THE OUT PUT
187.      C
188.            IF(IPRINT .NE. 1) GO TO 19
189.            PRINT 104,X(I,J),F4(I),F5(I),DEV,ENER(I,J)
190.      19     CONTINUE
191.      C
192.      C
193.      C      **************END OF REGION 1 ***************
194.      C
195.      C
196.            TOTE = TOTE + F5(I) * DELX
197.            X(I+1,J) = X(I,J) + DELX
198.      20    CONTINUE
199.            PRINT 105,TOTE
200.      105    FORMAT(//,'    TOTAL INTEGRATED DEPOSITED ENERGY IS'5X,F9.3,
201.           1       ' KEV',///)
202.            DO 209 I = 1,51
203.      209   XPT(I) = X(I,J)
204.              IF(IPLOT .NE. 1 ) GO TO 25
205.            IF(J .NE. 1) GO TO 21
206.            CALL GRAPH(XPT,'LINEAR',F5,'LINEAR',51,'NONE','SOLID',
207.           &'DISTANCE(MICRONS)$$',' DEPOSITION(KEV/MICRON)$$','NORMAL',
208.           &'SPATIAL DEPOSITION$$','FULL','NORMAL')
209.      21      CONTINUE
210.            IF(J .EQ. 1 ) GO TO 25
211.            CALL GRAPHM(XPT,'SCAL1',F5,'SCAL1',51,'NONE',1)
212.      25     CONTINUE
213.            IF(IPLOT .NE. 1) GO TO 30
214.            DO 28 J = 1,NUM
215.            DO 26 I = 1,51
216.            XPT(I) = X(I,J)
217.      26     PLTE(I) = ENER(I,J)
218.            IF(J .NE. 1) GO TO 27
219.            CALL GRAPH(XPT,'LINEAR',ENER,'LINEAR',51,'NONE','SOLID',
220.           &'DISTANCE MICRONS)$$',' LOCAL ENERGY (KEV)$$','NORMAL',
221.           &'INSTANTANE1US ENERGY$$','FULL','NORMAL')
222.      27      CONTINUE
223.           IF(J .EQ. 1 ) GO TO 28
224.            CALL GRAPHM(XPT,'SCAL1',PLTE,'SCAL1',51,'NONE',1)
225.      28     CONTINUE
226.      30      CONTINUE
227.
228.      103     FORMAT(///, '         SPATIAL DEPOSITION PRO'
```

```
      =====  LOWZDEP  =====

229.        &'FILE',///,12X,' FOR INCIDENT ENERGY OF ',3X,F12.4,3X,'KEV',//,
230.        &'  LOCATION       F4          F5          F5           ENERGY',//)
231.   104      FORMAT(8(3X,F12.4))
232.         STOP DONE
233.         END
```

DEPOSITION FUNCTION CREATION



```
=====  DEPFUNCREATE  =====

 1.
 2.
 3.
 4.
 5.                    DEPOSITION FUNCTION CREATION
 6.                    SECTION OF T*DAMEN CODE
 7.
 8.
 9.          ABSOLUTE
10.               DEPFUN
11.               11-READER
12.
13.
14.          READER
15.               READ/DEPFUN
16.
17.
18.          MAPPER
19.               MAP/DEPFUN
20.
21.
22.          LISTER
23.               LIST/DEPFUN
24.
25.
26.          SYMBOLICS
27.               DEPFUN --MASTER ROUTINE FOR CALLING EITHER BRYOUT,
28.                        GENOR,OR REGEN,AND PLOT
29.               BRYOUT --READS FILE 26 FOR DEPOSITION PROFILE FROM
30.                        BRICE CODE,FILLS D AND Z ARRAYS IN
31.                        KEV/MICRON,MICRON
32.               GENOR  --READS TABULATED DATA FROM FILES OF DEPOSITION
33.                        DATA FILLS D AND Z ARRAYS IN KEV/MICRON,MICRON
34.               LSTFIT --FITS THE DATA FROM GENOR OR BRYOUT WITH A 4TH
35.                        DEGREE POLYNOMIAL UP TO AND BEYOND PEAK FOR
36.                        EACH ENERGY
37.               LSTFEL --FITS DATA FROM GENOR OR BRYOUT (ASSUMING PEAK
38.                        AT X=0),WITH UP TO 4TH DEGREE POLYNOMIAL FOR
39.                        EACH ENERGY
40.               GENCOF --GENERATES THE ENERGY COEFFICIENT MATRIX FROM
41.                        LSTFIT DATA AND WRITES IN FILE 11
42.               REGEN  --PRODUCES A DEPOSITION PROFILE FROM DATA
43.                        ON FILE 11 FOR SPECIFIED ENERGY VALUES
44.               DAFDAT --READS DAMAGE FUNCTIONS FROM FILE 11 BY
45.                        A NUMBER WHICH INDICATES A CERTAIN ION-TARGET
46.                        COMBINATION
47.               DAME   --CALCULATES COEFFICIENTS OF DAMAGE FUNCTIONS
48.                        FOR A GIVEN VALUE OF ENERGY
49.               DAMX   --CALCULATES THE DAMAGE FUNCTION AT A POSITION
50.                        FOR A PREVIOUS CALL TO DAME
51.               PLOT   --MAKES PLOTS OF EITHER:
52.                              BRYOUT DATA
53.                              GENOR DATA
54.                              REGEN DATA
55.               11-READER--READS COEFFICIENTS STORED IN FILE 11 DIRECTRY
56.               UNFORM/FORM--CHANGE BRICE DATA FROM UNFORMATTED TO
57.                            FORMATTED
```

```
=====  READ/DEPFUNCREATE  =====

                *****READ STATEMENT FOR THE DEPFUNCREATE CODE*****


                  FOR THE DEPFUN SUBROUTINE:

1          READ(5,10,END = 99) INST
10            FORMAT(A6)
           IF(INST  .EQ.'STOP')  GO TO 99
           IF(INST  .EQ.'BRICE')  GO TO 100
           IF(INST  .EQ.'GENOR')  GO TO 200
           IF(INST  .EQ.'REGEN')  GO TO 300
           IF(INST  .EQ.'PLOT')  GO TO 400
100         READ 10, INST2
           IF (INST2 .NE. 'CREATE') GO TO 150
200         READ 10, INST2
           READ 10,INST3
           IF(INST2 .NE. 'CREATE') GO TO 250
           IF(INST3.EQ.'ELECT')GO TO 231
300           CALL REGEN
400         READ, NDPLOT,ND1,ND2,ND3,NYPLOT,NY1,NY2,NY3
            READ, CONVER


                  FOR THE BRYOUT/DEPFUN SUBROUTINE:

             READ , IPRINT
             READ (26,100) NDAM,NE,NSTP,NDIV,NDIVE,NMULT,NTYPE,JSTP,
           1NXTND,KE,ITER,ALAT,PR,X3,(P2(J),Z2(J),J=1,4)
             READ (26,90) ITER,(E(I),I = 1,NDAM)
90             FORMAT(I5,(E12.6))
            READ (26,110)(RP(I,1),DELTA(I,1),DP(I,1),R(I,1),F(I),G(I
           1),XI00(I),XI01(I),XI02(I),XI03(I),XI20(I),XI21(I),XI30(I),XMOM(I)
           2,XMOM2(I),XMOM3(I),(DAMG(I,J),J=1,101),I=1,NDAM)
100            FORMAT(1115,/,11D12.6)
110            FORMAT(10D12.6)


                  FOR THE GENOR/DEPFUN SUBROUTINE:

             READ ,  NUME,NUMX
             READ,IPRINT
            READ,C1,C2
            READ,(E(I),RP(I),EDRP(I),I=1,NUME)
            DO 10 I=1,NUMX
10           READ,XRP(I),(Q(I,J),J=1,NUME)


                  FOR THE LSTFIT/DEPFUN SUBROUTINE:

            READ,ORDER
            READ,STO
            READ,INK
            READ,INK
            READ,ORDER


                  FOR THE LSTFEL/DEPFUN SUBROUTINE:
```

```
                =====  READ/DEPFUNCREATE  =====

58.
59.               READ,ORDER
60.               READ,INK
61.
62.
63.                  FOR THE GENCOF/DEPFUN SUBROUTINE:
64.
65.               READ , NUMFL
66.       243      FORMAT(9A6)
67.               READ 243,IDENT
68.       5       READ(11,END = 10)
69.
70.
71.                  FOR THE REGEN/DEPFUN SUBROUTINE:
72.
73.               READ , LOCDFN,CONV
74.               READ, NUM,(E(I),I = 1,NUM)
75.               READ , IPRINT
76.
77.
78.                  FOR THE DAFDAT/DEPFUN SUBROUTINE:
79.
80.       5        READ (11,END = 10)
81.       777     READ(11,END = 100) LOC,IDENT,CG
82.
83.
84.                  FOR THE 11-READER SUBROUTINE:
85.
86.               READ,I
87.       5        READ (11,END = 10)
88.       777     READ(11,END = 100) LOC,NE,IDENT,CG
```

```
=====  MAP/DEPFUN  =====

1.      @MAP,I    ,T*DAMEN.DEPFUN
2.      IN T*DAMEN.DEPFUN,.BRYOUT/DEPFUN,.GENOR/DEPFUN
3.      IN T*DAMEN.LSTFIT/DEPFUN,.GENCOF/DEPFUN,.REGEN/DEPFUN,.LSTFEL/DEPFUN
4.      IN T*DAMEN.DAME/DEPFUN,.DAMX/DEPFUN,.DAFDAT/DEPFUN,.PLOT/DEPFUN
5.      END
```

```
=====  LIST/DEPFUNCREATE  =====

 1.     @PRT,S  T*DAMEN.LIST/DEPFUNCREATE
 2.     @PRT,S  T*DAMEN.DEPFUN
 3.     @PRT,S  T*DAMEN.BRYOUT/DEPFUN
 4.     @PRT,S  T*DAMEN.GENOR/DEPFUN
 5.     @PRT,S  T*DAMEN.LSTFIT/DEPFUN
 6.     @PRT,S  T*DAMEN.LSTFEL/DEPFUN
 7.     @PRT,S  T*DAMEN.GENCOF/DEPFUN
 8.     @PRT,S  T*DAMEN.REGEN/DEPFUN
 9.     @PRT,S  T*DAMEN.DAFDAT/DEPFUN
10.     @PRT,S  T*DAMEN.DAME/DEPFUN
11.     @PRT,S  T*DAMEN.DAMX/DEPFUN
12.     @PRT,S  T*DAMEN.PLOT/DEPFUN
13.     @ PRT,S T*DAMEN.11-READER
```

```
=====  BRYOUT/DEPFUN  =====

      SUBROUTINE BRYOUT
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL E,Z,D,Y,DMAX,ZMAX,ZLAST,XX
      DIMENSION P2(4),Z2(4),RP(20,1),DELTA(20,1),DP(20,1),F(20)
      DIMENSION G(20),XI00(20),XI01(20),XI02(20),XI03(20),R(20,1)
      DIMENSION XI20(20),XI21(20),XI30(20),XMOM(20),XMOM2(20)
      DIMENSION XMOM3(20),DAMG(20,101)
C
C    OUTPUT OF THIS CODE IS THE D AND Z AND NDAM
C
      COMMON/GENO/D(75,20),Y(75,20),Z(75,20),DMAX(20),ISTOP(20)
     1,E(50),ZMAX(20),ZLAST(20),JSTOP(20),XX(75,20)
      COMMON/LIMITS/NDAM,NUMX
      REWIND 26
      PRINT 22
22    FORMAT(1X,'BRYOUT/DEPFUN IS CALLED TO READ FILE 26.'/)
      READ , IPRINT
C
C  INPUT FROM FILE 26  ---  A PREVIOUS  BRICE RUN AFTER
C    TRANSLATION TO FIELD DATA FORMATTED
C
      READ (26,100) NDAM,NE,NSTP,NDIV,NDIVE,NMULT,NTYPE,JSTP,
     1NXTND,KE,ITER,ALAT,PR,X3,(P2(J),Z2(J),J=1,4)
      READ (26,90) ITER,(E(I),I = 1,NDAM)
90      FORMAT(I5,(E12.6))
      READ (26,110)(RP(I,1),DELTA(I,1),DP(I,1),R(I,1),F(I),G(I
     1),XI00(I),XI01(I),XI02(I),XI03(I),XI20(I),XI21(I),XI30(I),XMOM(I)
     2,XMOM2(I),XMOM3(I),(DAMG(I,J),J=1,101),I=1,NDAM)
C
C    CHANGE THE  LOCATION FROM NORMAL TO RP TO MICRONS
C
C     AND CHANGE THE DEPOSITED ENERGY FROM EV/A TO KEV/MICRON
C
      NUMX = 51
      DO 10 I=1,NDAM
      DO 5 J=1,101,2
      ZZ=(J+1)/2
      NN=ZZ
      D(NN,I)=DAMG(I,J) * 10.
      AA=.04*(ZZ-1.)
5     Z(NN,I)=AA*RP(I,1)*1.E-4
10     CONTINUE
      IF(IPRINT .NE. 1) GO TO 87
      DO 30 J = 1,NDAM
      PRINT 477,E(J)
477     FORMAT(///,'   INCIDENT ENERGY OF  ',F12.2,'  KEV',//,
     1'  X(MICRON)  ','  DE/DX(KEV/MICRON) ',/)
      PRINT 75,(Z(I,J),D(I,J),I=1,51)
75      FORMAT(2(3X,E12.5))
30     CONTINUE
      PRINT,(RP(I,1),I=1,NDAM)
      PRINT,NDAM,NE,NSTP,NDIV,NDIVE,NMULT,NTYPE,JSTP
87     CONTINUE
100     FORMAT(11I5,/,11D12.6)
110     FORMAT(10D12.6)
      RETURN
      END
```

===== DAFDAT/DEPFUN =====

```
      SUBROUTINE DAFDAT(I)
C
C    A PROGRAM TO READ FROM BINARY FILE 11 THE DPA CROSS SECTIONS
C CORRESPONDING TO POSITION I  IN THAT FILE  THIS ESTABLISHES THE
C  ARRAY CG AND A TITLE ARRAY OF NINE WORDS AND  LOCATION MARK
      COMMON/GCOF/ CG(15,10)
      COMMON/LIMITS/NE,NUMX
      DIMENSION IDENT(13)
       REWIND  11
      M = I-1
      IF(M .EQ. 0) GO TO 777
      IE = 0
5      READ (11,END = 10)
10     IE = IE+ 1
      IF(IE .LT. M) GO TO 5
777   READ(11,END = 100) LOC,NE,IDENT,CG
100      CONTINUE
      PRINT 423,IDENT,LOC
423    FORMAT(//,3X,'DATA FOR',13A6,//,' WAS READ FROM BLOCK',3X,I5,//)
      RETURN
      END
```

```
=====  DAME/DEPFUN  =====

      SUBROUTINE DAME(E)
C
C     A ROUTINE TO GENERATE THE POLYNOMIAL COEFFICIENTS FOR THE VALUE E
C
C     OUTPUT OF THIS ROUTINE
C
      COMMON/XFL/ C(20),NC
C
C     INPUT FROM FILE11 THRU DAFDAT
C
      COMMON/GCOF/CG(15,10)
      COMMON/LIMITS/NE,NUMX
      DATA NC/5/
      LIST = 2*NC + 2
      MEC = LIST + 1
      DO 10 I = 1,20
10     C(I) = 0.
C
C GENERATE THE COEFFICIENTS
C
      DO 20 I = 1,NE
      IF(E .LE. CG(MEC,I)) GO TO 21
20     CONTINUE
21      IF(E .LT. CG(MEC,1)) GO TO 300
        IF(I .EQ. 1) GO TO 3000
      IF(E .GT. CG(MEC,NE)) GO TO 400
C
C  LINEAR INTERPOLATION IS USED FOR ALL CO-EFFICIENTS
C
      DO 15 J = 1,LIST
      ALNA=CG(J,I)
      ALNB=CG(J,I-1)
      C(J) = (ALNA - ALNB)/(CG(MEC,I) - CG(MEC,I-1))
     1  * ( E-CG(MEC,I-1))  + ALNB
15      CONTINUE
      RETURN
300    PRINT  500
      DO 275 J = 1,LIST
      IF(ABS(CG(J,I)).LE.1.E-38)GO TO 275
      C(J) = (CG(J,2) - CG(J,1))/(CG(MEC,2) - CG(MEC,1))
     1  * ( E-CG(MEC,1))  + CG(J,1)
275   CONTINUE
500    FORMAT('   E IS BELOW THE LOWEST ENERGY IN THE TABLE')
          RETURN
400     PRINT   600
      DO 276 J = 1,LIST
      IF(ABS(CG(J,I)).LE.1.E-38)GO TO 276
      C(J) = (CG(J,NE) - CG(J,NE-1))/(CG(MEC,NE) - CG(MEC,NE-1))
     1  * ( E-CG(MEC,NE))  + CG(J,NE)
276   CONTINUE
600    FORMAT('   E IS LARGER THAN THE MAXIMUM IN THE TABLE')
      RETURN
3000   DO  277 J = 1,LIST
      C(J)   = CG(J,1)
277    CONTINUE
      RETURN
      END
```

```
=====  DAMX/DEPFUN  =====

 1.              FUNCTION DAMX(X)
 2.      C
 3.      C    A FUNCTION TO CALCULATE THE LOCAL DPA CROSS SECTION AT POSITION X
 4.      C    FOR AN ENERGY E FROM THE PREVIOUS CALL TO  DAME
 5.      C
 6.              COMMON/XFL/ C(20),NC
 7.              DAMX = 0.
 8.              NJK = 2
 9.              Y = X
10.              IF(X .GT.  C(2) ) GO TO 200
11.              IF(X .LE. 1.E-8) GO TO 300
12.              IF(X .GT. C(1) ) NJK = NC+ 2
13.              IF(C(1).LE.1.E-38)NJK=2
14.              IF(X .GE. C(1)) Y = X-C(1)
15.              IF(Y .LE. 1.E-8) GO TO 400
16.              DO 100 I = 1,NC
17.              DAMX = DAMX + C(I + NJK) * Y **(I-1)
18.      100      CONTINUE
19.              RETURN
20.      200      CONTINUE
21.              RETURN
22.      300      DAMX = C(3)
23.              RETURN
24.      400      DAMX = C(3+NC)
25.              RETURN
26.              END
```

===== DEPFUN =====

```
C   A MAIN PROGRAM FOR CALLING PROGRAM SEGMENTS BRYOUT,GENCOF AND REGEN
C  AND PLOT
C
      COMMON/SCALE/CONVER
1      READ(5,10,END = 99) INST
10        FORMAT(A6)
C
C    READ AND INSTRUCTION
C
        IF(INST  .EQ.'STOP')  GO TO 99
        IF(INST  .EQ.'BRICE')  GO TO 100
        IF(INST  .EQ.'GENOR')  GO TO 200
        IF(INST  .EQ.'REGEN')  GO TO 300
        IF(INST  .EQ.'PLOT')  GO TO 400
        PRINT 11
11      FORMAT(//,' LABEL NOT RECOGNIZED -- TRY AGAIN ',//)
        GO TO 1
C
C       READ A FILE 26 AND GET D AND Z ARRAYS
C
100       READ 10, INST2
        READ 10,INST3
        CALL BRYOUT
        IF (INST2 .NE. 'CREATE') GO TO 150
        IF(INST3.EQ.'ELECT')GO TO 230
        CALL LSTFIT
        GO TO 240
230      CALL LSTFEL
240     CALL GENCOF
150        CONTINUE
        GO TO 1
C
C      READ A MANUAL FILE AND FILL D AND Z ARRAY
C
200        READ 10, INST2
        READ 10,INST3
           CALL GENOR
        IF(INST2 .NE. 'CREATE') GO TO 250
        IF(INST3.EQ.'ELECT')GO TO 231
        CALL LSTFIT
        GO TO 241
231     CALL LSTFEL
241     CONTINUE
        CALL GENCOF
250       CONTINUE
         GO TO 1
C
C    GENERATE A SET OF VALUES FROM THE STORED COEFFICIENTS
C
300        CALL REGEN
        GO TO 1
C
C        PLOT A SET OF VALUES
C
400        READ, NDPLOT,ND1,ND2,ND3,NYPLOT,NY1,NY2,NY3
        READ, CONVER
        CALL PLOT(NDPLOT,ND1,ND2,ND3,NYPLOT,NY1,NY2,NY3)
```

```
                =====   DEPFUN   =====

58.                GO TO 1
59.       99        STOP
60.                END
```

```
=====  GENCOF/DEPFUN  =====

      SUBROUTINE GENCOF
      COMMON/GEN0/D(75,20),Y(75,20),Z(75,20),DMAX(20),ISTOP(20)
     1,E(50),ZMAX(20),ZLAST(20),JSTOP(20),XX(75,20)
C
C       INPUT FROM LSTFIT
C
      COMMON/ECOF/C1(10,10),C2(10,10)
C
C       AVAILABLE FOR OUTPUT
C
      COMMON/GCOF/ CG(15,10)
      DIMENSION IDENT(13)
      COMMON/LIMITS/NE,NX
      DATA NC/5/
      DO 63 JJ = 1,10
      DO 63 KK = 1,15
63     CG(KK,JJ) = 0.0
      PRINT 425
      READ , NUMFL
243    FORMAT(13A6)
      READ 243,IDENT
      MEC = 2*NC + 3
      DO 437 I=1,15
      DO 436 J=1,10
436   CG(I,J)=0.
437   CONTINUE
      DO 200 J = 1,NE
      DO 100 I = 1,NC
      CG(I+2,J) = C1(J,I)
      CG(I+2+NC,J) = C2(J,I)
100     CONTINUE
      CG(1,J) = ZMAX(J)
      CG(2,J) = ZLAST(J)
      CG(MEC,J) = E(J)
200     CONTINUE
      IF(NUMFL .LE. 1) GO TO 77
C
C     SEARCH THE FILE FOR THE NUMFL'TH BLOCK OF DATA
C
       REWIND 11
       M = NUMFL - 1
       IE = 0
5      READ(11,END = 10)
10      IE = IE + 1
       IF(IE .LT. M) GO TO 5
77     WRITE(11) NUMFL,NE,IDENT,CG
       END  FILE 11
       PRINT   423,IDENT, NUMFL
423      FORMAT(//,'  DATA FOR ',/,3X,13A6,//,' IN BLOCK ',I5,//)
425   FORMAT(1X,'GENCOF HAS BEEN CALLED TO STORE THE FITTED CO-EFFICIENTS
     1S ON FILE 11',/)
       RETURN
       END
```

```
            =====  GENOR/DEPFUN  =====

 1.           SUBROUTINE GENOR
 2.           DIMENSION RP(51),EDRP(50),XRP(100),Q(75,20),XP(50)
 3.           COMMON/GEN0/D2(75,20),Y2(75,20),Z2(75,20),DMAX(20),ISTOP(20),E(50)
 4.          1,ZMAX(20),ZLAST(20),JSTOP(20),XX(75,20)
 5.           COMMON/LIMITS/NUME,NUMX
 6.           DIMENSION TDIS(50)
 7.            PRINT 99
 8.     99     FORMAT(1X,'GENOR/DEPFUN HAS BEEN CALLED TO READ A SYMBOLIC ELEMEN
 9.          1T OF DATA'/)
10.            READ ,  NUME,NUMX
11.            READ,IPRINT
12.           PRINT 100
13.     100    FORMAT (1X,'PLEASE INPUT YOUR  DATA SET')
14.           READ,C1,C2
15.           READ,(E(I),RP(I),EDRP(I),I=1,NUME)
16.           DO 10 I=1,NUMX
17.     10      READ,XRP(I),(Q(I,J),J=1,NUME)
18.     C
19.     C    THE DATA ARRAYS ARE FILLED
20.     C
21.     51       DO 21 I=1,NUME
22.             XP(I)=RP(I)*C1
23.            TDIS(I)=EDRP(I)*C2
24.            DO 15 J=1,NUMX
25.            D2(J,I)=Q(J,I)*TDIS(I)
26.              Z2(J,I) = XRP(J) *  XP(I)
27.     15     CONTINUE
28.     21      CONTINUE
29.             IF(IPRINT .NE. 1) GO TO 87
30.             DO 30 J = 1,NUME
31.             PRINT 477,E(J)
32.     477       FORMAT(///,'   INCIDENT ENERGY OF  ',F12.2,'  KEV',//,
33.          1'  X(MICRON)  ',' DE/DX(KEV/MICRON) ',/)
34.            PRINT 75,(Z2(I,J),D2(I,J),I=1,51)
35.     75        FORMAT(2(3X,E12.5))
36.     30       CONTINUE
37.     87      CONTINUE
38.     C
39.     C     THE DATA HAS BEEN CONVERTED
40.     C
41.             RETURN
42.             END
```

```
      =====  LSTFEL/DEPFUN  =====

      SUBROUTINE LSTFEL
C     THIS SUBROUTINE FITS A LEAST SQUARE NTH ORDER POLYNOMIAL
C     THRU THE RANGE OF POINTS FOR THE ELECTRONIC LOSS DATA. NO
C     PLOTTING IS DONE.
      DIMENSION CURV(100),XAXIS(100)
      COMMON/GENO/D(75,20),Y(75,20),Z(75,20),DMAX(20),ISTOP(20)
     1,E(50),ZMAX(20),ZLAST(20),JSTOP(20),XX(75,20)
      COMMON/ECOF/C1(10,10),C2(10,10)
      COMMON/LIMITS/NUME,NUMX
      DIMENSION C(10),TEMP(500),ACCUR(5)
      INTEGER ORDER
      PRINT 725
      READ,ORDER
      DO 733 I=1,10
      DO 732 J=1,10
      C1(I,J)=0.
732   C2(I,J)=0.
733   CONTINUE
C
C     FIND THE MAXIMA FOR EACH CURVE
C
      DO 21 I=1,NUME
      ZMAX(I)=Z(1,I)
21    DMAX(I)=D(1,I)
      READ,INK
      DO 200 J=1,NUME
      DO 180 I=1,NUMX
      CURV(I)=D(I,J)
      XAXIS(I)=Z(I,J)
      IF(D(I,J).LE..01*D(1,J))GO TO 240
180   CONTINUE
240     ZLAST(J)=Z(I,J)
      KO = I
      CALL AILSQP(XAXIS,CURV,'ONE',KO,ORDER,ORDER,C,NORD,TEMP,ACCUR,INK,
     10,2H..,$1)
C
C   THE CONSTANTS ARE CALCULATED FOR A CURVE
C
      NORD=ORDER+1
      DO 195 I=1,NORD
195   C1(J,I)=C(I)
200   CONTINUE
C
C     THE CONSTANTS ARE FOUND & STORED.
C
1     CONTINUE
      PRINT 669
      DO 250 J=1,NUME
250     PRINT 700,E(J),ZMAX(J),(C1(J,I),I=1,5),ZLAST(J)
      RETURN
669     FORMAT(1X,'ENERGY',5X,'PEAK',9X,'CURVE COEFFICIENTS',
     127X,'LAST'/2X,'(KEV)',5X,'POINT',6X,'1',7X,'2',7X,'3',
     27X,'4',7X,'5',14X,'POINT'//)
700     FORMAT(1X,F6.0,3X,E7.3,3X,5E9.3,3X,E7.3)
725   FORMAT(1X,'LSTFEL HAS BEEN CALLED TO FIT YOUR DATA WITH A LEASTSQUARES
     1UARES METHOD'/)
      END
```

```
      =====  LSTFIT/DEPFUN  =====

 1.             SUBROUTINE LSTFIT
 2.      C      THIS PROGRAM FITS A LEAST SQUARE NTH ORDER POLYNOMIAL THRU THE
 3.      C      CURVES UP TO THE PEAK, AND THEN FITS THE POINTS TO THE LEFT
 4.      C      OF THE PEAK WITH A POLYNOMIAL THAT WILL REPRODUCE ITS
 5.      C      GAUSSIAN SHAPE USING ANOTHER LEAST SQUARES FIT. NO PLOTTING IS DONE.
 6.             DIMENSION CURV(100),XAXIS(100)
 7.             COMMON/GENO/D(75,20),Y(75,20),Z(75,20),DMAX(20),ISTOP(20)
 8.            1,E(50),ZMAX(20),ZLAST(20),JSTOP(20),XX(75,20)
 9.             COMMON/ECOF/C1(10,10),C2(10,10)
10.             COMMON/LIMITS/NUME,NUMX
11.             DIMENSION C(10),TEMP(500),ACCUR(5)
12.             INTEGER ORDER
13.             PRINT 725
14.             READ,ORDER
15.             DO 733 I=1,10
16.             DO 732 J=1,10
17.             C1(I,J)=0.
18.      732    C2(I,J)=0.
19.      733    CONTINUE
20.      C
21.      C      FIND THE MAXIMA FOR EACH CURVE
22.      C
23.             READ,STO
24.             DO 21 I = 1,NUME
25.             DMAX(I)  = 1.E-35
26.            DO 25 J=1,NUMX
27.             DMAX(I) = AMAX1( DMAX(I),D(J,I))
28.      25     CONTINUE
29.             CRIT = 1.E-8* DMAX(I)
30.             DO 27 J = 1,NUMX
31.             DIF = DMAX(I) - D(J,I)
32.             DIF = ABS(DIF)
33.      27      IF(DIF .LE. CRIT) GO TO 20
34.      20       CONTINUE
35.            ISTOP(I)=J-STO
36.            K=ISTOP(I)
37.            ZMAX(I)=Z(K,I)
38.      21     CONTINUE
39.      C
40.      C    DATA CAN NOW BE FITTED
41.      C
42.             READ,INK
43.      149    DO 200 J=1,NUME
44.            LI=ISTOP(J)
45.             DO 180 I=1,LI
46.            CURV(I)=D(I,J)
47.      180    XAXIS(I)=Z(I,J)
48.             CALL AILSQP(XAXIS,CURV,'ONE',LI,ORDER,ORDER,C,NORD,TEMP,ACCUR,INK,
49.           10,2H..,$1)
50.      C
51.      C    THE CONSTANTS ARE CALCULATED FOR A CURVE
52.      C
53.             NORD=ORDER+1
54.             DO 195 I=1,NORD
55.      195    C1(J,I)=C(I)
56.      200    CONTINUE
57.      C
```

```
      =====  LSTFIT/DEPFUN  =====

C    THE POLYNOMIAL ESTIMATES ARE NOW OBTAINED FOR THE LEFT OF THE PEAK
C
C
C    THE POINTS TO THE LEFT OF THE PEAK ARE FINISHED
C
      DO 210 I=1,10
210   C(I)=0.
       READ,INK
       READ,ORDER
      DO 240 J=1,NUME
      LI=ISTOP(J)
      DO 220 I=LI,NUMX
      CURV(I-LI+1)=D(I,J)
       KKK = ISTOP(J)
      XAXIS(I-LI+1)=Z(I,J)-Z(KKK,J)
      IF(D(I,J).LE..01*D(LI,J))GO TO 225
220   CONTINUE
225   JSTOP(J)=I
      ZLAST(J)=Z(I,J)
      IL=JSTOP(J)-ISTOP(J)+1
      CALL AILSQP(XAXIS,CURV,'ONE',IL,ORDER,ORDER,C,NORD,TEMP,ACCUR,INK,
     10,2H..,$1)
C
C    THE CONSTANTS FOR THE CURVE TO THE RIGHT OF THE PEAK ARE KNOWN
C
      NORD=ORDER+1
      DO 235 I=1,NORD
235    C2(J,I)=C(I)
C
C     THE RIGHT HAND SIDE LEAST SQUARE FIT IS OBTAINED
C
240    CONTINUE
1     CONTINUE
      PRINT 669
      DO 250 J=1,NUME
250    PRINT 700,E(J),ZMAX(J),(C1(J,I),I=1,5),ZLAST(J),(C2(J,I),I=1,5)
      RETURN
500    FORMAT(2E10.3,12X,E10.3)
669   FORMAT(1X,'ENERGY',5X,'PEAK',4X,'UP TO PEAK COEFFICIENTS',27X,'LAST',4X,
     1T',4X,'FROM PEAK COEFFICIENTS'/2X,'(KEV)',5X,'POINT',6X,'1',7X,'2'
     2,7X,'3',7X,'4',7X,'5',14X,'POINT',6X,'1',7X,'2',7X,'3',7X,'4',7X,'
     35',//)
700   FORMAT(1X,F6.0,3X,E7.3,3X,5E9.3,3X,E7.3,3X,5E9.3)
725    FORMAT(1X,'LSTFIT HAS BEEN CALLED TO FIT YOUR DATA WITH A LEASTSQUARES
     1UARES METHOD'/)
       END
```

===== PLOT/DEPFUN =====

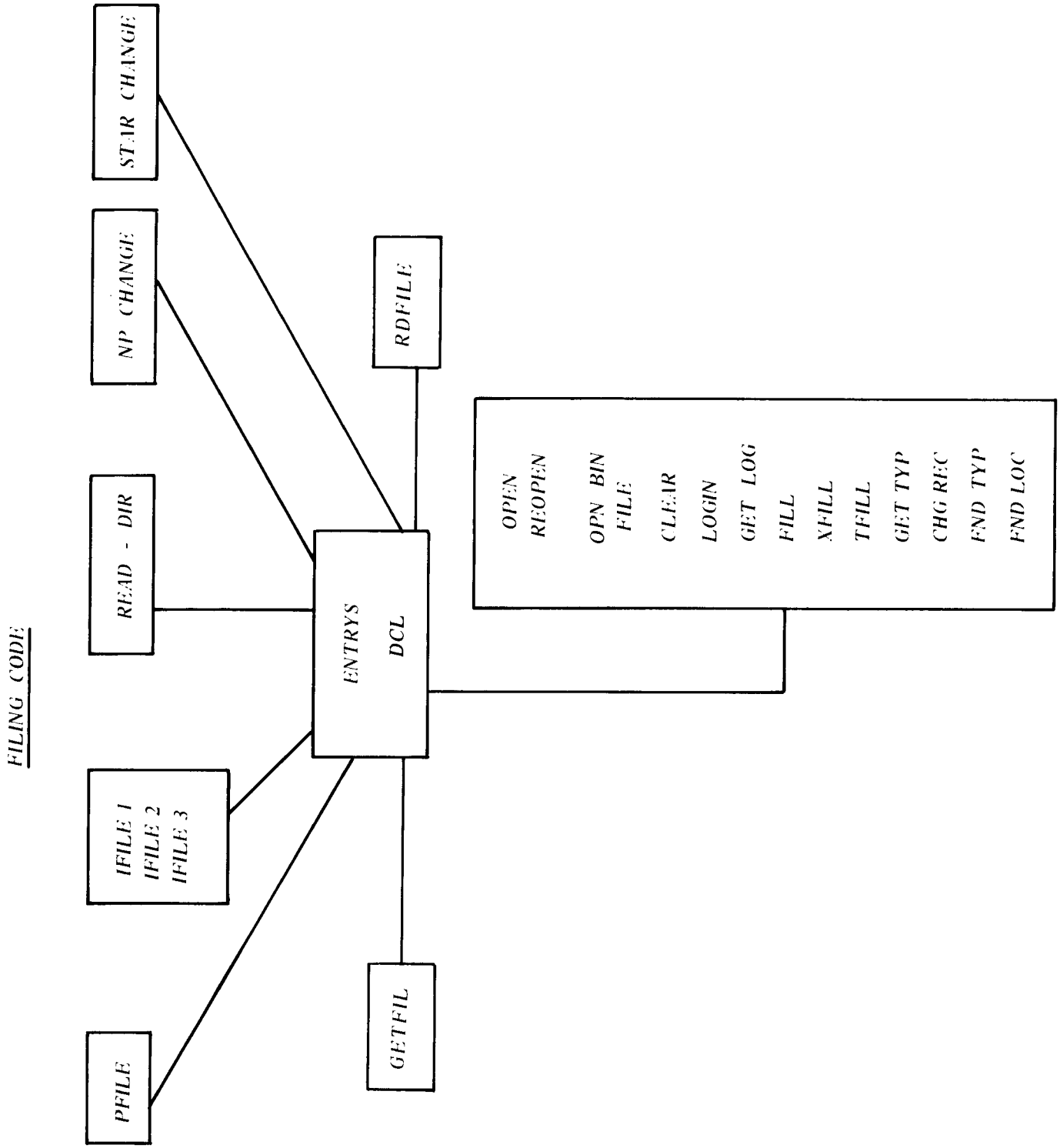
```
      SUBROUTINE PLOT(NDPLOT,ND1,ND2,ND3,NYPLOT,NY1,NY2,NY3)
      COMMON/GENO/D(75,20),Y(75,20),Z(75,20),DMAX(20),ISTOP(20)
     1,E(50),ZMAX(20),ZLAST(20),JSTOP(20),XX(75,20)
      DIMENSION XAXIS(75),YAXIS(75),MLINE(20),YUP(2),XUP(2)
      DIMENSION YPTS(75,20),XPTS(75,20)
      COMMON/SCALE/CONVER
      COMMON/LIMITS/NUME,NUMX
      K=1
C
C     FILL CURVES FOR PLOTS FROM ORIGINAL DATA
C
      IF(NDPLOT.NE.1)GO TO 50
      DO 10 I=ND1,ND2,ND3
      DO 5 J=1,51
      YPTS(J,K)=D(J,I)*CONVER
5     XPTS(J,K)=Z(J,I)
      MLINE(K)='SOLID'
      K=K+1
10    CONTINUE
C
C     FILL CURVES FOR PLOTS FROM FITTED DATA
C
50    IF(NDPLOT.NE.1.AND.NYPLOT.NE.1)GO TO 800
      IF(NYPLOT.NE.1)GO TO 80
      DO 20 I=NY1,NY2,NY3
      DO 15 J=1,51
      YPTS(J,K)=Y(J,I)*CONVER
15    XPTS(J,K)=XX(J,I)
      MLINE(K)=1
      K=K+1
20    CONTINUE
C
C   FIND MAXIMUM FOR LIMITS OF GRAPH
C
80    LIMIT=K-1
      YUP(2)=1.E-35
      XUP(1)=1.E-35
      YUP(1)=0.
      XUP(2)=0.
      DO 100 J=1,LIMIT
      DO 60 I=1,51
      YUP(2)=AMAX1(YUP(2),YPTS(I,J))
60    XUP(1)=AMAX1(XUP(1),XPTS(I,J))
100     CONTINUE
       CALL DEVSET('PLTTR')
        CALL GRAPH (XUP,1,YUP,1,2,'NONE','BLANK','Z(CM)$$',
     1'DE/DZ(KEV/MICRON)$$',3,'ION ENERGY LOSS VS. DISTANCE$$',4HFULL,5H
     2TRNSP)
C
C   THE CO-ORDINATES OF D VS. Y OR D VS.Z ARE DRAWN
C
C
      DO 200 J=1,LIMIT
      DO 150 I=1,51
      XAXIS(I)=XPTS(I,J)
150   YAXIS(I)=YPTS(I,J)
      ISR=MLINE(J)
```

```
      =====  PLOT/DEPFUN  =====

58.         CALL GRAPHM(XAXIS,5HSCAL1,YAXIS,5HSCAL1,NUMX,4HNONE,ISR)
59.   200       CONTINUE
60.   800     CONTINUE
61.           RETURN
62.          END
```

===== REGEN/DEPFUN =====

```
      SUBROUTINE REGEN
     COMMON/GENO/D2(75,20),Y2(75,20),Z2(75,20),DMAX(20),ISTOP(20),E(50)
    1,ZMAX(20),ZLAST(20),JSTOP(20),XX(75,20)
      COMMON/GCOF/CG(15,10)
      COMMON /XFL/C(20),NC
     COMMON  /LIMITS/NE,NUMX
      READ , LOCDFN,CONV
      READ, NUM,(E(I),I = 1,NUM)
      DATA NC/5/
      READ , IPRINT
C
C   GET COEFFICIENTS FROM FILE 11
C
      CALL DAFDAT(LOCDFN)
C
C
C    CALCULATE THE DEPOSITION PROFILES FOR EACH INCIDENT ENERGY
C
      NE=10
      DO 473 I=1,10
      IF (CG(13,I).LE. 1.0E-38)GO TO 474
473   CONTINUE
      GO TO 475
474   NE=I-1
475   CONTINUE
      DO 25 J = 1,NUM
      CALL DAME(E(J))
      DX  = C(2)/50.
      X = 0.
      DO 50 I = 1,51
      Y2(I,J) = DAMX(X) * CONV
      XX(I,J) = X
      X  = X + DX
50     CONTINUE
C
C   PRINT THE OUTPUT
C
      IF(IPRINT.NE. 1) GO TO 81
      PRINT 55, E(J)
      DO 58 I = 1,51
      PRINT 56, I,XX(I,J),Y2(I,J)
58      CONTINUE
55     FORMAT(//,15X,'NUCLEAR ENERGY DEPOSITION',//,
     1  5X,'ION DEPOSITION',5X,F10.2,//,5X,
     2 'LOCATION',7X,'DE/DX(KEV/MICRON)',//)
56    FORMAT(1X,I2,3X,4E15.3)
81     CONTINUE
25    CONTINUE
      END
```

```
      =====  11-READER  =====

      C
      C  A PROGRAM TO READ PARTICULAR DATA BLOCKS IN FILE 11 THAT
      C  ARE FILED WITH BRICE CO-EFFICIENTS
      C
            DIMENSION CG(15,10),IDENT(13)
      27     REWIND  11
            READ,I
            M = I-1
            IF(M .EQ. 0) GO TO 777
            IE = 0
      5      READ (11,END = 10)
      10     IE = IE+ 1
            IF(IE .LT. M) GO TO 5
      777   READ(11,END = 100) LOC,NE,IDENT,CG
      100      CONTINUE
            PRINT 423,IDENT,LOC
      423    FORMAT(//,3X,'DATA FOR',13A6,//,' WAS READ FROM BLOCK',3X,I5,//)
            PRINT 669
      669   FORMAT(1X,'ENERGY',5X,'PEAK',4X,'UP TO PEAK COEFFICIENTS',27X,'LAST',4X,
           1T',4X,'FROM PEAK COEFFICIENTS'/2X,'(KEV)',5X,'POINT',6X,'1',7X,'2'
           2,7X,'3',7X,'4',7X,'5',14X,'POINT',6X,'1',7X,'2',7X,'3',7X,'4',7X,'
           35',//)
            DO 250 J=1,NE
      250   PRINT 700,CG(13,J),CG(1,J),(CG(K,J),K=3,7),CG(2,J),(CG(KK,J),KK=8
           1,12)
      700   FORMAT(1X,F6.0,3X,E7.3,3X,5E9.3,3X,E7.3,3X,5E9.3)
            PRINT 25
      25    FORMAT('DO YOU WANT MORE DATA? SAY YES.')
            READ 12,INST
      12    FORMAT(A6)
            IF(INST.EQ.'YES')GO TO 27
            END
```


FILING CODE
STAR CHANGE
NP CHANGE
READ - DIR
IFILE 1
IFILE 2
IFILE 3
PFILE
ENTRYS
DCL
RDFILE
GETFIL
OPEN
REOPEN
OPN BIN
FILE
CLEAR
LOGIN
GET LOG
FILL
XFILL
TFILL
GET TYP
CHG REC
END TYP
END LOC

===== IODR =====

IODR
SECTION OF T*DAMEN CODE

ABSOLUTE
READ-DIR/IODR
RDFILE/IODR

READER
READ/IODR

MAPPER
MAP/RDFILE

LISTER
LIST/IODR

SYMBOLICS
IFILE/IODR--ROUTINE FOR FILING COMMON BLOCKS IN ION CODE
PFILE/IODR--ROUTINE FOR FILING COMMON BLOCKS IN PHOTON CODE
ENTRYS/IODR--BASIC IODR HANDLING SUBROUTINES
READ-DIR/IODR--WILL READ COMPLETE DIRECTORY OF ANY IODR FILE
ENTRYSNUM/IODR--SAME AS ENTRYS/IODR EXCEPT SUPPORTS PLOTNUM
DCL/IODR--STRUCTURE OF DATA BLOCKS IN IODR FILES
NPCHANGE/IODR--WILL CHANGE # OF POINTS IN ANY FILE
STARCHANGE/IODR--WILL CHANGE X,T ARRAYS IN A FILE
RDFILE/IODR--READS ION DATA AND PRINTS OUT
GETFIL/IODR--READS ION DATA INTO COMMON BLOCKS

RUNSTREAMS

```
=====  LIST/IODR  =====

 1.     @PRT,S  T*DAMEN.LIST/IODR
 2.     @PRT,S  T*DAMEN.IFILE/IODR
 3.     @ .     SEE ION
 4.     @PRT,S  T*DAMEN.PFILE/IODR
 5.     @ .     SEE PHOTON
 6.     @PRT,S  T*DAMEN.ENTRYS/IODR
 7.
 8.     @PRT,S  T*DAMEN.READ-DIR/IODR
 9.     @PRT,S  T*DAMEN.ENTRYSNUM/IODR
10.     @PRT,S  T*DAMEN.DCL/IODR
11.
12.     @PRT,S  T*DAMEN.NPCHANGE/IODR
13.     @PRT,S  T*DAMEN.STARCHANGE/IODR
14.     @PRT,S  T*DAMEN.RDFILE/IODR
15.     @PRT,S  T*DAMEN.GETFIL/IODR
```

```
=====  READ/IODR  =====

                *****READ STATEMENTS FOR THE IODR CODE*****


           FOR THE ENTRYS/IODR SUBROUTINE:

      READ 3,FLNAME
3     FORMAT(13A6)
      CALL IODROP(FCT,NUMFCT,FLNAME,BLOCK,RECORD,ICON,ID)
      READ 3,FLNAME
      CALL IODRRO(FCT,NUMFCT,FLNAME,BLOCK,RECORD,ICON,ID)
      READ 3,MSG
      READ(5,3,END=14)LOGMSG
      CALL IODRRW (FCT,LOGLOC,LOGMSG)


           FOR THE ENTRYSNUM/IODR SUBROUTINE:

      READ 3,FLNAME
3     FORMAT(13A6)
      READ 3,FLNAME
      READ 3,MSG
      READ(5,3,END=14)LOGMSG


           FOR THE NPCHANGE/IODR SUBROUTINE:

      READ  (-,-,END=99) NBIN
      READ 4,INST
4     FORMAT(A6)
      IF(INST .NE. 'Y') GO TO 1


           FOR THE STARCHANGE/IODR SUBROUTINE:

      READ(-,-,END=99)NBIN
      READ 4,INST
4     FORMAT(A6)
      IF(INST .EQ. 'Y')


           FOR THE RDFILE/IODR SUBROUTINE:

      READ(5,72,END = 1000) INST
72    FORMAT(A6)
      READ , IBIN,INOUT,IA,IB,IC
C
      IF(INST .EQ. 'RDFIL1') GO TO 10
      IF(INST .EQ. 'RDFIL2') GO TO 20
      IF(INST .EQ. 'RDFIL3') GO TO 30
```

```
=====  DCL/IODR  =====

 1.      IODR    PROC
 2.      C       ************************************************************
 3.
 4.      C         THIS IS THE BOOKEEPING SECTION OF TOM HUNTER'S DATA BANK
 5.
 6.      C       ************************************************
 7.
 8.      C
 9.              ------------------------------------------------------
10.      C       SET UP THE FILE CONTROL TABLE(FCT) AND ITS ASSOCIATED
11.      C       PARAMETERS
12.
13.      C        INTEGER RECORD,BLOCK,NUMBUF,FCT,LOC,NUMFCT
14.              INTEGER ICON/0/,  ID/'(NONE)'/,   FLNAME(2)/'IPDATA   '/
15.              PARAMETER RECORD=50,RECP1=RECORD+1
16.              PARAMETER BLOCK=14
17.              PARAMETER NUMBUF=1
18.              PARAMETER NUMFCT=30 + NUMBUF*(2 + RECORD*BLOCK)
19.              DIMENSION FCT(NUMFCT)
20.
21.
22.      C       ------------------------------------------------------
23.
24.      C          THIS AREA SETS UP THE STORAGE BINS FOR THE DATA
25.      C       AND THE DIRECTORY INFORMATION NECESSARY TO RETRIEVE
26.      C       THE DATA STORED
27.
28.              INTEGER HEADER(100),DIRLOC/0/,   HLOC/1/
29.              PARAMETER BINSZE=308,BINLOC=0
30.              INTEGER MXBIN/0/
31.              EQUIVALENCE (MXBIN,HEADER(1))
32.
33.              INTEGER DIREC(100),TYPE,MSG(13)
34.               EQUIVALENCE (TYPE,DIREC) , (MSG,DIREC(2))
35.
36.
37.              INTEGER LOG(RECP1),LOGMSG(13),LOGLOC/201/
38.              EQUIVALENCE (LOGMSG,LOG(2)) , (LOGLOC,LOG)
39.      C       ------------------------------------------------
40.
41.              INTEGER NP(50),IWID(4)
42.              EQUIVALENCE (IWID,NP(4))
43.
44.      C       ------------------------------------------------------
45.
46.      C       SET UP TABLES TO IMPLIMENT STORAGE-LOCATION-INDEPENDANT
47.      C       DATA RETRIEVAL. RETRIEVAL WILL BE BY
48.      C       (BIN# , VARIABLE NAME , INDEPENDANT VARIABLE INDEX) TRIPLETS.
49.
50.              PARAMETER BNAMES=4
51.              INTEGER BINS(BNAMES)/ 'PHOTON' , 'ION-1' , 'ION-2' , 'ION-3' /
52.              INTEGER BTYPES(BNAMES)/   1      ,   2  ,     3    ,    4     /
53.
54.      C       CURRENTLY THERE ARE 4 'TYPES' OF BINS
55.              PARAMETER NUMTYP=4
56.      C       LOCATION OF THE 'BIG'   T AND X   NUM PTS IN THE NP RECORD
57.              INTEGER BTLOC(NUMTYP)/  3   ,      2    ,      2  ,     4   /
```

```
=====  DCL/IODR  =====

      INTEGER BXLOC(NUMTYP)/  2      ,   1     ,     1   ,    3  /

      PARAMETER MXVB=10
      INTEGER VABS(MXVB,NUMTYP),VABLOC(MXVB,NUMTYP),VABREC(MXVB,NUMTYP)
     .       ,VABNPT(MXVB,NUMTYP)

C     SET UP FOR PHOTON BIN TYPE
      PARAMETER PN=9
      INTEGER PVAR(PN),PVLOC(PN),PVREC(PN),PVNPT(PN)
      EQUIVALENCE (PVAR,VABS),(PVLOC,VABLOC),(PVREC,VABREC),
     .            (PVNPT,VABNPT)
      DATA PVAR/ 'F','E','U','Y1','X','ADBTMP','TEMP','TM','Y'/
      DATA PVLOC/257,267,269, 277, 285, 293,      1,   251, 256/
      DATA PVREC/ 2,  2,  2,   2,   2,   2,      1,     2, 1/
      DATA PVNPT/ 1 , 1 , 1 ,'IX','IX','IX',  'BIG',    3 ,  2 /

C     ION-1 TYPE BIN
      PARAMETER I1N=6
      INTEGER I1VAR(I1N),I1LOC(I1N),I1REC(I1N),I1NPT(I1N)
      EQUIVALENCE (I1VAR,VABS(1,2)),(I1LOC,VABLOC(1,2))
     .           ,(I1REC,VABREC(1,2)),(I1NPT,VABNPT(1,2))
      DATA I1VAR/'FT','FF','E','T' ,'X','DEPX'/
      DATA I1LOC/ 201, 205,209,213 ,217,  1  /
      DATA I1REC/  4 ,  4 , 4 ,  4 , 1 ,  1  /
      DATA I1NPT/  2 ,  2 , 2 ,  2 , 1 , 'BIG' /

C     ION-2 TYPE BIN
      PARAMETER I2N=6
      INTEGER I2VAR(I2N),I2LOC(I2N),I2REC(I2N),I2NPT(I2N)
      EQUIVALENCE (I2VAR,VABS(1,3)),(I2LOC,VABLOC(1,3))
     .           ,(I2REC,VABREC(1,3)),(I2NPT,VABNPT(1,3))
      DATA I2VAR/'TEMP',  'FT',  'FF',  'E',   'T' ,  'X'   /
      DATA I2LOC/  1 ,     251 , 255 ,  259 ,   263 ,  267  /
      DATA I2REC/  1 ,       4 ,   4 ,    4 ,     4 ,    1  /
      DATA I2NPT/'BIG',      2 ,   2 ,    2 ,     2 ,    1  /

C     ION-3 TYPE BIN
      PARAMETER I3N=7
      INTEGER I3VAR(I3N),I3LOC(I3N),I3REC(I3N),I3NPT(I3N)
      EQUIVALENCE (I3VAR,VABS(1,4)),(I3LOC,VABLOC(1,4))
     .           ,(I3REC,VABREC(1,4)),(I3NPT,VABNPT(1,4))
      DATA I3VAR/'TEMP',  'TA',  'Y' ,  'FT' ,  'FF' ,  'E' ,  'T'/
      DATA I3LOC/  1 ,     267 , 271   ,251 ,   255  ,  259  ,  263 /
      DATA I3REC /  1  ,     4  ,  1  ,    4  ,    4  ,    4  ,    4 /
      DATA I3NPT /'BIG',     4  ,  3  ,    2  ,    2  ,    2  ,    2 /
       END
```

```
=====  ENTRYS/IODR  =====

 1.                SUBROUTINE ENTRYS(NUMREC,ARY,ITYPE,NBIN,NAMVAR,IVAR,NPTS)
 2.          INCLUDE IODR,LIST
 3.
 4.         INTEGER NUMREC,ARY(50,1),ITYPE
 5.          INTEGER NBIN,NAMVAR,IVAR,NPTS,BUFFER(50),BTYPE
 6.          INTEGER TYP NDX, VBL NDX
 7.
 8.     C   ------------------------------------------------------------
 9.         ENTRY OPEN
10.          READ 3,FLNAME
11.     3    FORMAT(13A6)
12.         CALL IODROP(FCT,NUMFCT,FLNAME,BLOCK,RECORD,ICON,ID)
13.          RETURN
14.
15.     C    ------------------------------------------------------------
16.
17.
18.
19.     C   ------------------------------------------------------------
20.         ENTRY REOPEN
21.          READ 3,FLNAME
22.         CALL IODRRO(FCT,NUMFCT,FLNAME,BLOCK,RECORD,ICON,ID)
23.     C     **   RESET PARAMETERS **
24.          CALL IODRRR(FCT,0,HEADER)
25.          HLOC=MXBIN+1
26.          DIRLOC=MXBIN*2
27.     C    ** UPDATE THE LOG LOCATION POINTER
28.          CALL IODRRR(FCT,200,LOG)
29.          RETURN
30.
31.     C    ------------------------------------------------------------
32.
33.
34.     C    ------------------------------------------------
35.                       ENTRY OPN BIN(ITYPE)
36.     C    ********   DO THE BOOKEEPING  ********************
37.
38.     C   MARK DOWN THE TYPE AND FILE THE HEAD RECORD
39.          TYPE=ITYPE
40.         MXBIN=MXBIN+1
41.         HLOC=MXBIN+1
42.         HEADER(HLOC)=TYPE
43.         LOC=0
44.         CALL IODRRW(FCT,0,HEADER)
45.         CALL IODRRW(FCT,1,HEADER(51))
46.
47.     C   WRITE THE DIRECTORY INFO AND FILE
48.     C    ** READ THE MESSAGE **
49.          READ 3,MSG
50.          DIRLOC=MXBIN*2
51.          CALL IODRRW( FCT,DIRLOC,DIREC)
52.          DIRLOC=DIRLOC+1
53.          CALL IODRRW(FCT,DIRLOC,DIREC(51))
54.          CALL IODRCL(FCT)
55.
56.
57.     C   SET THE LOC COUNTER FOR SEQUENTIAL FILING
```

```
                =====  ENTRYS/IODR  =====

 58.            LOC=MXBIN*BINSZE
 59.
 60.             RETURN
 61.      C      ************  END OF SUBROUTINE BOOK KP  ******************
 62.      C      ------------------------------------------------
 63.
 64.
 65.      C       ------------------------------------------------
 66.                        ENTRY FILE (ARY,NUMREC)
 67.
 68.            DO 100 I=1,NUMREC
 69.            CALL IODRRW(FCT,LOC,ARY(1,I))
 70.            LOC=LOC+1
 71.      100   CONTINUE
 72.            RETURN
 73.
 74.      C      *******   END OF SUBROUINE FILE ************************
 75.
 76.      C      ------------------------------------------------------
 77.
 78.
 79.
 80.      C      ------------------------------------------------------
 81.             ENTRY CLEAR
 82.             CALL IODRCL(FCT)
 83.             RETURN
 84.
 85.      C      ------------------------------------------------------------
 86.
 87.
 88.
 89.
 90.      C      ------------------------------------------------------
 91.             ENTRY LOGIN
 92.            DO 12 I=1,50
 93.            READ(5,3,END=14)LOGMSG
 94.            CALL IODRRW (FCT,LOGLOC,LOGMSG)
 95.            LOGLOC=LOGLOC+1
 96.      12    CONTINUE
 97.
 98.      C      ** FILE THE LOGLOC **
 99.      14     CONTINUE
100.            CALL IODRRW (FCT,200,LOG)
101.             CALL IODRCL(FCT)
102.           RETURN
103.
104.     C      ------------------------------------------------------
105.
106.
107.     C      ------------------------------------------------
108.            ENTRY GET LOG
109.
110.            PRINT 131
111.     131    FORMAT('1')
112.
113.     C      ** PRINT THE LOG **
114.            DO 133 LOC=201,LOGLOC
```

```
115.              CALL IODRRR(FCT,LOC,LOGMSG)
116.              PRINT 132,LOGMSG
117.       132    FORMAT(13X,13A6)
118.       133    CONTINUE
119.
120.       C      ** PRINT THE DIRECTORY **
121.              PRINT 140
122.       140    FORMAT(' ',//,'     BIN    TYPE                          CONTENTS')
123.              DIRLOC=2*MXBIN
124.              DO 141 LOC=2,DIRLOC,2
125.              CALL IODRRR(FCT,LOC,DIREC)
126.              I=LOC/2
127.              PRINT 142,I,TYPE,MSG
128.       142    FORMAT(' ',I6,2X,A6,'  ----   ',13A6)
129.       141    CONTINUE
130.              RETURN
131.       C      ------------------------------------------------------------
132.
133.
134.       C      ------------------------------------------------------------
135.                       ENTRY FILL(ARY,NBIN,NAMVAR,IVAR,NPTS,BTYPE)
136.       C      *** FILL THE ARRAY,ARY, FROM BIN NUMBER NBIN
137.       C      AND VARIABLE OF NAME NAMVAR AND FROM INDEPENDANT VARIABLE VALUE
138.       C      OF IVAR.  RETURN THE NUMBER OF VALID POINTS,NPTS, AND BIN TYPE
139.
140.              CALL FND TYP(NBIN,BTYPE)
141.              CALL FND LOC(NAMVAR)
142.
143.       C       ** SET THE NUM OF PTS
144.              NN=VABNPT(VBL NDX,TYP NDX)
145.              IF (NN .LT. 50) NPTS=NP(NN)
146.              IF (NN .EQ. 'IX' ) NPTS= IWID(IVAR)
147.              IF (NN .NE. 'BIG') GO TO 210
148.              NN=BXLOC(TYP NDX)
149.              NPTS=NP(NN)
150.       210    CONTINUE
151.
152.
153.       C      CALCULATE LOCATION OF VARIABLE IN FILE
154.              LOC= VABLOC(VBL NDX,TYP NDX) + (IVAR - 1)*VABREC(VBL NDX,TYP NDX)
155.              LOC=NBIN*BINSZE + LOC
156.       C      FILL THE ARRAY
157.              N R PER V=VABREC(VBL NDX , TYP NDX)
158.              DO 250 J=1,N R PER V
159.              CALL IODRRR(FCT,LOC,ARY(1,J))
160.              LOC=LOC+1
161.       250    CONTINUE
162.
163.              RETURN
164.       C      ------------------------------------------------------
165.
166.
167.
168.       C      ------------------------------------------------------------
169.                       ENTRY XFILL(ARY,NBIN,NAMVAR,IVAR,NPTS,BTYPE)
170.       C      *** FILL THE ARRAY,ARY, FROM BIN NUMBER NBIN
171.       C      AND VARIABLE OF NAME NAMVAR AND FROM INDEPENDANT VARIABLE VALUE
```

```
172.      C       OF IVAR.  RETURN THE NUMBER OF VALID POINTS,NPTS, AND BIN TYPE
173.
174.              CALL FND TYP(NBIN,BTYPE)
175.              CALL FND LOC(NAMVAR)
176.
177.      C        **  SET THE NUM OF PTS
178.              NN=BTLOC(TYP NDX)
179.              NPTS=NP(NN)
180.
181.      C       CALCULATE LOCATION OF VARIABLE IN FILE
182.              LOC=NBIN*BINSZE + VABLOC(VBL NDX , TYP NDX)
183.
184.      C       FILL THE ARRAY
185.              DO 350 JJ=1,NPTS
186.              CALL IODRRR(FCT,LOC,BUFFER)
187.              ARY(JJ,1)=BUFFER(IVAR)
188.              LOC=LOC+1
189.      350     CONTINUE
190.
191.              RETURN
192.      C       ------------------------------------------------
193.
194.
195.      C       ----------------------------------------
196.                        ENTRY TFILL(ARY,NBIN,BTYPE,NPTS,NTPTS)
197.
198.              CALL FND TYP(NBIN,BTYPE)
199.      C       **FIND THE BINLOC AND VECTOR SIZE OF VARIABLE
200.      C        ** NOTE THAT THE BIG ARRAY SHOULD ALWAYS START IN
201.      C       LOCATION #1. ALSO NOTE THAT 250 RECORDS ARE ALWAYS RETURNED
202.              LOC=NBIN*BINSZE + 1
203.              CALL  IODMRR(FCT,LOC,ARY,250)
204.
205.      C        ** SET THE NUM OF PTS
206.              NN=BXLOC(TYP NDX)
207.              NPTS=NP(NN)
208.
209.              NN=BTLOC(TYP NDX)
210.              NTPTS=NP(NN)
211.
212.              RETURN
213.      C       ------------------------------------------------
214.
215.
216.      C       ----------------------------------
217.                 ENTRY GET TYP(NBIN,BTYPE)
218.              CALL FND TYP(NBIN,BTYPE)
219.              RETURN
220.      C       ----------------------------------------
221.
222.
223.      C       ----------------------------------
224.                 ENTRY CHG REC(NBIN,NLOC,ARY)
225.              LOC=NBIN*BINSZE + NLOC
226.              CALL IODRRW(FCT,LOC,ARY)
227.              IIIB=BINSZE
228.              PRINT 1,NBIN,NLOC,LOC,IIIB
```

```
                =====  ENTRYS/IODR  =====

229.      1       FORMAT(' BIN-# ',I3,'  LOCATION',I3,'  WAS CHANGED'/
230.             .  ,' IODR LOCATION IS ',I4,'  BIN SIZE IS',I4)
231.              RETURN
232.      C       ------------------------------------
233.
234.
235.      C       ------------------------------------------------------
236.                              SUBROUTINE FND TYP(NBIN,BTYPE)
237.              INTEGER NBIN,BTYPE
238.      C       FIND THE TYPE OF BIN
239.              HLOC=NBIN+1
240.              BTYPE=HEADER(HLOC)
241.
242.      C       FIND THE BINDEX FROM THE TYPE
243.              DO 310 TYP NDX=1,BNAMES
244.              IF(BTYPE .EQ. BINS(TYP NDX) )  GO TO 320
245.      310     CONTINUE
246.              TYP NDX=0
247.              PRINT 311,BTYPE
248.      311     FORMAT(' BIN OF TYPE ',A6,' NOT FOUND')
249.              STOP BADBIN
250.
251.      320     CONTINUE
252.              TYP NDX=BTYPES(TYP NDX)
253.      C        ** RETRIEVE THE NUM OF PTS RECORD
254.              LOC=NBIN*BINSZE
255.              CALL IODRRR(FCT,LOC,NP)
256.              RETURN
257.
258.      C       ------------------------------------------------
259.
260.
261.
262.      C       ------------------------------------------------
263.                        SUBROUTINE FND LOC(NAMVAR)
264.              INTEGER NAMVAR
265.      C       **FIND THE BINLOC AND VECTOR SIZE OF VARIABLE
266.              DO 330 VBL NDX=1,MXVB
267.              IF ( NAMVAR .EQ. VABS(VBL NDX,TYP NDX) ) RETURN
268.      330     CONTINUE
269.              PRINT 331,NAMVAR,NBIN,BTYPE
270.      331     FORMAT(' NO VARIABLE ',A6,'  IN BIN# ',I3,' OF TYPE ',A6)
271.              STOP BADVAR
272.
273.      C       ------------------------------------------------
274.              END
```

```
      =====   ENTRYSNUM/IODR   =====

 1.                SUBROUTINE ENTRYS(NUMREC,ARY,ITYPE)
 2.             INCLUDE IODR
 3.
 4.            INTEGER NUMREC,ARY(50,1),ITYPE
 5.
 6.      C     ------------------------------------------------------------
 7.            ENTRY OPEN
 8.             READ 3,FLNAME
 9.      3      FORMAT(13A6)
10.            CALL IODROP(FCT,NUMFCT,FLNAME,BLOCK,RECORD,ICON,ID)
11.             RETURN
12.
13.      C      ------------------------------------------------------------
14.
15.
16.
17.      C     ------------------------------------------------------------
18.            ENTRY REOPEN
19.             READ 3,FLNAME
20.            CALL IODRRO(FCT,NUMFCT,FLNAME,BLOCK,RECORD,ICON,ID)
21.      C       **    RESET PARAMETERS **
22.             CALL IODRRR(FCT,0,HEADER)
23.             HLOC=MXBIN+1
24.             DIRLOC=MXBIN*2
25.      C      ** UPDATE THE LOG LOCATION POINTER
26.             CALL IODRRR(FCT,200,LOG)
27.             RETURN
28.
29.      C      ------------------------------------------------------------
30.
31.
32.      C      ------------------------------------------------
33.                          ENTRY OPN BIN(ITYPE)
34.      C      ********    DO THE BOOKEEPING  ********************
35.
36.      C     MARK DOWN THE TYPE AND FILE THE HEAD RECORD
37.             TYPE=ITYPE
38.            MXBIN=MXBIN+1
39.            HLOC=MXBIN+1
40.            HEADER(HLOC)=TYPE
41.            LOC=0
42.            CALL IODRRW(FCT,0,HEADER)
43.            CALL IODRRW(FCT,1,HEADER(51))
44.
45.      C     WRITE THE DIRECTORY INFO AND FILE
46.      C      ** READ THE MESSAGE **
47.             READ 3,MSG
48.             DIRLOC=MXBIN*2
49.             CALL IODRRW( FCT,DIRLOC,DIREC)
50.             DIRLOC=DIRLOC+1
51.             CALL IODRRW(FCT,DIRLOC,DIREC(51))
52.             CALL IODRCL(FCT)
53.
54.
55.      C     SET THE LOC COUNTER FOR SEQUENTIAL FILING
56.            LOC=MXBIN*BINSZE
57.
```

```
            =====  ENTRYSNUM/IODR  =====

 58.             RETURN
 59.      C      ************  END OF SUBROUTINE BOOK KP  ******************
 60.      C      ------------------------------------------------
 61.
 62.
 63.      C      ------------------------------------------------
 64.                        ENTRY FILE (ARY,NUMREC)
 65.
 66.             DO 100 I=1,NUMREC
 67.             CALL IODRRW(FCT,LOC,ARY(1,I))
 68.             LOC=LOC+1
 69.      100    CONTINUE
 70.             RETURN
 71.
 72.      C      *******   END OF SUBROUINE FILE ************************
 73.
 74.      C      ------------------------------------------------------
 75.
 76.
 77.
 78.      C      ------------------------------------------------------
 79.             ENTRY CLEAR
 80.             CALL IODRCL(FCT)
 81.             RETURN
 82.
 83.      C      ------------------------------------------------------------
 84.
 85.
 86.
 87.
 88.      C      ------------------------------------------------------
 89.             ENTRY LOGIN
 90.             DO 12 I=1,50
 91.             READ(5,3,END=14)LOGMSG
 92.             CALL IODRRW (FCT,LOGLOC,LOGMSG)
 93.             LOGLOC=LOGLOC+1
 94.      12     CONTINUE
 95.
 96.      C      ** FILE THE LOGLOC **
 97.      14     CONTINUE
 98.             CALL IODRRW (FCT,200,LOG)
 99.              CALL IODRCL(FCT)
100.             RETURN
101.
102.      C      ------------------------------------------------------
103.
104.
105.      C      ------------------------------------------------
106.             ENTRY GET LOG
107.
108.             PRINT 131
109.      131    FORMAT('1')
110.
111.      C      ** PRINT THE LOG **
112.             DO 133 LOC=201,LOGLOC
113.             CALL IODRRR(FCT,LOC,LOGMSG)
114.             PRINT 132,LOGMSG
```

```
      =====  ENTRYSNUM/IODR  =====

115.      132     FORMAT(13X,13A6)
116.      133     CONTINUE
117.
118.      C       ** PRINT THE DIRECTORY **
119.              PRINT 140
120.      140     FORMAT(' ',//,'     BIN    TYPE                          CONTENTS')
121.              DIRLOC=2*MXBIN
122.              DO 141 LOC=2,DIRLOC,2
123.              CALL IODRRR(FCT,LOC,DIREC)
124.              I=LOC/2
125.              PRINT 142,I,TYPE,MSG
126.      142     FORMAT(' ',I6,2X,A6,'   ----   ',13A6)
127.      141     CONTINUE
128.              RETURN
129.      C       ------------------------------------------------------------
130.              END
```

```
=====  GETFIL/IODR  =====

      SUBROUTINE GETFIL(IBIN,INST)
      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
      COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
      COMMON/OUT/NT,TA(250),NY,Y(250)
      INTEGER BTYPE
C
C
      IF(INST .EQ. 'RDFIL1') GO TO 10
      IF(INST .EQ. 'RDFIL2') GO TO 20
      IF(INST .EQ. 'RDFIL3') GO TO 30
C
C    ION 1 TYPES  DEPOSITED ENERGY , DPA RATES ,ETC
C
10     CONTINUE
      CALL TFILL(TEMP,IBIN,IHEAD,IX,IT)
      CALL FILL(FT,IBIN,'FT',1,NP,BTYPE)
      CALL FILL(FF,IBIN,'FF',1,NP,BTYPE)
      CALL FILL(E,IBIN,'E',1,NP,BTYPE)
      CALL FILL(T,IBIN,'T',1,NP,BTYPE)
      NE = NP
      CALL FILL(X,IBIN,'X',1,NP,BTYPE)
      NX = NP
      DO 100 I = 1,200
      DO 100 J = 1,50
100   DEPX(J,I) = TEMP(J,I)
      PRINT 200, IBIN,BTYPE,NE,NX
200    FORMAT(//,'  FROM BIN # ',I5,'  OF TYPE  ',A6,//,
     1  3X,I5,'  TIME POINTS AND  ',I5,'  X POINTS WERE READ',//)
      GO TO 1000
C
C    ION 2 TYPES INT DEPOSITED ENERGY, DPA, ETC
C
20     CONTINUE
      CALL TFILL(TEMP,IBIN,IHEAD,IX,IT)
      CALL FILL(FT,IBIN,'FT',1,NP,BTYPE)
      CALL FILL(FF,IBIN,'FF',1,NP,BTYPE)
      CALL FILL(E,IBIN,'E',1,NP,BTYPE)
      CALL FILL(T,IBIN,'T',1,NP,BTYPE)
      NE = NP
      CALL FILL(X,IBIN,'X',1,NP,BTYPE)
      NX = NP
      PRINT 200, IBIN,BTYPE,NE,NX
      GO TO 1000
C
C    ION 3 TYPES TEMPERATURES
C
30     CONTINUE
      CALL TFILL(TEMP,IBIN,IHEAD,IX,IT)
      CALL FILL(FT,IBIN,'FT',1,NP,BTYPE)
      CALL FILL(FF,IBIN,'FF',1,NP,BTYPE)
      CALL FILL(E,IBIN,'E',1,NP,BTYPE)
      CALL FILL(T,IBIN,'T',1,NP,BTYPE)
      NE = NP
      CALL FILL(TA,IBIN,'TA',1,NP,BTYPE)
```

```
=====  GETFIL/IODR  =====

58.              NST = NP
59.              CALL FILL(Y,IBIN,'Y',1,NP,BTYPE)
60.              NSX = NP
61.              PRINT 200, IBIN,BTYPE,NST,NSX
62.      1000     RETURN
63.              END
```

===== NPCHANGE/IODR =====

```
 1.              INTEGER  BTYPE,NPTS(50)
 2.              NPTS(3)=15
 3.              NPTS(4)=115
 4.              NPTS(1)=15
 5.              NPTS(2)=115
 6.              CALL REOPEN
 7.
 8.      1       CONTINUE
 9.              PRINT 2
10.      2       FORMAT(' TYPE IN A BIN NUMBER')
11.              READ  (-,-,END=99) NBIN
12.              CALL GET TYP(NBIN,BTYPE)
13.              PRINT 3,NBIN,BTYPE
14.      3       FORMAT(' OK TO CHANGE BIN',I3,'OF TYPE',A6,'?')
15.              READ 4,INST
16.      4       FORMAT(A6)
17.              IF(INST .NE. 'Y') GO TO 1
18.              CALL CHG REC(NBIN,0,NPTS)
19.              GO TO 1
20.
21.      99      CONTINUE
22.              CALL CLEAR
23.              STOP DONE
24.              END
```

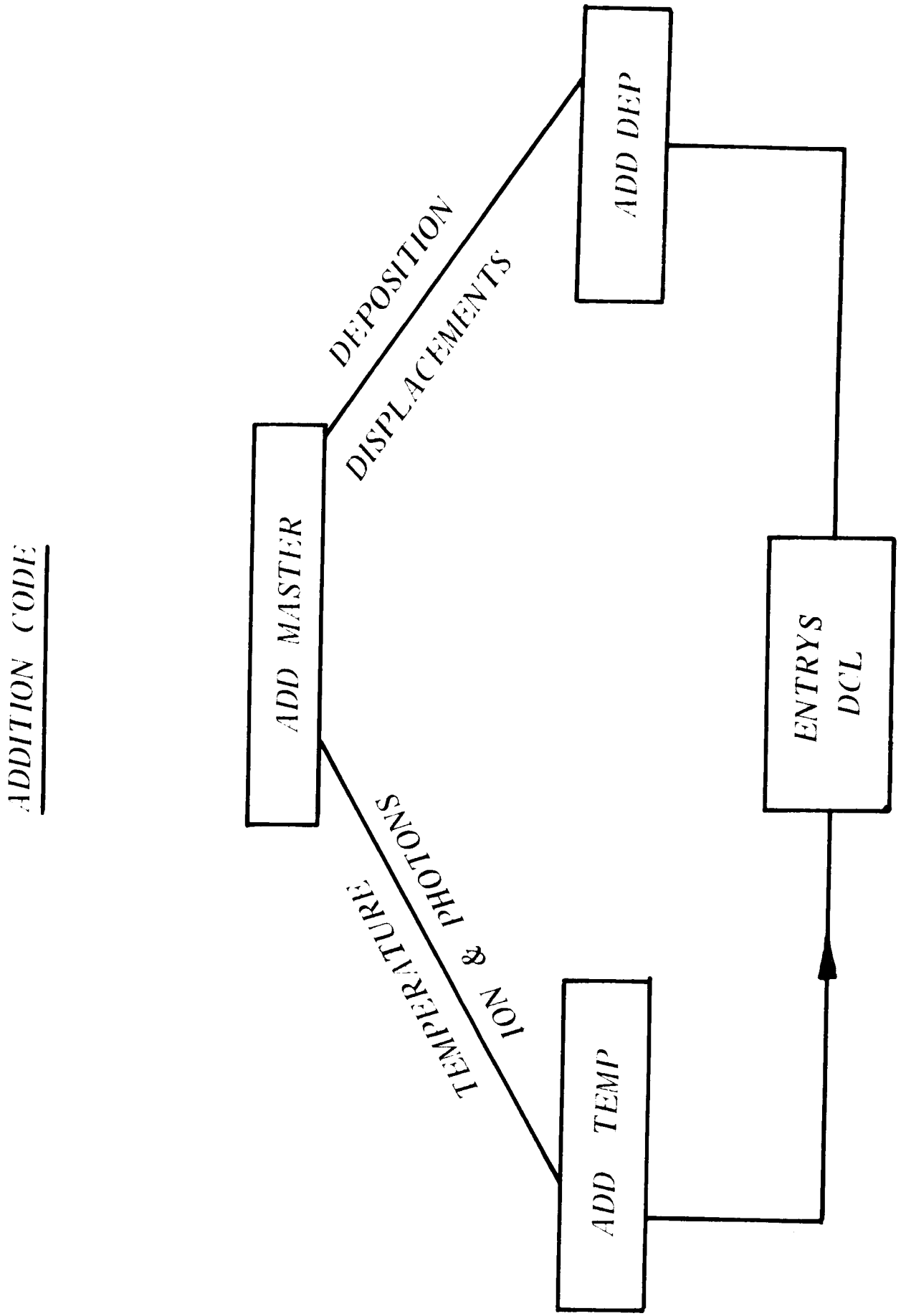
```
      =====  RDFILE/IODR  =====

      COMMON/BAS/T(200),E(200),FT(200),POW(200),PR(200),TEMP(50,250),
     &          DEPX(50,200),DEL(200),X(200),FF(200)
      COMMON/VAR/DT,DX,BB,NX
      COMMON/ION/ISPEC,EMN,SIG,EMIN,EMAX,NE,FL,R,A,W,NPLUS
      COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
      COMMON/OUT/NT,TA(250),NY,Y(250)
      INTEGER BTYPE
C
      CALL REOPEN
C
4      CONTINUE
      READ(5,72,END = 1000) INST
72     FORMAT(A6)
      READ , IBIN,INOUT,IA,IB,IC
C
C
      IF(INST .EQ. 'RDFIL1') GO TO 10
      IF(INST .EQ. 'RDFIL2') GO TO 20
      IF(INST .EQ. 'RDFIL3') GO TO 30
C
C    ION 1 TYPES  DEPOSITED ENERGY , DPA RATES ,ETC
C
10     CONTINUE
      CALL TFILL(TEMP,IBIN,IHEAD,IX,IT)
      CALL FILL(FT,IBIN,'FT',1,NP,BTYPE)
      CALL FILL(FF,IBIN,'FF',1,NP,BTYPE)
      CALL FILL(E,IBIN,'E',1,NP,BTYPE)
      CALL FILL(T,IBIN,'T',1,NP,BTYPE)
      NE = NP
      CALL FILL(X,IBIN,'X',1,NP,BTYPE)
      NX = NP
      DO 100 I = 1,200
      DO 100 J = 1,50
100   DEPX(J,I) = TEMP(J,I)
      PRINT 200, IBIN,BTYPE,NE,NX
200    FORMAT(//,'  FROM BIN # ',I5,'  OF TYPE  ',A6,//,
     1  3X,I5,'  TIME POINTS AND  ',I5,'  X POINTS WERE READ',//)
      IF(INOUT .LT. 1) GO TO 4
       GO TO (1,2,3),INOUT
1      PRINT 400
400    FORMAT('1',//,'  I      TIME      ENERGY      FLUX     SPECTRUM',//)
      DO 410 I = 1,NE
410    PRINT 401,I,T(I),E(I),FT(I),FF(I)
401    FORMAT(I5,7E10.4)
      GO TO 4
2      PRINT 500
500    FORMAT('1',//,'  TIME FUNCTIONS AT VARIOUS X VALUES',//)
      PRINT 501, (X(J),J = IA,IB,IC)
501    FORMAT(//,15X,6E10.4,//)
      DO 510 I = 1,NE
510    PRINT 401,I,T(I),(DEPX(J,I),J = IA,IB,IC)
       GO TO 4
3      PRINT 600
600    FORMAT('1',//,'  X FUNCTIONS AT VARIOUS TIMES',//)
      PRINT 501,(T(J),J = IA,IB,IC)
      DO 610 I = 1,NX
610    PRINT 401,I,X(I),(DEPX(I,J),J = IA,IB,IC)
```

```
58.              GO TO 4
59.       C
60.       C    ION 2 TYPES INT DEPOSITED ENERGY, DPA, ETC
61.       C
62.       20      CONTINUE
63.              CALL TFILL(TEMP,IBIN,IHEAD,IX,IT)
64.              CALL FILL(FT,IBIN,'FT',1,NP,BTYPE)
65.              CALL FILL(FF,IBIN,'FF',1,NP,BTYPE)
66.              CALL FILL(E,IBIN,'E',1,NP,BTYPE)
67.              CALL FILL(T,IBIN,'T',1,NP,BTYPE)
68.              NE = NP
69.              CALL FILL(X,IBIN,'X',1,NP,BTYPE)
70.              NX = NP
71.              PRINT 200, IBIN,BTYPE,NE,NX
72.              IF(INOUT .LT. 1 ) GO TO 4
73.               GO TO (11,22,33),INOUT
74.       11       PRINT 400
75.              DO 710 I = 1,NE
76.       710     PRINT 401,I,T(I),E(I),FT(I),FF(I)
77.              GO TO 4
78.       22      PRINT 500
79.              PRINT 501, (X(J),J = IA,IB,IC)
80.              DO 810 I = 1,NE
81.       810     PRINT 401,I,T(I),(TEMP(J,I),J = IA,IB,IC)
82.               GO TO 4
83.       33      PRINT 600
84.              PRINT 501,(T(J),J = IA,IB,IC)
85.              DO 910 I = 1,NX
86.       910     PRINT 401,I,X(I),(TEMP(I,J),J = IA,IB,IC)
87.              GO TO 4
88.       C
89.       C    ION 3 TYPES TEMPERATURES
90.       C
91.       30      CONTINUE
92.              CALL TFILL(TEMP,IBIN,IHEAD,IX,IT)
93.              CALL FILL(FT,IBIN,'FT',1,NP,BTYPE)
94.              CALL FILL(FF,IBIN,'FF',1,NP,BTYPE)
95.              CALL FILL(E,IBIN,'E',1,NP,BTYPE)
96.              CALL FILL(T,IBIN,'T',1,NP,BTYPE)
97.              NE = NP
98.              CALL FILL(TA,IBIN,'TA',1,NP,BTYPE)
99.              NST = NP
100.             CALL FILL(Y,IBIN,'Y',1,NP,BTYPE)
101.             NSX = NP
102.             PRINT 200, IBIN,BTYPE,NST,NSX
103.             IF(INOUT .LT. 1 ) GO TO 4
104.              GO TO (111,222,333),INOUT
105.      111       PRINT 400
106.             DO 210 I = 1,NE
107.      210     PRINT 401,I,T(I),E(I),FT(I),FF(I)
108.             GO TO 4
109.      222       PRINT 500
110.             PRINT 501, (Y(J),J = IA,IB,IC)
111.             DO 310 I = 1,NST
112.      310     PRINT 401,I,TA(I),(TEMP(J,I),J = IA,IB,IC)
113.             GO TO 4
114.      333      PRINT 600
```

```
      =====  RDFILE/IODR  =====

115.            PRINT 501,(TA(J),J = IA,IB,IC)
116.            DO 1310 I = 1,NSX
117.     1310     PRINT 401,I,Y(I),(TEMP(I,J),J = IA,IB,IC)
118.            GO TO 4
119.     1000    CONTINUE
120.            END
```

```
=====  READ-DIR/IODR  =====

1.          CALL REOPEN
2.          CALL GET LOG
3.          STOP DONE
4.          END
```

```
=====  STARCHANGE/IODR  =====

      INTEGER  BTYPE,NPTS(50)
      COMMON/STAN/ST(250),SX(50),NST,NSX,STP(50),NTP
      NPTS(3)=15
      CALL REOPEN
      CALL STARAY

1     CONTINUE
C       DO 200 NBIN=1,27
      PRINT 1522
1522  FORMAT(' TYPE IN A BIN#')
      READ(-,-,END=99)NBIN
      CALL GET TYP(NBIN,BTYPE)
      IF (BTYPE .NE. 'ION-3') GO TO 200
      PRINT 3,NBIN,BTYPE
3     FORMAT(' OK TO CHANGE BIN',I4,'  OF TYPE  ',A6,'?')
      READ 4,INST
4     FORMAT(A6)
      IF(INST .EQ. 'Y')
     .CALL CHG REC(NBIN,271,SX)
200   CONTINUE

      GO TO 1
99    CONTINUE
      CALL CLEAR
      STOP DONE
      END
```


ADDITION CODE
ADD MASTER
DEPOSITION
DISPLACEMENTS
ADD DEP
ENTRYS
DCL
TEMPERATURE
ION & PHOTONS
ADD TEMP

```
      =====  ADDITION  =====

 1.
 2.
 3.
 4.
 5.
 6.
 7.                    ADDITION
 8.              SECTION OF T*DAMEN CODE
 9.
10.
11.          ABSOLUTE
12.              ADDITION
13.
14.
15.          READER
16.              READ/ADD
17.
18.
19.          MAPPER
20.              MAP/ADDITION
21.
22.
23.          LISTER
24.              LIST/ADDITION
25.
26.
27.          SYMBOLICS
28.              ADDMASTER  --CALLS EITHER ADDTEMP OR ADDDEP
29.              ADDTEMP    --READS AN ION-3 FILE OR A PHOTON FILE AND
30.                           ADDS THE RESULTS TOGETHER AND FILLING
31.                           THE PHOTON ARRAY; FILES RESULT AS AN
32.                           ION-3 FILE
33.              ADDDEP     --READS AN ION-1 FILE OR ION-2 FILE AND ADDS
34.                           RESULTS TOGETHER, ALSO ADDS DISPLACEMENTS
35.                           FILES RESULT AS ION-2 FILE
36.              ENTRYS/IODR--BASIC SUBROUTINE FOR IODR HANDLING
37.              DCL/IODR   --BASIC FILE STRUCTURE FOR IODR DATA BLOCKS
38.
39.
40.          RUNSTREAMS
41.              RUNADDITION
```

```
=====  LIST/ADDITION  =====

1.      @PRT,S  T*DAMEN.LIST/ADDITION
2.      @PRT,S  T*DAMEN.ADDMASTER
3.      @PRT,S  T*DAMEN.ADDTEMP
4.      @PRT,S  T*DAMEN.ADDDEP
5.      @PRT,S  T*DAMEN.ENTRYS/IODR
6.      @  .    SEE FILING
7.      @PRT,S  T*DAMEN.DCL/IODR
8.      @  .    SEE FILING
```

```
=====  READ/ADDITION  =====

                  *****READ STATEMENT FOR THE ADDITION CODE*****



             FOR THE ADDMASTER SUBROUTINE:

           READ(5,10,END=99)INST
10          FORMAT(A6)
           IF(INST .EQ. 'ADDTEM') CALL ADDTEM
           IF(INST .EQ. 'ADDDEP') CALL ADDDEP


             FOR THE ADDTEMP SUBROUTINE:

C    READ IN BIN NUMBER

1          READ(5,-,END = 400) IBIN


             FOR THE ADDDEP SUBROUTINE:

1          READ(5,-,END = 400) IBIN


             FOR THE ENTRYS/IODR SUBROUTINE:

           READ 3,FLNAME
3          FORMAT(13A6)
           READ 3,FLNAME
           READ 3,MSG
           READ(5,3,END=14)LOGMSG
```

```
=====  MAP/ADDITION  =====

1.    @MAP*MAP.MAP    ,T*DAMEN.ADDITION
2.    IN T*DAMEN.ADDMASTER,.ADDDEP,.ADDTEMP
3.    IN T*DAMEN.ENTRYS/IODR
4.    END
```

```
=====  ADDDEP  =====

      SUBROUTINE ADDDEP
      DIMENSION DEP(50,250),SUMDEP(50,250)
     1  ,T(200),X(200)
      INTEGER  NPTS(50)
      COMMON/ADDVAR/ TEMP(50,250),SUMTEM(50,250),ATT(50)
     1    ,TA(200),Y(50)
      EQUIVALENCE (SUMTEM,SUMDEP),(TEMP,DEP)
C
C     SET SUM ARRAY EQUAL TO ZERO
C
      DO 333 I = 1,50
      DO 333 J = 1,250
333     SUMDEP(I,J) = 0.0
C    READ IN BIN NUMBER
C
       I = 1
1     READ(5,-,END = 400) IBIN
C
C     READ HEADER AND TEMPERATURE ARRAY FROM  STORAGE
C
      CALL TFILL(DEP,IBIN,IHEAD,IX,IT)
C
C    IS  THE BIN ONE CREATED BY A DEPOSITION RUN
C      IS THE BIN ION OR PHOTON
C
      IF(I .EQ.1) IFIRST = IHEAD
      IF(IHEAD.NE.IFIRST) GO TO 1000
      IF(IHEAD .EQ. 'ION-1') GO TO 200
      IF(IHEAD .EQ. 'ION-2') GO TO 200
      PRINT 10
10     FORMAT(' A BIN WAS CHOSEN FOR SUMMATION WHICH DID NOT ',/
     1 ,'  COME FROM A DEPOSITION OR TIME INTEGRATED DEPOSITION')
      RETURN
200    CALL FILL(T,IBIN,'T',1,NP,BTYPE)
      NPTS(2) = NP
       CALL FILL(X,IBIN,'X',1,NP,BTYPE)
      NPTS(1) = NP
         DO 300 I = 1,50
      DO 250 J = 1,200
      SUMDEP(I,J) = SUMDEP(I,J) + DEP(I,J)
250    CONTINUE
300     CONTINUE
C
C GO BACK AND READ ANOTHER BIN NUMBER
C
      I = I+1
        GO TO 1
400     CONTINUE
      PRINT 337,NPTS(1),NPTS(2)
337   FORMAT(' THE VALUES FOR THE NUMBER OF POINTS ARE  ',I5,'AND',I5)
      DO 555 J = 50,90
      PRINT 556,(SUMDEP(K,J),K = 1,10)
556    FORMAT(10(2X,E10.4))
555    CONTINUE
C
C    FILE THE ADDED VALUE AS AN ION-2 BIN
C
```

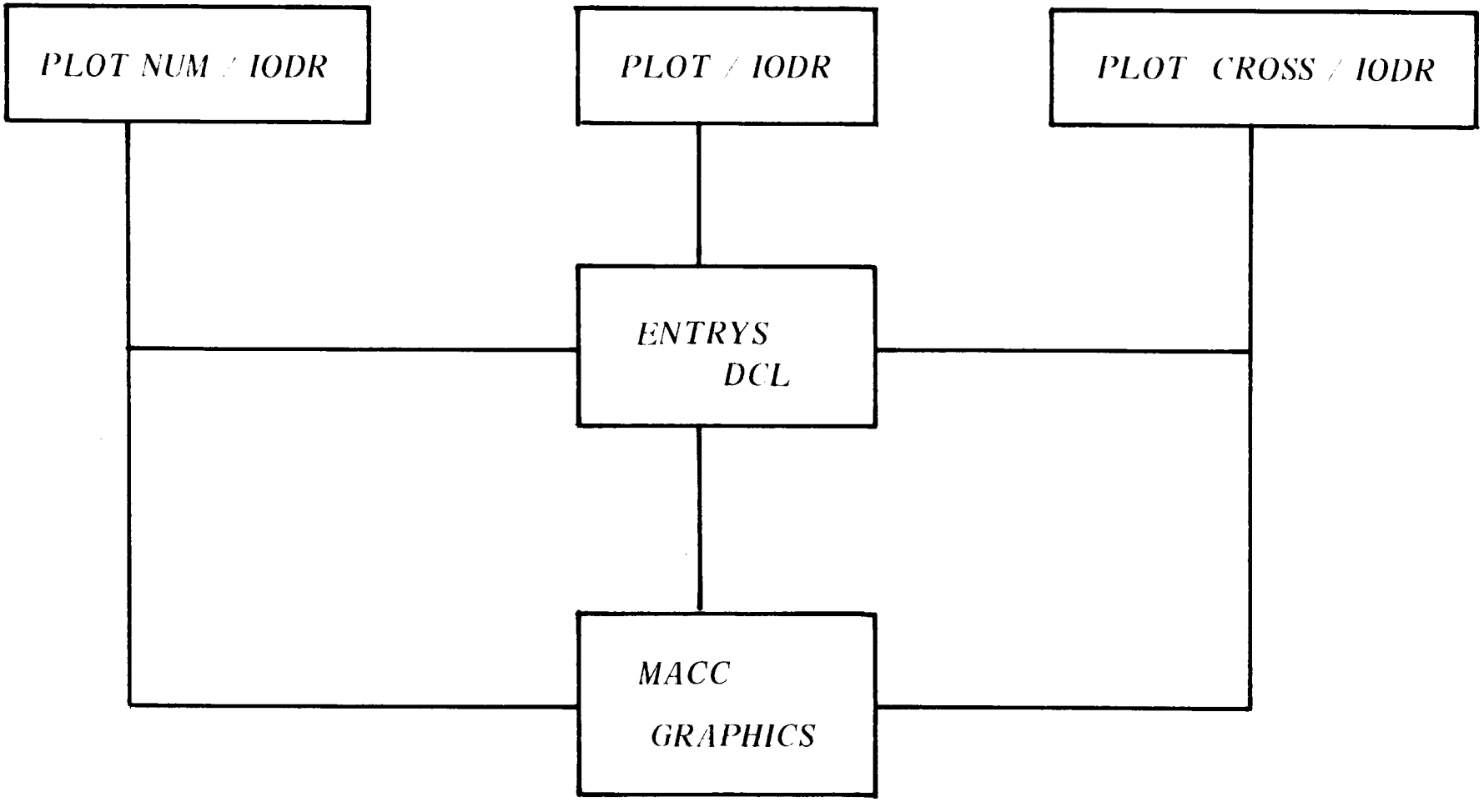
```
      =====  ADDDEP  =====

58.            CALL OPN BIN('ION-2')
59.      C     __  FILE THE DATA
60.            CALL FILE(NPTS,1)
61.           CALL FILE(SUMDEP,250)
62.            CALL FILE(T,4)
63.           CALL FILE(T,4)
64.            CALL FILE(T,4)
65.            CALL FILE(T,4)
66.            CALL FILE(X,4)
67.            CALL CLEAR
68.            RETURN
69.      1000   PRINT 11
70.      11     FORMAT('     ALL BINS WERE NOT OF THE SAME TYPE TRY AGAIN')
71.            RETURN
72.            END
```

```
      =====   ADDMASTER   =====

 1.      C     ***************************************
 2.      C
 3.      C   THIS ISTHE DRIVER FOR THE SUMMING ROUTINE
 4.      C
 5.      C     ***********************************************
 6.              INTEGER INST
 7.      C
 8.              CALL REOPEN
 9.      C
10.      1          CONTINUE
11.      C
12.      C    FINDOUT WHAT TODO
13.      C
14.              READ(5,10,END=99)INST
15.              PRINT  1000,INST
16.      1000    FORMAT(' ',A6)
17.      10       FORMAT(A6)
18.              IF(INST .EQ. 'ADDTEM') CALL ADDTEM
19.              IF(INST .EQ. 'ADDDEP') CALL ADDDEP
20.      C
21.              GO TO 1
22.      C
23.      99      STOP
24.              END
```

```
=====  ADDTEMP  =====

      SUBROUTINE ADDTEM
      COMMON/ADDVAR/ TEMP(50,250),SUMTEM(50,250),ATT(50)
     1    ,TA(200),Y(50)
      INTEGER NPTS(50)
C
C     SET SUM ARRAY EQUAL TO ZERO
C
      DO 333 I = 1,50
      DO 333 J = 1,250
333     SUMTEM(I,J) = 0.0
C    READ IN BIN NUMBER
C
1       READ(5,-,END = 400) IBIN
C
C       READ HEADER AND TEMPERATURE ARRAY FROM  STORAGE
C
        CALL TFILL(TEMP,IBIN,IHEAD,IX,IT)
        IF(IHEAD .EQ. 'PHOTON') GO TO 30
        NP = 200
        CALL FILL(TA,IBIN,'TA',1,NP,BTYPE)
        NP = 50
        CALL FILL(Y,IBIN,'Y',1,NP,BTYPE)
30        CONTINUE
C
C      IS THE BIN ION OR PHOTON
C
        IF(IHEAD .EQ. 'ION-3') GO TO 200
        IF(IHEAD .EQ. 'PHOTON') GO TO 40
C
C   FILL IN THE PHOTON ARRAY
C
        PRINT 111
111      FORMAT('  A BIN WAS CHOSEN WHICH WAS NOT A PHOTON TEMPERATURE',/
     1    ,'  OR ION TEMPERATURE CALCULATION',//)
        RETURN
40       DO 100 I = 1,50
        DO 50 J = 1,22
50       ATT(J) = TEMP(I,J)
        DO 60 J = 1,19,2
        K = (J+1)/2
        TEMP(I,J)   = ATT(K)
        TEMP(I,J+1) = (ATT(K+1) + ATT(K))/2.
60       CONTINUE
        DO 70 J = 19,99,10
        M = 9 + (J+1)/10
        DO 65 K = 1,10
        N = J + K
        X = K
        S = X/10.
        TEMP(I,N) = (ATT(M) - ATT(M-1))*S + ATT(M-1)
65       CONTINUE
70       CONTINUE
        TEMP(I,110) = (ATT(20) + ATT(19))/2.
        TEMP(I,111) = ATT(20)
        TEMP(I,112) = (ATT(21) + ATT(20))/2.
        TEMP(I,113) = ATT(21)
        TEMP(I,114) = (ATT(22) + ATT(21))/2.
```

```
=====  ADDTEMP  =====

 58.              TEMP(I,115) = ATT(22)
 59.      100       CONTINUE
 60.               DO 150 J = 1,115
 61.       150     PRINT 500,J,(TEMP(I,J),I = 1,10)
 62.      500      FORMAT(I5,10(2X,E9.3))
 63.      200      CONTINUE
 64.      C
 65.      C    ADD THE ARRAYS TOGETHER **********************
 66.      C
 67.              DO 300 I = 1,50
 68.              DO 250 J = 1,115
 69.              SUMTEM(I,J) = SUMTEM(I,J) + TEMP(I,J)
 70.      250      CONTINUE
 71.      300       CONTINUE
 72.      C
 73.      C GO BACK AND READ ANOTHER BIN NUMBER
 74.      C
 75.              GO TO 1
 76.      C       WRITE SUMTEM BACK INTO THE STORAGE
 77.      400      CONTINUE
 78.              DO 555 I = 1,90
 79.              PRINT 500,I,(SUMTEM(K,I),K = 1,10)
 80.      555      CONTINUE
 81.      C
 82.      C    MAKE THE FILING THE SAME AS ION-3
 83.      C
 84.      C        **  NOTE THET THE FOLLOWING SET OF NUMBERS
 85.      C       FOR THE NUMBER OF VALID POINTS IN EACH DIMENSOIN
 86.      C       IS SUBJECT TO CHANGE
 87.              NPTS(3)=15
 88.              NPTS(4)=115
 89.
 90.              CALL OPNBIN('ION-3')
 91.              CALL FILE(NPTS,1)
 92.              CALL FILE(SUMTEM,250)
 93.              CALL FILE(TA,4)
 94.              CALL FILE(TA,4)
 95.              CALL FILE(TA,4)
 96.              CALL FILE(TA,4)
 97.              CALL FILE(TA,4)
 98.              CALL FILE(Y,1)
 99.              CALL CLEAR
100.              RETURN
101.              END
```

PLOT CODE


PLOT NUM / IODR
PLOT / IODR
PLOT CROSS / IODR
ENTRYS
DCL
MACC
GRAPHICS

```
=====  PLOTTING  =====


                    PLOTTING
                SECTION OF T*DAMEN CODE


ABSOLUTE
    PLOT/IODR    --PLOTTER FROM IODR FILES
    PLOTNUM/IODR--PLOTTER FROM IODR FILES SPECIFY #OF PTS
    PLOTCROSS/IODR--SELECTS DATA FROM DIFFERENT IODR FILES


READER
    READ/PLOT


MAPPER
    MAP/PLOT


LISTER
    LIST/PLOT


SYMBOLICS
    PLOT/IODR--BASIC PLOT PACKAGE FOR PLOTTING FROM ANY
                IODR FILE 1-15 POSSIBLE
    PLOTNUM/IODR--SAME AS PLOT/IODR EXCEPT # OF POINTS
                  IN EACH LINE MUST BE SPECIFIED
    PLOTCROSS/IODR--SAME AS PLOT/IODR EXCEPT DATA ARE FROM
                    DIFFERENT FILES
    ENTRYS/IODR--BASIC IODR HANDLING SUBROUTINES
    ENTRYSNUM/IODR--SAME AS ENTRYS/IODR EXCEPT SUPPORTS
                    PLOTNUM
    DCL/IODR--STRUCTURE OF DATA BLOCKS IN IODR FILES
    PLT-ID--A LISTING OF THE 15 PLOT COMBINATIONS


RUNSTREAMS
    TOTALPLOT
```

```
=====  READ/PLOTTING  =====

                *****READ STATEMENTS FOR THE PLOTTING CODE*****


           FOR THE PLOT/IODR SUBROUTINE:

      NAMELIST  /IN/ NP,PRINT,PLOT,IDEV,LONG Y,PRT LNS
      READ (5,IN)
      IF (PRT LNS) CALL LNS OUT
      IF(LONG Y)TILT='NORMAL'
      CALL DEVSET(IDEV)
5     READ (-,-,END=10)NUMBIN,NUMLN,INDVAR
      IF(NUMLN .NE. 15) GO TO 26
      READ (-,-,END=99)ILOGX,ILOGY
12    FORMAT(13A6)
      READ  12,XLAB,YLAB,TITLE
122   FORMAT(A6)

      IF (INS .EQ. 'PLT' .AND. IDEV .NE. 1)    CALL PLTTR
      IF (INS .NE. 'PLT') PRINT 323
323   FORMAT('  PAGE WILL NOT BE PLOTTED')
      READ   122,INS


           FOR THE PLOTNUM/IODR SUBROUTINE:

      NAMELIST  /IN/ NP,PRINT,PLOT,IDEV
      READ (5,IN)
      CALL DEVSET(IDEV)
5     READ (-,-,END=10)NUMBIN,NUMLN,INDVAR
      IF(NUMLN .NE. 15) GO TO 26
     IF(NUMLN.NE.  8 .AND. NUMLN .NE. 14 ) GO TO 28
      READ ,ILOGX,ILOGY
      IF(ILOGX .EQ. 1 )LOGX='LINEAR'
      IF(ILOGY .EQ. 1 ) LOGY='LINEAR'
      READ  12,XLAB,YLAB,TITLE
12    FORMAT(6A6)


           FOR THE ENTRYS/IODR SUBROUTINE:

      READ 3,FLNAME
3     FORMAT(13A6)
     CALL IODROP(FCT,NUMFCT,FLNAME,BLOCK,RECORD,ICON,ID)
      READ 3,FLNAME
     CALL IODRRO(FCT,NUMFCT,FLNAME,BLOCK,RECORD,ICON,ID)
      READ 3,MSG
      READ(5,3,END=14)LOGMSG
     CALL IODRRW (FCT,LOGLOC,LOGMSG)


           FOR THE ENTRYSNUM/IODR SUBROUTINE:

      READ 3,FLNAME
3     FORMAT(13A6)
     CALL IODROP(FCT,NUMFCT,FLNAME,BLOCK,RECORD,ICON,ID)
      READ 3,FLNAME
```

```
            =====  READ/PLOTTING  =====

58.          READ 3,MSG
59.          READ(5,3,END=14)LOGMSG
60.
61.
62.             FOR THE NPCHANGE/IODR SUBROUTINE:
63.
64.          READ  (-,-,END=99) NBIN
65.          READ 4,INST
66.     4    FORMAT(A6)
67.          IF(INST .NE. 'Y') GO TO 1
68.
69.
70.             FOR THE STARCHANGE/IODR SUBROUTINE:
71.
72.          READ(-,-,END=99)NBIN
73.          READ 4,INST
74.     4    FORMAT(A6)
75.          IF(INST .EQ. 'Y')
76.
77.
78.             FOR THE PLOTCROSS/IODR SUBROUTINE:
79.
80.          NAMELIST  /IN/ NP,PRINT,PLOT,IDEV,LONG Y,PRT LNS
81.          READ (5,IN)
82.          IF (PRT LNS) CALL LNS OUT
83.          IF(LONG Y)TILT='NORMAL'
84.     5    READ (-,-,END=10)NUMBIN,NUMLN,INDVAR
85.          IF(NUMLN .NE. 15) GO TO 26
86.         IF(NUMLN.NE.  8 .AND. NUMLN .NE. 14 ) GO TO 28
87.          READ (-,-,END=99)ILOGX,ILOGY
88.          IF(ILOGX .EQ. 1 )LOGX='LINEAR'
89.          IF(ILOGY .EQ. 1 ) LOGY='LINEAR'
90.          READ  12,XLAB,YLAB,TITLE
91.     12   FORMAT(13A6)
92.          READ   122,INS
93.     122  FORMAT(A6)
94.
95.          IF (INS .EQ. 'PLT' .AND. IDEV .NE. 1)    CALL PLTTR
96.          IF (INS .NE. 'PLT') PRINT 323
97.     323  FORMAT('  PAGE WILL NOT BE PLOTTED')
```

```
=====  LIST/PLOTTING  =====

 1.     @PRT,S  T*DAMEN.LIST/PLOTTING
 2.     @PRT,S  T*DAMEN.PLOT/IODR
 3.     @PRT,S  T*DAMEN.PLOTNUM/IODR
 4.     @PRT,S  T*DAMEN.READ-DIR/IODR
 5.     @ .     SEE FILING
 6.     @PRT,S  T*DAMEN.ENTRYS/IODR
 7.     @ .     SEE FILING
 8.     @PRT,S  T*DAMEN.ENTRYSNUM/IODR
 9.     @ .     SEE FILING
10.     @PRT,S  T*DAMEN.DCL/IODR
11.     @ .     SEE FILING
12.     @PRT,S  T*DAMEN.NPCHANGE/IODR
13.     @ .     SEE FILING
14.     @PRT,S  T*DAMEN.STARCHANGE/IODR
15.     @ .     SEE FILING
16.     @PRT,S  T*DAMEN.PLT-ID
17.     @PRT,S  T*DAMEN.PLOTCROSS/IODR
```

```
      =====  PLOT/IODR  =====

C       *********  THIS ROUTINE DOES THE IODR FILE PLOTTING *****

C        ** FIRST SET UP THE PLOT TABLE ETC.
        PARAMETER NL=14
        INTEGER YVARB(NL)/'FT','FF','DEPX','DEPX','TEMP','TEMP','TEMP',
     .     'TEMP','F','U','Y1','ADBTMP','TEMP','TEMP'/

        INTEGER XVARB(NL)/'T','E','T','X','T','X','TA','Y',
     .         'E','E','X','X','TM','Y'/

        LOGICAL YCROSS(NL)/.FALSE.,.FALSE.,.TRUE.,.FALSE.,.TRUE.,.FALSE.,
     . .TRUE.,.FALSE., .FALSE.,.FALSE.,.FALSE.,.FALSE.,.TRUE.,.FALSE./

        LOGICAL XCROSS(NL)/NL*.FALSE./

C        *** NOW MAKE SPACE TO STORE THE VARIOUS LINES ***
        PARAMETER MXLNS=20
        REAL XPTS(250,MXLNS),YPTS(250,MXLNS)
        INTEGER NPTS(MXLNS)

C       *********       INPUT PARAMETERS     ******
        INTEGER NP/50/      IDEV/5/
        LOGICAL PRINT/.FALSE./      PLOT/.TRUE./
        LOGICAL LONG Y /.FALSE./
        LOGICAL PRT LNS/.FALSE./
        NAMELIST  /IN/ NP,PRINT,PLOT,IDEV,LONG Y,PRT LNS

C       ********************   PLOT VARIABLES   *******
        REAL       XSCALE(10),YSCALE(10)
        INTEGER XL(14)/13*' ','$$'/YL(14)/13*' ','$$'/TI(14)/13*' ','$$'/
        INTEGER  XLAB(13),YLAB(13),TITLE(13)
        EQUIVALENCE (XLAB,XL) , (YLAB,YL)  , (TITLE,TI)
        INTEGER TILT/'TRNSP'/
C       ---------------------------- ACTIVE CODE ------------

        READ (5,IN)
        IF (PRT LNS) CALL LNS OUT
        IF(LONG Y)TILT='NORMAL'
        CALL DEVSET(IDEV)
        CALL REOPEN


C        *****       MAIN PLOTTING LOOP
1       CONTINUE
C       FIRST READ IN THE LINES TO BE PLOTTED
        LNCUR=1
5       READ (-,-,END=10)NUMBIN,NUMLN,INDVAR
C
C    SPECIAL CALCULATION FOR CONVERSION OF ION SPECTRUM PLOT
C
        IORG = 1
        IF(NUMLN .NE. 15) GO TO 26
        IORG = 15
        NUMLN = 2
26       CONTINUE
        IF(XCROSS(NUMLN))
     .    CALL XFILL(XPTS(1,LNCUR),NUMBIN,XVARB(NUMLN),1,NP,BTYPE)
```

```
                =====  PLOT/IODR  =====

              IF(.NOT.XCROSS(NUMLN))
             .   CALL FILL(XPTS(1,LNCUR),NUMBIN,XVARB(NUMLN),1,NP,BTYPE)
              IF(YCROSS(NUMLN))
             .  CALL XFILL(YPTS(1,LNCUR),NUMBIN,YVARB(NUMLN),INDVAR,NP,BTYPE)
              IF(.NOT.YCROSS(NUMLN))
             .    CALL FILL(YPTS(1,LNCUR),NUMBIN,YVARB(NUMLN),INDVAR,NP,BTYPE)
      C
              NPTS(LNCUR)=NP

      C    SPECIAL PROCESS FOR FILLING THE ZEROS IN THE Y ARRAY
      C
             IF(NUMLN.NE.  8 .AND. NUMLN .NE. 14 ) GO TO 28
              IPEND = NP-1
              DO 27 I = 2,IPEND
              IF(YPTS(I,LNCUR) .LE. 1.E-30)
            1    YPTS(I,LNCUR) = (YPTS(I+1,LNCUR) + YPTS(I-1,LNCUR))/2.
      C      2   /(XPTS(I+1,LNCUR) - XPTS(I-1,LNCUR))
      C      3    * (XPTS(I,LNCUR) - XPTS(I-1,LNCUR))
      27      CONTINUE
              XPTS(1,LNCUR) = 1.E-8
      28      CONTINUE
      C
      C    FINISH THE ZERO FILL
      C
      C NOW CONVERT THE ION SPECTRA FROM PARTICLES TO JOULES
      C
              IF(IORG .NE. 15) GO TO 30
              DO 29 I = 1,NP
              YPTS(I,LNCUR) = YPTS(I,LNCUR) * 1.6E-16*XPTS(I,LNCUR)
      29      CONTINUE
              IORG = 1
      30      CONTINUE
              LNCUR=LNCUR+1
              GO TO 5
      10      CONTINUE


      C       ** NOW PRINT OUT ALL THE LINES
              LNCUR=LNCUR-1


              IF ( .NOT. PRINT) GO TO 200
              DO 100 I=1,LNCUR
              NP=NPTS(I)
              PRINT 123,(  (  XPTS(K,I)    , YPTS(K,I)  ),K=1,NP)
      123     FORMAT(' ',E15.7,E15.7)
      100     CONTINUE
      200     CONTINUE

      C       ****************** DO THE PLOTS *****************
              IF (.NOT. PLOT) GO TO 500

      C        ** FIRST SCALE **
              READ (-,-,END=99)ILOGX,ILOGY
              LOGX='LOG'
              LOGY='LOG'
              IF(ILOGX .EQ. 1 )LOGX='LINEAR'
```

```
115.              IF(ILOGY .EQ. 1 ) LOGY='LINEAR'
116.
117.              IF (LOGY .EQ. 'LINEAR') GO TO 225
118.        C      ** CHECK FOR ZERO VALUES
119.              DO  205 I=1,LNCUR
120.        C     FIRST FIND THE CURRENT MIN
121.              NP=NPTS(I)
122.              PMIN=1.E35
123.
124.              DO 220 J=1,NP
125.              IF(YPTS(J,I) .GT. 0.0) PMIN=AMIN1(PMIN,YPTS(J,I))
126.        220   CONTINUE
127.
128.        C     NEXT RECTIFY NEGS AND ZEROS
129.              DO 210 J=1,NP
130.              IF(YPTS(J,I) .LE. 0.) YPTS(J,I)=PMIN
131.        210   CONTINUE
132.
133.        205    CONTINUE
134.        225   CONTINUE
135.              CALL SCALE(XPTS,2,XSCALE,LOGX,5,'NONE')
136.              CALL SCALE(YPTS,2,YSCALE,LOGY,10,'NONE')
137.              DO 400 I=1,LNCUR
138.              CALL MSCALE(XPTS(1,I),NPTS(I),XSCALE)
139.              CALL MSCALE(YPTS(1,I),NPTS(I),YSCALE)
140.        400   CONTINUE
141.        C      ** DO THE SCALING **
142.        C      **  AND THE LABELS  ***
143.              READ  12,XLAB,YLAB,TITLE
144.        12    FORMAT(13A6)
145.              CALL PLOTIT
146.
147.              PRINT    121
148.        121   FORMAT('    PLT  OR  X   ?')
149.              READ   122,INS
150.        122   FORMAT(A6)
151.
152.              IF (INS .EQ. 'PLT' .AND. IDEV .NE. 1)    CALL PLTTR
153.              IF (INS .NE. 'PLT') PRINT 323
154.        323   FORMAT('  PAGE WILL NOT BE PLOTTED')
155.
156.        500   CONTINUE
157.              GO TO 1
158.
159.
160.        99    STOP DONE
161.
162.        C     ------------------------------------------
163.                   SUBROUTINE PLOTIT
164.        C      ** SCALE AND AXES
165.              CALL GRAPH(
166.             .   XPTS,XSCALE,YPTS,YSCALE,1,
167.             .    'NONE','BLANK',XLAB,YLAB,
168.             .   'NORMAL',TITLE,
169.             .   'FULL',TILT)
170.
171.
```

```
      C       ** DO THE LINES ***
              DO 300 I=1,LNCUR
              CALL GRAPHM(XPTS(1,I),'SCAL1',YPTS(1,I),'SCAL1',NPTS(I),'NONE',
             .  'SOLID')
      300     CONTINUE
              RETURN
      C       -------------------------------


      C       ----------------------------------------
                   SUBROUTINE PLTTR
              PRINT 1
      1       FORMAT('   PAGE WILL BE PLOTTED')
              CALL DEVSET(1)
              CALL PLOTIT
              CALL DEVSET (IDEV)
              RETURN
      C       ---------------------------------------


      C       ----------------------------------------
              SUBROUTINE LNS OUT
      C        ** PRINT OUT THE LIST OF LINES THAT MAY BE PLOTTED
              PRINT 1
      1       FORMAT('1',///'  THESE LINES MAY BE PLOTTED')
              PRINT 2
      2       FORMAT(' ','---------------------------------'/
             .       '        YVAR           XVAR')
              PRINT 3,(I,YVARB(I),XVARB(I),I=1,NL)
      3       FORMAT(' ',I4,2X,A6,'  VS.   ',A6)
              STOP DONE
      C       ------------------------------------------------------
              END
```

```
      =====  PLOTCROSS/IODR  =====

      C       *********  THIS ROUTINE DOES THE IODR FILE PLOTTING *****
      C        ** FIRST SET UP THE PLOT TABLE ETC.
            PARAMETER NL=14
            INTEGER YVARB(NL)/'FT','FF','DEPX','DEPX','TEMP','TEMP','TEMP',
           .    'TEMP','F','U','Y1','ADBTMP','TEMP','TEMP'/

            INTEGER XVARB(NL)/'T','E','T','X','T','X','TA','Y',
           .       'E','E','X','X','TM','Y'/

            LOGICAL YCROSS(NL)/.FALSE.,.FALSE.,.TRUE.,.FALSE.,.TRUE.,.FALSE.,
           . .TRUE.,.FALSE., .FALSE.,.FALSE.,.FALSE.,.FALSE.,.TRUE.,.FALSE./

            LOGICAL XCROSS(NL)/NL*.FALSE./

      C        *** NOW MAKE SPACE TO STORE THE VARIOUS LINES ***
            PARAMETER MXLNS=20
            REAL XPTS(250,MXLNS),YPTS(250,MXLNS)
            INTEGER NPTS(MXLNS)

      C       *********       INPUT PARAMETERS     ******
            INTEGER NP/50/      IDEV/5/
            LOGICAL PRINT/.FALSE./      PLOT/.TRUE./
            LOGICAL LONG Y /.FALSE./
            LOGICAL PRT LNS/.FALSE./
            NAMELIST  /IN/ NP,PRINT,PLOT,IDEV,LONG Y,PRT LNS

      C       ********************   PLOT VARIABLES   *******
            REAL       XSCALE(10),YSCALE(10)
            INTEGER XL(14)/13*' ','$$'/YL(14)/13*' ','$$'/TI(14)/13*' ','$$'/
            INTEGER  XLAB(13),YLAB(13),TITLE(13)
            EQUIVALENCE (XLAB,XL) , (YLAB,YL)  , (TITLE,TI)
            INTEGER TILT/'TRNSP'/
      C       ---------------------------- ACTIVE CODE ------------

            READ (5,IN)
            IF (PRT LNS) CALL LNS OUT
            IF(LONG Y)TILT='NORMAL'
            CALL DEVSET(IDEV)


      C        *****       MAIN PLOTTING LOOP
      1       CONTINUE
      C       FIRST READ IN THE LINES TO BE PLOTTED
            LNCUR=1
      5       READ (-,-,END=10)NUMBIN,NUMLN,INDVAR
      C
      C     READ IN FILE FOR LNEXT PLOT
      C
            CALL REOPEN
      C
      C    SPECIAL CALCULATION FOR CONVERSION OF ION SPECTRUM PLOT
      C
            IORG = 1
            IF(NUMLN .NE. 15) GO TO 26
            IORG = 15
            NUMLN = 2
```

```
 26       CONTINUE
          IF(XCROSS(NUMLN))
         .   CALL XFILL(XPTS(1,LNCUR),NUMBIN,XVARB(NUMLN),1,NP,BTYPE)
          IF(.NOT.XCROSS(NUMLN))
         .   CALL FILL(XPTS(1,LNCUR),NUMBIN,XVARB(NUMLN),1,NP,BTYPE)
          IF(YCROSS(NUMLN))
         .  CALL XFILL(YPTS(1,LNCUR),NUMBIN,YVARB(NUMLN),INDVAR,NP,BTYPE)
          IF(.NOT.YCROSS(NUMLN))
         .   CALL FILL(YPTS(1,LNCUR),NUMBIN,YVARB(NUMLN),INDVAR,NP,BTYPE)
C
          NPTS(LNCUR)=NP

C    SPECIAL PROCESS FOR FILLING THE ZEROS IN THE Y ARRAY
C
         IF(NUMLN.NE.  8 .AND. NUMLN .NE. 14 ) GO TO 28
          IPEND = NP-1
          DO 27 I = 2,IPEND
          IF(YPTS(I,LNCUR) .LE. 1.E-30)
        1    YPTS(I,LNCUR) = (YPTS(I+1,LNCUR) + YPTS(I-1,LNCUR))/2.
C      2    /(XPTS(I+1,LNCUR) - XPTS(I-1,LNCUR))
C      3     * (XPTS(I,LNCUR) - XPTS(I-1,LNCUR))
 27       CONTINUE
          XPTS(1,LNCUR) = 1.E-8
 28       CONTINUE
C
C    FINISH THE ZERO FILL
C
C NOW CONVERT THE ION SPECTRA FROM PARTICLES TO JOULES
C
          IF(IORG .NE. 15) GO TO 30
          DO 29 I = 1,NP
          YPTS(I,LNCUR) = YPTS(I,LNCUR) * 1.6E-16*XPTS(I,LNCUR)
 29       CONTINUE
          IORG = 1
 30       CONTINUE
          LNCUR=LNCUR+1
          GO TO 5
 10       CONTINUE


C         ** NOW PRINT OUT ALL THE LINES
          LNCUR=LNCUR-1


          IF ( .NOT. PRINT) GO TO 200
          DO 100 I=1,LNCUR
          NP=NPTS(I)
          PRINT 123,(  (  XPTS(K,I)    , YPTS(K,I)  ),K=1,NP)
 123      FORMAT(' ',E15.7,E15.7)
 100      CONTINUE
 200      CONTINUE

C         ****************** DO THE PLOTS *****************
          IF (.NOT. PLOT) GO TO 500

C          ** FIRST SCALE **
          READ (-,-,END=99)ILOGX,ILOGY
```

```
115.            LOGX='LOG'
116.            LOGY='LOG'
117.            IF(ILOGX .EQ. 1 )LOGX='LINEAR'
118.            IF(ILOGY .EQ. 1 ) LOGY='LINEAR'
119.
120.            IF (LOGY .EQ. 'LINEAR') GO TO 225
121.      C      ** CHECK FOR ZERO VALUES
122.            DO  205 I=1,LNCUR
123.      C     FIRST FIND THE CURRENT MIN
124.            NP=NPTS(I)
125.            PMIN=1.E35
126.
127.            DO 220 J=1,NP
128.            IF(YPTS(J,I) .GT. 0.0) PMIN=AMIN1(PMIN,YPTS(J,I))
129.      220   CONTINUE
130.
131.      C     NEXT RECTIFY NEGS AND ZEROS
132.            DO 210 J=1,NP
133.            IF(YPTS(J,I) .LE. 0.) YPTS(J,I)=PMIN
134.      210   CONTINUE
135.
136.      205    CONTINUE
137.      225   CONTINUE
138.            CALL SCALE(XPTS,2,XSCALE,LOGX,5,'NONE')
139.            CALL SCALE(YPTS,2,YSCALE,LOGY,10,'NONE')
140.            DO 400 I=1,LNCUR
141.            CALL MSCALE(XPTS(1,I),NPTS(I),XSCALE)
142.            CALL MSCALE(YPTS(1,I),NPTS(I),YSCALE)
143.      400   CONTINUE
144.      C      ** DO THE SCALING **
145.      C      **  AND THE LABELS  ***
146.            READ  12,XLAB,YLAB,TITLE
147.      12    FORMAT(13A6)
148.            CALL PLOTIT
149.
150.            PRINT    121
151.      121   FORMAT('    PLT  OR  X   ?')
152.            READ   122,INS
153.      122   FORMAT(A6)
154.
155.            IF (INS .EQ. 'PLT' .AND. IDEV .NE. 1)    CALL PLTTR
156.            IF (INS .NE. 'PLT') PRINT 323
157.      323   FORMAT('  PAGE WILL NOT BE PLOTTED')
158.
159.      500   CONTINUE
160.            GO TO 1
161.
162.
163.      99    STOP DONE
164.
165.      C     ------------------------------------------
166.                 SUBROUTINE PLOTIT
167.      C      ** SCALE AND AXES
168.            CALL GRAPH(
169.           .   XPTS,XSCALE,YPTS,YSCALE,1,
170.           .    'NONE','BLANK',XLAB,YLAB,
171.           .   'NORMAL',TITLE,
```

```
     .    'FULL',TILT)
C      ** DO THE LINES ***
       DO 300 I=1,LNCUR
       CALL GRAPHM(XPTS(1,I),'SCAL1',YPTS(1,I),'SCAL1',NPTS(I),'NONE',
     .   'SOLID')
300    CONTINUE
       RETURN
C      -------------------------------

C      ----------------------------------------
            SUBROUTINE PLTTR
       PRINT 1
1      FORMAT('   PAGE WILL BE PLOTTED')
       CALL DEVSET(1)
       CALL PLOTIT
       CALL DEVSET (IDEV)
       RETURN
C      ----------------------------------------

C      ----------------------------------------
       SUBROUTINE LNS OUT
C       ** PRINT OUT THE LIST OF LINES THAT MAY BE PLOTTED
       PRINT 1
1      FORMAT('1',///'  THESE LINES MAY BE PLOTTED')
       PRINT 2
2      FORMAT(' ','---------------------------------'/
     .        '       YVAR          XVAR')
       PRINT 3,(I,YVARB(I),XVARB(I),I=1,NL)
3      FORMAT(' ',I4,2X,A6,'  VS.   ',A6)
       STOP DONE
C      ------------------------------------------------------
       END
```

```
=====  PLOTNUM/IODR  =====

C       *********  THIS ROUTINE DOES THE IODR FILE PLOTTING *****

C        ** FIRST SET UP THE PLOT TABLE ETC.
        PARAMETER NL=14
        INTEGER YVARB(NL)/'FT','FF','DEPX','DEPX','TEMP','TEMP','TEMP',
     .     'TEMP','F','U','Y1','ADBTMP','TEMP','TEMP'/

        INTEGER XVARB(NL)/'T','E','T','X','T','X','TA','Y',
     .        'E','E','X','X','TM','Y'/

        LOGICAL YCROSS(NL)/.FALSE.,.FALSE.,.TRUE.,.FALSE.,.TRUE.,.FALSE.,
     . .TRUE.,.FALSE., .FALSE.,.FALSE.,.FALSE.,.FALSE.,.TRUE.,.FALSE./

        LOGICAL XCROSS(NL)/NL*.FALSE./

C        *** NOW MAKE SPACE TO STORE THE VARIOUS LINES ***
        PARAMETER MXLNS=10
        REAL XPTS(250,MXLNS),YPTS(250,MXLNS)

        INTEGER NPTS(MXLNS)

C       *********      INPUT PARAMETERS     ******
        INTEGER NP/50/     IDEV/5/
        LOGICAL PRINT/.FALSE./     PLOT/.TRUE./
        NAMELIST  /IN/ NP,PRINT,PLOT,IDEV

C       ********************   PLOT VARIABLES   *******
        REAL       XSCALE(10),YSCALE(10)
        INTEGER XL(7)/6*' ','$$'/ YL(7)/6*' ','$$'/  TI(7)/6*' ','$$'/
        INTEGER  XLAB(6),YLAB(6),TITLE(6)
        EQUIVALENCE (XLAB,XL) , (YLAB,YL)   , (TITLE,TI)
C       ---------------------------- ACTIVE CODE -------------

        CALL REOPEN
        READ (5,IN)
        CALL DEVSET(IDEV)
C       FIRST READ IN THE LINES TO BE PLOTTED
        LNCUR=1
5       READ (-,-,END=10)NUMBIN,NUMLN,INDVAR
C
C    SPECIAL CALCULATION FOR CONVERSION OF ION SPECTRUM PLOT
C
        IORG = 1
        IF(NUMLN .NE. 15) GO TO 26
        IORG = 15
        NUMLN = 2
26       CONTINUE
        IF(XCROSS(NUMLN))
     .    CALL XFILL(XPTS(1,LNCUR),NUMBIN,XVARB(NUMLN),1,NP,BTYPE)
        IF(.NOT.XCROSS(NUMLN))
     .    CALL FILL(XPTS(1,LNCUR),NUMBIN,XVARB(NUMLN),1,NP,BTYPE)
        IF(YCROSS(NUMLN))
     .   CALL XFILL(YPTS(1,LNCUR),NUMBIN,YVARB(NUMLN),INDVAR,NP,BTYPE)
        IF(.NOT.YCROSS(NUMLN))
     .     CALL FILL(YPTS(1,LNCUR),NUMBIN,YVARB(NUMLN),INDVAR,NP,BTYPE)
C
C    SPECIAL PROCESS FOR FILLING THE ZEROS IN THE Y ARRAY
```

```
            =====  PLOTNUM/IODR  =====

      C
            IF(NUMLN.NE.  8 .AND. NUMLN .NE. 14 ) GO TO 28
             IPEND = NP-1
             XPTS(1,LNCUR) = 1.E-8
             DO 27 I = 2,IPEND
             IF(YPTS(I,LNCUR) .LE. 1.E-4)
           1    YPTS(I,LNCUR) = (YPTS(I+1,LNCUR) - YPTS(I-1,LNCUR))
           2   /(XPTS(I+1,LNCUR) - XPTS(I-1,LNCUR))
           3    * (XPTS(I,LNCUR) - XPTS(I-1,LNCUR))
      27     CONTINUE
      28     CONTINUE
      C
      C   FINISH THE ZERO FILL
      C
      C NOW CONVERT THE ION SPECTRA FROM PARTICLES TO JOULES
      C
             IF(IORG .NE. 15) GO TO 30
             DO 29 I = 1,NP
             YPTS(I,LNCUR) = YPTS(I,LNCUR) * 1.6E-16*XPTS(I,LNCUR)
      29     CONTINUE
             IORG = 1
      30     CONTINUE
             LNCUR=LNCUR+1
             GO TO 5
      10     CONTINUE


      C      ** NOW PRINT OUT ALL THE LINES
             LNCUR=LNCUR-1


             IF ( .NOT. PRINT) GO TO 200
             DO 100 I=1,LNCUR
             PRINT 123,(  (  XPTS(K,I)   , YPTS(K,I)  ),K=1,NP)
      123    FORMAT(' ',E15.7,E15.7)
      100    CONTINUE
      200    CONTINUE

      C      ****************** DO THE PLOTS *****************
             IF (.NOT. PLOT) GO TO 500

      C       ** FIRST SCALE **
             READ ,ILOGX,ILOGY
             LOGX='LOG'
             LOGY='LOG'
             IF(ILOGX .EQ. 1 )LOGX='LINEAR'
             IF(ILOGY .EQ. 1 ) LOGY='LINEAR'

             CALL SCALE(XPTS,2,XSCALE,LOGX,5,'NONE')
             CALL SCALE(YPTS,2,YSCALE,LOGY,10,'NONE')
             DO 400 I=1,LNCUR
             CALL MSCALE(XPTS(1,I),NP,XSCALE)
             CALL MSCALE(YPTS(1,I),NP,YSCALE)
      400    CONTINUE

      C       ** DO THE SCALING **
      C       **  AND THE LABELS  ***
```

```
      =====  PLOTNUM/IODR  =====

115.            READ  12,XLAB,YLAB,TITLE
116.     12     FORMAT(6A6)
117.            CALL GRAPH(
118.           .  XPTS,XSCALE,YPTS,YSCALE,1,
119.           .   'NONE','BLANK',XLAB,YLAB,
120.           .  'NORMAL',TITLE,
121.           .  'FULL','NORMAL')
122.
123.
124.     C       ** DO THE LINES ***
125.            DO 300 I=1,LNCUR
126.            CALL GRAPHM(XPTS(1,I),'SCAL1',YPTS(1,I),'SCAL1',NP,'NONE',
127.           .  'SOLID')
128.     300    CONTINUE
129.     500    CONTINUE
130.            STOP END
131.            END
```

```
      =====  PLT-ID  =====

 1.     PLOT#  Y-AXES X-AXES     INDEP-VAR #PTS-VAR
 2.     IONS
 3.      1. FT         T          1         NE
 4.      2. FF         E          1         NE
 5.      3. DEPX       T          X         NE
 6.      4. DEPX       X          T         NX
 7.      5. INTDEPX    T          X         NE
 8.      6. INTDEPX    X          T         NX
 9.      7. TEMP       TA         Y         NT
10.      8. TEMP       Y          TA        NY
11.     PHOTONS
12.      9. F          E          LAYER#    JK
13.     10. U          E          LAYER#    JK
14.     11. Y1         X          LAYER#    IWID
15.     12. ADBTMP     X          LAYER#    IWID
16.     13. TEMP       TM         Y         NUMT
17.     14. TEMP       Y          TM        NUMY
18.     PSEUDO
19.     15. FF         (SAME AS 2., EXCEPT J/CM**2/KEV)
```

IV. INPUT INSTRUCTIONS

The section contains a description of the input data necessary to operate each routine. This information is presented in conjunction with the read statements in the previous section and the example in the next section.

IV.A. Photon Code

IV.A.1. [*P-1*] Spectral Deposition and Temperature - This routine is executed by the statement:

@ XQT T*DAMEN. M/E

This command activates a master program which will call various subroutines upon encountering a recognized command (see list in read statements). If X-rays are to be considered the command

@ ADD T*DAMEN. ELT/ATOM

is inserted, this calls INITIA and reads in all the X-ray cross section library.

IV.A.1.a Spectra

The spectrum is generated by the statement

P-SPECTRUM

of which follow the data (in free format unless otherwise specified):

	Variable Name	Description	Units	Real/Integer
	1 KEV	Blackbody Temperature	keV	real
(A)	2 JK	# of Points in Spectrum	---	integer
	3 FLUX	Total X-ray Energy	J	real
	4 RADIUS	Position of Exposed Material	Meters	real

if the variable KEV is less than 0 the following data for a histogram spectra are read in:

	1 NMHIST	# of Points In HISTOGRAM	---	integer
(B)	2 EHIST(I)	ENERGY Points	keV	real
	3 AMP(I)	Spectrum amplitude	J/keV	

repeat 2 and 3 up through NMHIST

IV.A.1.b. Material

To set-up the material description the command of

P-LAYER

is used. This is followed by parameters for each material in the first layer as:

(A)	1 NUM	layer #	---	integer
	2 Width(NUM)	layer thickness	CM	real
	3 IWID(NUM)	# of logarithmic intervals in layer (for plots 11,12)	---	integer
	1 IZ	atomic number	---	integer
	2 RO	mass density	g/cm^3	real
	3 C(NUM)	specific heat	cal/gm-L	real
(B)	4 Alpha	thermal diffusivity	cm^2/sec	real
	5 IG	0=not gas, 1=gas	---	integer
	6 AM	atomic mass (not used if IG=0)	AMU	real
	7 P	gas pressure (not used if IG=0)	Torr	real
	8 GTMP	gas temperature (not used if IG=0)	K	real

This card may be repeated if the layer contains more than one constituent. Each layer must be terminated by:

@ EOF

After all layers (Max=4) have been described input to the layers is terminated by another:

@ EOF

IV.A.1.c. Deposition

A deposition profile for the spectrum and material chosen can be obtained by the statement:

P-DEPOSITION

followed by the following data:

(A)	1 LAYLO	first layer considered	---	integer
	2 LAYHI	last layer considered	---	integer

A printed output is obtained of the spectrum, absorption parameters, and energy deposition by the command.

P-OUTPUT

If instead of X-rays a monoenergetic absorption is required the code has a simple model of laser absorption in a black body which is initiated by

P-LASER

followed by the data

	1 WAVLTH	laser wavelength	microns	real
(A)	2 FLUX	total energy	J	real
	3 RADIUS	surface radius	meters	real

IV.A.1.d. Temperature Response

The temperature response for one of the layers described above is obtained from either of two subroutines. One routine determines the temperature at user specified times and locations while the other selects times from a predetermined time and space grid which is initiated by the command:

STARAY .

Temperature response at user specified times are obtained by the command and following data:

P-TIME

(A)	1 NLAYR	layer number for analysis	---	integer
	2 IMOD	temperature model used 1 = impulse 2 = finite duration 3 = finite slab, impulse	---	integer
	3 SIZE	width of slab for model 3	cm	real
(B)	1 NUMY	# of locations for analysis	---	integer
	2 Y(I)	locations	cm	real
	3 NUMT	# of times for analysis	---	integer
	4 TM	times (from photon arrival)	sec	real
	5 TD	pulse duration (model 2 only)*	sec	real
	6 W	pulse spacing } (model 3 only)*	sec	real
	7 NPLUS	# of pulses } (model 3 only)*	---	integer

*Some value still required in data field.

for temperature analysis at the standard times and locations the following command and data are needed:

PSTEMP

	1	NLAYR	layer number for analysis	---	integer
	2	IMOD	temperature model (see P-TIME)	---	integer
(A)	3	TD	pulse duration (model 2 only)*	sec	real
	4	W	pulse spacing } (model 3 only)*	sec	real
	5	NPLUS	# of pulses } (model 3 only)*	---	integer
	6	SIZE	width of slab } (model 3 only)*	cm	real
	1	IXS	beginning standard location	---	integer
	2	IXF	final standard location	---	integer
(B)	3	IXD	increment of index for standard location	---	integer

IV.A.1.e. Filing

Data transfer from the code to the data files are accomplished by the following commands:

OPEN - sets up an IODR file heretofore unused

REOPEN - reopens a previously used file

LOGIN - places information in the log for the file directory

P-FILE - transfers photon data from program to file

IV.B. Ion Code

IV.B.1. [*I-1*] Spectral Deposition and Temperature Response

This routine is activated by the statement

@ XQT T*DAMEN. IONCODE

IV.B.1.a. Spectra

The command for setting up the ion spectrum and the associated data are:

I-SPECTRUM

	1	ISPEC	type of spectrum (1=Max, 2=Gaus)	---	integer
	2	EMN	characteristic energy	keV	real
	3	SIG	Gaussian std deviation	keV	real
(A)	4	EMIN	minimum energy (0=default)	keV	real
	5	EMAX	maximum energy (0=default)	keV	real
	6	NE	number of energy intervals	---	integer
	7	FL	total number of particles	---	real
	8	R	exposed surface radius	meters	real
	9	A	ion mass	amu	real
	10	W	pulse spacing	sec	real
	11	NPLUS	number of pulses	---	integer
	12	IPRI	0=print output, 1=don't	---	integer

If the parameter ISPEC is 3, a histogram spectrum is required and the data are:

	1	NMHIST	# of points in Histogram	---	integer
(B)	2	EHIST(I)	energy points	keV	real
	3	AMP(I)	spectrum points	J/keV	real

(repeat 2 and 3 through NMHIST)

IV.B.1.b. Material

The material properties are read in as:

	1 IMODEL	deposition response model (1 = region 1 light ions) (2 = heavy ions) (5 = general ions)		
	2 ALPHA	thermal diffusivity	cm^2/sec	real
	3 RHO	density	gm/cm^2	real
	4 CP	specific heat	cal/gm-K	real
	5 X2	layer thickness (not used)	cm	real
	6 DX	# of points for deposition if standard array not chosen	---	integer
(B)	7 C	stopping power for model 1 0'th order range coefficient (model 2)	keV/μ cm	real
	8 EO	reference energy for C (model 1) 1st order range coefficient (model 2)	keV cm/keV	real
	9 A2	2nd order range coefficient (model 2)	cm/keV^2	real
	10 A3	3rd order range coefficient (model 2)	cm/keV^3	real

if IMODEL=5, then the following data must be given:

	1 E1	reference energy (same as E2)	keV	real
	2 E2	reference energy (same as E3)	keV	real
(C)	3 S1	reference stopping power (same as S2)	keV	real

4 S2	reference stopping power (same as S3)	keV	real
5 EM	reference energy (estimate of energy loss intersection)	keV	real

IV.B.1.c. Gas Spectra Modification

If the spectra are to be modified by a gaseous layer the following commands are required:

LITMOD - for light ion spectra

SPEMOD - for heavy ions

The LITMOD subroutine has the following data:

(A)	1 INST1	if = FILE, results are filed	---	integer
(B)	1 INST2	if = RANGE, range table is created	---	integer
	1 E0	reference energy	keV	real
	2 S0	stopping power at E0	keV/μ	real
	3 E1	reference energy	keV	real
	4 S1	stopping power at E1	keV	real
(C)	5 E2	reference energy	keV	real
	6 S2	stopping power at E2	keV/μ	real
	7 E3	reference energy	keV	real
	8 S3	stopping power at E3	keV/μ	real
	9 SMAX	maximum stopping power	keV/μ	real
(D)	1 REFP	reference pressure ions in gas	Torr	real
	2 REFT	reference temperatrue for ions in gas	K	real
	1 RBUF	gas layer thickness	cm	real
(E)	2 GASP	gas layer pressure	Torr	real
	3 GAST	gas layer temperature	K	real
	4 IRPI	= 1 if print is suppressed	---	integer

For heavy ions, the input data for the SPEMOD program are:

(A)	1 INST1	if = FILE, results are filed	---	integer
(B)	1 INST2	if = RANGE, range table is created	---	integer
	1 E1	reference energy for ion	keV	real
	2 R1	range of E1 in gas	cm	real
(C)	3 SIG1	ΔR of E1 in gas	cm	real
	4 E2	reference energy for ion	keV	real
	5 R2	range of E2 in gas	cm	real
	6 SIG2	ΔR of E2 in gas	cm	real
	7 REFP	reference pressure for above data	Torr	real
	8 REFT	reference temperature for above data	K	real
	1 RBUF	thickness of gas layer	cm	real
(D)	2 GASP	pressure of gas layer	Torr	real
	3 GAST	temperature of gas layer	K	real
	4 IPRI	= 1 to suppress print	---	integer

IV.B.1.d. Energy Deposition in Gas

If the volumetric deposition of energy from the ion spectrum into the gas is desired. For light ions the command and data are:

DEPLIT

(A) (Same as card C following LITMOD)

(B) (Same as card D following LITMOD)

(C) (Same as card E following LITMOD)

(D)	1 LAYOUT	if = 0, logarithmic grid	---	integer
		if = 1, linear grid	---	
	2 RINNER	innermost radial position	cm	real
	3 ROUTER	outermost radial position	cm	real
	4 NSPOTS	number of positions to evaluate deposition	---	integer
(E)	INST	if = I-1FILE file deposition rate	---	
		if = I-2FILE file deposition	---	
		(Each must be followed by a one line Bin heading. An ·@EOF must to terminate filing) (repeat (E) sequence NSPOTS timer)		

for heavy ions the commands and data are

DEPHVY

(A) (Same as card C following SPEMOD)

(B)	GASP	pressure of gas layer	Torr	real
	GAST	temperature of gas layer	K	real

(C) (Same as card D following DEPLIT)

(D) (Same as card E following DEPLIT)

IV.B.1.e. Spectral Energy Deposition

The deposition subroutine for the ion response calculations may be performed on different spatial formats. If a standard format is used a command to generate the spatial grid must be made, i.e.: (followed by one data card)

STARA3

	1 BIGX	maximum X of deposition region	cm	real

A command to perform the deposition and associated data then follow as for light ions:

I-DEPOSITION (must use model 5)

(A)	1 IXMATX	if = 50 standard X array is used	---	integer
	1 EO	reference energy for ions in target	keV	real
	2 SO	reference stopping power at EO	keV/μ	real
	3 E1	reference energy	keV	real
(B)	4 S1	reference stopping power at E1	keV/μ	real
	5 E2	reference energy	keV	real
	6 S2	reference stopping power at E2	keV/μ	real
	7 E3	reference energy	keV	real
	8 S3	reference stopping power at E3	keV/μ	real
	9 SMAX	maximum stopping power	keV/μ	real

for heavy ions the sequence is

I-DEPOSITION

(A)	1 IXMATX	if = 50 standard X array is used	---	integer

(B)	1 LOCDFN	block number of nuclear deposition cross sections in file 11	---	integer
	2 LOCDFE	block number of electronic deposition cross sections in file 11	---	integer

IV.B.1.f. Temperature

As in the photons, the ion temperature routine has two basic forms: a) locations and times for temperature analysis are user selected, and b) locations and times are selected from a standard array. In addition, the temperature calculations can be performed independently from the deposition calculations. If the temperature is calculated without calculating the depositions, it is necessary to generate the polynomial coefficients for model 5 by the commands:

HCOEF - for heavy ions

LCOEF - for light ions

(followed by same data as 2nd card after I-Deposition)

for temperature calculations selected by the user the appropriate command and data are:

I-TEMP

1 NY	# of analysis locations	---	integer
1 Y(I)	locations (repeat to NY)	cm	real
1 NT	# of analysis times	---	integer
1 TA(I)	times (repeat to NT)	sec	real

if the locations and times are selected from the standard array commands must first be made to set-up the array as:

STARAY - 15 locations from 10^{-7} to 10^{-2} and
115 times from 10^{-10} to 10^{-2}

STARA2 - 15 locations from 10^{-7} to 10^{-2} and
115 times from 10^{-7} to 10^{-4}

STARA3 - 50 values of X (where X is read in) and
115 times from 10^{-10} - 10^{-2}
(must be followed by BIGX, as in deposition)

the standard temperature routine can then be called with the command and data as follows:

I-STEMP

1 IXS	first standard location considered	---	integer
2 IXF	last standard location considered	---	integer
3 IXD	increment in standard location	---	integer

(if IXS is given as 777, a IMODEL becomes IXF and the above 3 must be repeated)

IV.B.1.f.(1). Reference Locations and Timer

The locations and timer which are generated by the STARAY and STARA2 are listed in tables IV.1 and IV.2.

Table IV.1

STANDARD TIMES AND LOCATION FOR T*DAMEN CODE (STARAY)

ION TIMES

.100-09	.147-09	.215-09	.316-09	.464-09	.681-09	.100-08
.147-08	.215-08	.316-08	.464-08	.681-08	.100-07	.147-07
.215-07	.316-07	.464-07	.681-07	.100-06	.108-06	.117-06
.126-06	.136-06	.147-06	.158-06	.171-06	.185-06	.200-06
.215-06	.233-06	.251-06	.271-06	.293-06	.316-06	.341-06
.369-06	.398-06	.430-06	.464-06	.501-06	.541-06	.584-06
.631-06	.681-06	.736-06	.794-06	.858-06	.926-06	.100-05
.108-05	.117-05	.126-05	.136-05	.147-05	.158-05	.171-05
.185-05	.200-05	.215-05	.233-05	.251-05	.271-05	.293-05
.316-05	.341-05	.369-05	.398-05	.430-05	.464-05	.501-05
.541-05	.584-05	.631-05	.681-05	.736-05	.794-05	.858-05
.926-05	.100-04	.108-04	.117-04	.126-04	.136-04	.147-04
.158-04	.171-04	.185-04	.200-04	.215-04	.233-04	.251-04
.271-04	.293-04	.316-04	.341-04	.369-04	.398-04	.430-04
.464-04	.501-04	.541-04	.584-04	.631-04	.681-04	.736-04
.794-04	.858-04	.926-04	.100-03	.147-03	.215-03	.316-03
.464-03	.681-03	.100-02				

PHOTON TIMES

.100-09	.215-09	.464-09	.100-08	.215-08	.464-08	.100-07
.215-07	.464-07	.100-06	.215-06	.464-06	.100-05	.215-05
.464-05	.100-04	.215-04	.464-04	.100-03	.215-03	.464-03
.100-02						

LOCATION

.100-06	.100-05	.100-04	.200-04	.500-04	.700-04	.100-03
.150-03	.200-03	.300-03	.500-03	.700-03	.100-02	.300-02
.100-01						

Table IV.2

STANDARD TIMES AND LOCATION
FOR T*DAMEN CODE
(STARA2)

ION TIMES

.100-06	.106-06	.113-06	.120-06	.127-06	.135-06	.144-06
.153-06	.162-06	.173-06	.183-06	.195-06	.207-06	.220-06
.234-06	.248-06	.264-06	.280-06	.298-06	.316-06	.336-06
.357-06	.379-06	.403-06	.428-06	.455-06	.483-06	.513-06
.546-06	.580-06	.616-06	.654-06	.695-06	.739-06	.785-06
.834-06	.886-06	.941-06	.100-05	.106-05	.113-05	.120-05
.127-05	.135-05	.144-05	.153-05	.162-05	.173-05	.183-05
.195-05	.207-05	.220-05	.234-05	.248-05	.264-05	.280-05
.298-05	.316-05	.336-05	.357-05	.379-05	.403-05	.428-05
.455-05	.483-05	.513-05	.546-05	.580-05	.616-05	.654-05
.695-05	.739-05	.785-05	.834-05	.886-05	.941-05	.100-04
.106-04	.113-04	.120-04	.127-04	.135-04	.144-04	.153-04
.162-04	.173-04	.183-04	.195-04	.207-04	.220-04	.234-04
.248-04	.264-04	.280-04	.298-04	.316-04	.336-04	.357-04
.379-04	.403-04	.428-04	.455-04	.483-04	.513-04	.546-04
.580-04	.616-04	.654-04	.695-04	.739-04	.785-04	.834-04
.886-04	.941-04	.100-03				

PHOTON TIMES

.100-09	.215-09	.464-09	.100-08	.215-08	.464-08	.100-07
.215-07	.464-07	.100-06	.215-06	.464-06	.100-05	.215-05
.464-05	.100-04	.215-04	.464-04	.100-03	.215-03	.464-03
.100-02						

LOCATION

.100-06	.100-05	.100-04	.200-04	.500-04	.700-04	.100-03
.150-03	.200-03	.300-03	.500-03	.700-03	.100-02	.300-02
.100-01						

IV.B.1.g. Filing

The ion routine contains the basic instructions for transferring calculated data to the IODR mass storage system. The basic commands are:

I-OPEN	sets up an IODR file heretofore unused
I-REOPEN	reopens a previously used file
LOGIN	places information in the log for the file directory
I-1FILE	files the flux, spectra, and energy deposition rates
I-2FILE	files the time integrated energy deposition
I-3FILE	files the results of the temperature calculation

IV.B.1.h. Auxiliary Routines

In addition to the subroutine mentioned above, the ion code contains several other routines which can perform calculations associated with the general response of ions.

IV.B.1.h.(1). Heat Flux

If a thermal response calculation is described which is based on a finite duration heat flux on a semi-infinite slab, the following command and data are required:

HTFLUX

(A)

1 F	total fluence	J/cm^2	real
2 TD	pulse duration	sec	real
3 ALPHA	thermal diffusivity	cm^2/sec	real
4 TK	thermal conductivity	cal/cm/sec/K	real
5 M	number of pulses	---	integer
6 W	pulse spacing	sec	real

IV.B.1.h.(2). Residual Temperature

If the temperature profile after many ion pulses is required, the solution for a finite with slab can be obtained by the following command and data:

I-RESID

(A)	1 NY	number of locations to evaluate	---	integer
(B)	1 Y(I)	location (NY values)	cm	real
(C)	1 NT	number of times to evaluate	---	integer
(D)	1 TA(I)	evaluation times (NT values)	sec	real
(E)	1 NCONV	number of terms in series solution	---	integer

IV.B.1.h.(3). Implantation Distribution

An estimate of the implantation distributions for light ions can be made with the commands and data

LOWIMP

(followed by same data as 2nd card after I-Deposition)

IV.B.1.h.(4). Sputtering

The sputtering calculation for a heavy ion spectrum is contained in a separate absolute element. This routine reads data in the IODR data file to determine the total temperature history at the front surface. This information is then folded with the temperature and energy dependent sputtering model and the total sputtering rates are calculated. The commands and data are:

@ XQT T*DAMEN. SPUTTER

(A)	1 FILENAME	file name of IODR data file	---	integer
(B)	1 BINT	Bin number of surface temperature	---	integer

(C)	1 BINF	Bin number of surface ion flux	---	integer
	1 SPTEMC	constant in sputtering-temperature relation	---	real
(D)				
	2 SPTEME	activation energy of sputtering	eV	real
	3 TREF	ambient surface temperature	K	real
	1 CONST	constant in energy-sputtering relation	---	real
	2 FIFTY	constant in energy-sputtering relation	---	real
(E)	3 Z1	ion atomic number	---	real
	4 Z2	target atomic number	---	real
	5 AM1	ion atomic mark	amu	real
	6 AM2	target atomic mark	amu	real

for plotting (reference section IV.C.3).

Data is stored in file in the array DEP(I,J) and is plotted as a deposition versus time (NUMLIN =) as follows:

Independent Variable

1 Sputtering rate with temperature effect

2 Sputtering rate without temperature effect

3 Total sputtering with temperature effect

4 Total sputtering without temperature effect

5 Ratio of 3 and 4

IV.B.1.h.(5). Surface evaporation

To calculate the evaporation rate the data file containing the surface temperature must be accessed. The command string and data are as follows:

@XQT T*DAMEN. EVAP

	FILE NAME			
(A)	1 BINT	Bin # of surface temperature	---	integer
	1 STIPR	surface sticking probability	---	real
(B)	2 AMASS	atomic mass of surface	AMU	real
	3 TAMB	ambient surface temperature	K	real
	1 PZERO	coefficient of vapor pressure curve	Torr	real
(C)	2 HSUB	sublimation energy	eV	real
(D)	1 INSTF	if = I-1 FILE file data as ion-1 type	---	integer
(E)	1 FILE NAME	name of file for output data	---	integer
(F)	- ---	directory entry for bin title	---	---

for plotting, data is stored in file in the array DEPX(I,J) and is plotted as a deposition versus time (NUMLIN = 3) as follows:

1 evaporation rate vs. time

2 total evaporation vs. time

3 temperature vs. time

4 vapor pressure vs. time

IV.B.2. [*I-2*] Displacement

The displacement calculation is initiated by the command

@ XQT T*DAMEN. DISPLACEMENT

The input requirements for generating the spectra and any modification by gaseous layers for the displacement code are identical to that for I-1 (see sections IV.B.1.a. and IV.B.1.c.). Any displacement calculation must be preceded by a call to generate the standard time and location array (STARAY, STARA2, STARA3). The user can then elect to perform a calculation for light ions or heavy ions. The appropriate command and data are:

IV.B.2.a. Light Ions

DISPLACEMENT

(A)	1 M1	ion mass	amu	real
	2 Z1	ion atomic number	---	real
	3 M2	target mass	amu	real
	4 Z2	target atomic	---	real
	5 E0	displacement energy	keV	real

followed by a card for the ion-target stopping reference values identical to IV.B.1.d.

IV.B.2.b. Heavy Ions

The heavy ion code requires the use of the nuclear deposition functions stored in data file 11. The subroutine is initiated by the following command and data:

HIDPA

(A)	1 LOCOFN	block location of nuclear deposition functions in file 11	---	integer
	2 CONVER	convert keV/μ to barn-dpa [$.4 \times 10^{28}/E_d(\text{keV})/N(\text{a/cm}^3)$]	barn-dpa	real

At this point the code has calculated the displacement rates based on a time frame determined by the ion arrival times. If data are to be converted to a standard time base the following command is necessary:

BASALT

The filing instruction for the displacement calculation are identical with those for the deposition routine with the following two commands:

I-1FILE	files flux, spectra, times, and displacement rate data (dpa/s)
I-2FILE	files time integrated displacements (dpa)

IV.B.3. [*I-3*] Monoenergetic Deposition

This deposition routine provides stopping power energy deposition and local ion energies for light ions and energy deposition data for heavy ions based on data from file 11. The light ion subroutine has an internal plotting package for data display.

IV.B.3.a. Light Ion

The light ion code is initiated by the following command:

@ XQT T*DAMEN. LOWZDEP

if plots are desired the statement must be preceded by

@ GSP,P

PLOTTER PEN/LIQ

data following the XQT statement are:

(A)	1	E0	reference energy	keV	real
	2	S0	stopping power at E0	keV/μ	real
	3	E1	reference energy	keV	real
	4	S1	stopping power at E1	keV/μ	real
	5	E2	reference energy	keV	real
	6	S2	stopping power at E2	keV/μ	real
	7	E3	reference energy	keV	real
	8	S3	stopping power at E3	keV/μ	real
	9	SMAX	maximum stopping power	keV/μ	real
(B)	1	Z	first Brice coefficient	---	real
	2	AP	second Brice coefficient	---	real
	3	AN	third Brice coefficient	---	real
	4	MI	ion atomic mass	amu	real

	5 RHO	target density	gm/cm^3	real
	6 M2	target atomic mass	amu	
	7 Z1	ion atomic number	---	real
	8 Z2	target atomic number	---	real
	1 IND1	1=perform stopping power calculation	---	integer
(C)	2 IND2	1=perform spatial distribution	---	integer
	3 IND3	not used	---	integer
	4 IPLOT	1=make plots	---	integer
	5 IPRINT	1=print output (in all cases 0=no)	---	integer
	1 FIRSTE	smallest energy in stopping power calculation	keV	real
(D)	2 FINE	largest energy in stopping power calculation	keV	real
	1 NUM	number of energies for spatial profiles	---	integer
(E)	2 X3(I)	energies for spatial profiles (NUM values)	keV	real

IV.B.3.b. Heavy Ions

The heavy ion code must be preceded by assigning the deposition function data file and then executing as:

```
@ ASG,AX  11
@ XQT T*DAMEN. HIGHZDEP
```

these instructions are followed by the following data:

(A)	1 LOCDFN	block location of nuclear deposition functions in file 11	---	integer
	2 LOCDFE	block location of electron deposition functions in file 11	---	integer
	3 CONV	conversion factor if data are to be different than keV/μ	---	real
(B)	1 NUM	number of incident energy to get deposition for	---	integer
	2 E(I)	incident energies (NUM values)	keV	real
(C)	1 IPRINT	1=print output	---	integer
	2 IPLOT	1=plot output (0=no in all cases)	---	integer

IV.B.4. [*I-4*] Deposition Function Creation

This routine generates deposition functions from either output data from the Brice codes or from tabulated data. If Brice code data are used the data must be processed into a form compatible with the reading programs. This processing is done by assigning the input file (Brice data) and a temporary output file and converting from unformatted to formatted data as:

```
@ ASG,AX B*26.              (Brice code file)
@ ASG, T   13.
@ XQT. T*DAMEN. UNFOR/FORM
@ FREE B*26.                (Brice code file)
@ ASG,T  26.                (another 26 with different qualifier)
@ EDIT,F 13.,26.
```

The master code (DEPFUN) is initiated by the commands (assuming data it to be written in file 11):

```
@ ASG,AX  11.
@ XQT. T*DAMEN. DEPFUN
```

if input data are from a Brice calculation the command and data are:

```
BRICE
1 INST2      if=CREATE, deposition          ---          integer
             functions will be developed
1 INST3      if=ELECT, data will be for     ---          integer
             electronic deposition
1 IPRINT     if=1 data will be printed      ---          integer
```

if the data are to be read from tabulated values the following data are needed:

```
GENOR
```

(A)	1 INST2	(same as above)		
(B)	1 INST3	(same as above)		
(C)	1 NUME	number of incident energies considered (MAX=10)	---	integer
	2 NUMX	number of locations for data	---	integer
(D)	1 IPRINT	(same as above)		
(E)	1 C1	conversion for range data	---	real
	2 C2	conversion for ED/RP data	---	real
(F)	1 E(I)	incident energies	keV	real
	2 RP(I)	normal range for each E(I)	---	real
	3 EDRP(I)	damage/range factor of E(I) (all repeated NUME times)	---	real
(G)	1 XRP	location/normal range	---	real
	2 Q(I,J)	normalized damage at XRP	---	real
	. .	.		.
	. .	.		.
	. .	.		.
	NUME Q(I)NUME	normalized damage at XRP (repeat all for I-NUMX times)	---	real

If the parameter for INST2 was "CREATE" for either of the above two cases, a call will be made to the least squares fitting program which requires the following data for nuclear deposition values:

	1 ORDER	order of fit to the left of the peak	---	integer
	1 STO	position of match point to left of peak	---	real
	1 INK	1=print out fitting characteristic	---	integer
	1 INK	1=print out (right of peak)		integer
	1 ORDER	order of fit to the right of the peak	---	integer

for the electron deposition values:

	1 ORDER	order of fit	---	integer
	1 INK	1=fitting characteristics printed out	---	integer

after fitting the coefficients are stored in file 11 which require:

	1 NUMFL	the data block number in file 11	---	integer
	1 IDENT	a 54 column identifier for the ion target combination, etc.	---	integer

if desired this routine can recreate deposition profiles from the data in file 11 with the following commands and data:

REGEN

(A)	1 LOCDFN	data block location in file 11	---	integer
	2 CONV	conversion factor if desired results are different than keV/μ	---	real
(B)	1 NUM	number of incident energies to consider	---	integer

	2 E(I)	incident energies (NUM values)	keV	real
	1 IPRINT	1=print output		

if plots are required the following commands and data are needed:

PLOT

	1 NDPLOT	1=plot original deposition profiles	---	integer
	2 ND1	INDEX for first plot of original profiles	---	integer
	3 ND2	INDEX for last plot	---	integer
(A)	4 ND3	increment for above plots	---	integer
	5 NYPLOT	1=plot regenerated deposition profiles	---	integer
	6 NY1	index for first plot of regenerated profiles	---	integer
	7 NY2	index for last plot of regenerated profiles	---	integer
	8 NY3	increment for regenerated plots	---	integer
(B)	1 CONVE	conversion factor for plot ordinate	---	real

At any time the use may also check the contents of file 11 by:

@ XQT T*DAMEN. 11-READER

	1 I	data block number in file 11	---	integer

IV.C. Supporting Routines

IV.C.1. Filing

Portions of the filing system have already been introduced in previous sections. This section will repeat the instructions used by the photon [P-1], ion [I-1], and displacement [I-2] codes and will also present information on related routines.

All filing commands must be preceded by an initialization command of either:

```
OPEN , or I-OPEN
1 FILENAME    file name of data file          ---          integer
```

or

```
REOPEN or I-REOPEN
1 FILENAME    file name of data file          ---          integer
```

A descriptive narrative may be placed in the "log" of each data file with:

```
LOGIN
(A)
(B)          up to 50 lines of
.            narrative to be placed
.            at the top of the file
.
(N)
```

the list is terminated by

```
@ EOF
```

When data is placed in a particular "bin" as with a command of P-FILE or IlFILE, there must be followed by a single line of narrative which identifies the data being placed in the file, e.g.:

I-1FILE

ENERGY DEPOSITION FROM IONS, etc.

Once data has been placed in a file a directory of the file's contents (log and individual entries) can be obtained by the following commands:

```
@ ASG,AX FILE.
@ XQT T*DAMEN. READ-DIR/IODR
 FILE
```

In addition, the data in any bin created by the ion codes may be read and printed out by the commands:

```
@ ASG,AX FILE.
@ XQT T*DAMEN. RDFILE/IODR
 FILE
```

(A)	1 INST	RDFIL1-Ion-1 Type	---	integer
		RDFIL2-Ion-2 Types		
		RDFIL3-Ion-3 Types		
	1 IBIN	bin number of data	---	integer
	2 INOUT	1=print times energy, flux	---	integer
		and spectra		
		2=print deposition or temper-	---	integer
		ature vs. time		
(B)		3=print deposition or temper-	---	integer
		ature vs. location		
	3 IA	first time or location in	---	integer
		printout		
	4 IB	last time or location in		
		printout		
	5 IC	increment in time or location		

IV.C.2 [*S-2*] Superposition

The addition code is used once data is placed in an IODR file. Here data which has been generated on a common time and location basis can be superimposed if it was produced in a linear process.

Two versions are called from the master routine for either: a) temperatures from any combination of photons or ions, or b) energy deposition or displacement calculations. The command is:

```
@ ASG,AX DATAFILE.
@ XQT T*DAMEN. ADDITION
 DATAFILE
```

followed by either

```
ADDTEM
```

or

```
ADDDEP
```

and

(A)	1	bin # to be added to total	---	integer
(B)	1	bin # to be added to total		
.	.			
.	.			
(N)	N	bin # to be added to total	---	integer

```
@ EOF
```

1 MSG	directory notation for total bin	---	integer

IV.C.3. Plotting

The basic plotting package is contained in the absolute element initiated by the instructions:

```
@ ASG,AX  DATAFILE.
@ GSP,P
  PLOTTER PEN/LIQ
@ XQT T*DAMEN. PLOT/IODR
```

followed by a "namelist" input of the following parameters:

```
    1 NP          number of points (not used)
    2 PRINT       if true, data is printed
                  default=false
(A) 3 PLOT        if true, data is plotted
                  default=true
    4 IDEV        output device, default=5 (interactive) (1=direct plots)
    5 LONG Y      if true, Y axis is long
                  axis default=false
    6 PRT LNS     if true, available line table
                  is printed, default=false
```

then the filename is given:

```
(B) 1 FILENAME    name of IODR data file          ---          integer
```

followed by instructions for up to 20 lines plotted per page specified by three parameters per line

```
    1 NUMBIN      bin number from which data      ---          integer
                  is taken
(C) 2 NUMLIN      code for variable selection     ---          integer
    3 INDVAR      index for independent variable  ---          integer
```

The NUMLIN and INDVAR variable are selected from the following table:

Ions

1 Flux vs. time	Ind. var. = 1
2 Spectra vs. energy	Ind. var. = 1
3 Deposition vs. time	Ind. var. = X index
4 Deposition vs. X	Ind. var. = time
5 Integrated deposition vs. time	Ind. var. = X index
6 Integrated deposition vs. X	Ind. var. = time
7 Temperature vs. time	Ind. var. = X index
8 Temperature vs. X	Ind. var. = time

Photons

9 Spectra vs. energy	Ind. var. = layer #
10 Absorption coef. vs. energy	Ind. var. = layer #
11 Energy absorption vs. X	Ind. var. = layer #
12 Adiabatic temperature vs. X	Ind. var. = layer #
13 Temperature vs. time	Ind. var. = X index
14 Temperature vs. X	Ind. var. = time
15 same as 2 except units = $J/cm^2/keV$	

After the desired number of lines are specified the sequence is terminated with:

@ EOF

and

1 ILOGX	if 0 X axis = log	---	integer
(E)	if 1 X axis = linear		
2 ILOGY	if 0 Y axis = log	---	integer
	of 1 Y axis = linear		

(F) 1 XLAB label for X axis

(G) 1 YLAB label for Y axis

(H) 1 TITLE label for graph

The process above is repeated for additional plot pages. If the number of points is to be user specified these alternative commands are used:

```
@ XQT T*DAMEN. PLOTNUM/IODR
DATAFILE
  $ NAMELIST, where NP is specified and all other data are identical.
```

If data are to be taken from several data files another routine is used as:

```
@ ASG,AX      DATAFILE1.
@ ASG,AX      DATAFILE2.
@ GSP,P
PLOTTER   PEN/LIQ
@ XQT T*DAMEN. PLOTCROSS/IODR
  $ NAMELIST (same as PLOT/IODR)

  1 NUMBIN
(A) 2 NUMLIN      (same as PLOT/IODR)
  3 INDVAR

(B) 1 FILENAME    Data file for above data

  1 NUMBIN
(C) 2 NUMLIN
  3 INDVAR
```

(D) 1 FILENAME Data file for above data

(etc.)

The balance of the instructions are identical to the normal PLOT/IODR routine.

Chapter V

Sample Data and Reference Values for Various Materials

```
=====  EXAMPLE/PHOTON  =====

                        SAMPLE INPUT FOR THE PHOTON CODE


@ASG,AX T-NEW*SPEC-2-CU-7.,///600
@XQT  T*DAMEN.M/E
@ADD T*DAMEN.ELT/ATOM
STARAY   GENERATE THE STANDARD TIMES & LOCATIONS
OPEN      THE IODR FILE
SPEC-2-CU-7
LOGIN   BEGIN THE FILE LOG
                SPECTRUM-2
  TRIAL FOR REVISED ION CODE
SPECTRUM HAS TWO PHOTONS AND SIX IONS. MATERIAL IS COPPER.
 FEBRUARY 25,1978
RADIUS IS SEVEN METERS. TOTAL ENERGY IS 100MJ
@EOF
  ******   COMPONENT #1   **********
P-LAYER
 1, 1.E-3  11
       29,    8.96,  .11,  0.903,  0,  63.5,  1.,  2000.
@EOF       OF THE WALL LAYER
@EOF       OF MATERIAL COMPOSITION
P-LASER ABSORBTION
  10.6   .2E+6  7.
PSTEMP
 1  2   1.E-8    .1   100     1.
 1   15   3
P-FILE
10.6 MICRON LASER ONTO COPPER
      ******   COMPONENT  #2  ******************
P-SPECTRUM
    1.0  100    2.E+6  7.
P-LAYER
1  700. 3
10  9.002E-4  1.25 .5  1 20.2  .5  273.
@EOF
 2, 1.E-3  11
       29,    8.96,  .11,  0.903,  0,  63.5,  1.,  2000.
@EOF       OF THE WALL LAYER
@EOF       OF MATERIAL COMPOSITION
P-DEPOSITION
    1   2
PSTEMP
 2 1  1.E-8  .1  100   1.
   1  15  3
P-FILE
1. KEV BLACKBODY XRAYS ONTO CU AFTER ATTENUATION BY 1 TORR OF NEON

@EOF    END OF THE PHOTON DATA GENERATION
```

```
=====  EXAMPLE/IONCODE  =====

                    SAMPLE INPUT FOR IONCODE

     @ASG,AX T-NEW*SPEC-2-CU-7.
     @ASG,AX 11.
     @XQT  T*DAMEN.IONCODE
     I-REOPEN THE IODR FILE
     SPEC-2-CU-7
     **********COMPONENT #2  **********
     I-SPECTRUM CALCULATIONS
     1 240. 10. 0. 0. 96 1.30E+20 7. 3. .10 100 0
     5 .903 8.96 .111 1. 12 155. 130. 0. 0.
     500.  1000.  207.  178.  267.
     LITMOD
     DONT FILE
     RANGE
     120.  .0296  240.  .0375  500.  .0393  1200.  .0289  370.
     760. 273.
     700. .5 273. 0
     STARA3
     5.E-4
     I-DEPOSITION CALCULATIONS
     50
     120.  150.  240.  190.  660.  200.  1350.  157.  207.
     I-1FILE
     TRITIUM IONS ONTO CU 240 KEV MAX SPEC AFTER 7M OF .5 TORR NE
     I-2FILE
     TRITIUM IONS ONTO CU 240 KEV MAX SPEC AFTER 7M OF .5 TORR NE
     STARAY
     I-STEMP  CALL THE STANDARD ION TEMPERATURE ROUTINE
     1  15  3
     I-3FILE
     TEMP FOR T IONS ONTO CU 240 KEV MAX AFTER 7M OF .5 TORR NE
     **********COMPONENT #4**********
     I-SPECTRUM CALCULATIONS
     1 800. 10. 0. 0. 96 1.5E+19  7. 28. .10 100 0
     2 .903 8.96 .111 1. 12 0. .705E-7 0. 0.
     SPEMOD
     FILE
     RANGE
     1000. 210.7  29.03  2500.  417.7  36.2 1. 273.
     700. .500  273. 0
          MODIFIED SI SPECTRUM AFTER 200 CM OF .5 TORR NE GAS
     STARA3
     5.E-4
     I-DEPOSITION CALCULATIONS
     50
     1  2
     I-1FILE
       SI IONS ONTO CU  800 KEV MAX SPEC AFTER 7 M OF .5 TORR NE
     I-2FILE
       SI IONS ONTO CU  800 KEV MAX SPEC AFTER 7 M OF .5 TORR NE
     STARAY
     I-STEMP  CALL THE STANDARD ION TEMPERATURE ROUTINE
     1  15  3
     I-3FILE
      TEMP FOR SI IONS ON CU  800 KEV MAX AFTER 7 M OF .5 TORR NE
     @EOF       THIS IS THE END
```

```
          =====  EXAMPLE/DISPLACEMENT  =====

 1.     @ASG,AX 11.
 2.     @ASG,AX DPA*SPEC-2-CU-7.
 3.     @XQT T*DAMEN.DISPLACEMENT
 4.     STARA3
 5.     5.E-4
 6.         CALCULATE THE DISPLACEMENTS FOR SPEC-1-CU
 7.         **********COMPONENT ION -1  DEUTERIUM**********
 8.     I-SPECTRUM
 9.     1 160. 10. 0. 0. 96 1.30E+20 7. 2. .10 100  0
10.     5 .903 8.96 .111 1. 15 200. 200. 0. 0.
11.       440.  1000.     200.     149.    237.
12.     DISPLACEMENT DO A DISPLACEMENT CALCULATION
13.     2.  1.  63.5  29. .030
14.     80.  150.  160.  190.  440.  200.  900.  157.  207.
15.     BASALT
16.     OPEN
17.     SPEC-2-CU-7
18.     LOGIN
19.             DISPLACEMENT CALCALCULATIONS
20.                  SPEC-2-CU-7
21.             NEW CODE
22.         MARCH 5 1978
23.     @EOF
24.     I-1FILE
25.         DISPLACEMENTS FROM DEUTERIUM ON COPPER 160KEV MAXWELLIAN
26.     I-2FILE
27.     DISPLACEMENTS FROM A DEUTERIUM ON COPPER 160 KEV MAXWELLIAN
28.         ********** COMPONENT - 4   SILICON ON COPPER SPEC-2-CU-7
29.     I-SPECTRUM
30.     1 800. 10. 0. 0. 96 1.5E+19  7. 28. .10 100 0
31.     2 .903 8.96 .111 1. 15 0. .705E-7 0. 0.
32.     HIDPA   DO A LOW ENERGY HIGH MASS CALCULATION FOR DPA
33.     1   1.57E+6
34.     BASALT
35.     I-1FILE
36.        DISPLACEMENTS FOR SILICON IONS ON COPPER SPEC-2-CU-7
37.     I-2FILE
38.     DISPLACEMENTS FOR SILICON IONS ON COPPER  SPEC-2-CU-7
39.     @EOF
```

===== EXAMPLE/DEPOSITION =====

```
 1.
 2.
 3.                SAMPLE INPUT FOR SINGLE ION DEPOSITION
 4.
 5.
 6.                          LIGHT IONS
 7.
 8.
 9.      @ . HELIUM ON CARBON
10.      @XQT T*H.LOWZDEP
11.      160.  47.2  320.  59.8  1000.  52.2  3600.  19.3  164.
12.      3.461  0.3707  3.639  4.  1.75  12.  2.  6.
13.      1  0  0  0  1
14.      40.  4040.
15.      1  100.
16.
17.
18.
19.
20.                         HEAVY IONS
21.
22.
23.
24.      @ASG,AX 11.
25.      @XQT T*DAMEN.HIGHZDEP
26.      1 2 1.0
27.      5 1000. 2000. 3000. 4000. 5000.
28.      1 1
29.
30.
31.
32.
33.
```

```
=====  EXAMPLE/DEPFUNCREATE  =====

 1.
 2.
 3.
 4.                  SAMPLE INPUT FOR THE
 5.              DEPOSITION FUNCTION CREATION CODE
 6.
 7.               CREATING COEFFICIENTS FROM TABULATED DATA
 8.                 AND PLACING IN FILE 11
 9.
10.
11.
12.      @ASG,AX 11.
13.      @XQT T*M.DEPFUN
14.      GENOR
15.      CREATE
16.      NUCLEAR
17.      9 51
18.      0
19.      @ADD T*M.ALNIN
20.      4
21.      2
22.      0
23.      0
24.      4
25.      1
26.        NUCLEAR ENERGY COEFFICIENTS AL - NI 540 TO 2700 KEV
27.      GENOR
28.      CREATE
29.      ELECT
30.      9 51
31.      0
32.      @ADD T*M.ALNIE
33.      4
34.      0
35.      2
36.        ELECTRONIC ENERGY COEFFICIENTS AL-NI 540 TO 2700 KEV
37.      STOP
```

```
=====  EXAMPLE/FILING  =====

 1.
 2.
 3.              SAMPLE INPUT FOR THE FILING ROUTINES
 4.
 5.          DATA ARE FILED FOR A PHOTON AND ION RUN AND
 6.          THE DIRECTORY IS READ AFTER COMPLETION
 7.
 8.
 9.      @ASG,AX T-NEW*SPEC-2-CU-7.,///600
10.      @XQT  T*DAMEN.M/E
11.      OPEN       THE IODR FILE
12.      SPEC-2-CU-7
13.      LOGIN   BEGIN THE FILE LOG
14.                    SPECTRUM-2
15.        TRIAL FOR REVISED ION CODE
16.      SPECTRUM HAS TWO PHOTONS AND SIX IONS. MATERIAL IS COPPER.
17.       FEBRUARY 25,1978
18.      RADIUS  IS SEVEN METERS. TOTAL ENERGY IS 100MJ
19.      @EOF
20.
21.
22.               PERFORM PHOTON TEMPERATURE CALCULATIONS
23.
24.
25.      P-FILE
26.      1. KEV BLACKBODY XRAYS ONTO COPPER  AFTER  1 TORR OF NEON
27.
28.
29.
30.
31.      @XQT  T*DAMEN.IONCODE
32.      I-REOPEN THE IODR FILE
33.      SPEC-2-CU-7
34.
35.
36.               PERFORM ION DEPOSITION CALCULATION
37.
38.
39.      I-1FILE
40.        SI IONS ONTO CU  800 KEV MAX SPEC AFTER 7 M OF .5 TORR NE
41.      I-2FILE
42.        SI IONS ONTO CU  800 KEV MAX SPEC AFTER 7 M OF .5 TORR NE
43.
44.
45.               PERFORM ION TEMPERATURE CALCULATIONS
46.
47.
48.      I-3FILE
49.       TEMP FOR SI IONS ON CU  800 KEV MAX AFTER 7 M OF .5 TORR NE
50.      @EOF       THIS IS THE END
51.
52.
53.
54.      @XQT T*DAMEN.READ-DIR/IODR
55.      SPEC-2-CU-7
56.      @EOF
57.
```

```
            =====  EXAMPLE/ADDITION  =====

 1.
 2.
 3.
 4.                SAMPLE INPUT FOR THE SUPERPOSITION CODE
 5.
 6.           DATA ARE TAKEN FROM 7 BINS AND SUMMED
 7.             AND FILED IN THE NEXT AVAILABLE BIN
 8.
 9.             THE DIRECTORY IS READ AFTER COMPLETION
10.
11.
12.
13.       @ASG,AX T-NEW*SPEC-2-CU-7.
14.       @PRT,S  R*S.RUNADDITION/SPEC-2-CU-7
15.       @XQT T*DAMEN.ADDITION
16.       SPEC-2-CU-7
17.       ADDTEMP
18.       1
19.       2
20.       5
21.       8
22.       11
23.       14
24.       17
25.       @EOF
26.         SUMMATION OF 2 PHOTONS AND 5 IONS FROM RUN-1/SPEC-2-CU-7 (RERUN)
27.
28.
29.
30.
31.       @XQT   T*DAMEN.READ-DIR/IODR
32.       SPEC-2-CU-7
33.
34.
35.
36.
```

===== EXAMPLE/PLOTTING =====

```
 1.
 2.
 3.              SAMPLE INPUT FOR THE PLOTTING CODE
 4.
 5.      C
 6.      @ASG,AX T*SPEC-2-CU-7.
 7.      @GSP,P
 8.      PLOTTER PEN/LIQ
 9.      @XQT T*H.PLOT/IODR
10.       $IN  IDEV = 1, $END
11.      SPEC-2-CU-7
12.      3 1 1
13.      6 1 1
14.      9 1 1
15.      12 1 1
16.      15 1 1
17.      18 1 1
18.      @EOF
19.      0 0
20.        TIME,SEC
21.        PARTICLE FLUX, PART/SEC/CM2
22.      PARTICLE FLUX FOR IONS ON SPEC-2-COPPER-7M
23.      PLT
24.      3 2 1
25.      6 2 1
26.      9 2 1
27.      12 2 1
28.      15 2 1
29.      @EOF
30.      0 0
31.          ENERGY,KEV
32.          F,PART/KEV/CM2
33.       ION SPECTRA FOR IONS FOR SPEC-2-COPPER-7M
34.      PLT
35.      1 13 1
36.      2 13 1
37.      5 7 1
38.      8 7 1
39.      11 7 1
40.      14 7 1
41.      17 7 1
42.      @EOF
43.      0 1
44.        TIME,SEC
45.      TEMPERATURE RISE ,C
46.        TEMP RESPONSE AT X = 0 SPEC-2-COPPER-7M
47.      PLT
48.      @EOF
49.      @EOF
50.
51.
52.
53.
```

===== MATL/CARBON =====

*****CARBON*****

I. THERMAL PROPERTIES

Z	RHO (GM/CM3)	CP (CAL/GM/K)	ALPHA (CM2/SEC)	T(MELT) (K)	A (AMU)
6	2.25	.5	.082	3950	12

II.ION INTERACTION PARAMETERS

---BRICE PARAMETERS

	Z	AP	N
D	1.5274	0.5273	3.461
T	1.5274	0.5273	3.461
HE	1.444	0.3707	3.639

---DEPOSITION

	E0	S0	E1	S1	E2	S2	E3	S3	SMAX
D	60.	144.	120.	180.	260.	173.	1000.	83.	185.
T	90.	144.	180.	180.	390.	173.	1500.	83.	185.
HE	320.	343.	640.	420.	1000.	404.	3600.	198.	422.

---MOD 5

	C	E0	E1	E2	S1	S2	EM
D	135.	100.	240.	1000.	130.	64.5	116.
T	140.	180.	300.	1500.	138.	64.5	200.
HE	376.	400.	1000.	3600.	404.	198.	583.

===== MATL/COPPER =====

*****COPPER*****

I. THERMAL PROPERTIES

Z	RHO (GM/CM3)	CP (CAL/GM/K)	ALPHA (CM2/SEC)	T(MELT) (K)	A (AMU)
29	8.96	.111	.903	1356	63.5

II.ION INTERACTION PARAMETERS

---BRICE PARAMETERS

	Z	AP	N
D	4.9593	0.3503	3.038
T	4.9593	0.3503	3.038
HE	2.390	0.3327	2.962

---DEPOSITION

	E0	S0	E1	S1	E2	S2	E3	S3	SMAX
D	80.	155.	160.	197.	440.	207.	1000.	154.	164.
T	100.	144.	200.	187.	660.	207.	1500.	154.	164.
HE	320.	482.	640.	598.	1880.	589.	3800.	474.	164.

---MOD 5

	C	E0	E1	E2	S1	S2	EM
D	200.	200.	440.	1000.	200.	149.	237.
T	155.	130.	500.	1000.	207.	178.	267.
HE	570.	500.	1580.	3220.	609.	501.	703.

===== MATL/MOLYBDENUM =====

*****MOLYBDENUM*****

I. THERMAL PROPERTIES

Z	RHO (GM/CM3)	CP (CAL/GM/K)	ALPHA (CM2/SEC)	T(MELT) (K)	A (AMU)
42	10.2	0.0888	.283	2883	95.9

II.ION INTERACTION PARAMETERS

---BRICE PARAMETERS

	Z	AP	N
D	2.831	0.5140	3.057
T	2.831	0.5140	3.057
HE	1.945	0.3731	3.235

---DEPOSITION

	E0	S0	E1	S1	E2	S2	E3	S3	SMAX
D	50.	226.	100.	274.	360.	237.	1000.	143.	164.
T	80.	231.	160.	277.	400.	262.	1500.	143.	164.
HE	200.	537.	400.	710.	1560.	681.	3600.	462.	164.

---MOD 5

	C	E0	E1	E2	S1	S2	EM
D	257.	80.	260.	1000.	257.	139.	104.
T	257.	120.	390.	1500.	257.	139.	156.
HE	582.	240.	1160.	3200.	740.	490.	505.

===== MATL/NEON =====

*****NEON*****

I. THERMAL PROPERTIES

Z	RHO (GM/CM3)	CP (CAL/GM/K)	ALPHA (CM2/SEC)	T(MELT) (K)	A (AMU)
10	9.002E-4 (STP)	.246		24.5	20.2

II.ION INTERACTION PARAMETERS

---BRICE PARAMETERS

	Z	AP	N
D	3.0194	0.3874	3.323
T	3.0194	0.3874	3.323
HE	1.677	0.3581	3.230

---DEPOSITION (AT STP)

	E0	S0	E1	S1	E2	S2	E3	S3	SMAX
D	80.	.0296	160.	.0375	333.3	.0393	800.	.0289	.0399
T	120.	.0296	240.	.0375	500.	.0393	1200.	.0289	.0399
HE	240.	.0833	480.	.112	1200.	.120	3000.	.0870	.123

---MOD 5

	C	E0	E1	E2	S1	S2	EM
D							
T							
HE							

===== MATL/NEON =====

RANGE/DECTA RANGE (P = 1 TORR, T = 273 K)

KEV	D	T	HE	SI	HG
20	0.8452/.2876	0.9455/.3381	0.4288/.1900		
50	1.8529/.3622	2.1370/.4382	--	0.0997/.0563	0.0283/.0119
100	3.1146/.4082	3.6319/.5015	1.7906/.3298	0.2239/.0852	0.0577/.0166
150	4.1692/.4358	4.8503/.5374	--	0.3501/.1118	0.0840/.0216
200	5.1403/.4560	5.9326/.5620	2.9096/.3670	0.4753/.1349	0.1087/.0266
250	6.0800/.4727	6.9404/.5809	--	0.5978/.1550	0.1326/.0314
300	7.0157/.4874	7.9071/.5966	3.7776/.3870	0.7169/.1723	0.1560/.0361
400	8.9332/.5140	9.7909/.6226	4.5255/.4001	0.9460/.2008	0.2019/.0454
500	10.9640/.5394	11.6779/.6449	5.2080/.4099	1.1631/.2234	0.2471/.0545
600	13.1402/.5647	13.6151/.6655	5.8537/.4179	1.3693/.2417	0.2921/.0635
700	15.4772/.5909	15.6287/.6854	6.4788/.4247	1.5658/.2569	0.3371/.0724
800	17.9827/.6184	17.7342/.7052	7.0938/.4307	1.7535/.2697	0.3821/.0813
900	20.6600/.6473	19.9415/.7252	7.7055/.4362	1.9335/.2808	0.4273/.0900
1000	23.5100/.6780	22.2563/.7457	8.3185/.4414	2.1065/.2904	0.4727/.0987
1200		27.2222/.7888	9.5610/.4510	2.4343/.3064	0.5640/.1158
1400			10.8388/.4601	2.7414/.3193	0.6561/.1327
1600			12.1614/.4689	3.0311/.3300	0.7490/.1494
1800			13.5343/.4777	3.3061/.3390	0.8426/.1657
2000			14.9609/.4866	3.5682/.3467	0.9369/.1819
2500			18.7723/.5095	4.1773/.3621	1.1753/.2209
3000			22.9394/.5342	4.7336/.3736	1.4166/.2581
3500			27.4598/.5610	5.2487/.3827	1.6598/.2936
4000			32.3264/.5900	5.7306/.3901	1.9042/.3274
4500				6.1847/.3963	2.1493/.3595
5000				6.6155/.4016	2.3946/.3901
6000				7.4193/.4102	2.8846/.4465
7000				8.1606/.4168	3.3713/.4978
8000				8.8521/.4222	3.8529/.5446
9000				9.5026/.4267	4.3284/.5877
10000				10.1186/.4305	4.7972/.6273

===== MATL/NICKEL =====

*****NICKEL*****

I. THERMAL PROPERTIES

Z	RHO (GM/CM3)	CP (CAL/GM/K)	ALPHA (CM2/SEC)	T(MELT) (K)	A (AMU)
28	8.9	.130	.140		

II.ION INTERACTION PARAMETERS

---BRICE PARAMETERS

	Z	AP	N
D	3.4772	0.4351	3.106
T	3.4772	0.4351	3.106
HE	2.236	0.3318	3.232

---DEPOSITION

	E0	S0	E1	S1	E2	S2	E3	S3	SMAX
D	60.	199.	120.	249.	320.	250.	1000.	152.	164.
T	90.	199.	180.	249.	480.	250.	1500.	152.	164.
HE	260.	512.	520.	670.	1400.	699.	3680.	481.	164.

---MOD5

	C	E0	E1	E2	S1	S2	EM
D							
T							
HE							

===== MATL/TANTALUM =====

*****TANTALUM*****

I. THERMAL PROPERTIES

Z	RHO (GM/CM3)	CP (CAL/GM/K)	ALPHA (CM2/SEC)	T(MELT) (K)	A (AMU)
73.	16.6	.0400	.242	3269	181.

II.ION INTERACTION PARAMETERS

---BRICE PARAMETERS

	Z	AP	N
D	7.2270	0.3581	2.7406
T	7.2270	0.3581	2.7406
HE	3.166	0.3223	3.065

---DEPOSITION

	E0	S0	E1	S1	E2	S2	E3	S3	SMAX
D	100.	174.	200.	206.	400.	206.	1000.	155.	164.
T	150.	174.	300.	206.	600.	206.	1500.	155.	164.
HE	280.	516.	560.	640.	1640.	621.	3600.	464.	164.

---MOD5

	C	E0	E1	E2	S1	S2	EM
D							
T							
HE							

===== MATL/XENON =====

*****XENON*****

I. THERMAL PROPERTIES

Z	RHO (GM/CM3)	CP (CAL/GM/K)	ALPHA (CM2/SEC)	T(MELT) (K)	A (AMU)
54	5.862E-3 (STP)	.246		162.2	131.3

II.ION INTERACTION PARAMETERS

---BRICE PARAMETERS

	Z	AP	N
D	3.4675	0.4566	3.089
T	3.4675	0.4566	3.089
HE	1.627	0.4565	2.985

---DEPOSITION (AT STP)

	E0	S0	E1	S1	E2	S2	E3	S3	SMAX
D	60.	.113	120.	.130	320.	.124	880.	.0783	.135
T	90.	.113	180.	.130	480.	.124	1320.	.0783	.135
HE	160.	.295	320.	.375	1040.	.375	3200.	.250	164.

---MOD5

	C	E0	E1	E2	S1	S2	EM
D							
T							
HE							

===== MATL/XENON =====

RANGE/DECTA RANGE (P = 1 TORR, T = 273 K)

KEV	D	T	HE	SI	HG
20	0.1485/.0923	0.1633/.1114	0.0757/.0589		
50	0.3435/.1318	0.3862/.1602	--	0.0413/.0464	0.0099/.0107
100	0.6224/.1680	0.7012/.2046	0.3152/.1136	0.0737/.0609	0.0183/.0093
150	0.8830/.1914	0.9835/.2315	--	0.1100/.0763	0.0258/.0122
200	1.1426/.2102	1.2519/.2516	0.5432/.1400	0.1472/.0916	0.0328/.0151
250	1.4093/.2269	1.5160/.2682	--	0.1847/.1061	0.0395/.0178
300	1.6873/.2426	1.7811/.2829	0.7410/.1552	0.2219/.1205	0.0460/.0205
400	2.2856/.2732	2.3268/.3092	0.9264/.1661	0.2964/.1455	0.0586/.0256
500	2.9467/.3043	2.9039/.3338	1.1073/.1749	0.3702/.1672	0.0709/.0306
600	3.6733/.3368	3.5196/.3576	1.2878/.1826	0.4429/.1863	0.0832/.0354
700	4.4655/.3713	4.1772/.3825	1.4699/.1896	0.5141/.2035	0.0954/.0401
800	5.3225/.4077	4.8781/.4077	1.6551/.1962	0.5839/.2192	0.1075/.0447
900	6.2428/.4464	5.6232/.4338	1.8441/.2024	0.6521/.2335	0.1196/.0493
1000	7.2247/.4871	6.4121/.4610	2.0374/.2085	0.7189/.2467	0.1318/.0539
1200		8.1200/.5188	2.4383/.2204	0.8483/.2700	0.1562/.0628
1400			2.8591/.2321	0.9726/.2901	0.1807/.0715
1600			3.3000/.2439	1.0924/.3075	0.2054/.0801
1800			3.7610/.2558	1.2081/.3229	0.2302/.0886
2000			4.2418/.2679	1.3199/.3365	0.2552/.0969
2500			5.5278/.2993	1.5850/.3648	0.3181/.1174
3000			6.9280/.3324	1.8323/.3872	0.3817/.1370
3500			8.4358/.3673	2.0649/.4055	0.4459/.1559
4000			10.0450/.4038	2.2850/.4208	0.5104/.1739
4500				2.4943/.4339	0.5753/.1913
5000				2.6944/.4453	0.6403/.2080
6000				3.0709/.4641	0.7705/.2396
7000				3.4211/.4792	0.9005/.2688
8000				3.7499/.4916	1.0302/.2959
9000				4.0607/.5021	1.1590/.3211
10000				4.3561/.5111	1.2869/.3447