NUREG/CR-4396
LA-10549-M
Manual

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

SIMMER Postprocessor Manual

Los Alamos
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Prepared by Wendy Rowley, Group Q-9

NUREG/CR-4396
LA-10549-M
Manual

AN

SIMMER
Postprocessor Manual

Frederick R. Parker

Manuscript submitted: September 1985
Date published: October 1985

Prepared for
Division of Accident Evaluation
Office of Nuclear Regulatory Research
US Nuclear Regulatory Commission
Washington, DC 20555

NRC FIN No. A7015

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

CONTENTS

SIMMER POSTPROCESSOR MANUAL

by

Frederick R. Parker

ABSTRACT

The T6P postprocessor analyzes SIMMER-II TAPE6 and TAPE36 data. The processor calculates new variables, integrates variables over regions of the mesh, compares values of a variable of a given problem at selected times, compares values of a variable at selected times between problems for parameter studies, calculates derivatives of a variable with respect to time or another variable, traces a variable at a requested location in the mesh over time. The data are presented to the user graphically, using two-dimensional graphs and three-dimensional perspective or contour plots. Interactive graphic techniques may be used with perspective plots.

I. INTRODUCTION

The postprocessor T6P is a graphics tool to assist SIMMER-II (Ref. 1) users in diagnosing the results of a problem. The processor plots data from SIMMER-II such as RHO, the reactivity from the neutronics section, or ALPG, the volume fraction of vapor from the fluid dynamics section. The processor calculates and plots integrals of variables over the mesh. A new variable may be calculated from any variables or constants known to the processor. The new variable will be plotted. A trace over time of any variable in a specified cell of the mesh can be made. The trace could simulate a sensor in an experiment. Contours of a variable may be plotted to assist in analyzing the motion of a material or the change in temperature in regions of the mesh. Comparisons may be made at different times to demonstrate the losses and gains of a variable. To assist with parameter studies, comparisons of variables between problems are available. Movies of perspective and contour plots of mesh variables are useful in analyzing various physical phenomena. The perspective plots may be rotated at specified times to give different views of the mesh.

The T6P postprocessor is designed to make expansion easy. The input for the processor has a free format form to simplify the construction of input decks and the adding of new options. The reading and saving of data from the SIMMER-II files has been segmented to such a high degree that if a new type of data is added to the SIMMER-II file, the only changes necessary would be an additional routine to save the data and a change to an existing computed GO TO statement.

The T6P processor is a production code and is difficult to use for experimental methods of analysis. The P2M processor has been developed to remedy this handicap. P2M is a stand-alone processor designed to use data files generated by the T6P postprocessor and is intended to be used for development of experimental analysis procedures. P2M should be used to test all new analysis procedures before they are added to the T6P postprocessor. See Appendix A for a description of the P2M processor.

II. METHODS

The numerical methods used in the T6P processor are described in this chapter.

A. Interpolation

All temporal interpolations in the T6P processor are linear. It is assumed the time steps of the data are small enough to make the error of the linear interpolation acceptable. The interpolation formula is

$$V = V_1 + (V_2 - V_1)(T - T_1)/(T_2 - T_1) \ ,$$

where V is the desired value of the variable at the requested time T, V_i, i=1,2, is the value of the variable at time T_i, i=1,2, and

$$T_1 \leq T < T_2 \ .$$

B. Integrations and Other Functions

T6P will currently perform 11 types of integration. Other types of integrations may be added by modifying the routines INGRAT and FGRAT. The user must make sure all the data for the new type of integration are available.

In the integration formulas listed below, $v_{i,j}$ is the value of the variable for the cell (i,j), $w_{i,j}$ is the value of the weight function for the cell (i,j), and $V_{i,j}$ is the volume of the cell (i,j). In the double summations, the first summation is over j and the second summation is over i. The ranges of the indexes i and j may be any square region of the mesh. See Appendix B for the description of a region as used by the INTEG card. The integration formulas are as follows:

1. $\sum\sum v_{i,j} w_{i,j} V_{i,j} / \sum\sum w_{i,j} V_{i,j}$

   The volume weighted average of v. An example is the mass average of the temperature.

2. $\sum\sum (v_{i,j} - w_{i,j}) V_{i,j} / \sum\sum V_{i,j}$

   The volume average of the difference of $v_{i,j}$ and $w_{i,j}$.

3. $\sum\sum v_{i,j}^2 w_{i,j} V_{i,j}$

   The volume weighted total of the square of v. An example would be the axial kinetic energy.

4. $\sum\sum v_{i,j}^2 w_{i,j} V_{i,j} / \sum\sum w_{i,j} V_{i,j}$

   The volume weighted average of the square of v. An example would be the axial kinetic energy per unit mass.

5. $\sum\sum$ (for all $v_{i,j} w_{i,j} > 0$) $V_{i,j}$ The volume weighted total of the product of v and w that is greater than zero. An example would be the upward momentum.

6. $\sum\sum$ (for all $v_{i,j} w_{i,j} > 0$) $V_{i,j} / \sum\sum w_{i,j} V_{i,j}$ The weighted volume average of the product of v and w that is greater than zero. An example would be the upward momentum per unit mass.

7. $\sum\sum$ (for all $v_{i,j} w_{i,j} < 0$) $V_{i,j}$

   The volume weighted total of the product of v and w that is less than zero. An example would be the downward momentum.

8. $\sum\sum$ (for all $v_{i,j} w_{i,j} < 0$) $V_{i,j} / \sum\sum w_{i,j} V_{i,j}$

   The weighted volume average of the product of v and w that is less than zero. An example would be the downward momentum per unit mass.

9. $\sum\sum v_{i,j} V_{i,j}$

   The volume weighted total of v. An example would be the total mass over a region.

10. The maximum of $v_{i,j}$

    The maximum value of v over a region. An example would be the peak pressure in a region.

11. $\sum\sum v_{i,j} V_{i,j} / \sum\sum w_{i,j} V_{i,j}$

    The volume weighted total of v divided by the volume weighted total of w. An example would be region enrichment, the ratio of the mass of fertile fuel to the total mass of fuel.

C. Cell Trace

Cell tracing is useful when SIMMER-II is used to simulate an experiment. A cell at a given location in the mesh represents a sensor in the experiment. The cell trace of the SIMMER-II variable representing the type of sensor would give the calculated values for the quantity read at that sensor.

Cell tracing traces the variables of a given type in a specified cell over time. Each requested variable of the type is traced separately. A cell is specified by a pair of numbers (I,J), where I is the cell number in the radial direction and J is the cell number in the axial direction. At most, ten pairs of numbers may be given. If neither I nor J is zero, a single cell is

specified. If either I or J is zero, a string of cells is specified. In this case, the next pair of numbers will specify the first cell in the string. If I = 0, the string of cells is in the axial direction. If J = 0, the string of cells is in the radial direction. The plot of a string of cells is a perspective. The plot of individual cells is a 2-d plot. All contiguous pairs requesting individual cells are plotted on the same frame.

D. Contours

A contour is a plot of a trace of a constant value of a variable in a region of the mesh. A maximum of 20 contours may be requested. The values of the contours are

$$C_N = V_{min} + N \, (V_{max} - V_{min}) \, / \, NC \quad .$$

where N is the number of the contour being plotted, NC is the number of contours being requested, V_{min} is the minimum value of the variable, and V_{max} is the maximum value of the variable. Contour plots may be drawn for any mesh variable. A mirror-image of the contours can be requested. If a mirror-image is requested, the mirror-image contours and the regular contours are plotted on the same frame abutted to each other.

E. Derivatives

The derivative calculation is an approximation of $\frac{dv_1}{dv_2}$, $\frac{dv_1}{dt}$, or dv_1 and is allowed only for system variables. The formulas of approximation are

1. $$\frac{dv_1}{dv_2} = \frac{v_{1,i} - v_{1,i-1}}{v_{2,i} - v_{2,i-1}} \; ,$$

2. $$\frac{dv_1}{dt} = \frac{v_{1,i} - v_{1,i-1}}{t_i - t_{i-1}} \; , \text{ and}$$

3. $$dv_1 = v_{1,i} - v_{1,i-1} \; .$$

where $v_{1,i}$ and $v_{2,i}$ are the values of any variables at time step i, and t_i is the value of the time at time step i.

F. Variable Comparison

Two types of comparisons may be made.

1. Compare the values of a variable at a specified time with the values of the same variable at a second specified time.

2. Compare the values of a variable from problem one at a specified time with the values of the same variable from problem two at a specified time.

Type 1 comparison is used only for mesh variables and could be used to trace the gain or loss of the variable. A perspective of the difference is plotted.

Type 2 comparison is used for mesh and system variables and would be used for parameter studies. A perspective plot is made of the differences for the mesh variables. Times are not specified for the system variables. For system variables, the curves of the variable from each problem are plotted on one frame and the differences of the two curves are plotted on a second frame.

G. New Variables

The new variable calculation is valid only for the mesh variables. The calculation may use any variable or constant known to the T6P processor during the current execution period. The operations are + (plus), - (minus), * (multiply), / (divide), RECP (the reciprocal), SQRT (the square root), SQ (the square), and LOGN (the logarithm to the base e). The result of all previous operations is an operand for the next operation. The operations +, -, *, and / require a second operand. This operand may be a system variable, a mesh variable, or a constant. For example, a new variable card could specify,

```
NEWV +V1 +V2 RECP *V3 LOGN /V4
```

which would define a new variable, in FORTRAN notation, as

```
ALOG(V3/(V1+V2))/V4 .
```

III. PROGRAM DESCRIPTION AND USAGE

The techniques required to use and to modify the T6P processor are described in this chapter. The first section describes the preprocessor needed to create a compiler source. In addition, the code structure, including the source description and options, is described in the first section. Programming is described in the second section; data management and file structure are described in the third section.

A. Overall Program Structure

1. Creating a FORTRAN Compiler Source. The T6P processor was developed on the CDC 7600 with a LTSS system. The program was written in ANSI X3.9-1978 FORTRAN with the full language set. The CDC UPDATE[2] utility was used to maintain, to modify, and to make corrections to the code. The user will need access to UPDATE or HISTORIAN by Opcode, Inc.[3] to use the T6P processor.

a. Preprocessor. UPDATE and HISTORIAN are general maintenance programs and are used to manage and maintain the T6P program.

The directives used in the T6P processor are *DECK, *COMDECK, *CALL, *IF, and *ENDIF. The directives *IDENT, *DEFINE, *ADDFILE, *INSERT, *BEFORE, *DELETE, and *READ are used to maintain the code and to generate a FORTRAN source. See a manual for UPDATE or HISTORIAN for a description of these directives.

b. The T6P Source File Description. The T6P source file furnished the user has been constructed to be used with one of the maintenance processors listed above. The order of the decks in the file are

1. common decks,
2. programs, and
3. subroutines and functions.

The common decks contain all the FORTRAN common, dimension, and equivalence statements that are common among the programs, subroutines, and functions. The program decks contain only the main program. If the processor is to be used on a CDC 7600 with the NOS or SCOPE operating system,* the programs needed to drive the different overlays would be inserted in this section after the main program. The subroutine and function decks, which are

*NOS is the interactive system for the Control Data Corporation's CDC 7600. SCOPE is the batch system for the CDC 7600.

intended to be placed in a user library, contain all the routines necessary for the T6P program except for the system and graphic library routines.

To create a set of binary files for execution on a system that will not force load the programs for the overlays, you must make two compilations. First, extract all the programs necessary for the overlays from the source file and compile. Then extract all subroutines and functions and compile. Make a user library from the second compilation. Next, load the binary file containing the programs with the user library as the primary library. The plot utility library, the system graphic library, and the general system library are the secondary libraries. The resulting binary file is now ready for execution.

c. Code Options. The update directive "*IF" allows many options in the T6P processor. The options for machines and systems other than those used at Los Alamos National Laboratory have not been completed or debugged. The options in the present T6P source file are listed below.

T6P Compiler Options

1) FTN - ON - create a code for the CDC FTN compiler.
   OFF - do not create a code for the CDC FTN compiler.

2) CFT - ON - create a code for the Cray CFT compiler.
   OFF - do not create a code for the Cray CFT compiler.

3) SCOPE - ON - create a code for the CDC NOS or SCOPE operating system.
   OFF - do not create a code for the CDC NOS or SCOPE operating system.

4) FOURB - ON - create a code for a machine with eight bits/byte and four bytes/word.
   OFF - do not create a code for a machine with eight bits/byte and four bytes/word.

5) EIGHTB - ON - create a code for a machine with eight bits/byte and eight bytes/word.
   OFF - do not create a code for a machine with eight bits/byte and eight bytes/word.

6) TENB - ON - create a code for a machine with six bits/byte and ten bytes/word.
   OFF - do not create a code for a machine with six bits/byte and ten bytes/word.

7) OVLAY - ON - create a code with overlays.

OFF - do not create a code with overlays.

8) IBM - ON - create a code for the IBM machines.

OFF - do not create a code for the IBM machines.

9) VAX - ON - create a code for the VAX 11/780 machine.

OFF - do not create a code for the VAX 11/780 machine.

10) PAC - ON - the TAPE36 produced by the program is packed.

OFF - the TAPE36 produced by the program is not packed.

11) NOLCM - ON - create a code for a machine with only one memory.

OFF - create a code for a machine with a small core memory and a large core memory.

d. Deck descriptions. Appendix C presents a list of the programs, subroutines, and functions. These routines are listed in the order they appear in the source file.

e. Libraries. A plot utility library is furnished for the T6P processor. Appendix D gives a brief description of the routines in the library. The user will have to modify or rewrite the subroutines COLOR, WLCH, PLT, DRV, and ADV to communicate with the graphic system at his installation. A user library is to be created from these plotting routines. In the loading sequence, the plot utility library should be accessed before the graphic system library and the general system library.

2. Flow of the Calculation. The T6P processor was developed on the CDC 7600 computer. For the CDC 7600, the code has an overlay structure; this is removed for other computers with virtual or very large memories. The modular structure required for overlays is retained for nonoverlaid codes. The processor contains three primary overlays, each of which may have 63 secondary overlays. The main overlay, which is always in memory, calls the primary overlays into memory as needed, and they may then call the secondary overlays as needed. For nonoverlaid codes, all routines are in memory and are called as needed. A description of the overlays follows.

OVERLAY (0,0). Overlay (0,0) is the main overlay. This routine reads the names of the TAPE6 and TAPE36 files, initializes the graphic system, and calls the primary overlays (n,0), n=1,2,3. See Fig. 1 for the structure of overlay (0,0).

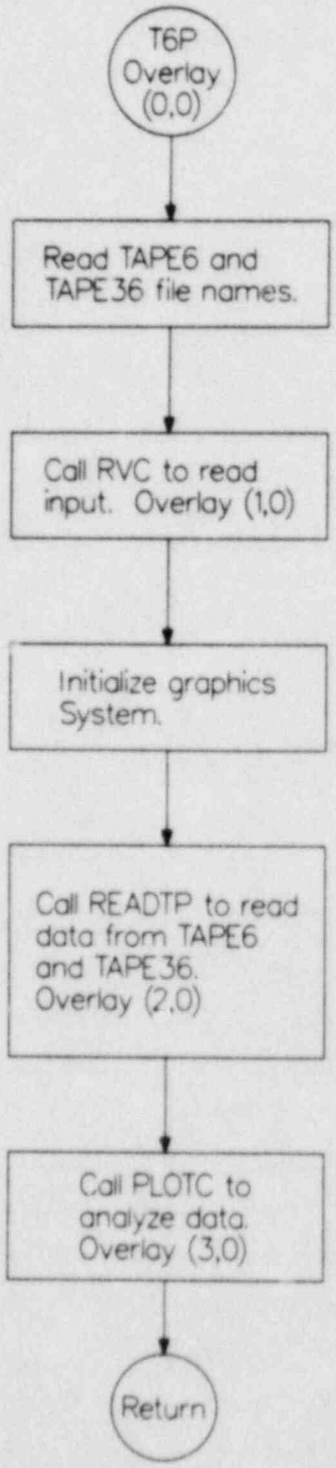

Fig. 1.
Overlay (0,0) flow diagram.

OVERLAY (1,0). Overlay (1,0) reads the card image input. There are no secondary overlays for overlay (1,0).

If the current run is a restart, overlay (1,0) will read the BASE and the BASET files generated by a previous run of the T6P processor.

OVERLAY (2,0). Overlay (2,0) reads the TAPE6 files generated by SIMMER-II . Overlay (2,0) currently has 11 secondary overlays. Overlays (2,j), j=10,11,12, are reserved for future use. Overlays (2,j), j = 1, 2, ···, 9, 13, 14, save the data read from the TAPE6 files. Overlay (2,14) will save the fluid dynamics mesh data for types 14 to 26 when SIMMER-II is modified to write the data to TAPE6. Overlay (2,28) reads and saves the fluid dynamics data from the TAPE36. Only the data for the variables for which the user requested analysis are saved. See Fig. 2 for the structure of overlay (2,0).

OVERLAY (3,0). Overlay (3,0) controls the analysis and plotting of the data read by overlay (2,0). Overlay (3,0) currently has three secondary overlays. See Fig. 3 for the structure of overlay (3,0).

Overlay (3,1) does the analysis for the system data from types 1, 2, 7, and 9. Overlay (3,4) draws a picture of the mesh for types 4 and 13. Overlay (3,5) does the analysis for the mesh type data from types 5, 6, 8, and 14 to 26. See Fig. 4 and 5 for the structure of overlays (3,1) and (3,5).

B. Programming

1. Data Types and Data Reading. The data on the SIMMER-II TAPE6 and TAPE36 files are divided into two general types of variables. System variables are scalar variables defined for the system. Mesh variables are matrix elements with one element defined for each mesh cell. The general types of variables are divided into 26 specific types of variables. Types 1, 2, 3, 7, and 9 are the system variables. Types 4, 5, 6, 8, and 13 to 26 are mesh variables. Types 1, 3, 4, 5, 6, 8, and 9 are from the neutronics portion of SIMMER-II. Types 2, 7, and 13 to 26 are from the fluid dynamics portion of SIMMER-II. Types 10, 11, and 12 are reserved for future use.

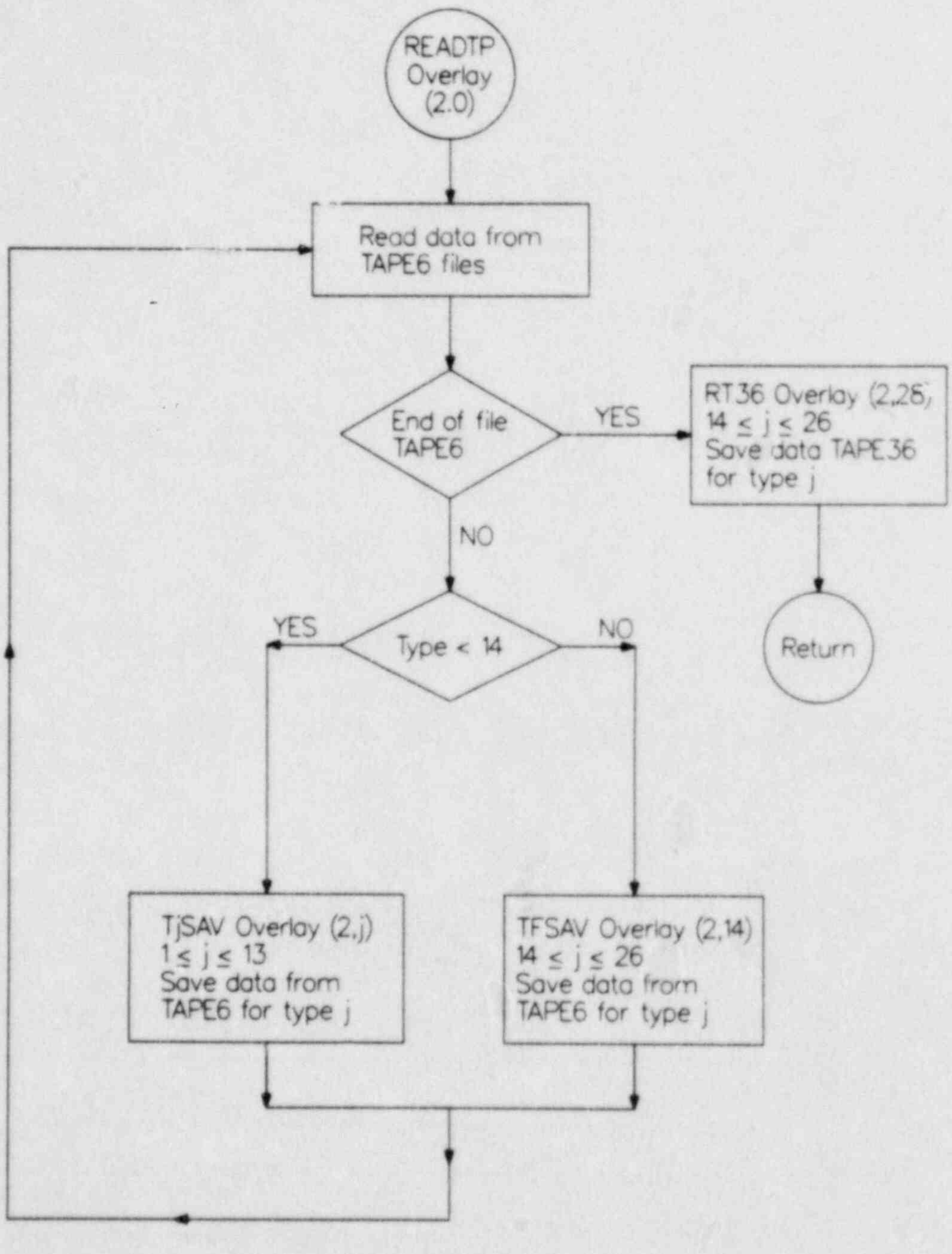

Fig. 2.
Overlay (2,0) flow diagram.

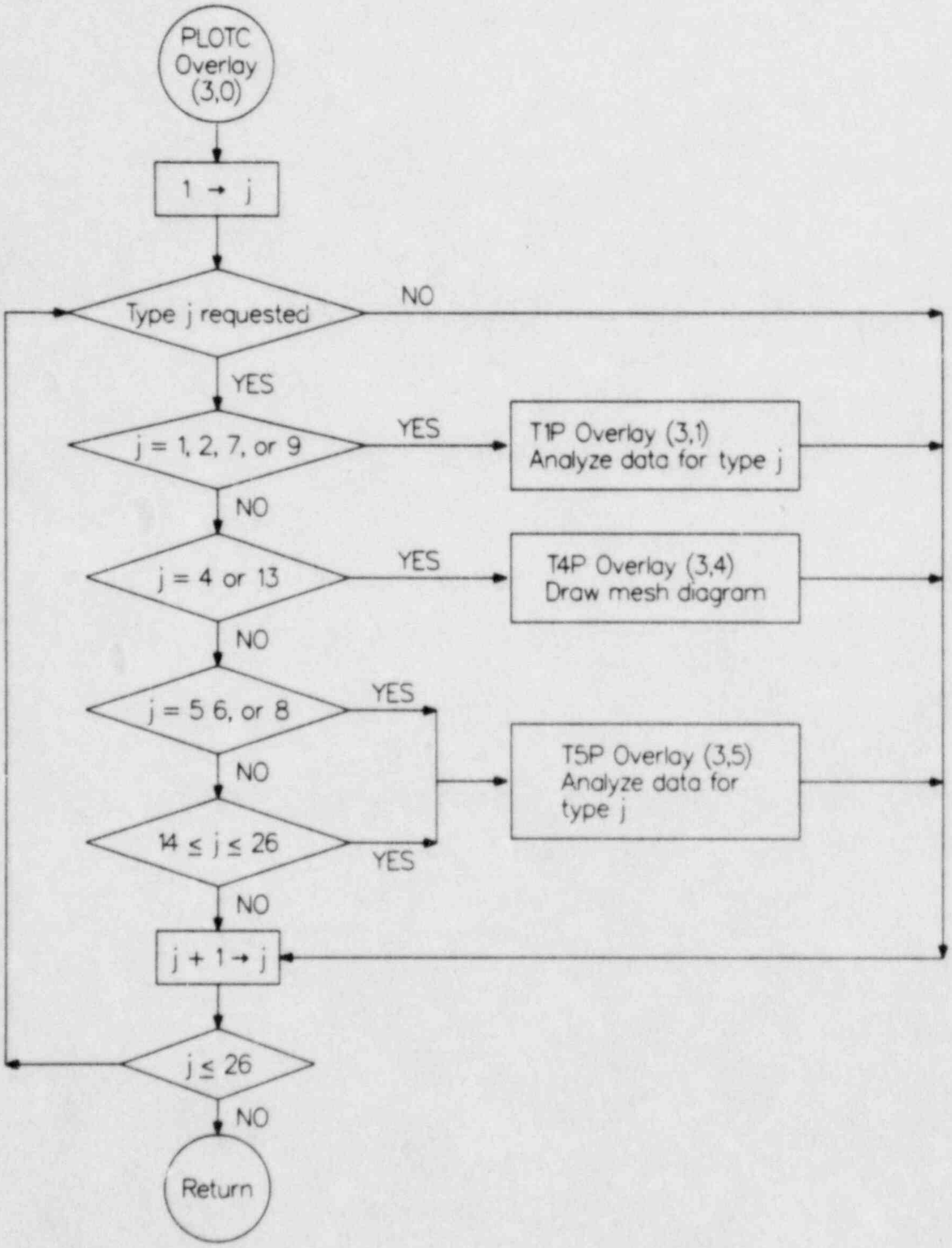

Fig. 3.
Overlay (3,0) flow diagram.

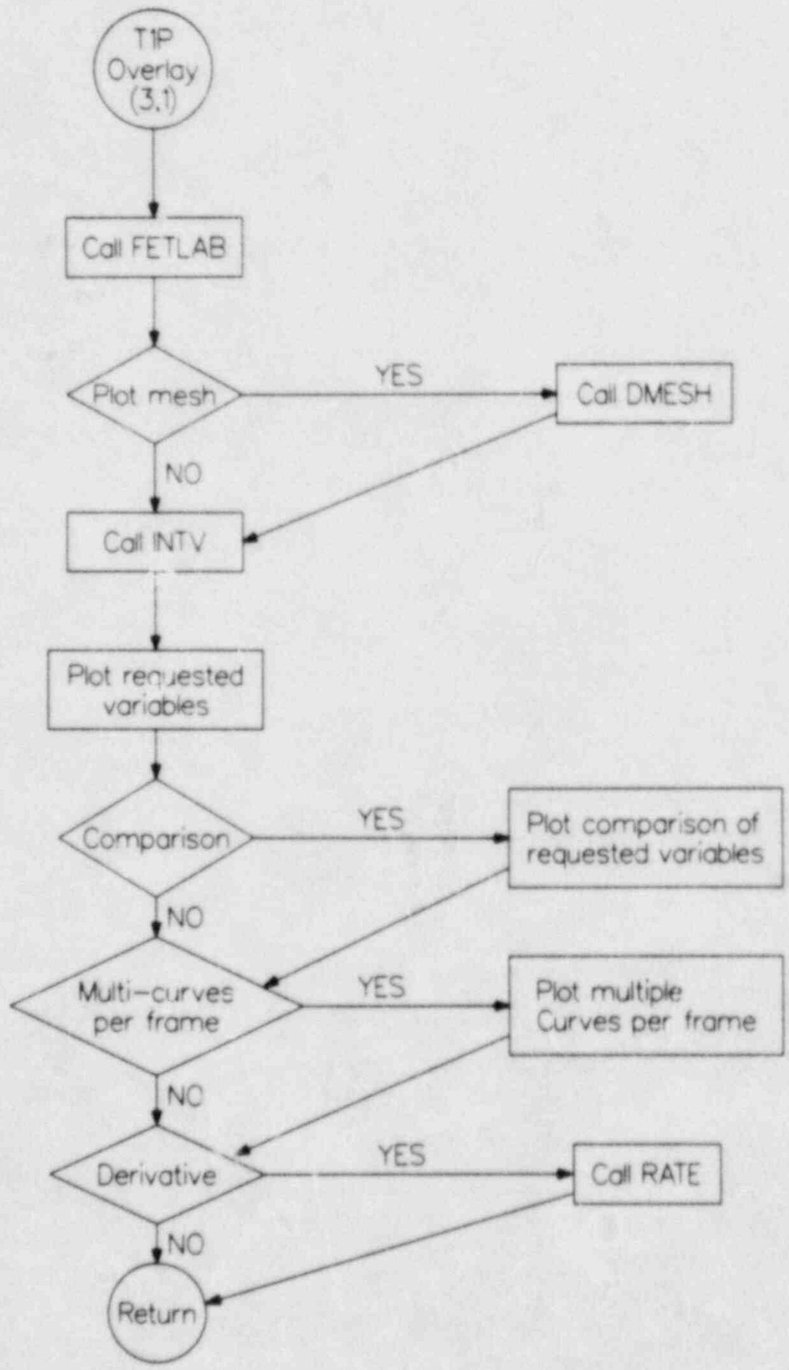

Fig. 4.
Overlay (3,1) flow diagram.

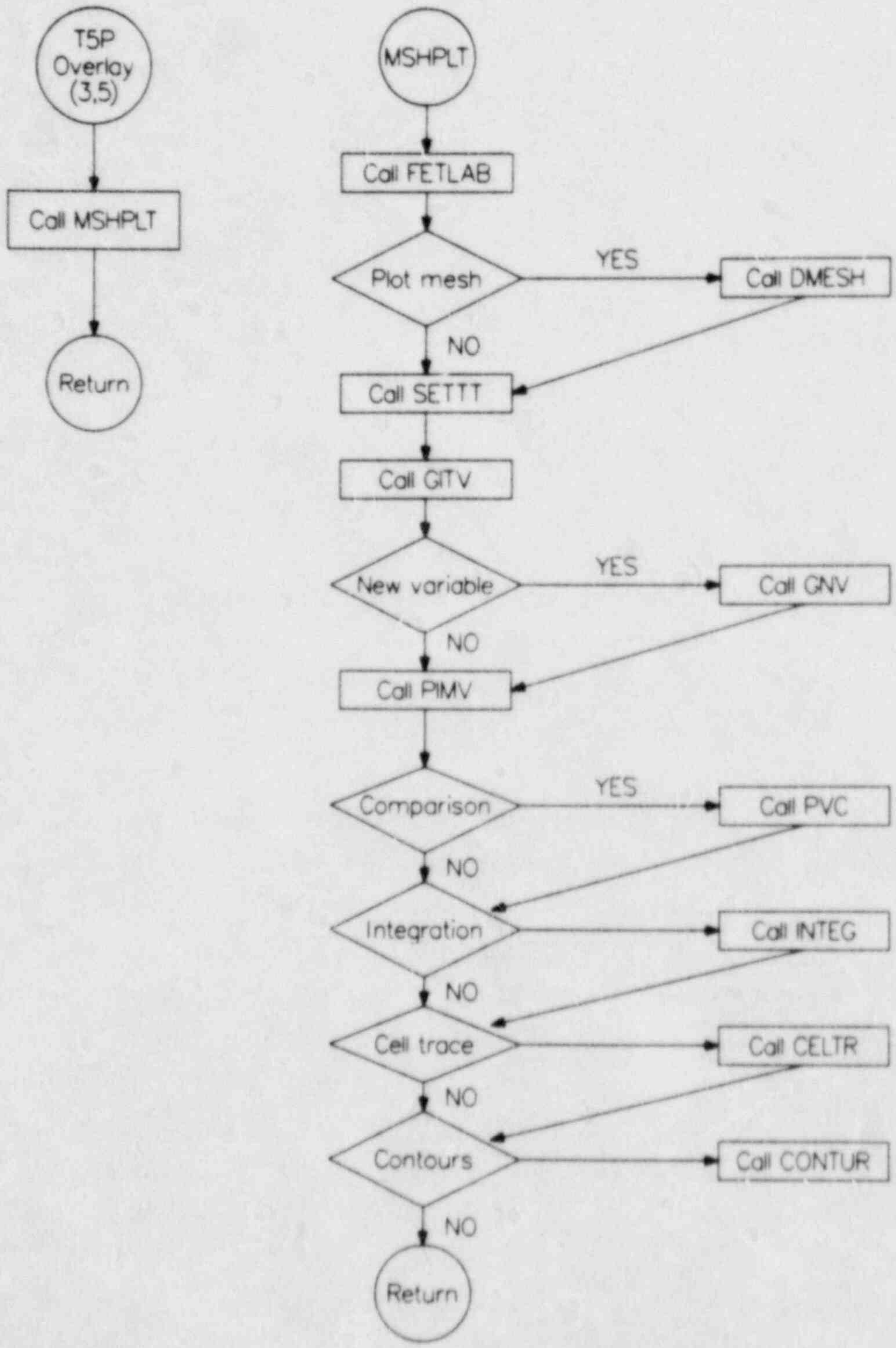

Fig. 5.
Flow diagram of overlay (3,5) and MSHPLT.

The data from the SIMMER-II TAPE6 and TAPE36 files are read with buffering statements. T6P is designed to read data generated by the CDC 7600 LTSS system on the Cray 1 CTSS system and data generated by the Cray 1 CTSS system on the CDC 7600 LTSS system.

2. Input. The first two cards have a fixed format and give the names of the TAPE6 and TAPE36 data files from SIMMER II and the machine on which the data were generated. All other cards are free format delimited by blanks.

The next card gives the names of the BASE and BASET files needed for a restart. If these files are not needed, a blank card or a card with an asterisk as the first character should be used.

The remaining cards name the variables to be analyzed and define the type of analysis. A set of cards is needed for each type of variable to be analyzed. The first card of the set must be a TYPE card. The last card of the set is an END card, the TYPE card for the next set, or an end of file condition from the input file. The remaining cards in the set may be in any order. A detailed discussion of the input can be found in Appendix A.

3. Data Analysis. The methods of analysis are described in Sec. II.

Analysis will be performed for all variables requested for a given type at the times requested. If times are not requested, the analysis will be done at the times the variables were generated by SIMMER-II. At most, ten variables of one type may be requested. The plotting of requested variables may be omitted. Plots of the results of special analysis will be made even if the plotting of the requested variable is omitted. Plotting and analysis of specific regions of the variable may be requested. For the system variables, the regions are a range of times. For the mesh variables, the regions are specified areas of the mesh. Several system variables of a given type may be put on the same frame using the scale of the right or left grid. If a variable has a range of values larger than three decades, the variable will be plotted using a logarithmic scale. If a new variable is requested for a type of variable, all analysis requested for this type of variable will be done for the new variable.

The types of analysis are:

1. Compare a variable at a specified time with the value of the variable at another time. All variables requested for a given type are compared.

2. Compare a variable at a specified time for one problem with the same variable at a specified time from a second problem. All variables requested for a given type are compared.

3. Compute a rate of change of one variable with respect to another variable or with respect to time. This will approximate a derivative.

4. Integrate a variable over the mesh.

5. Make contour plots of the mesh variables.

6. Trace a variable in a specified cell of the mesh.

Analysis types 1, 2, and 3 are used for the system variables. Analysis types 1, 2, 4, 5, and 6 are used for variables defined on a mesh.

C. Data Management

1. SIMMER-II Data Files. The SIMMER-II data files are described in Appendix E. All the SIMMER-II data needed for the T6P processor are extracted from these files. For the mesh type variables, only the data for the requested variables are extracted. All the data available for the system type variables are extracted.

2. T6P Data Files. All T6P data files are direct access files. Files BASE and HYDI are standard direct access files. All other T6P data files are a family of direct access files. These files should be saved so additional data from continuation runs of SIMMER-II can be added without reading the first set of SIMMER-II files again. The names of these files are listed in Appendix F. See Sec. C of Appendix F for a description of a family of files.

3. Internal Files. Two internal files are used by the T6P processor. The files are NV for the new variables and VEF for the interpolated data. These files are created only when analyzing mesh type data and should not be saved. NV and VEF are a family of direct access files.

IV. T6P GRAPHICS

The graphics in the T6P processor were designed to be used with the Common Graphic System (CGS) developed at Los Alamos National Laboratory. Other users will have to furnish the routines listed in Appendix G, Sec. A.

The T6P processor may generate two graphic files for processing on a graphic device or generate plots on a graphic terminal. A plot may be sent to any combination of these three devices simultaneously. One of the graphic files contains still or snapshot pictures and the second one is used for motion pictures. The graphic terminal is used only for interactive graphics for mesh variables. Interactive graphics may be requested for each type of variable only at input time. The interactive graphics may be turned off for each variable or type of variable any time after initiation.

The grid for the frame used in the T6P processor is 1024 × 1024 with the point (0,0) being the upper left corner. For graphic systems with a different grid, a transformation will have to be made. In a commonly used grid for a frame, each side of the frame has length of unity--that is, the value of each coordinate of a point is less than 1. The point (0,0) is the lower left corner. For this grid, the transformation of any point (IX,IY), from the T6P grid would be (IX/1024,1-IY/1024).

A plotting utility library is furnished with the T6P processor. The routines in the library are high-level graphic routines for drawing perspectives, rotating perspectives, crosshatching between two curves, drawing muotiple curves on one frame, and drawing and labeling grids. The routines in this library are listed in Appendix E. The user will have to supply the routines needed for communications with the graphic system at his installation. These routines are used only in the subroutines COLOR, WLCH, PLT, DRV, and ADV.

REFERENCES

1. "SIMMER-II: A Computer Program for LMFBR Disrupted Core Analysis," Los Alamos National Laboratory report LA-7515-M, Rev. (NUREG/CR-0453), June 1980.

2. "Update Reference Manual," Control Data Corporation Publications No. 60342500 (undated).

3. "Historian Plus Users Manual," Opcode, Inc., April 1983.

APPENDIX A

P2M POSTPROCESSOR

The T6P processor is a production code and is difficult to use for experimental methods of analysis. The P2M processor has been developed to remedy this handicap. P2M is a stand-alone processor designed to use data files generated by the T6P post processor and is intended to be used for development of expermental analysis procedures. P2M should be used to test all new analysis precedures before adding them to the T6P post processor.

The data for the P2M processor are from the BASE, BASET, and the variable files generated by the T6P postprocessor. The P2M processor may read files that are foreign to the T6P processing system. To accomplish the reading of the foreign files, the process needing the data would read the files.

There are provisions in the P2M processor for 63 distinct processes. Each process can expect to use all the available memory, that is data for one process would not be available in memory for the next process. During one execution period, one or more of the processes may be executed any number of times with the same or different data sets.

The P2M processor was developed on the CDC 7600 using the UPDATE utility. The user must have access to UPDATE or HISTORIAN, to use P2M (see Sec. III). The P2M processor uses five UPDATE compiler options defined in Sec. III.A.c: FTN, CFT, FOURB, EIGHTB, and TENB. The system-dependent routines defined in Appendix D and the plot utiltiy library defined in Appendix E are also used by P2M.

The routines and common blocks used by the P2M processor are defined below in Secs. I and II. Section III defines the input needed by the P2M processor. Figure G.1 is a flow diagram of the main control program for the P2M processor. Figure G.2 is the flow diagram of the PROCES subroutine, which selects the process to be used.

I. Routines: The routines used by the P2M processor ordered by their appearance on the P2M source file.

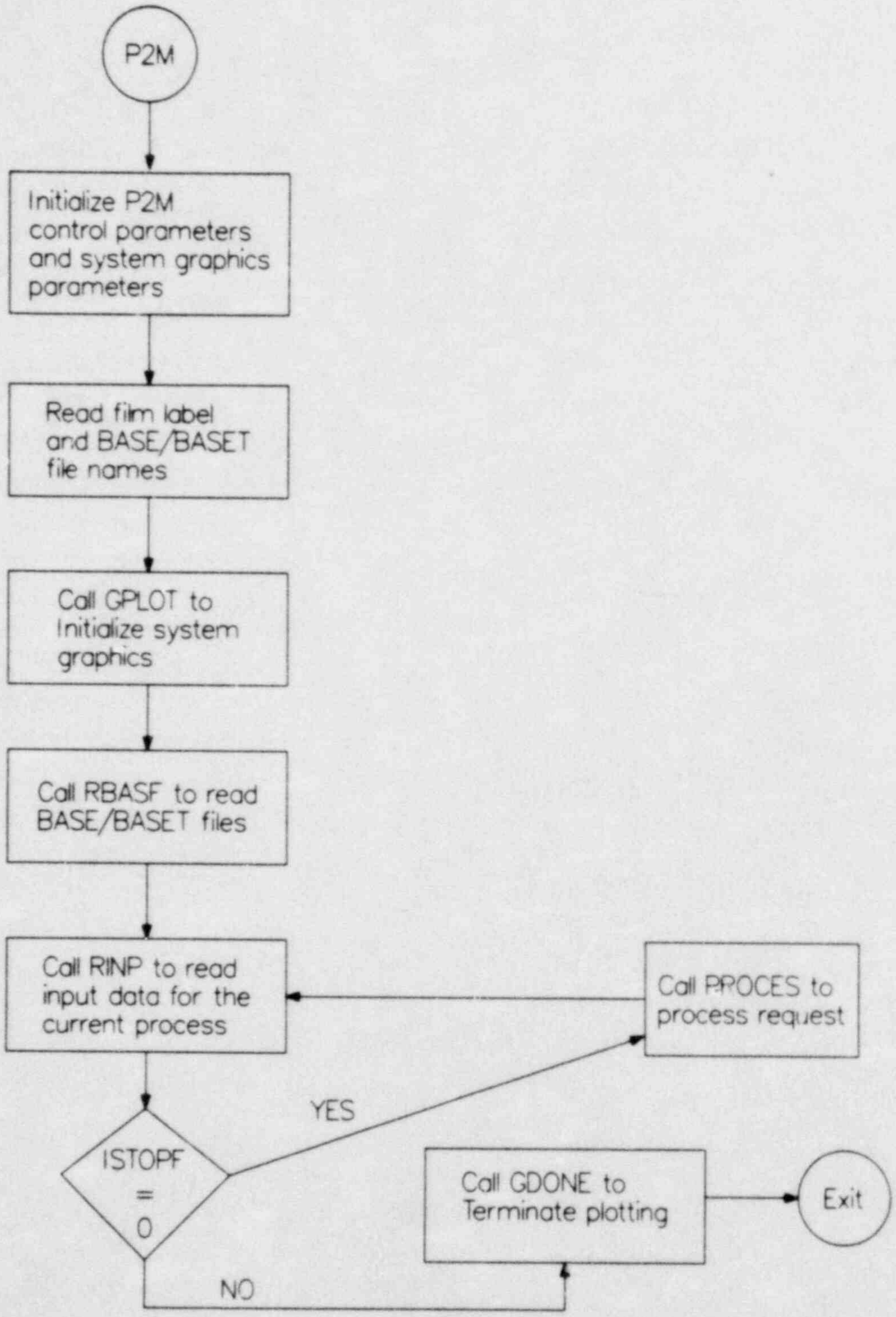

Fig. A.1.
P2M flowchart.

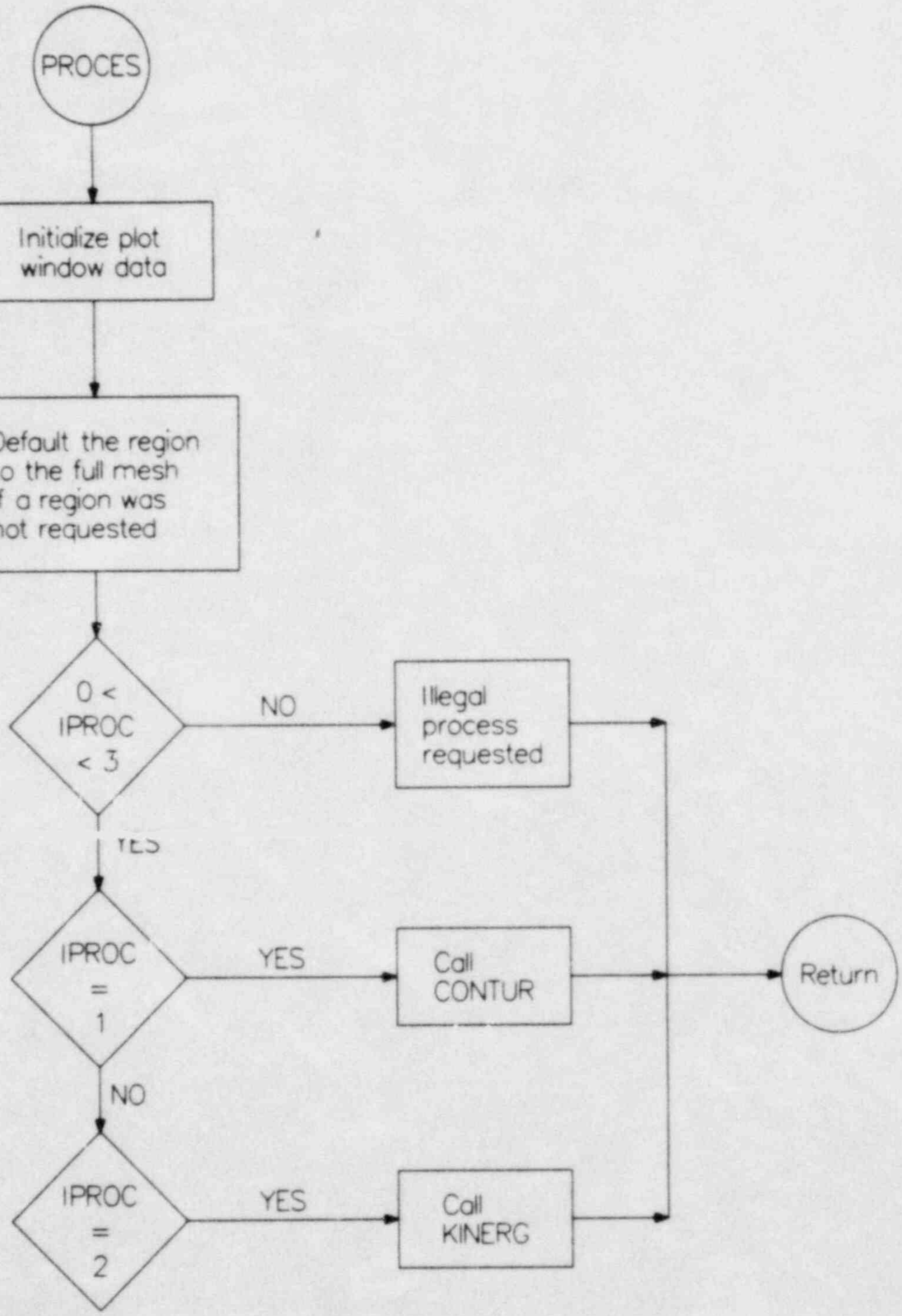

Fig. A.2.
PROCES flowchart.

1. P2M The main program for P2M.

2. RBASF The routine to read the BASE/BASET files.

3. RINP The routine to read the input for the processes.

4. PROCES The routine to select and execute the requested process.

5. KSADDR A routine to calculate the number of words between two variables (inclusively).

6. FSEP A routine to separate an array into N fields of 10 or fewer characters.

7. IRJZ A function to right justify and blank fill a 10-character field.

8. ILJZ A function to left justify and blank fill a 10-character field.

9. SEPFL A routine to separate an array of blank-delimited fields into N fields left justified and blank filled.

10. DNBDL A routine to discard all leading blank characters and the first N blank-delimited fields.

11. ILCP A function to determine if the rightmost character of a right-justified character string is a).

12. IFCP A function to determine if leftmost character of a right-justified character string is a (.

13. RCONV A function to convert a right-justified blank-filled field of 10 characters to a real number.

14. ICONV A function to convert a right-justified blank-filled field of 10 characters to an integer.

15. IDONEQ A routine to test the completion of an I/O operation.

16. SETI A routine to initialize an array to a specified field.

17. LCSETI A routine to initialize an array to a specified field. On the CDC 7600 the array is in LCM.

18. SETC A routine to initialize an array to a specified field of characters.

19. EXLAB A routine to skip N blank delimited field and extract the next m characters ($m \leq 50$) for a label.

20. LPIC A routine to label a picture.

21. CHARCO A routine to delete the leading and trailing blanks of a label and to count the remaining characters.

22. IFTP A function to find a pointer in the time table for a given t such that $T_i \leq t \leq T_{i+1}$

23. TTIVAR A routine to recreate the interpolated time tables.

24. XYV A routine to calculate the mesh dependent variables.

25. CONTUR The driver for Process 1.

26. DRCONT A routine to draw the contours of a variable on a given mesh.

27. KINERG The driver for Process 2. KINERG calculates the kinetic energy for the SIMMER-II fluid dynamics mesh.

II. Common Blocks: The common blocks used by the P2M processor. If T6P has a common biock with the same name as the P2M common block, they will be the same except where stated.

1. ARRLIM The limits to arrays in P2M.

2. BASE IBASE The BASE and IBASE common blocks used to receive data from the BASE and BASET files from T6P. This common block is different from the one used by T6P.

3. CONTOR Process 1 common block to pass data from the driver routine to the other routines.

4. CJE07 A common block of data for the graphics system.

5. DATA LCM data buffer common block.

6. FRAM A common block of variables defining the window in a picture.

7. INPUT The input for P2M. This common block differs from the T6P common block INPUT.

8. LABEL A common block to pass labels to LPIC.

9. PICBOU A common block of data for PICTUR.

10. PLTCOM A plotting common block.

11. PLTPAR A common block of data for the graphics system.

12. QMBUFFC A system common block used for receiving messages from a controller.

13. UNIT I/O parameters.

III. Input: The input to the P2M processor is a free-format design with blank-delimited fields. The number of blanks between fields is important only on LABEL cards. Each LABEL card must start in column one. Each pair of blanks will count as a field on a LABEL card. The input is divided into three sections. Section 1 contains the film header card and the BASE/BASET file name card. Only one set of Section 1 cards is used per execution. Section 2

contains input of the type that can be used by all processes. A set of Section 2 input must be read for each process. Section 3 describes the processes and the input that is unique for a given process. The input of Section 2 and Section 3 is positioned between a PROCESS card and an END card. A card will be continued if its last field is an ampersand (&).

Section 1 Input

CARD 1 The film header label card of 30 characters. All blank characters count. The label must start in column 1.

CARD 2 The BASE/BASET/KP card.

BASE The name of the BASE file from T6P.
BASET The name of the BASET file from T6P.
KP The problem number of the data to be read from the files. $KP \leq 2$.

Example:

BAS1 BAST1 2 BAS2 BAST2 1

NOTE: Only six sets of files may be read, and only one problem may be read from each set. Only the sets of files needed need to be listed.

CARD 3 The process name card.

PROCESS The card name.
K The process number. $0 < K \leq 63$
(See Section 3 for a list of the available processes.)

Example:

PROCESS 1

CARD 4 The end card.

END

NOTE: Cards 3 and 4 are needed for each process to be executed. All input from Sections 2 and 3 needed for each process are positioned between the PROCESS card and the END card.

Section 2 Input

CARD 1 File names input

FILES FN_1 FN_2 $\cdots$ FN_{10}

FN_i is a file name or a variable name. At most, 10 FN_is may be given. If FN_i is a file name it may be a T6P processor variable file or a foreign file. The FILES card is used with the TYPES and PROBNUM cards. All foreign files will have a type of 0 and a problem number of 0. If the type is 1, 2, 4, 9 or 13, FN_i is a variable name. If the type is 2, the name of the file containing the variable will be T2VF. If the type is 5, 6, 8 or $\geq$ 14, Fn_i will be a file name.

Example:

FILES TOTPU RHO DENNA DSCAT

CARD 2 The type of variable card.

TYPES TY_1 TY_2 $\cdots$ TY_{10}

TY_i is the type of variable for FN_i given on the FILES card. If TY_i=0, the FN_i is a foreign file name. Otherwise $1 \leq TY_i \leq 26$, the types defined by SIMMER-II. TY_i are integers. Default is TY_i=0. If TY_i is defaulted, all TY_j, $j > i$ are also defaulted.

Example:

TYPES 28 1 5 9

CARD 3 The problem numbers of FN_i.

PROBNUM PN_1 PN_2 ··· PN_{10}

PN_i is the problem number as defined by the order the BASE/BASET files were read in. If PN_i = j, other data, such as the time tables, will be found in the jth BASE/BASET files read. j is not the KP read from card 2 of Section 1 input. Default is PN_i = 1. If PN_i is defaulted then all PN_j, j > i are also defaulted.

Example:

PROBNUM 2 1 1 2

CARD 4 Clip the maximum values.

CLIPMAX MAX_1 MAX_2 ··· MAX_{10}

MAX_i is the maximum values of the i^{th} variable. The default is the maximum on the variable file. The default is denoted by a "-" character or a blank. If a blank is used for the default of the i^{th} field, then the i^{th} + 1 field is also defaulted.

Example:

CLIPMAX 1.85 - 2

CARD 5 Clip the minimum values.

CLIPMIN MIN_1 MIN_2 ··· MIN_{10}

CARD 8 The region definition cards.

$$\text{REGIONS IP } (IL_{1,1}\ IU_{1,1}\ JL_{1,1}\ JU_{1,1} \cdots IL_{20,1}\ IU_{20,1}\ JL_{20,1}\ JU_{20,1})\ \&$$

$$\vdots$$

$$(IL_{1,20}\ IU_{1,20}\ JL_{1,20}\ JU_{1,20} \cdots IL_{20,20}\ IU_{20,20}\ JL_{20,20}\ JU_{20,20})$$

IP is the problem number the regions are defined for. Regions will be used only with mesh variables. $IL_{i,j}$, $IU_{i,j}$, $JL_{i,j}$, and $JU_{i,j}$ define a given region in the mesh. A pair (I,J) defines a cell in the mesh, I the radial position and J the axial position. The L and U are for the lower (L) and upper (U) index ranges. There can be eight regions in a set and eight sets in a process. The minimum and maximum of a variable is found for all regions in a set.

Example:

```
REGIONS 2 (3 5 3 6 7 9 3 6 11 13 3 6) &
          (3 5 11 14 7 9 11 14 11 1 3 7 9) &
          (1 13 17 20)
```

CARD 9 The frame label cards.

LABEL I (54 characters)

LABEL must start in column 1. There can be only one blank between LABEL and I. The first blank after I is a field separator. The next 54 characters including the leading blanks, are part of the label. $1 \leq I \leq 4$. I is an integer. Only those labels given will be used. If only one label is given and I>1, the label will be spaced I-1 lines from the top of the frame. If more than one label is given and the Is are not contiguous, the lines for the missing Is will be blank lines. All space after the line for the largest I will be used for the picture.

CARD 10 The grid definition card.

GRID FHC FH FVC FV FPFH PFH FPFV PFV &
UOX UX FPUX PUX UOY UY FPUY PUY

FH, FV, PFH, PFV, UX, PUX, UY, and PUY are real numbers described by the FORTRAN format operators Fw.d or Ew.d. w ≤ 10. A frame is the entire viewing surface. A window is that part of the frame used for a picture, including grid labels. A grid is that part of the window used for drawing a picture excluding the grid labels. The length of a side of the frame is 1. The (0,0) point of the frame is the lower left corner.

FHC First horizontal coordinate of the window. Default = .12.
0 ≤ FHC ≤ .75

FVC First vertical coordinate of the window. Default = .12.
0 ≤ FVC ≤ .75.

NOTE: The point (FHC,FVC) in the frame is the lower left corner of the window.

FPFH The fractional part of the frame used as the horizontal boundary of the window. Default = .85. 0 ≤ FPFH ≤ 1.

FPFV The fractional part of the frame used as the vertical boundary of the window. Default = .85. 0 ≤ FPFV ≤ 1.

NOTE: FHC + FPFH ≤ 1 and FHV + FPFV ≤ 1.

UOX A range of the x-axis values to be used as a unit of the x-axis. A reasonable value will be chosen for a default.

FPUX the fractional part of the x-axis to be used for a unit of x. A reasonable value will be chosen for a default.

UOY A range of the y-axis values to be used as a unit of the y-axis. A reasonable value will be chosen for a default.

FPUY The fractional part of the y-axis to be used for a unit of y. A reasonable value will be chosen for a default.

Example:

GRID FHC .5 FVC .5 FPFH .5 FPHV .5 &
UOX .01 FPUX .1 UOY .5 FPUY .1

Section 3 Input

PROCESS 1

Process 1 will draw contours of the variables contained in the files listed on the FILES card. NPIC sets of contours will be drawn on each frame. The input card for process 1 is a card with "CONTOURS" in the first field of the card. The input card is

CONTOURS NCT NC NPIC NP ZCF ZC

NC and NP are integers. ZC is a character string.

NCT Number of contour interval lines. NC is the number of lines. If $NC < 0$, the mirror image contour will be added as part of the picture.

NPIC Number of contour pictures per frame. The pictures are side by side. $1 \leq NP \leq 4$ is the number of pictures.

ZCF Plot the zero contour if zero exists flag. ZC may be YES or NO. The default is YES, plot the zero contour.

Example:

CONTOUR NCT -5 NPIC 4 ZCF YES

PROCESS 2

Process 2 calculates the kinetic energy using the densities and velocities of the liquid and gaseous materials defined by the fluid dynamics mesh. The data are contained on 18 files from the T6P processor. There is no special input for this process. The TYPE card from Sec. 2 is not used. Only PN_1, the first problem number, is used from the PROBNUM card. For process 2 , the FILES card has been extended and 18 file names must be read in. The order of the file names on the FILES card is

1. density of fertile fuel vapor (RG1Fi),
2. density of fissele fuel vapor (RG2Fi),
3. density of steel vapor (RG3Fi),
4. density of sodium vapor (RG4Fi),
5. density of control vapor (RG5Fi),
6. density of fission gas vapor (RG6Fi),
7. density Of liquid fertile fuel (RL1Fi),
8. density of liquid fissile fuel (RL2Fi),
9. density of liquid steel (RL3Fi),
10. density of liquid sodium (RL4Fi),
11. density of liquid control (RL5Fi),
12. density of fertile fuel particles (RL6Fi),
13. density of fissile fuel particles (RL7Fi),
14. density of steel particles (RL8Fi),
15. radial velocity of vapor (UGFi),
16. axial velocity of vapor (VGFi),
17. radial velocity of liquid (ULFi), and
18. axial velocity of liquid (VLFi).

The names enclosed in parentheses are the default names of the files generated by the T6P processor.

Sample Input

```
WPB SPACE SHUTTLE TEST
SPBA SPBTA 1
PROCESS 1
FILES TOTGF TOTLF TOTSF TOTNA TOTPU
TYPES 28 28 28 28 28
PROBNUM 1 1 1 1 1
TIMES 0.  .01 .1
LABEL 2 SIMULATION OF PYRO TEST 118
LABEL 4 OXYGEN DENSITY
CONTOURS NCT -10 NPIC 3
END
```

APPENDIX B

INPUT FOR THE T6P POSTPROCESSOR

The input for the T6P postprocessor, listed below, could be divided into three sections. Cards 1, 2, and 3, the first section, give the SIMMER-II and T6P data files necessary for the execution of T6P. Cards 4 and 5 would be the second section. A card 4, the TYPE card, must be given for each type of variable to be analyzed. Card 5, the END card, terminates the reading of the parameter cards. If card 5 is not used, the next TYPE card or a end-of-file condition on the input file will terminate the reading of the parameter cards. Cards 6 to 22 constitute the third section. These are the parameter cards and give all the data necessary to control the analysis of the variables of a given type. The parameter cards for a type of variable must follow the TYPE card for the variables. The parameter cards are followed by another TYPE card, an END card, or are the last cards in the input file. Cards 1 and 2 have fixed formats. Cards 3 to 22 have free formats, with their items separated by at least one blank. The items on cards 3 to 22 may be in any order except for the several instances in which an order is specified. Continuations cards are allowed for a TYPE card and all parameter cards. An ampersand as the last field of a card designates the next card as continuation of the current card. For a sample of the input file, see Appendix H.

CARD 1 - Table of the SIMMER-II TAPE6 files to be used as input. The format is (3(A8,A6,I6)). All items on thi card are left justified and blank filled. Twenty-one files may be given. As many as seven cards may be used. There are three fields per card. Each field contains three items. If there are not any TAPE6's, insert a null line. The items for this card are:

FILE NAME - Eight or fewer characters, left justified and blank filled. This is the first file of a family of files. Each member of the family is a complete TAPE6 from SIMMER II. The next member of the family is found by incrementing the rightmost character by one. Numeric characters cycle from 0 to 9 back to 0; alphabetic characters cycle from a to z back to

a. When a complete cycle 0-9 or a-z occurs, the next character to the left will be tested and modified in the same manner. The first missing member of a family of files will terminate processing of the current family. The first blank field for this item or the reading of the seventh card terminates reading.

SYSTEM NAME - Six or fewer characters, left justified and blank filled. The name of the system used to generate the file. Each problem may have files generated on different systems. The systems at LANL are

LTSS - the CDC 7600 LTSS system.

CTSS - the Cray 1 CTSS system.

PN - The problem number. Up to two problems may be given for analysis. All files must be ordered first by the problem and then by the problem real time.

EXAMPLE: X4T6 CTSS 1

NOTE: If there are five family members for the file X4T6, the fourth and fifth members would be X4T9 and X4U0.

CARD 2 - Table of SIMMER-I TAPE36 files to be used as input. The format is (3(A8,A6,I6)). All items on this card are left justified and blank filled. Twenty-one files may be given. As many as seven cards may be used. There are three fields per card. Each field contains three items. If there are not any TAPE36's, insert a null line. The items for this card are the same as for CARD 1 above.

EXAMPLE: X4T36 CTSS 1

NOTE: If there are five family members for file X4T36, the fourth and fifth members would be X4T39 and X4T40.

NOTE: Only cards 1 and 2 have a fixed format. All other cards are free-formatted with the items delimited by one or more blanks.

CARD 3 - The names of the base files required for a restart. The file names have eight or fewer characters. This card may be a blank card or the first item on the card may be an asterisk or a file name. If base files are not to be used, then the card is blank or the first item is an asterisk. If the first item on this card is a file name, then a set of base files is to be read. At most, two sets of base files may be read.

BASEi BASETi PNi

BASEi - Name of the base files for problem i.

BASETi - Name of the base file containing the table times for problem i. Only the first member of the family of files needs to be specified, but all family members must be local.

PNi - The number of the problem in the base file to be used as the i^{th} problem of the current run.

EXAMPLE: X4TBASA X4TBASTA 1

CARD 4 - The TYPE card. The items on this card may be in any order except that the item TYPE must be first and the item VAR must be last. The card is free-formatted and blank delimited. A new TYPE card with any combination of parameter cards must be given for each type of variable to be processed. Only one TYPE card with a set of parameter cards may be given for each type. Reading of the TYPE and parameter cards is terminated by reading another TYPE card or an END card.

TYPE ii - $1 \leq ii \leq 28$. The type of record to be processed. TYPE ii must be the first item on the card. All other items may be in any order, except VAR, which must be the last. All items are blank delimited. See Appendix F for a list of the types available.

CPS - Compare the variables of problem 1 with the same variable of problem 2 for each pair of times specified on the CPT card.

CTS - Compare between the times specified on the CTT card each variable specified on the TYPE card. This option applies only to the mesh variables.

VT T - Rotate the viewing positions of perspective plots at time increment T. If T = 0, only the first and the last times will be rotated.

INACT - Do the 3-d plots interactively. This flag will negate the movie and regular still options until the flag is turned off for the current variable by the user.

CONTOUR N - Make a contour plot of N equally spaced curves over the regions defined by the REGION card. Maximum is 20 contours and the default is 10 contours. For color devices, the contour colors will range from red for the minimum through yellow, green, and blue to purple for the maximum. For a mirror-image plot, set N to negative.

MFR j - j ≥ 1. Perform the requested analysis for material j, flux for group j, or reactivity j, depending upon the type. MFR is a substitute for the VAR element for types 5, 6, and 8. MFR may be used in conjunction with VAR. For type 6, DEN, POW, TEMP, and SPOW will be analyzed. Only one variable may be requested for each MFR. If more than one variable is desired, add another MFR, one for each variable.

VAR - List of variables given by name to be processed. See Appendix E for a list of the variable names for each type.

MOVIE (OP) - The list of options for making movies. Movie options do not apply to cell traces or to integrations. If the movie option S is turned on, all still options will be used for every N^{th} time. If the S option is not used, all plot options except cell trace and integration will be made inoperative. If CP plotting is requested, the first times on the CPT cards will be used as offsets. The other times on the CPT cards will be ignored. Movie options are one or all of the following options, blank delimited and enclosed in parentheses.

OP - V Plot individual variables.
CP Plot the comparison plots.
NV Plot the NEWV variables.
NCP Plot the comparison of the NEWV variables.
CV Contour plots of the variable.
NCV Contour plots of the new variables.
S N Take a still picture every N frames.

DONOT FLAG - Do not plot the individual variables as defined by FLAG. This option does not apply to the movie, contour, cell trace, or the integration options. FLAG is

V - The regular variables.
NV - The new variables.
BV - Both the regular and new variables.

EXAMPLE: TYPE 28 CTS VT 0 CONTOUR -10 MOVIE (CV) DONOT V &
VAR RSF1 RSF2 RLF1 RLF2 RLB2 RLB3 RLB7 UL VL TRFU

CARD 5 - The END card. Terminates the reading of the parameter cards for TYPE ii.

NOTE: The cards from this point on are parameter cards giving the parameters necessary to process the variables for the current type. These cards may be in any order. The items on a parameter card may be in any order.

CARD 6 TIMES - The range of times for which the analysis is to be done.

TIMES T_0 DT_1 T_1 $\cdots$ DT_n T_n

where T_i is the starting time for region i, DT_i is the delta T used for region i, and T_{i+1} terminates region i. The number of regions, n, is less than or equal to 5.

EXAMPLE: TIMES 0. .0001 .01 1.E-3 .1 .1 1.

CARD 7 REGION - The regions to be analyzed. For mesh variables, five regions may be given. For system variables 10 regions of time may be given.

Mesh variables:

REGION (I_{11} I_{21} J_{11} J_{21}) $\cdots$ (I_{1n} I_{2n} J_{1n} J_{2n})

where I_{1i} is the left cell of region i,
I_{2i} is the right cell of region i,
J_{1i} is the bottom cell of region i,
and J_{2i} is the top cell of region i, i = 1, 2, $\cdots$, n.

System variables:

REGION (T_{11} T_{21}) (T_{12} T_{22}) $\cdots$ (T_{1n} T_{2n})

where T_{1i} is the lower time limit of region i and T_{2i} is the upper time limit of region i.

EXAMPLE:

For the mesh variables

REGION (1 4 4 7) (6 7 4 7)

For the system variables

REGION (.0 .005) (.006 .009) (.01 .0113)

CARD 8 CTT - A table of times for the CT-type comparisons. At most, five comparisons may be made. These times must be in the range of times selected by the TIMES card.

CTT T_{11} T_{21} ··· T_{1n} T_{2n}

where T_{1i} and T_{2i} are the times for the i^{th} comparison.

EXAMPLE: CTT 0.1 0.9 0.2 0.8 0.5 0.6

CARD 9 CPT - A table of times for the CP-type comparisons. The CPT card is used only for mesh type variables. At most, five comparisons may be made. These times must be in the range of times selected by the TIMES card.

CPT T_{11} T_{21} ··· T_{1n} T_{2n}

where T_{1i} and T_{2i} are the i^{th} times for problems 1 and 2 respectively.

EXAMPLE: CPT .775 .5 .875 .75 .975 .9

CARD 10 GRID - Parameters necessary to define the desired grid for a plot. A frame is the entire viewing surface. A window is that part of the frame used for a picture, including the grid labels. A grid is that part of the window used for drawing a picture excluding the grid labels. The length of a side of the frame is 1. The (0,0) point of the frame is the lower left corner. The grid label character size is

.012 × .015. The grid variable name character size is .018 × .021.

FHC - First horizontal coordinate of the window. Default is .12. $0 \leq FHC \leq .75$

FVC - First vertical coordinate of the window. Default is .12. $0 \leq FVC \leq .75$

<u>NOTE</u>: The point (FHC,FVC) in the frame is the lower left corner of the window.

FPFH - The fractional part of the frame used as the horizontal boundary of the window. Default is .85. $0 \leq FPFH \leq 1$.

FPFV - The fractional part of the frame used as the vertical boundary of the window. Default is .85. $0 \leq FPFV \leq 1$.

<u>NOTE</u>: $FHC + FPFH \leq 1$ and $FHV + FPFV \leq 1$.

UOX - A range of the x-axis values to be used as a unit of the x-axis. A reasonable value will be chosen for a default.

FPUX - The fractional part of the x-axis to be used for a unit of x. A reasonable value will be chosen for a default.

UOY - A range of the y-axis values to be used as a unit of the y-axis. A reasonable value will be chosen for a default.

FPUY - The fractional part of the y-axis to be used for a unit of y. A reasonable value will be chosen for a default.

EXAMPLE: GRID FHC .25 FVC .25 FPFH .5 FPFV .5 &
UOX .2 FPUX .1 UOY .5 FPUY .2

CARD 11 VIEW - Data needed to control the view of the perspective plots. IJC must be the last item on the card.

R - The distance from the center of the box to the eye. The minimum radius is

$$rm = (xl^2 + yl^2 + zl^2)^{(1/2)} + \min0\ (xl, yl, zl)/2 \ ,$$

where xl, yl, and zl are the lengths of the edges of the box. The default is 3*rm.

HA - The horizontal angle. The origin is considered to be the center of the box. $0 \leq HA \leq 360$ rotated counter-clockwise from the positive x-axis.

DHA - The HA increment. DHA is used for multiple views of the same picture.

THA - The horizontal angle used to terminate rotation.

VA - The vertical angle. $|VA| \leq 90$ degrees.

$VA < 0$ - The perspective is viewed from below the origin.

$VA > 0$ - The perspective is viewed from above the origin.

DVA - The VA increments. DVA is used for multiple views of the same picture.

TVA - The vertical angle used to terminate rotation.

RZA - The ratio of the axial length to the radial length. The default is

RZA = yl/xl

RFA - The ratio of the function to the axial length. The default is

RFA = zl/xl

REP - The number of times to repeat a frame. The default is 0.

IJC - The color region definition fields. A maximum of 20 regions may be given. Each region has five fields as defined below:

- N - The color array index. N = 0 terminates the entries in IJC.
- IL - The left radial cell number.
- IR - The right radial cell number.
- JB - The lower axial cell number.
- JT - The upper axial cell number.

EXAMPLE:

VIEW HA 225 DHA 0 THA 225 VA 15 DVA 0 TVA 15 RZA .5 RFA 2 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7

CARD 12 COLOR - The colors to be used for plotting. A maximum of 15 colors may be given. The elements of the COLOR array may be used differently for the different type of plots. There are 60 distinct colors to choose from. The colors are defined as real numbers below. Intermediate colors may be chosen by linear interpolation. The basic colors are

0.0 The background color as defined by your system.

0.5 Red

1.5 Yellow

2.0 Green

2.5 Cyan (Blueish green)

3.0 Blue

3.5 Magenta (Reddish purple)

4.0 Pale red

4.5 Red

EXAMPLE: COLOR 2.5 2.0 1.5 3.0 0.0 0.5

CARD 13 MULTV - A group of variables to be plotted on the same frame. Each group of variables is enclosed in parentheses. MULTV applies only to system variables. Variables names with a minus sign (-) preceding them are to be plotted using the right scale. All others are to be plotted using the left scale. At most, 10 variables may be listed on a MULTV card. That is, the sum of the number of variables in each group must be less than or equal to 10. Example:

MULTV (V_1 V_2 $-V_3$ $-V_4$)

where V_1, V_2, V_3, and V_4 are variable names. V_1 and V_2 will be plotted using the left scale. V_3 and V_4 will be plotted using the right scale.

EXAMPLE: MULTV (GENT BETA AMP -RHO) (OM -RHO)

CARD 14 NEWV - Calculate a new variable from the variables from this problem. The card is of the form

NEWV $*V_1$ $/V_2$ $+V_3$ $-V_4$ $\cdots$ $/V_7$ $*V_8$ $+V_9$ $-V_{10}$

Other operations are RECP (reciprocal), SQRT (square root), SQ (square), and LOGN (logarithm base e). At most, 10 operations may be given. A V may be the same as any other V. All Vs must be from the same problem. V may be a system variable of any type, a mesh variable of the current type, VOL (the volume of the current cell), AREA (the area of the current cell), or a constant (C). All operations are executed from left to right and operate on the results of all previous operations. NEWV is for mesh variables only.

EXAMPLE:

NEWV $+V_1$ $+V_2$ RECP $*V_3$ LOGN $/V_4$

would yield, in FORTRAN notation

$ALOG(V_3/(V_1+V_2))/V_4$

CARD 15 LABNV - A 40-character label for the new variable.

CARD 16 LABIV - A 40-character label for the integrated variable or the rate variable.

CARD 17 RATE - Generates a 2-d plot using the ratio of the difference of VARA at successive time steps and the difference of VARB at successive time steps vs time. The RATE function is for system variables only. The form for the rate card is:

RATE VARA VARB

where VARA is any system variable and VARB is any system variable or TIME if the difference in time values is desired. When TIME is selected as VARB, the difference between successive interpolated time steps is used. If VARB = NONE, the denominator is set to the constant 1. The equation for the rate is

$$RATE = (VARA_{t_n} - VARA_{t_{n-1}})/(VARB_{t_n} - VARB_{t_{n-1}})$$

EXAMPLE: RATE SMS1 SMS2

CARD 18 FILE - The name of files to be used for this type. The names are blank delimited and of eight or fewer characters. The file names must be in the same order as the variable names listed on the TYPE card and ordered by problem number. The character "/" terminates the list of files for a problem. The files for each type are listed below, ordered by type, using default names.

TYPE 1 - BASE
2 - HYDIk k=1, 2
3 - BASE
4 - BASE
5 - REACTmk, m=1, 2, ... ,6 and k=1, 2
6 - DENmk, TEMPmk, POWmk, SPOWk, m=1,2,...,5 and k=1,2. This group is order by m and then by k.
7 - No files.
8 - FLUXmk - m=1,2,...,IGM, IGM < 20, k=1,2
9 - BASE
10 - No files.
11 - No files.
12 - No files.

13 - BASE

14 to 26 - File names resemble variable names. See Appendix F for the names.

EXAMPLE: FILE RSF11 RSF21 RLF11 RLF21 RL2F1 RL3F1 RL7F1 & ULF1 VLF1 UO2F1

<u>NOTE</u>: Note the correlation between these file names and the variable names on the TYPE card example.

CARD 19 INTEG - Integrate variables of this type. The card is of the form

INTEG W VAR MP GT VN GR E VALUE TYPE (N) REGIONS (IL IU JL JU)

where

W VAR - VAR is a variable from this problem or a constant to be used as W in the integration. If VAR is a mesh variable and not of the current type, the dimensions of the mesh for VAR must be the same as the dimensions of the mesh for the variables of the current type. If VAR is NEWV, it must be the new variable calculated for this type. If W VAR is not given, VAR will default to the constant 1.0.

MP GT VN GR - Requests variable VN be plotted on the same frame as the integration results. GT is the group type for the variable name VN. GR is the grid type for variable VN. For values of GR see GRID i below.

E VALUE - VALUE is the value to be used as the minimum value for the absolute value of the variables. The default is 1.0e-20.

GRID i - The type of grid for the results of this integration. Only one type of grid is allowed for each INTEG card.

i = 1 - linear.
  2 - logarithmic.
  3 - minimum to -1, logarithmic. -1 to maximum, linear.
  4 - minimum to 1, linear. 1 to maximum, logarithmic.
  5 - minimum to -1, logarithmic. -1 to +1, linear. +1 to maximum, logarithmic.

TYPE (N) - The type of integration to be performed. If TYPE (N) is not given, the default integration is type 1. Five types of integrations may be requested. The sums loop first over i and then over j. The types of integration are:

1) (SUM (VAR(i,j)*W(i,j)*VOL(i,j)))/(SUM (W(i,j)*VOL(i,j)))
2) (SUM (VAR(i,j)-W(i,j))*VOL(i,j))/(SUM VOL(i,j))
3) SUM (VAR(I,J)**2*W(I,J)*VOL(I,J))
4) (SUM (VAR(I,J)**2*W(I,J)*VOL(I,J)))/(SUM (W(I,J)*VOL(I,J)))
5) SUM (VAR(I,J)*W(I,J) > 0)*VOL(I,J)
6) (SUM (VAR(I,J)*W(I,J) > 0)*VOL(I,J))/(SUM (W(I,J)*VOL(I,J)))
7) SUM (VAR(I,J)*W(I,J) < 0)*VOL(I,J)
8) (SUM (VAR(I,J)*W(I,J) < 0)*VOL(I,J))/(SUM (W(I,J)*VOL(I,J)))
9) SUM (VAR(i,j)*VOL(i,j))
10) Maximum of the VAR(i,j)
11) SUM(VAR(i,j)*VOL(i,j))/SUM(W(i,j)*VOL(i,j))

REGIONS (IL IU JL JU) - The regions to be integrated. Five sets of regions may be given. Ten regions may be given for each set. For each region in a set, the integration is performed. Then the integration is performed over the entire set of regions. Each set of regions is

enclosed by parentheses. The boundaries of the regions are:

IL - The left radial cell.
IU - The right radial cell.
JL - The lower axial cell.
JU - The upper axial cell.

EXAMPLE: INTEG W RHO E 1.e-10 TYPE (2 4 6) &
REGIONS (1 4 4 7) (6 7 4 7)

CARD 20 CELL - Trace the variables of this type for the requested cells over time. The card is of the form

CELL I_1 J_1 I_2 J_2 $\cdots$ I_n J_n

where $n \leq 10$ and each item is blank delimited.

1) For each pair (I,J) where $I > 0$, $J = 0$, using the next pair (M,N), trace the cells (M+k-1,N) until $k = I$, $k = 1, 2, \cdots, I$. $I \leq 10$. A perspective plot is made.

2) For each pair (I,J) where $I = 0$, $J > 0$, using the next pair (M,N), trace the cells (M,N+k-1) until $k = J$, $k = 1, 2, \cdots, J$. $J \leq 10$. A perspective plot is made.

3) For each pair (I,J) that does not satisfy conditions 1 and 2 above, a 2-d plot is made. All pairs are plotted on the same graph.

EXAMPLE: CELL 4 0 3 4 2 6 0 4 1 6

CARD 21 SIDBAR - Plot a bar representing a variable at a given time at the right side of the perspective plot of a variable of this type. The card format is

SIDBAR TYPE VAR GRID

Where the fields are

1) TYPE - The type of the variable to be used.

2) VAR - The name of the variable to be used.

3) GRID - The type of grid to be used. Where GRID is one of the integers described below.

- 1 - Scale the grid linearly
- 2 - Scale the grid logarithmically
- 3 - minimum to -1, use a log scale. -1 to maximum, use a linear scale. minimum > -99.999, maximum < 4.
- -3 - minimum to 1, use a linear scale. 1 to maximum, use a log scale. minimum > -4, maximum < 99.999

EXAMPLE: SIDBAR 1 RHO 3

CARD 22 MESH - Table to define regions of the reactor mesh for drawing a 2-d picture of the reactor. Variables for each region are enclosed in parentheses. At most, 50 regions may be defined. The card format is

MESH (I_{11} I_{21} J_{11} J_{21} IRC_1) ··· (I_{1n} I_{2n} J_{1n} J_{2n} IRC_n)

1) I_{1j} - The left boundary cell. First I_{1j} = 0 stops the reading of data. I_{1j} < 0 draw the left boundary of the cell.

2) I_{2j} - The right boundary cell. I_{2j} < 0 draw the right boundary of the cell.

3) J_{1j} - The bottom boundary cell. J_{1j} < 0 draw the bottom boundary Of the cell.

4) J_{2j} - The top boundary cell. $J_{2j} < 0$ draws the top boundary.

5) IRC_j - $IRC_j \leq 15$. Index to the color array.

EXAMPLE: MESH (-1 -4 -3 -7 2) (-5 -5 -1 -9 3) (-6 -7 -3 -7 2) &
(-8 -9 -1 -9 3) (-1 -4 -1 -2 4) (-6 -7 -1 -2 4) &
(-1 -4 -8 -9 5) (-6 -7 -8 -9 5)

APPENDIX C

ROUTINES FOR THE T6P POSTPROCESSOR

The routines of the T6P processor are listed below. These routines are listed in the same order as they occur in the T6P program file. The routines depend on the plot utility library described in Appendix E. A system graphics library and a general system library are also needed.

PROGRAM T6POST(TAPE5=1024,TAPE6=1024,TTY,TAPE59=TTY,TAPE63=1024) : A routine to control the processing of a SIMMER-II TAPE6.

SUBROUTINE XYV(X,Y,IM,JM,CCX,CCY,DX,DY,V,IFLG) : A routine to calculate mesh dependent variables.

SUBROUTINE FASS(IF,NAME,BUF,IBUF,IT,ICON) : A routine to assign the file NAME to unit number IF.

SUBROUTINE FREAD(IF,NAME,BUF,NWDS,IT,ICON,IHFW,IHN) : A routine to read NWDS words from the next record on the file IF. $0 < IF \leq 99$.

SUBROUTINE INCFN(FN) : A routine to find the next family member of a file set by incrementing the last character of the file name FN.

FUNCTION ITTCAL(IR,T,KP) : A function to calculate the location of the time T in the TTAB table

SUBROUTINE FSEP(IN,OUT,N) : A routine to separate the array IN into N fields of ten or fewer characters and store them into the array OUT, left justified and blank filled.

FUNCTION IRJZF(I) : A routine to right justify and blank fill a ten-character left-justified and blank-filled field.

FUNCTION ILJZF(I) : A routine to left justify and blank fill a ten-character right-justified and blank-filled field.

SUBROUTINE SEPFL(IN,N,IO,NO) : A routine to separate N words in array IN into NO fields of at most ten characters each, right justified and blank filled, and stored into array IO.

SUBROUTINE DNBDF(IN,M,N,IO,L) : A routine to discard the first N blank-delimited fields from M words of characters, keeping the remaining characters intact left justified. The first blank field stops the transfer from IN to IO.

FUNCTION ILCP(IV) : A function to determine if the last character of IV is the character).

FUNCTION IFCP(IV) : A function to determine if the first character of IV is the character).

SUBROUTINE SETI(I,N,IS) : A routine to set the N words of array I to IS.

SUBROUTINE SETC(CA,N,CS) : A routine to set the N character fields of CA to CS.

SUBROUTINE LCSETI(I,N,IS) : A routine to set the N words of array I to IS.

FUNCTION RCONV(IR) : A function to convert IR, a variable of ten or fewer characters, to a real number using E10.0.

FUNCTION ICONV(I) : A function to convert I, a variable of ten or fewer characters, to an integer using I10($I < 2^{36}$).

FUNCTION IFNUM(I) : A function to test I, a variable of ten or fewer characters, for non-numeric characters.

FUNCTION IVARN(IV,IT) : A function to extract the variable number for the variable IV of type IT.

SUBROUTINE BKOPF(IOP,NOP) : A routine to interpret the operation field IOP for generating a new variable.

SUBROUTINE UNQFIL(NBASE) : A routine to generate a unique file name.

SUBROUTINE RVC : A routine to read the variable input cards.

SUBROUTINE READTP : A routine to read a SIMMER-II TAPE6.

SUBROUTINE T1SAV : A routine to save the data from a record of type 1.

SUBROUTINE T2SAV : A routine to save the data from a record of type 2.

SUBROUTINE T3SAV : A routine to save the data from a record of type 3.

SUBROUTINE T4SAV : A routine to save the data from a record of type 4.

SUBROUTINE T5SAV : A routine to save the data from a record of type 5.

SUBROUTINE T6SAV : A routine to save the data from a record of type 6.

SUBROUTINE T7SAV : A routine to save the data from a record of type 7.

SUBROUTINE T8SAV : A routine to save the data from a record of type 8.

SUBROUTINE T9SAV : A routine to save the data from a record of type 9.

SUBROUTINE T13SAV : A routine to save the data from a record of type 13.

SUBROUTINE TFSAV : A routine to save the data from a record of type 14 to 26.

SUBROUTINE RT36 : A routine to read the SIMMER-II TAPE36 and save the data for types 14 to 26.

SUBROUTINE UNSTF(IF,NRPS,NRS,IS,IFO,N,IO,NB,IB,NR,MACH) : A routine to unpack the data from a SIMMER-II packed TAPE36.

FUNCTION XOR(I,J) A routine to calculate the exclusive OR of I and J.

SUBROUTINE ECRD(SM,A,N,IER) : The routine has an entry point ECWR. Routines to transfer data to/from array SM from/to array A.

SUBROUTINE LMCMR(ARRAY,ILSUB,RARRAY,NW) : A routine to convert 60-bit LTSS data to 64-bit CTSS data.

SUBROUTINE LMCMI(ARRAY,ILSUB,IARRAY,NW) : A routine to convert LTSS integer data to CTSS integer data.

SUBROUTINE LCMASQ(FNO,NAME,BUF,LBUF,IER) : A routine to assign a LTSS monitor file to a CTSS unit number.

SUBROUTINE LMCMC(IARR,ILS,CARR,NW,NCPW) : A routine to convert LTSS 6-bit ASCII data to CTSS 8-bit ASCII data.

SUBROUTINE LMCMM(BUF,WORD,NW,PRF,IERR) : A routine to transfer the next NW 60-bit words from the current file to the 64-bit array WORD.

SUBROUTINE LCMREW : A routine to rewind the current file.

SUBROUTINE PLOTC : A routine to control the plotting.

SUBROUTINE SETTT(IT,INTF) : A routine to calculate TT, a table of times, from the TIMES or TTAB tables.

SUBROUTINE TTIVAR(ITS,IT,IR) : A routine to recreate the next interpolated time from TTAB.

FUNCTION IFTP(T,ITY,IR) : A routine to find the pointer to the TTAB table for type ITY and problem IR time T such that
TTAB(IFTP,ITY,IR) $\leq$ T $<$ TTAB(IFTP+1,ITY,IR).

SUBROUTINE CNVPP(X,Y,I,J) : A routine to convert the point (X,Y) to the grid coordinate (I,J)

SUBROUTINE SBASE : A routine to save the BASE common block.

SUBROUTINE RDBAS(NB) : A routine to read the BASE file for either of two problems.

SUBROUTINE RDVIE(II) : A routine to read data for the VIEW array.

SUBROUTINE INTV(IT,JV,VI,VO,VMI,VMX,INTF) : A routine to interpolate variables for a given set of times.

SUBROUTINE INTVST (IT,JV,VO,TSV,IR) : A routine to interpolate the variable JV from type IT for the time TSV.

SUBROUTINE VINT(IR,ITY1,ITY2,JV,W) : A routine to interpolate the variable JV from type ITY1 using the time table from ITY2. ITY1 is equal to 1, 2, or 9.

SUBROUTINE FETLAB(ITY,IV,IL,NCL) : A routine to fetch the label for the type ITY variable IV.

SUBROUTINE LPIC(IB,JB,T,FM) : A routine to label a picture.

SUBROUTINE CHARCO(NW,LAB,NCL) : A routine to count the characters in LAB, discarding the leading and trailing blanks.

SUBROUTINE DMESH(IM,JM,X,Y,ITYP,MES,NAM) : A routine to draw a 2-d diagram of a mesh by regions.

SUBROUTINE GITV(IT,IFDO,NFN,MIJRS) : A routine to generate a interpolated variable data file. Data is ordered by variable and then problem.

SUBROUTINE GNV(ITY,IFN,IFDO,NVF,VOL,AREA,MIJRS,NBLOCK) : A routine to generate the new variables for type ITY.

SUBROUTINE CONTUR(ITY,IVFN,NVFN,X,Y,IRZ,LAB) : A routine to plot the contours for the variables of type ITY.

SUBROUTINE INTEG(ITY,IVNF,IFDO,NVFN,IRS,VOL,LAB,INGRAT,FGRAT) : A routine to do the integrations over defined regions for the variables of type ITY.

SUBROUTINE CELTR(ITY,IVNF,IFDO,NVFN,IRS,LAB) : A routine to trace a cell over the defined regions for variables of type ITY.

SUBROUTINE PIMV(ITY,LAB,IFDO,IVFN,NVFN,IJRS,X,Y) : A routine to plot the individual mesh variables of type ITY.

SUBROUTINE PCV(ITY,LAB,IVFN,NVFN,ISZR,XX,YY,NBLOCK) : A routine to make comparison plots between mesh variables.

SUBROUTINE INGRAT(ASUM,DUMMY,AVAR,C,AVOL,EPS,JTYPE) : A routine to calculate the functions for the INTEG routine.

SUBROUTINE FGRAT(ASUM,DUMMY,ITYPE) : A routine to complete the integration for routine INTEG.

SUBROUTINE MSHPLT(ITY,LAB) : A routine to control the plotting of the mesh type variables.

SUBROUTINE RATE(ITY,IVARI1,IVART1,IVARI2,IVART2) : A routine to calculate the rate of change between two variables.

SUBROUTINE T1P : A routine to plot the system type variables.

SUBROUTINE T4P : A routine to draw a mesh diagram.

SUBROUTINE T5P : A routine to plot the mesh type variables.

SUBROUTINE KSADDR(IFW,ILW,LAR) : A routine to calculate the number of words between two variables (inclusively).

FUNCTION IDONEQ(IUN) : A routine to test completion of the I/O operation on unit IUN.

APPENDIX D

THE PLOTTING UTILITY LIBRARY

The Plot Utility Library was developed at the Los Alamos National Laboratory.

The library was programmed entirely in ANSI X3.9-1978 FORTRAN with the full !anguage set.

Features of the Control Data Corporation (CDC) utility UPDATE were used extensively throught out the library. A user must have access to UPDATE or HISTORIAN by Opcode, Inc. to use the library.

The subroutines COLOR, WLCH, PLT, DRV, and ADV were developed using the CGS graphic system at Los Alamos National Laboratory and are dependent on the grahics system at the user's computer center. The routines WLCH, PLT, DRV, and ADV will have to be reprogrammed. The routine COLOR calls a routine GPALET with one argument N, $1 \leq N \leq 60$. GPALET gives access to a palette of 60 colors. The palette of colors is cyclic and varies from red to orange to yellow to green to blue to purple and back to red. The user should furnish a routine equivalent to GPALET compatible with his graphics system.

The functions SHIFT, AND, OR, and COMPL are dependent on the user's compiler. These are functions for the CDC FTN compiler. The Cray CFT compiler will accept them. Users of any other system will have to furnish them.

This library uses a plotting frame with a grid of 1024 × 1024 addressable points. The point (0,0) is the upper left corner of the frame. All grid coordinates (IX,IY) must satisfy the conditions

$$0 \leq IX < 1024 \text{ and}$$
$$0 \leq IY < 1024.$$

Only named common blocks are used in the plotting utility library. The common blocks PICTC and DLXY are used internally in the subroutines PICTUR, INSECT, and DCDVR and should not concern the user.

The named common blocks CDR, RCDR, PICBOU, CJE07, and UNIT contain variables that the user will have to initialize. These common blocks are

1) COMMON/CDR/LBUF(5, 2000)

LBUF is a table of vector coordinates used when NPASS is defined for PICTUR. The second dimension of LBUF should be modified as defined in the writeup for PICTUR.

2) COMMON/RBUF/NBUF,IBUF,NSKIP,NP(13)

NBUF contains the value of the second dimension of LBUF in the named common block CDR. NBUF has to be initialized by the user. The variables IBUF, NSKIP, and NP are used internally by the subroutines PICTUR, CDRV, DCDRV, and CADV and must not be changed by the user.

3) COMMON/PICBOU/FXL, FXR, FYB, FYT

These variables will have to be initialized by the user before every call to the subroutines PICCON and PICTUR. All of these variables are real numbers.

FXL - The left boundary position of the picture in the frame.
FXR - The right boundary position of the picture in the frame.
FYB - The bottom boundary position of the picture in the frame.
FYT - The top boundary position of the picture in the frame.

4) COMMON/CJE07/IXL,IXR,IYT,IYB,XL,XR,YT,YB

These variables are defined by routines SUGRID, MIXSC, and SPLOT. The user will have to make sure this common block contains the correct values before calling any other routine.

IXL - The left boundary position of the grid in the frame.
IXR - The right boundary position of the grid in the frame.
IYT - The top boundary position of the grid in the frame.
IYB - The bottom boundary position of the grid in the frame.
XL - The minimum value for the x axis.
XR - The maximum value for the x axis.
YT - The maximum value for the y axis.
YB - The minimum value for the y axis.

5) COMMON/UNIT/NOUT

The user _must_ define NOUT before using the plotting utilities. NOUT is the unit number of the output file used for printed material.

To compile the code, one of the following options must be chosen during the preprocessing stage. The options are

IBM - The computer is an IBM, DEC-11, VAX, AMDHAL, or FACOM.

FTN - The computer is a CDC 7600 using the LTSS system with a Los Alamos FTN compiler.

CFT - The computer is a Cray-1 using the CFT compiler.

DBL - Double precision is used.

SUBROUTINE SUGRID(NOPB,NOPT,NOPL,NOPR,XB,XT,YL,YR,NXY,XYR)

A routine to set up a grid for multiple plots per frame. NOPB, NOPT, NOPL, and NOPR define the type of grid desired for the bottom, top, left, and right grid. For NOPB, NOPT, NOPL, or NOPR < 0 or > 2 draw the same grid that was used on the opposite side but do not label the grid. For NOPB, NOPT, NOPL, or NOPR

= 0 - Do not draw or label the grid.
= 1 - Draw a linear grid.
= 2 - Draw a logarithmic grid.

XB - The range of the X coordinate for the scale on the bottom.
XT - The range of the X coordinate for the scale on the top.
YL - The range of the Y coordinate for the scale on the left side.
YR - The range of the Y coordinate for the scale on the right side.
NXY - A vector defining the limit of the grid on a 1024 × 1024 grid. To use the defaults set NXY(j) < 0, j = 1, 2, 3, 4.

$$0 \leq |NXY(j)| < 1024, \ j = 1, 2, 3, \text{ or } 4$$

NXY(1) - The grid point on the x axis for the minimum X. The default is 120.
NXY(2) - The grid point on the x axis for the maximum X. The default is 980.
NXY(3) - The grid point on the y axis for the maximum Y. The default is 76.
NXY(4) - The grid point on the y axis for the minimum Y. The default is 936.

XYR - The output vector defining the range of the X and Y values used for plotting. XYR(j), j=1,2 is the adjusted range for the bottom. XYR(j), j=3,4 is the adjusted range for the top. XYR(j), j=5,6 is the adjusted range for the left side. XYR(j), j=7,8 is is the adjusted range for the right side.

SUBROUTINE MIXSC (A,N,RA,AM,ISIDE,IRL,NXY,XYR)

A routine to generate a mixed logarithmic-linear grid.

A(N) - The variable used to define the grid and scaling. A(i), depending upon the value of IRL, may be recalculated as -1.-ALOG10(ABS(A(i))) or 1.+ALOG10(A(i)).

AM(2) - AM(1) is the minimum of A(i). AM(2) is the maximum of A(i).

N - The number of elements in A.

RA(2) - The range of A(i). A(i) is modified such that RA(1) ≤ A(i) ≤ RA(2). The |RA(i)| ≤ 999.9999. If IRL = 1, RA(2) ≤ 4. If IRL = 2, RA(1) ≥ -4.

IRL - The type of scale mixture.

= 0 - Do not scale.

= 1 - Scale from the minimum of A(i) to -1. logarithmically and from -1. to the maximum of A(i) linearly.

= 2 - Scale from the minimum of A(i) to 1. linearly and from 1. to the maximum of A(i) logarithmically.

= 3 - Scale from the minimum of A(i) to -1. logarithmically, from -1. to 1. linearly, and from 1. to the maximum of A(i) logarithmically.

ISIDE - The side of the grid to be scaled.

ISIDE < 0 - Do not label the grid.

ISIDE > 0 - Label the grid.

|ISIDE| = 1 Scale the bottom side.

= 2 Scale the top side.

= 3 Scale the left side.

= 4 Scale the right side.

NXY(4) - The frame boundary points. The default is used if NXY(i) < 0.

NXY(1) = IXL, the left boundary point. The defult is 120.

NXY(2) = IXR, the right boundary point. The default is 960.

NXY(3) = IYT, the top boundary point. The default is 76.

NXY(4) = IYB, the bottom boundary point. The default is 936.

XYR(2) - The range of the scale for the axis to be drawn.

SUBROUTINE SCURV(N,X,Y,ICH,ITYPE)

A routine to draw a curve of type ITYPE.

N - The number of points on the curve Y versus X.

X - A contiguously stored array containing the X coordinate.

Y - A contiguously stored array containing the Y coordinate.

ICH - The character to be used as the plotting character. The allowable characters are 0,1,2,...,9,a,b,c,...,z,*,/ . The characters will be plotted 80 bits apart.

ITYPE - The type of curve to be drawn.

= 0 - Draw a solid curve.

= 1 - Draw a curve with one long dash and one short dash.

= 2 - Draw a curve with one long dash and two short dashes.

= 3 - Draw a curve with only short dashes.

SUBROUTINE DGRID(GRID,XM,YM,IXY)

A routine to define the grid on a plotting frame.

GRID - An array of variables used to define the desired grid on an axis. The maximum number of bits per axis is 1024. The elements of GRID are:

FHC - in GRID(1). The first horizontal coordinate of the window. Default is .75 (1 - FPFH). 0 ≤ FHC ≤ .75.

FVC - in GRID(2). The first vertical coordinate of the window. Default is .45 (1 - FPFV). 0 ≤ FVC ≤ .75.

NOTE: The point (FHC,FVC) in the frame is the lower left corner of the window.

FPFH - in GRID(3). The fractional part of the frame used as the horizontal boundary of the window. Default is 860/1024. 0 ≤ FPFH ≤ 1.

FPFV - in GRID(4). The fractional part of the frame used as the vertical boundary of the window. Default is 860/1024. 0 ≤ FPFV ≤ 1.

NOTE: FHC + FPFH ≤ 1 and FHV + FPFV ≤ 1.

UOX - in GRID(6). A range of the x axis values to be used as a unit of the x axis. A reasonable value will be chosen for the default.

FPUX - in GRID(5). The fractional part of the number of bits assigned to the x axis to be used for a unit of x. A reasonable value will be chosen for the default.

UOY - in GRID(8). A range of the y axis values to be used as a unit of the y axis. A reasonable value will be chosen for the default.

FPUY - in GRID(7). The fractional part of the number of bits assigned to the y axis to be used for a unit of y. A reasonable value will be chosen for the default.

XM(2) - The minimum, XM(1), and maximum, XM(2), of the values to be plotted on the x axis. The XM array will be redefined by DGRID to be the minimum and maximum values of the grid for the x axis.

YM(2) - The minimum, YM(1), and maximum, YM(2), of the values to be plotted on the y axis. The YM array will be redefined by DGRID to be the minimum and maximum values of the grid for the y axis.

IXY(4) - A vector defining the grid limits on a 1024 × 1024 grid. To use the defaults set IXY(i) < 0, i = 1, 2, 3, 4. IXY(i) will be redefined to the default value by DGRID if IXY(i) < 0.

IXY(1) - The left x axis grid boundary. The default is 120.

IXY(2) - The right x axis grid boundary. The default is 980.

IXY(3) - The top y axis grid boundary. The default is 76.

IXY(4) - The bottom y axis grid boundary. The default is 936.

SUBROUTINE CRSHAT(IX,IY1,IY2,NMAX,INC,ISD,IANG)

A routine to draw curve 2 and to crosshatch between curve 1 and curve 2.

$$IY1(i) \leq IY2(i) \text{ for } i=1,2,3,\ldots,NMAX$$

The coordinates IX, IY1, and IY2 are defined on a grid of 1024 × 1024 points.

INPUT:

IX - The X coordinate

IY1 - The Y coordinate of the first curve.

IY2 - The Y coordinate of the second curve.

NMAX - The number of points on the curves.

INC - The spacing between the crosshatching.

ISD - A flag to determine the type of crosshatching line.
- = 0 Use a solid line for the crosshatching.
- = 1 Use a line of dashes for the crosshatching.

IANG - The angle of the crosshatching lines measured from the positive x axis.
- = 0 Do not crosshatch (draw curve 2 only).
- = 1 Draw the crosshatching at 0° (horizontal).
- = 2 Draw the crosshatching at 45° (upward to the right).
- = 3 Draw the crosshatching at 90° (vertical).
- = 4 Draw the crosshatching at 135° (upward to the left).

SUBROUTINE XHATCH(IX1,IY1,IX2,IY2,INC,IXS,IYS,ISD,IANG)

A routine to cross hatch a rectangle. All plotting is done on a 1024 × 1024 grid. All input except IANG is determined by this grid.

INPUT:

(IX1,IY1) and (IX2,IY2) - The diagonally opposite corners of the rectangle.

INC - The spacing between the crosshatching lines.

IXS - The number of bits from the lower left corner of the rectangle in the x direction to start the crosshatching.

IYS - The number of bits from the lower left corner of the rectangle in the y direction to start the crosshatching.

ISD - A flag to determine the type of crosshatching lines.
= 0 Use a solid line for the crosshatching.
= 1 Use a line of dashes for the crosshatching.

IANG - The angle of the crosshatching line measured from the positive x axis.
= 1 Draw the crosshatching at 0° (horizontal).
= 2 Draw the crosshatching at 45° (upward to the right)
= 3 draw the crosshatching at 90° (vertical).
= 4 draw the crosshatching at 135° (upward to the left)

SUBROUTINE SPLOT(INOP,N,X,Y,ICHAR,ICON,IXY)

A routine to plot N points from the X and Y arrays on a linear-linear, a linear-logarithmic, a logarithmic-linear, or a logarithmic-logarithmic grid.

INOP - The type of grid desired. If INOP < 0, do not advance frame before plotting. If INOP > 0, advance frame before plotting.

|INOP| = 1 - Draw a linear-linear grid.

= 2 - Draw a linear-logarithmic grid.

= 3 - Draw a logarithmic-linear grid.

= 4 - Draw a logarithmic-logarithmic grid.

N - The number of points to plotted.

X - The array of N values for the X coordinate.

Y - The array of N values for the Y coordinate.

ICHAR - The plotting character to label the curve.

ICON - If ICON = 1/0, do/do not connect the points.

IXY - A vector defining the limits of the grid on a 1024 × 1024 grid. To use the defaults set IXY(j) < 0, j = 1, 2, 3, 4.

$0 \leq |IXY(j)|$ 1024, j=1, 2, 3, or 4

IXY(1) - The leftmost point on the x axis. The default is 120.

IXY(2) - The rightmost point on the x axis. The default is 980.

IXY(3) - The topmost point on the y axis. The default is 76.

IXY(4) - The bottommost point on the y axis. The default is 936.

SUBROUTINE DLN(N,KM,K,KF)

A routine to draw and/or label a linear grid.

N - The grid is to be drawn on the right, the left, the bottom, or the top. If N < 0, do not label the grid. If N > 0, label the grid.

|N| = 1 - Draw the bottom grid.

2 - Draw the top grid.

3 - Draw the left grid.

4 - Draw the right grid.

K - The number of minor tick marks to be drawn.

KM - The number of major tick marks to be drawn.

KF - The format of the scale to be used for the grid. See the routine ASCL for the definition of KF.

SUBROUTINE DLG(N)

A routine to draw and/or label a logarithmic grid.

N - The grid is to be drawn on the right, the left, the bottom, or the top. If N < 0, do not label the grid. If N > 0, label the grid.

|N| = 1 - Draw the bottom grid.

2 - Draw the top grid.

3 - Draw the left grid.

4 - Draw the right grid

SUBROUTINE DLNLN(NX,NY)

A routine to draw a linear-linear grid and put tick marks on the x and y axes.

NX - The number of tick marks on the x axis.

NY - The number of tick marks on the y axis.

SUBROUTINE DLNLG(NX)

A routine to draw a linear-logarithmic, a logarithmic-linear, or a logarithmic-logarithmic grid and put tick marks on the x and y axes.

The entries are

- DLNLG - Draw a linear-logarithmic grid.
- DLGLN - Draw a logarithmic-linear grid.
- DLGLG - Draw a logarithmic-logarithmic grid.

NX - The number of tick marks for the linear axis.

SUBROUTINE MAXV(X,IX,N,M,Y)

MAXV determines the maximum value of an array of real numbers.

INPUT:

X - The source array.

IX - The spacing of the numbers in the X array to be examined.

N - The number of values in the X array to be examined.

OUTPUT:

M - The location of the maximum value of X in the X array. M has a value from 1 to N .

Y - The maximum value of X.

ENTRY MINV(X,IX,N,M,Y)

MINV determines the minimum value of an array of real numbers.

INPUT

X - The source array.

IX - The spacing of the numbers in the X array to be examined.

N - The number of values in the X array to be examined.

OUTPUT:

M - The location of minimum value of X in the X array. M has a value from 1 to N.

Y - The minimum value of X.

SUBROUTINE ASCL(M,Z1,Z2,MAJOR,MINOR,KF)

A automatic scaling routine.

INPUT:

M - The minimum number of major intervals. M should be fairly small to avoid the overlapping of the tick mark labels.

Z1 - The minimum value of the quantities to be plotted.

Z2 - The maximum value of of the quantities to be plotted.

OUTPUT:

Z1 - The minimum value of the axis.

Z2 - The maximum value of the axis.

MAJOR - The number of major intervals. Tick marks should be positioned at each major interval.

MINOR - The number of minor intervals. MINOR is a multiple of MAJOR. Tick marks should be positioned at each minor interval.

KF - The format code describing the number of digits needed to display the tick mark labels uniquely. KF is an integer ($0 \leq KF \leq 6$ or $10 \leq KF \leq 16$) such that the units digit specifies the number of digits to be printed to the right of the decimal point. A tens digit of zero indicates a fixed point format (Fm.k). A tens digit of one indicates a floating point format (Em.k).

SUBROUTINE DGA(IX1,IX2,IY1,IY2,X1,X2,Y1,Y2)

A routine to initialize the CJE07 common block. See page D-2 for the definition of the variables in the CJE07 common block.

SUBROUTINE SLLIN(NNY,NK)

A routine to scale the left or right grid linearly.

The entries are

SLLIN - Scale the left grid.

SRLIN - Scale the right grid.

NNY - The number of major interval for the grid.

NK - The format of the scale to be used for the grid. See routine ASCL variable KF definition for the definition of NK.

SUBROUTINE SBLIN(NNX,NK)

A routine to scale the top or bottom grid linearly.

The entries are

SBLIN - Scale the bottom grid.

STLIN - Scale the top grid.

NNX - The number of major interval for the grid.

NK - The format of the scale to be used for the grid. See routine ASCL variable KF definition for the definition of NK.

SUBROUTINE SBLOG

A routine to scale a grid logarithmically.

The entries are

SBLOG - Scale the bottom grid.

STLOG - Scale the top grid.

SRLOG - Scale the right grid.

SLLOG - Scale the left grid.

SUBROUTINE LPLOT(N,X,MX,Y,MY,ICHAR,ICON)

A routine to plot N points (X(1,i),Y(1,i)), i=1,2,...,N. X and Y are dimensioned X(MX,N) and Y(MY,N). The points are connected by vectors if ICON ≠ 0. The character ICHAR is plotted at each point.

SUBROUTINE PICCON(X,Y,F,NX,NY,NFX,FM,T,FLAB,BARV,COLR,VDAT,XE,FE,E)

A routine to generate a perspective as requested by the array VDAT. If NX = 1 or NY = 1, a 2-d plot will be made. If NX = 1 and NY = 1, a plot will not be made.

The user must initialize the common block

COMMON/PICBOU/FXL,FXR,FYB,FYT

before calling the subroutine PICCON. See page D-2 for the definitions of the variables in the PICBOU common block.

INPUT:

X - The X coordinate. X is dimensioned X(NX).

Y - The Y coordinate. Y is dimensioned Y(NY).

F - The Z coordinate. F is dimensioned F(NFX,NY).

NX - The number of points to be plotted in the x direction.

NY - The number of points to be plotted in the y direction.

NFX - The value of the first index of the array F in the dimension statement of the calling program. NFX may differ from NX when the caller is not using the full array.

FM - An array of dimension 2 containing the minimum and maximum values of F.

FM(1) - The minimum value of F.

FM(2) - The maximum value of F.

T - The time of the current picture in seconds. If T < 0 the time label will not be written.

FLAB - A string of 40 characters used to label the picture.

BARV - An array containing information needed to add a bar chart of a given variable to the right side of the perspective plot of the function F.

BARV(1) - The current value of the variable.

BARV(2) - The minimum value of the variable for all times.

BARV(3) - The maximum value of the variable for all times.

BARV(4) - The type of scale to use for the bar chart.

= 0 Do not draw a bar chart.

= 1 Use a linear scale on the bar chart.

= 2 Use a logarithmic scale on the bar chart.

= 3 Use a mixed scale on the bar chart. The mixed scale is a logarithmic scale for the range of BARV(2) to -1. A linear scale for the range of -1. to the BARV(3). $-999.99 \leq$ BARV(2) and BARV(3) ≤ 4.

= -3 Use a mixed scale on the bar chart. The mixed scale is a linear scale for the range of BARV(2) to 1. A logarithmic scale for the range of 1. to the BARV(3). $-4. \leq$ BARV(2) and BARV(3) ≤ 999.99

COLR - An array of colors to paint the picture. The use of color is described in the subroutine PICTUR under COLR.

VDAT - The data describing the picture desired. VDAT is dimensioned $11 + 5 \times N$, where N is the number of regions described below under IJC.

1) R - The radius is the distance from the center of the box to the eye. The minimum radius allowed is

$$RM = ((XL^2 + YL^2 + ZL^2)^{1/2} + MIN(XL,YL,ZL)/2 ,$$

where XL, YL, and ZL are the lengths of the edges of the box. The default is $3 \times RM$.

2) HA - The horizontal angle. The origin is considered to be the center of the box. $0 \leq HA \leq 360$ rotated counter clockwise from the positive x axis.

3) VA - The vertical angle. $|VA| < 90$ degrees. If $VA < 0$ the view is from below the origin. If $VA > 0$ the view is from above the origin.

4) DHA - The increment of HA. DHA is used to rotate the picture horizontally about the origin to make multiple views of the picture.

5) DVA - The increment of VA. DVA is used to rotate the picture vertically about the origin to make multiple views of the picture.

6) THA - The horizontal angle used to terminate rotation.

7) TVA - The vertical angle used to terminate rotation.

8) RZA - The ratio of the axial length to the radial length. The default is

$$RZA = YL/XL$$

9) RFA - The ratio of the function to the radial length. The default is

$$RFA = FL/XL$$

10) IREP - The number of times to repeat a frame. The default is 0.

11-110) IJC - The colored region definition fields. At most 20 regions may be given. Each region has five fields as defined below

N - The negative of the color array index. N = 0 terminates the entries in IJC.

IL - The left radial cell number.

IR - The right radial cell number.

JB - The lower axial cell number.

JT - The upper axial cell number.

XE - A temporary array dimensioned XE(NX).

FE - A temporary array dimensioned FE(NX,NY).

E - A temporary array dimensioned E(10 × NX + 10 × NY).

LROUT - A routine supplied by the user to label the picture. FXL, FXR, FYB, and FYT from the common block PICBOU will be recalculated in this routine such that the plot will fill up all the frame not used by the labels.

<u>SUBROUTINE</u> XYZCON(HA,VA,RD,XLEN,YLEN,ZLEN,XVU,YVU,ZVU)

A routine that calculates a viewpoint. HA (the horizontal angle), VA (the verticle angle), and RD (the radius) are converted to XVU (the x coordinate of the viewpoint), YVU (the y coordinate of the viewpoint), and ZVU (the z coordinate of the viewpoint) for PICTUR. XLEN, YLEN, ZLEN are the lengths of the sides of the box of the perspective.

```
SUBROUTINE PICTUR (F,NX,NY,NXD,X,Y,XVU,YVU,ZVU,XH,YH,NS,BA
1 ,RF,SY,SZ,LHIDE,LBOX,LABEL,CHAR,NINT,LBAR,LSURF,LINE,COLR,NADV
2 ,WIN,CENTER,NPASS,NE,FMIN,FMAX,XE,YE,ZE,CE,NCH,NEC,IE,NET)
```

PICTUR was written by M. L. Pruett of the Los Alamos National Laboratory.

The PICTUR routine produces 3-d plots that can have hidden lines removed. The named common block PICBOU must be defined by the user. The variables in PICBOU in order are FXL, FXR, FYB, and FYT.

FXL is the left grid boundary bit position.
FXR is the right grid boundary bit position.
FYB is the bottom grid boundary bit position.
FYT is the top grid boundary bit position.

The defaults are for

FXR ≤ 0, FXR = 945.
FXL < 0, FXL = FXR - 935.
FYB ≤ 0, FYB = 970.
FYT < 0, FYT = FYB - 935.

The grid is a 1024 × 1024 bit grid. The point (0,0) is the top left hand corner.

INPUT:

F - The NX × NY points to be plotted. F is dimension F(NXD,NY).

NX - The number of points in the x direction to be plotted.

NY - The number of points in the y direction to be plotted.

NXD - The value of the first index of array F in the dimension statement of the calling program. NXD may differ from NX when the caller is not using the full array.

X - A vector of length NX giving the x coordinates of the mesh lines. X must be monotonically increasing.

Y - A vector of length NY giving the y coordinates of the mesh lines. Y must be monotonically increasing.

XVU, YVU, ZVU - The viewing position in the same units as X, Y, and F, respectively. The viewpoint must be outside of grid area.

XH, YH - Two temporary storage arrays of size NS. NS should be at least 5(NX+NY). If the lines run in one direction only and point away from viewer, this number should be larger, at most 2 × NX × NY.

BA - The level of the base of the box in the same units as F. If BA = RF, only the base will be plotted.

RF - The level of the roof of the box in the same units as F. If RF = BA, only the base will be plotted.

SY - A scale factor to scale Y, CENTER(2), and YVU. Multipling Y by SY will make Y appropriate to X. Normally, SY = 1.

SZ - The scale factor for F, ZVU, BA, RF, and CENTER(3). It scales F so that surface features are of appropriate size for the width of the plot. If SZ = 0, a suitable scale is chosen by PICTUR.

LHIDE - A flag to determine the type of hidden lines to be removed. If LHIDE = 0, no lines are removed. If LHIDE = 1, the hidden surface lines are removed. If LHIDE = 2, the hidden surface lines and the hidden box lines are removed.

LBOX - A flag to determine the type of box to be plotted. If LBOX = 0, the box is not plotted. If LBOX = 1, only the base of the box is plotted. If LBOX = 2, the base of the box and z axis are plotted. If LBOX = 3, the complete box is plotted. If LBOX = 4 or 5, horizontal lines will be drawn on walls of box corresponding to the tick marks (see

NINT below). If LBOX = 5, only background lines will be drawn.

LABEL - A flag to determine if the axes will be labeled. If LABEL = 0, axis labels will not be written. If LABEL = 1, the axes with visible end points will be labeled. If LABEL = 2, all axes are labeled even if the ends are invisible.

CHAR - The axis labels. CHAR is 3 character strings each with 10 characters. When CHAR(i) = 0, the i^{th} axis is labeled with the appropriate X, Y, or Z. Example - Set CHAR(2)='TIME' to label the y axis with TIME.

NINT - A set of flags to determine if the axes are to have tick marks and tick mark labels. NINT, an array of 3, gives the number of intervals for tick marks and tick labels on the x, y, and z axes. Example - If NINT(3) = 1, numbers representing the magnitude of the base and roof and tick marks are placed to the left of the box. If NINT(3) = 2, three tick marks and numbers (two intervals) are placed at left of the box. If NINT(3) = 0, tick marks will not be put on the z axis. For the x and y axes, the tick marks are provided only when the axis is in the foreground. If NINT is negative, tick marks are emplaced but the tick mark labels are not. When NINT or LABEL is nonzero, the plot is made smaller so that there is room for the numbers and/or labels.

LBAR - A flag to determine if side bars are to be put in the sides of the box. If LBAR = 0, the side bars are not plotted. If LBAR = 1, visible non-obscuring side bars are plotted. If LBAR = 2, all visible side bars are plotted. If LBAR = 3, only visible sidebars going down from the roof to the top of the surface and up from the base to the bottom of the surface are plotted.

LSURF - A flag to determine what surfaces are to be plotted. For LSURF = 1, 2, or 3, the top surface, the bottom surface, or both surfaces will be plotted.

LINE - A flag to determine which direction the lines of the surface will run. If LINE = 1, only the lines running in the x direction are plotted. If LINE = 2, only the lines running in the y direction are plotted. If LINE = 3, the lines running in both directions are plotted.

COLR - A table of colors to paint the picture. COLR is an array of 4 words. The 4 words define, respectively, the color of the side bars, the box, the top of the surface, and the underside. If NPASS (see below) is not zero, COLR is used differently. See the routine COLOR for a selection of colors.

NADV - The number of frames to be advanced after plotting. Use 0 to prevent the advancing of the frame. For making duplicate frames when NPASS is nonzero (see below for definition of NPASS), set NADV to the negative of the number of frames wanted and set NBUF sufficiently large (see NBUF below). Features such as labels are not duplicated on the duplicate frames.

WIN - Defines the window of the picture to view. WIN, in same units as X, gives the ratio of the width and the height of window, the area to be plotted. Lines outside of the window are cut off. The window is centered at the x, y, and z coordinates given by CENTER(1), CENTER(2), and CENTER(3). The window is perpendicular to the base and faces the viewer. If WIN ≤ 0, CENTER is not used. If WIN = 0., the window is made just large enough to fit the image, including the box even if LBOX = 0.

If WIN = -1., the picture will fill the frame and some distortion will occur.

NPASS - The number of passes to make to plot different parts of the surface in different colors. The parts to be plotted in each pass are defined by FMAX and FMIN or by the integers set in F, see the instructions below under NE. If NPASS = 0, FMIN and FMAX are not checked. If NPASS is not zero, a buffer LBUF of length NBUF must be supplied in the labeled common CDR. The common statements are

```
COMMON/RCDR/NBUF
COMMON/CDR/LBUF(5,the value of NBUF)
```

The suggested buffer size is 2.5 × NX × NY. If the buffer is too small, the color filters will have to be changed excessively. When NPASS is nonzero, 13 colors are possible. See the subroutine COLOR for a selection of colors.

NE - If NPASS = 0 and NE = 0, all the arguments following NE will not be used. FMAX, FMIN, and COLR are arrays of length 15. FMAX and FMIN set the limits for the portion of the surface to be plotted at a specific color. Example, if FMAX(1) = 10., FMIN(1) = 0., and COLR(1) = .5, then all portions of the surface lying between F = 0 and F = 10 will be plotted in red. When NPASS is nonzero, the colors of sidebars and box, when requested, are defined by the last two passes, respectively. If selected segments are to be plotted in different colors, use the following procedure. First set NPASS to the negative of the number of colors wanted, then set COLR to the appropriate colors. The four lowest-order bits of F(i,j) determine in which pass to plot the segment between (i-1,j) and (i,j), where i is the x coordinate index and j is

the y coordinate index. The next four lowest-order bits determine in which pass to plot the segment between (i,j-1) and (i,j). Example, COLR(3) = 1.5 will actuate the yellow filter for the third pass. All segments for which the bits of F are set to the integer 3 are colored yellow. To set the bits to 3 for both segments of the point (i,j), zero out the 8 lowest order bits by the FORTRAN statement

```
F(I,J) = AND(F(I,J), COMPL(377B))
```

Now reset the bits with the statement

```
F(I,J) = OR(OR(F(I,J), SHIFT(3,4)), 3)
```

FMAX and FMIN will also be checked when NPASS is negative. If this is not desired for a particular pass, set FMAX = 0 and FMIN = 0. Setting the last eight bits of F to 0 causes the corresponding segment to be skipped. The use of the eight low order bits of F is not recommended for the FOURB option.

EXTRA POINTS - If extra points are to be plotted, set NE to the number of extra points. XE, YE, and ZE are point coordinates, CE is the color for each extra point (set all CE = -1. for no color), and NCH is the plotting character for each extra point. XE, YE, ZE, CE, and NCH are all arrays of size NE. XE, YE, and ZE are in same units as X, Y, and F, respectively. If NE = 0, all arguments following FMAX are not used. NEC, IE, and NET are temporary arrays of a length at least the larger of NE or NX + NY which are used for the extra points. If NCH(i) = 0, a line will connect the i^{th} point to the next point provided both points are visible.

SUBROUTINE INSECT (I,X3,Y3,X4,Y4)

An internal routine used by PICTUR to calculate the intersection of two lines.

SUBROUTINE CDRV(IX1,IY1,IX2,IY2,M)

An internal routine used by PICTUR.

SUBROUTINE DCDRV

An internal routine used by PICTUR.

SUBROUTINE CADV(N)

An internal routine used by PICTUR.

SUBROUTINE COLOR(F)

A routine to convert the old Los Alamos color system to the new CGS system.

INPUT:

F - A real number selecting a color from the old Los Alamos system. Below is a table of numbers giving the correspondence between the old Los Alamos system and the new CGS system.

i	OLD	NEW	COLOR
1	0.5	1	Red
2	1.5	13	Yellow
3	2.0	22	Green
4	2.5	27	Cyan (A bluish green)
5	3.0	39	Blue
6	3.5	49	Magenta (A reddish purple)
7	4.0	54	Pale Red
8	4.5	60	Red

For a F given to the color routine such that

$$OLD_{i-1} < F < OLD_i \quad i=2,8$$

The color selected will be an N interpolated such that

$$NEW_{i-1} < N < NEW_i \quad i=2,8.$$

If $0. < F < .5$, then the color selected will be between NEW_7 and NEW_8. If $F = 0.$, then a clear, white, or black color, the inverse of the background color, will be selected.

SUBROUTINE WLCH(IX,IY,N,CS,IS)

A routine to print the N characters in the character string CS in a horizontal direction starting at the point (IX,IY). The frame has a 1024 × 1024 grid with the point (0,0) being the upper left corner of the grid. IS is the size of the characters and is defined in the table below. VERTICAL and HORIZONTAL are the number of grid positions the characters will occupy in the vertical and horizontal directions.

IS	VERTICAL	HORIZONTAL	CHARACTERS/LINE
1	12	15	85
2	18	22	56
3	24	29	42
4	30	36	34
5	36	43	28
6	42	50	24

ENTRY WLCV(IX,IY,N,CS,IS)

A routine to print the N characters in the character string CS in a vertical direction starting at the point (IX,IY). The definitions of the variables in the argument list are the same as for WLCH.

SUBROUTINE PLT(IX,IY,ICH)

A routine to plot the character ICH at the point (IX,IY). The coordinates IX and IY are for a frame with a 1024 × 1024 grid and the point (0,0) is the upper left corner of the frame.

SUBROUTINE DRV(IX1,IY1,IX2,IY2)

A routine to draw a vector from the point (IX1,IY1) to the point (IX2,IY2). The coordinates of the two points are for a frame with a 1024 × 1024 grid and the point (0,0) is the upper left corner of the frame.

SUBROUTINE ADV(N)

A routine to advance the film N frames.

FUNCTION SHIFT(I,J)

A function to shift I right or to do a left circular shift on I according to the following conditions:

J < 0 - Shift the bits of the word I to the right J places. Bits shifted out the right end of the word are lost. The empty positions on the left are filled with zeroes.

J = 0 - I is unchanged.

J > 0 - Shift the bits of the word I to the left J places. Bits shifted out the left end of the word are inserted into the empty places created at the right end of the word.

FUNCTION AND(I,J)

The Boolean "AND" operator. A bit-by-bit comparison is made of I and J. If the bits of I and J are 1, the result is 1. For all other combinations, the result is zero. For example,

```
I        = 0101110
J        = 1100101
AND(I,J) = 0100100.
```

FUNCTION OR(I,J)

The Boolean "OR" operator. A bit-by-bit comparison is made of I and J. If the bits of either I or J are 1, the result is 1. If the bits of both I and J are zero, the result is zero. For example,

```
I       = 0101110
J       = 1100101
OR(I,J) = 1101111.
```

FUNCTION COMPL(I)

The Boolean "NOT" operator. All bits of I that are equal to 1 will be set to 0 and all bits of I that are equal to 0 will be set to 1. For example,

```
I        = 0101110
COMPL(I) = 1010001.
```

APPENDIX E

DATA FILE STRUCTURE

TAPE6 from has 10 types of variables in 2 types of records. All records in TAPE6 are writter with buffering statements. The first word of the first record of each record type gives the type of variable contained in the record. The length of the first type of record, which contains only one record, depends on the type of variable. The second type of record has a variable number of records. The second word of the first record contains the number of records for this type of variable. A list of the types of variables is given below.

The first TAPE36 record gives the problem name and dimensions. If the "DBL" option is defined in SIMMER-II, the remaining records contain single-precision data for 99 variables at specified time steps. If "DBL" option is not defined, the data in the remaining records are packed 2 pieces of data per 60 bits. If the machine that generated the TAPE36 had 64 bits per word, the low-order four bits are not used. With bit 0 being the high-order bit, the data are packed as follows:

BITS	0	The sign bit of the fraction of the first number.
	1	The sign bit of the exponent of the first number.
	2-7	The exponent of the first number.
	8-29	The fraction of the first number.
	30	The sign bit of the fraction of the second number.
	31	The sign bit of the exponent of the second number.
	32-37	The exponent of the second number.
	38-59	The fraction of the second number.

<u>NOTE:</u> A) $|\text{exponent}| < 64$

B) The fraction is truncated to 22 bits.

C) $-9 \times 10^{18} < \text{Data} < 9 \times 10^{18}$ is the approximate range of each piece of data.

The logical record length LRL is $(99 \times IB \times JB + 2)/(1 + MD)$, where IB is the number of cells in the radial direction, JB is the number of cells in the axial direction, and for MD = 0/1 the SIMMER-II "DBL" options is on/off. The physical record length PRL for a TAPE36 is set at 6000 words by SIMMER-II. Thus, the logical record consists of M physical records, where

M = I[LRL/PRL] + (0/1 if R =/≠ 0) ,

and where I[x] is the integer part of x, and R is the fractional part of x.

TYPE 1: The variables of type 1 are generated by the SIMMER-II routine PKDRIV. These variables are used by the T6P from the BASE common block array T1VAR and are saved in the BASE file. The variables are

VARIABLE	DESCRIPTION
AMP	The flux amplitude.
POW	The power.
PINTG	The integrated amplitude.
RHO	The reactivity.
BETA	The effective beta.
GENT	The generation time.
OM	The inverse period.

TYPE 2: The variables of type 2 are the integrals over the fluid dynamics mesh generated in the SIMMER-II routine EDFLPR. These variables are used by the T6P from the BASE common block array T2VAR and are saved in the file HYDIi, where i is the problem number. The variables are

VARIABLE	DESCRIPTION
ALP2G	The vapor void fraction.
ALP2L	The liquid void fraction.
ALP2S	The solid void fraction.
ALPLNF	The no flow liquid void fraction.
ALPT	The total void fraction.
TGMASS	The total mass of vapor.
TLMASS	The total mass of liquid.
TSMASS	The total mass of solid.
TGMOF	The total mass of vapor outflow.
TLMOF	The total mass of liquid outflow.

MOMGR	The radial momentum of vapor.
MOMLR	The radial momentum of liquid.
MOMSR	The radial momentum of solid.
MOMGZ	The axial momentum of vapor.
MOMLZ	The axial momentum of liquid.
MOMGZ	The axial momentum of solid.
TMOMG	The total momentum of vapor.
TMOML	The total momentum of liquid.
TMOMS	The total momentum of solid.
GKE	The kinetic energy of vapor.
LKE	The kinetic energy of liquid.
SKE	The kinetic energy of solid.
GKEOF	The kinetic energy of vapor outflow.
LKEOF	The kinetic energy of liquid outflow.
TGE	The total energy of vapor.
TLE	The total energy of liquid.
TSE	The total energy of solid.
TGEOF	The total energy of vapor outflow.
TLEOF	The total energy of liquid ouflow.
TES	The total energy of the system.
TIES	The total initial energy of the system.
TESOF	The total energy of the system outflow.
TNEA	The total nuclear energy added.
DEC	The deviation from energy conservation.
TMS	The total mass of the system.
TIMS	The total initial mass of the system.
TMSOF	The total mass of the system outflow.
DMC	The deviation from mass conservation.
SMG1	The mass of fertile fuel in the vapor field.
SMG2	The mass of fissile fuel in the vapor field.
SMG3	The mass of steel in the vapor field.
SMG4	The mass of sodium in the vapor field.
SMG5	The mass of control in the vapor field.
SMG6	The mass of fission gas in the vapor field.

SML1 The mass of fertile fuel in the liquid field.

SML2 The mass of fissile fuel in the liquid field.

SML3 The mass of steel in the liquid field.

SML4 The mass of sodium in the liquid field.

SML5 The mass of control in the liquid field.

SML6 The mass of fertile fuel particles in the liquid field.

SML7 The mass of fissile fuel particles in the liquid field.

SML8 The mass of steel particles in the liquid field.

SMS1 The mass of fabricated fertile fuel in the structure field.

SMS2 The mass of fabricated fissile fuel in the structure field.

SMS3 The mass of frozen fertile fuel in the structure field.

SMS4 The mass of frozen fissile fuel in the structure field.

SMS5 The mass of cladding in the structure field.

SMS6 The mass of can wall in the structure field.

SMS7 The mass of control in the structure field.

SMS8 The mass of intragranular fission gas in the structure field.

SMS9 The mass of intergranular fission gas in the structure field.

SEG1 The energy of fertile fuel in the vapor field.

SEG2 The energy of fissile fuel in the vapor field.

SEG3 The energy of steel in the vapor field.

SEG4 The energy of sodium in the vapor field.

SEG5 The energy of control in the vapor field.

SEG6 The energy of fission gas in the vapor field.

SEL1 The energy of fertile fuel in the liquid field.

SEL2 The energy of fissile fuel in the liquid field.

SEL3 The energy of steel in the liquid field.

SEL4 The energy of sodium in the liquid field.

SEL5 The energy of control in the liquid field.

SEL6 The energy of fertile fuel particles in the liquid field.

SEL7 The energy of fissile fuel particles in the liquid field.

SEL8 The energy of steel particles in the liquid field.

SES1 The energy of fabricated fertile fuel in the structure field.

SES2	The energy of fabricated fissile fuel in the structure field.
SES3	The energy of frozen fertile fuel in the structure field.
SES4	The energy of frozen fissile fuel in the structure field.
SES5	The energy of cladding in the structure field.
SES6	The energy of can wall in the structure field.
SES7	The energy of control in the structure field.
SES8	The energy of intragranular fission gas in the structure field.
SES9	The energy of intergranular fission gas in the structure field.
SMGO1	The mass of fertile fuel outflow in the vapor field.
SMGO2	The mass of fissile fuel outflow in the vapor field.
SMGO3	The mass of steel outflow in the vapor field.
SMGO4	The mass of sodium outflow in the vapor field.
SMGO5	The mass of control outflow in the vapor field.
SMGO6	The mass of fission gas outflow in the vapor field.
SMLO1	The mass of fertile fuel outflow in the liquid field.
SMLO2	The mass of fissile fuel outflow in the liquid field.
SMLO3	The mass of steel outflow in the liquid field.
SMLO4	The mass of sodium outflow in the liquid field.
SMLO5	The mass of control outflow in the liquid field.
SMLO6	The mass of fertile fuel particles outflow in the liquid field.
SMLO7	The mass of fissile fuel particles outflow in the liquid field.
SMLO8	The mass of steel particles outflow in the liquid field.
SEGO1	The energy of fertile fuel outflow in the vapor field.
SEGO2	The energy of fissile fuel outflow in the vapor field.
SEGO3	The energy of steel outflow in the vapor field.
SEGO4	The energy of sodium outflow in the vapor field.
SEGO5	The energy of control outflow in the vapor field.
SEGO6	The energy of fission gas outflow in the vapor field.
SELO1	The energy of fertile fuel outflow in the liquid field.

SELO2	The energy of fissile fuel outflow in the liquid field.
SELO3	The energy of steel outflow in the liquid field.
SELO4	The energy of sodium outflow in the liquid field.
SELO5	The energy of control outflow in the liquid field.
SELO6	The energy of fertile fuel particles outflow in the liquid field.
SELO7	The energy of fissile fuel particles outflow in the liquid field.
SELO8	The energy of steel particles outflow in the liquid field.
CVG	The cover gas volume.
THI	The impulse at head.
PEAKP	The peak pressure.
TPEAKP	The time of peak pressure.
CPEAKP	The cell where the peak pressure occurs.
APEAKP	The average peak pressure.
ATPEAKP	The average time of peak pressure.

TYPE 4: The variables of type 4 are the spacing of the neutronics mesh. The variables are used by T6P from the BASE common block and are saved in the BASE file. The variable to use when requesting type 4 is MESH4.

TYPE 5: The variables of type 5 are the mesh variables for the reactivity contributions. These records are generated in the SIMMER-II routine INPROD. The variables are used by T6P from the files RACTij, where j is the problem number, and i is the number listed below for the different variables. The variables are

VARIABLE	FILE	DESCRIPTION
SUMALL	RACT1j	The net reactivity.
PROMPT	RACT2j	The prompt fission reactivity contribution.
DELAYED	RACT3j	The delayed fission reactivity contribution.
SCATTER	RACT4j	The neutron scatter reactivity contribution.
TOTAL	RACT5j	The total reactivity contribution.
LEAKAGE	RACT6j	The leakage reactivity contribution.

TYPE 6: The variables of type 6 are the densities and temperatures for each component of material transmitted from the SIMMER-II fluid dynamics to the neutronics and the energies transmitted from the neutronics to the fluid dynamics. These records are generated in the SIMMER-II routine POWCAL. The variables are used by T6P from the files DENij, TEMPij, POWij, and SPOWj, where i is the component number and j is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
DEN1	DEN1j	The density of fertile fuel.
DEN2	DEN2j	The density of fissile fuel.
DEN3	DEN3j	The density of steel.
DEN4	DEN4j	The density of sodium.
DEN5	DEN5j	The density of control.
TEMP1	TEMP1j	The temperature of fertile fuel.
TEMP2	TEMP2j	The temperature of fissile fuel.
TEMP3	TEMP3j	The temperature of steel.
TEMP4	TEMP4j	The temperature of sodium.
TEMP5	TEMP5j	The temperature of control.
POW1	POW1j	The power of fertile fuel.
POW2	POW2j	The power of fissile fuel.
POW3	POW3j	The power of steel.
POW4	POW4j	The power of sodium.
POW5	POW5j	The power of control.
SPOW	SPOWj	The total power.

TYPE 7: The variables of type 7 are generated by SIMMER-II only when the SIMMER-II option URANUS is defined. These variables are used by T6P from the BASE common block array T7VAR and are saved in the BASE file. The variables are

VARIABLE	DESCRIPTION
PHI	Total relative power.
POWFS	Prompt fission relative power.
POWDK	Decay heat relative power.

PINTG	Time integral of the relative power.
REAPGM	Driving reactivity.
REATOT	Total reactivity.
READOP	Doppler feedback reactivity.
REAFER	Fertile fuel displacement reactivity.
REAFIS	Fissile fuel displacement reactivity.
REASTR	Steel displacement reactivity.
REACOL	Sodium displacement reactivity.
REACTL	Control displacement reactivity.

<u>TYPE 8:</u> The variables of type 8 are the flux for IGM groups generated in the SIMMER-II routine POWCAL. The variables are used by T6P from the files FLUXij, where i is the group number and j is the problem number. The variables are FLUXi, where i is the group number.

<u>TYPE 9:</u> The variables of type 9 are integrals of the reactivity contributions generated in the SIMMER-II routine INPROD. The variables are used by T6P from the BASE common array T9VAR and are saved in the file BASE. The variables are

VARIABLE	DESCRIPTION
PFIS	The prompt fission reactivity contribution.
DFIS	The delayed fission reactivity contribution.
DSCAT	The neutron scatter reactivity contribution.
RTOTL	The total cross section reactivity contribution.
RLEAK	The leakage reactivity contribution.
TEBETA	The effective beta.
GENTIM	The generation time.
REACT	The inverse period.

<u>TYPE 13:</u> The variables of type 13 are the spacing of the fluid dynamics mesh. The variables are used by T6P from the BASE common block and are saved in the file BASE. The variable to be used when requesting type 13 is MESH13.

TYPE 14: The variables of type 14 are the 4 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
ALPG	ALGFi	The volume fraction of vapor.
ALPS	ALSFi	The volume fraction of structure.
ALPL	ALLFi	The volume fraction of liquid.
P	PFi	The pressure.

TYPE 15: The variables of type 15 are the 9 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below witn the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
RSF1	RSF1i	The total solid field density of fertile fuel.
RSF2	RSF2i	The total solid field density of fissile fuel.
RLF1	RLF1i	The total liquid field density of fertile fuel.
RLF2	RLF2i	The total liquid field density of fissile fuel.
RGFB	RGFFi	The total density of fuel vapor.
TRFU	UO2Fi	The total density of fuel.
STEE	FEFi	The total density of steel.
TRSO	NAFi	The total density of sodium.
TRCN	CTRFi	The total density of control.

TYPE 16: The variables of type 16 are the 10 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
RSB1	RS1Fi	The density of solid fabricated fertile fuel.

RSB2	RS2Fi	The density of solid fabricated fissile fuel.
RSB3	RS3Fi	The density of frozen fertile fuel.
RSB4	RS4Fi	The density of frozen fissile fuel.
RLB1	RL1Fi	The density of liquid fertile fuel.
RLB2	RL2Fi	The density of liquid fissile fuel.
RGB1	RG1Fi	The density of fertile fuel vapor.
RGB2	RG2Fi	The density of fissile fuel vapor.
RLB6	RL6Fi	The density of fertile fuel particles in liquid.
RLB7	RL7Fi	The density of fissile fuel particles in liquid.

TYPE 17: The variables of type 17 are the 9 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
TS1	TS1Fi	The temperature of solid fabricated fuel.
TS2	TS2Fi	The temperature of frozen fuel.
TS3	TS3Fi	The temperature of cladding.
TS4	TS4Fi	The temperature of subassembly can wall.
TS5	TS5Fi	The temperature of solid control.
TSAF	TSTFi	The fuel saturation temperature.
TSAS	TSTSi	The steel saturation temperature.
TSAN	TSTNi	The sodium saturation temperature.
TSAC	TSTCi	The control saturation temperature.

TYPE 18: The variables of type 18 are the 8 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
RSB5	RS5Fi	The density of cladding.
RSB6	RS6Fi	The density of subassembly can wall.

RSB7	RS7Fi	The density of solid control.
RLB3	RL3Fi	The density of liquid steel.
RLB5	RL5Fi	The density of liquid control.
RLB8	RL8Fi	The density of steel particles in liquid.
RGB3	RG3Fi	The density of steel vapor.
RGB5	RG5Fi	The density of control vapor.

TYPE 19: The variables of type 19 are the 5 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
RSB8	RS8Fi	The density of intragranular fission gas.
RSB9	RS9Fi	The density of intergranular fission gas.
RLB4	RL4Fi	The density of liquid sodium.
RGB4	RG4Fi	The density of sodium vapor.
RGB6	RG6Fi	The density of fission gas in the vapor field.

TYPE 20: The variables of type 20 are the 8 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
TL1	TL1Fi	The temperature of liquid fuel.
TL2	TL2Fi	The temperature of liquid steel.
TL3	TL3Fi	The temperature of liquid sodium.
TL4	TL4Fi	The temperature of liquid control.
TL5	TL5Fi	The temperature of fuel particles in liquid.
TL6	TL6Fi	The temperature of steel particles in liquid.
TG	TGFi	The temperature of vapor mixture.
TVN	TVNFi	The temperature of vapor field(TGN).

TYPE 21: The variables of type 21 are the 7 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
DH	DHFi	The hydraulic diameter of the cell.
RPFS	RPSFi	The solid fuel particle radius in liquid field.
RPSS	RPSSi	The solid steel particle radius in liquid field.
RPLF	RPLFi	The droplet radius for liquid fuel.
RPLS	RPLSi	The droplet radius for liquid steel.
RPLN	RPLNi	The droplet radius for liquid sodium.
RPLC	RPLCi	The droplet radius for liquid control.

TYPE 22: The variables of type 22 are the 9 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
QG	QGFi	The total energy transfer to vapor.
EN1	EN1Fi	The fertile fuel internal energy rate.
EN2	EN2Fi	The fissile fuel internal energy rate.
EN3	EN3Fi	The steel internal energy rate.
EN4	EN4Fi	The sodium internal energy rate.
EN5	EN5Fi	The control internal energy rate.
KIJ	KIJFi	The liquid to vapor momentum exchange.
FIJL	FIJLi	The liquid to structure momentum exchange.
FIJG	FIJGi	The vapor to structure momentum exchange.

TYPE 23: The variables of type 23 are the 9 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
FS1	FS1Fi	The mass transfer of fabricated fuel to liquid.
FS2	FS2Fi	The mass transfer of frozen fuel to liquid.
FS3	FS3Fi	The mass transfer of cladding to liquid.
FS4	FS4Fi	The mass transfer of can wall to liquid.
FS5	FS5Fi	The mass transfer of control to liquid.
FS6	FS6Fi	The mass transfer of intragranular fission gas to intergranular fission gas in steel.
FS7	FS7Fi	The mass transfer of fission gas in fuel to vapor.
FL1	FL1Fi	The mass transfer of fuel particles to liquid.
FL2	FL2Fi	The mass transfer of steel particles to liquid.

TYPE 24: The variables of type 24 are the 6 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
QL1	QL1Fi	The total energy transfer to liquid fuel.
QL2	QL2Fi	The total energy transfer to liquid steel.
QL3	QL3Fi	The total energy transfer to liquid sodium.
QL4	QL4Fi	The total energy transfer to liquid control.
QL5	QL5Fi	The total energy transfer to fuel particles in the liquid field.
QL6	QL6Fi	The total energy transfer to steel particles in the liquid field.

TYPE 25: The variables of type 25 are the 5 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE FILE DESCRIPTION

QS1	QS1Fi	The total energy transfer to fabricated fuel.
QS2	QS2Fi	The total energy transfer to frozen fuel.
QS3	QS3Fi	The total energy transfer to cladding.
QS4	QS4Fi	The total energy transfer to can wall.
QS5	QS5Fi	The total energy transfer to solid control.

TYPE 26: The variables of type 26 are the 10 variables defined for each cell of the fluid dynamics mesh. The variables are used by T6P from the variable files listed below with the variables. The file names will be listed with the last character an i, where i is the problem number. The variables are

VARIABLE	FILE	DESCRIPTION
UG	UGFi	The radial velocity of vapor.
UL	ULFi	The radial velocity of liquid.
VG	VGFi	The axial velocity of vapor.
VL	VLFi	The axial velocity of liquid.
VLMG	VLGNi	The liquid to vapor velocity difference.
GAM1	GC1Fi	The mass transfer of fuel vapor to liquid.
GAM2	GC2Fi	The mass transfer of steel vapor to liquid.
GAM3	GC3Fi	The mass transfer of sodium vapor to liquid.
GAM4	GC4Fi	The mass transfer of control vapor to liquid.
FAL3	FAILi	The can wall failure.

APPENDIX F

COMMON VARIABLES FOR THE T6P POSTPROCESSOR

A list of all the common variables in the T6P postprocessor program are given below. The variables are ordered alphabetically. The name of the common block where the variable resides and a description of the variable are also given.

FORTRAN variables	Common blocks	Description of the variables
AA1	KLC	A buffer used to write TAPE36 variables to a direct access disk file.
AREAF	BASE	The area of a fluid dynamic cell.
AREAN	BASE	The area of a neutronic cell.
BUF	BLANK	A large buffer used for I/O.
CCXF	BASE	The cell centered radial coordinate for the fluid dynamic mesh.
CCXN	BASE	The cell centered radial coordinate for the neutronic mesh.
CCYF	BASE	The cell centered axial coordinate for the fluid dynamic mesh.
CCYN	BASE	The cell centered axial coordinate for the neutronic mesh.

COLR	INPUT1	The input array to define the colors for plotting.
CONST	INPUT	The array of constants to be used in the integrations.
CP	INPUT	An input array requesting a comparison between problems for a given type of variable.
CPC36	INPUT1	The number of the problem for the current TAPE36.
CPC6	INPUT1	The number of the problem for the current TAPE6.
CPC	RTPC	The current problem number.
CPT	INPUT	A table of times for the comparisons requested by CP.
CT	INPUT	An input array requesting comparisons of a variable at the times specified on the CTT card.
CTSBUF	BLANK	A buffer used for I/O.
CTT	INPUT	A table of times for the variable comparisons requested by the CT card.
DXF	BASE	The spacing for the radial coordinate of the fluid dynamic mesh.
DXN	BASE	The spacing for the radial coordinate of the neutronic mesh.
DYF	BASE	The spacing for the axial coordinate of the fluid dynamic mesh.
DYN	BASE	The spacing for the axial coordinate of the neutronic mesh.

E	PLTCOM	A temporary array used by the plotting routines.
EPSINT	INPUT	The smallest value of a variable to be used in integration.
FILAR	UNIT1	An array containing the names of the input and the output files.
FUNC	BASE	An array of function values.
FXL	PICBOU	The minimum value of the x-coordinate.
FXR	PICBOU	The maximum value of the x-coordinate.
FYB	PICBOU	The minimum value of the y-coordinate.
FYT	PICBOU	The maximum value of the y-coordinate.
GRID	INPUT1	An input array of data defining the plotting grid.
IB	IBASE	A table of the number of cells in the radial direction for a given type of variable for a given problem.
ICEL	INPUT	A table of cells to be traced.
IFAT	IBASE	A flag stating that a direct access file has been created.
IFCEL	INPUT1	An input array requesting cell tracing.
IFDF	IBASE	The analysis of fluid dynamics variable requested flag (0/1 - no/yes).
IFDT	IBASE	Fluid dynamics data is from TAPE6 flag (0/1 - no/yes).

IFF	UNIT	A unit number for the direct access files.
IGM	IBASE	An array defining the number of neutron energy groups.
IGM1	IBASE	IGM-1.
IJB	IBASE	A table of the total number of cells for a given type of variable of a given problem.
IJRS	UNIT	The length of a record of the neutronic mesh variables data on a direct access file.
IJRSH	UNIT	The length of a record of the fluid dynamic mesh variable data on a direct access file.
ILO	LABEL	The left radial cell index used in the current picture.
IMPLT	INPUT1	A flag to request a movie of variable plots.
INACT	INPUT	A flag requesting interactive graphics for the mesh plotting.
INP	UNIT	The input file unit number.
INTFLG	INPUT	A table of flags requesting the integration of a mesh variables.
INTGRD	INPUT	A table of flags requesting a grid type for the integration plots.
INTMP	INPUT	A flag requesting multiple curves per frame of plotting.

INTREG	INPUT	A table of regions for the integrations.
INTTYP	INPUT	A table of the types of integrations to be performed.
INTW	INPUT1	The variable and the type of variable to be used as the weight function in the integrations.
IRC	RTPC	The number of records of variables for this type of variable.
IREGIN	INPUT	A table of regions of a mesh variable to be analyzed.
IRS	UNIT	The length of a block of data in a direct access file record.
IRSZ	UNIT	The length of a direct access file.
IRT	RTPC	The type of data record from TAPE6 being read.
ISBTV	INPUT	A table of flags to request a side bar plot on a perspective plot.
ISF	UNIT	A unit number for the direct access files.
ISPLT	INPUT1	A flag requesting snap shot pictures.
ITF	UNIT	A unit number for a direct access file.
ITPLT	INPUT1	A flag requesting interactive graphics at the terminal.
ITT	IBASE	The number of time steps for a given type of variable.
ITTP	IBASE	The location in the TTAB table of times of the times for the variables of this type.

ITTY	UNIT	The unit number for I/O to the terminal.
ITYPE	INPUT	A table of the types of variables to be analyzed.
IUP	LABEL	The right radial cell index used in the current picture.
IVAR	INPUT	A table of variables for a given type to be analyzed.
IWINB	FRAM	The bottom boundary of the window in the frame.
IWINL	FRAM	The left boundary of the window in the frame.
IWINR	FRAM	The right boundary of the window in the frame.
IWINT	FRAM	The top boundary of the window in the frame.
IXL	CJE07	The left grid boundary of the window in the frame.
IXR	CJE07	The right grid boundary of the window in the frame.
IYB	CJE07	The bottom grid boundary of the window in the frame.
IYT	CJE07	The top grid boundary of the window in the frame.
JB	IBASE	A table of the number of cell in the axial direction for a given type of variable and a given problem.
JLO	LABEL	The lower axial cell index used in the current picture.
JUP	LABEL	The upper axial cell index used in the current picture.

LABIV	INPUT2	The label for the plot of the integration of the variables of a given type.
LABLV	LABEL1	The label of the grid on the left side of the plot.
LABNV	INPUT2	The label for the plots of the new variable calculated for this type of variable.
LABRV	LABEL1	The label of the grid on the right side of the plot.
LABT	LABEL1	The label for the top of the picture.
LABX	LABEL1	The label for the x axis.
LNC	LCMB	A buffer used to write the BASE file.
MASKC	UNSF	A table of masks used to unpack a TAPE36.
MAXPT	IBASE	The maximum number of points in the function space that can be plotted.
MDMP	UNIT	The maximum number of time steps that can be read for any specified type of variable.
MES	QMBUFFC	A buffer to receive the execute line message on LTSS and CTSS.
MESH	INPUT	A table of region indices and color indices to define a 2-d picture of a mesh.
MIDX	FRAM	The mid point of the x axis for a window in a frame.
MIDY	FRAM	The mid point of the y axis for a window in a frame.

MLTV	INPUT	A table of variables to be plotted on the same frame.
MMRX	PLTPAR	The maximum number of points in the x axis of a frame.
MMRY	PLTPAR	The maximum number of points in the y axis of a frame.
MNP	IBASE	The maximum number of problems that can be analyzed.
MNTT	PLTCOM	The number of time steps for a given type of variable.
MOVOP	INPUT	A table of operations defining a movie.
MOVSF	INPUT	Make a snap shot frame every MOVSF movie frames.
MPLN	UNIT1	The name of the plot file to be used for the movies.
MPX	PLTPAR	The mid point of the x axis of a frame.
MPY	PLTPAR	The mid point of the y axis of a frame.
MRX	FRAM	The number of points to use for the x axis of a window.
MRY	FRAM	The number of points to use for the y axis of a window.
MTT	PLTCOM	A table pointing to the requested times in the TTAB time table.
MTYP	IBASE	The number of types of variables allowed.
N6PFN	UNIT1	The name of the file used to load the overlays.
NAME	CBASE	A table of names of the problems to be analyzed.

NBPR	UNIT	The number of blocks per record for the neutronics direct access file.
NBPRH	UNIT	The number of blocks per record for the fluid dynamics direct access files.
NC	IBASE	The number of material components in SIMMER-II. NC is currently 5.
NCFLG	INPUT	A flag requesting contour plots for a given type of variables.
NCTUR	INPUT1	The number of contours to be used in a contour plot.
NFIL	DAT	A table of names for the direct access files of the fluid dynamic variables.
NFILE	INPUT2	A table of direct access file names to be used in the analysis of the variables of a given type.
NLAB	LABEL	The number of label lines to be used at the top of the picture.
NLV	LABEL	The number of ten character strings used to contain a variable label.
NODO	INPUT	A flag to turn off the plotting of the current mesh variable.
NOUT	UNIT	The unit number of the output file.
NP	IBASE	The number of problems to be analyzed.
NVARI	DAT	A table of names of the fluid dynamic variables.

NVFLG	INPUT	A table of flags requesting the calculation of a new variable.
NVOP	INPUT	A table of operations and variables used to calculate a new variable.
PMCB	PLTBUF	An I/O buffer used to make movies.
PPC	RTPC	A table of problem numbers.
PSCB	PLTBUF	An I/O buffer used for making snapshot pictures.
PTCB	PLTBUF	An I/O buffer used to draw plots on the terminal.
PVMCB	PLTPAR	A table of parameters for the LANL graphics system used in the making of movies.
PVSCB	PLTPAR	A table of parameters for the LANL graphics system used in the making of snapshot pictures.
PVTCB	PLTPAR	A table of parameters for the LANL graphics system used to draw plots on a graphic terminal.
RUNT	BASE	A table of execution times from SIMMER-II.
SPLN	UNIT1	The name of the plot file used for snapshot pictures.
T1VAR	BASE	An array of values for the type 1 variables.
T9VAR	BASE	An array of values for the type 9 variables.
TIMES	INPUT	A table of time increments and time region boundaries needed to generate the TT table of times.

TT	PLTCOM	The table of times specifying when the plots are to be drawn.
TTAB	TTB	A table of times at which the variables were generated by SIMMER-II.
VAR	PLTCOM	The function values of the variable to be plotted.
VAR1	LCMBUF	An I/O buffer used to read on the CDC 7600 the files generated on the Cray.
VIEW	INPUT	A table of parameters used to control the viewpoint of a perspective plot.
VIEWI	INPUT	A time increment to space the time of the rotation of a perspective plot.
VMAX	PLTCOM	A table of maximum values of the variables to be analyzed.
VMIN	PLTCOM	A table of minimum values of the variables to be analyzed.
VOLF	BASE	The volume of the fluid dynamic cells.
VOLN	BASE	The volume of the neutronic cells.
WFN36	INPUT2	A table of file names of the input TAPE36s.
WFN6	INPUT2	A table of file names of the input TAPE6s.
WFN	RTPC1	The name of the problem data file being read.
WMAC36	INPUT2	A table of machines or systems used to generate the TAPE36s.

WMAC6	INPUT2	A table of machines or systems used to generate the TAPE6s.
WMAC	RTPC1	The name of the machine or system used to generate the data file being read.
WPC	RTPC	The number of the problem that generated the data file being read.
XF	BASE	The radial coordinate of the left boundary of the cells of the fluid dynamic mesh.
XL	CJE07	The minimum value for the x axis on the current plot.
XN	BASE	The radial coordinate of the left boundary of the cells of the neutronic mesh.
XR	CJE07	The maximum value for the x axis on the current plot.
YB	CJE07	The minimum value for the y axis on the current plot.
YF	BASE	The axial coordinate of the bottom boundary of the cells of the fluid dynamic mesh.
YN	BASE	The axial coordinate of the bottom boundary of the cells of the neutronic mesh.
YT	CJE07	The maximum value for the y axis on the current plot.

APPENDIX G
SYSTEM-DEPENDENT ROUTINES

The postprocessor was developed at Los Alamos National Laboratory using the LTSS and CTSS computer systems. The processor depends on several routines from these systems. The routines are divided into four categories· plotting routines, LTSS/CDC 7600 routines, I/O routines, and general routines. The system-dependent routines will have to be supplied by the user.

A. Plotting Routines

The T6P gaphics for the T6P processor was developed using the Common Graphic System (CGS) at the Los Alamos National Laboratory. The CGS will allow you to write several graphics files and/or graphics devices simultaneously. The T6P processor uses one file for still or snapshot pictures, one file for motion pictures, and a graphics terminal. The graphics terminal was a Tektronics 4014E.

The following table is a list of the arguments with definitions used by the different routines.

BS - The size of the buffer BUF. BS is an integer.
BUF - An I/O buffer for a graphics device.
CB - An array used as a control block for a graphics device.
CHAR - A character string.
CN - An index to the color palette.
DEV - The device to be used.
DX - The x component of a vector.
DY - The y component of a vector.
FN - The name of the graphic file. FN is a character string.
LS - The speed of the line to the graphic terminal.
NCHAR - The number of characters in CHAR.
X - The x coordinate. $0 \leq X \leq 1$.
Y - The y coordinate. $0 \leq Y \leq 1$.

A list of the routines depending on the graphics system follow. The user will have to supply a set of these routines that are compatible with his graphics systems.

SUBROUTINE GDONE

A routine to terminate the plotting. GDONE must be called after all plotting is finished to empty all graphics I/O buffers.

SUBROUTINE GFNAME(CB,FN)

A routine to assign the file name FN to the device defined by the control block CB.

SUBROUTINE GLINA2(X,Y)

A routine to draw a line from the current position to the position specified by the point (X,Y).

SUBROUTINE GMOVA2(X,Y)

A routine to move from the current position to the position specified by the coordinates (X,Y).

SUBROUTINE GOFF(CB)

A routine to deactivate the device specified by the control block CB.

SUBROUTINE GOLAY

A routine to allocate the graphics common blocks for a program with overlays.

SUBROUTINE GON(CB)

A routine to activate the device specified by the control block CB.

SUBROUTINE GPAGE

A routine to advance the frame and prepare the plotting surface for a new plot.

SUBROUTINE GPALET(CN)

A routine to select the color CN from the CGS color palette.

SUBROUTINE GPLBEG(CB,BUF,BS)

A routine to begin the initialization of the device specified by the control block CB.

SUBROUTINE GPLEND(CB)

A routine to complete the initialization of the device specified by the control block CB.

SUBROUTINE GSTART

A routine to initialize the graphics system. GSTART must be the first call to the graphics systems.

SUBROUTINE GTBSLN(DX,DY)

A routine to set the base line vector for character strings.

SUBROUTINE GTEBEG(CB,BUF,BS)

A routine to begin the initialization of the graphics terminal.

SUBROUTINE GTEBPS(CB,LS)

A routine to set the line speed for the graphics terminal.

SUBROUTINE GTEEND(CB)

A routine to complete the initialization of the graphics system for the graphics terminal.

SUBROUTINE GTEMOD(CB,DEV)

A routine to specify the type of graphics terminal being used.

SUBROUTINE GTEXT(CHAR,NCHAR)

A routine to write the NCHAR character in the string CHAR along the base vector defined by GTBSLN.

SUBROUTINE GTFNTA

A routine to select the font and stroke mode to be used for the character string defined by GTEXT.

SUBROUTINE GTOLAY

A routine to allocate the common block storage to be used when plotting on a graphics terminal. This routine is used with programs using overlays.

SUBROUTINE GTSTRM

A routine to select the default mode for the next call to GTEXT.

SUBROUTINE GTWDTH(WIDTH)

A routine to select the character width for the next call to GTEXT.

B. LTSS/CDC 7600 Routines

All routines listed below except OVERLAY are used to read a file created by a Cray 1 on a CDC 7600 using the LTSS system. OVERLAY is the routine used to load an overlay into memory. These routines could be used on a CDC machine using either the NOS (interactive) or the SCOPE (batch) system.

The following table is a list of the arguments with definitions used by the different routines.

BS - The size of the buffer RBUF. BS is an integer.
ERR - An error flag.
FN - The I/O file name.
LRB - A pointer to the next Cray word in the buffer RBUF.
LUN - The I/O unit number.
NOB - The number of words to be moved to the output buffer OBUF.
OBUF - A output buffer.

P - The primary number of the overlay to be loaded.

R - The overlay recall flag.

RBUF - The I/O buffer.

S - The secondary number of the overlay to be loaded.

A list of the LTSS/CDC 7600 dependent routines follow.

<u>SUBROUTINE</u> CMPASGQ(LUN,FN,RBUF,BS)

A routine to assign a Cray 1 file FN to the unit number LUN. All I/O to this file will use the I/O buffer RBUF of length BS.

<u>SUBROUTINE</u> CMPCLS(LUN)

A routine to close the Cray 1 file assigned to the unit number LUN.

<u>SUBROUTINE</u> CMPLMPC(RBUF,LRB,OBUF,NOB,ERR)

A routine to convert Cray characters to LTSS characters.

<u>SUBROUTINE</u> CMPLMPF(RBUF,LRB,OBUF,NOB,ERR)

A routine to convert Cray floating point numbers to LTSS floating point numbers.

<u>SUBROUTINE</u> CMPLMPI(RBUF,LRB,OBUF,NOB,ERR)

A routine to convert Cray integers to CDC 7600 integers.

<u>SUBROUTINE</u> CMPREW(LUN)

A routine to rewind the Cray file LUN.

<u>SUBROUTINE</u> CMPRD(LUN,RBUF,BS)

A routine to read the Cray file LUN filling the buffer RPUF.

<u>SUBROUTINE</u> OVERLAY(FN,P,S,R)

A routine to load a overlay P,S from file FN.

C. I/O Routines

T6P was developed on the CDC 7600 using LTSS (the Livermore Time Sharing System). The LTSS system restricts the use of files by 1) limiting the number of files a program can have active, and 2) stipulating that the size of a file cannot be extended during execution.

To satisfy 1, the files are opened when needed and closed when not in use. To satisfy 2, the system of a family of files is used. A file is created of a fixed length. When the file is full, the next family member is created. All I/O to the file is directed to the current family member.

The following table is the list of arguments with definitions used by the different routines.

- BS - The size of the buffer BUF. BS is an integer.
- BUF - The I/O buffer.
- DPT - A pointer to a word in a direct access file.
- FL - The length of a file. FL is an integer.
- FLG - A file exists flag. FLG is an integer.
- FN - A file name. FN is a character string.
- LUN - A file unit number.

SUBROUTINE CHGFNL(FN,LUN)

A routine to change the name associated with file LUN to FN.

SUBROUTINE CLOSE(LUN)

A routine to close the file associated with unit LUN.

SUBROUTINE FEXIST(FN,FLG)

A routine to determine if file FN exists.

SUBROUTINE FILSIZ(FN,FL)

A routine to determine the size of file FN.

SUBROUTINE QASSIGN(LUN,FN,BUF,BS)

A routine to assign the unit number LUN to file FN. The I/O buffer will be BUF with a length of BS.

SUBROUTINE RDABSF(LUN,BUF,BS,DPT)

A routine to read a family member of the direct access file LUN beginning at word DPT.

SUBROUTINE RDISK(LUN,BUF,BS,DPT)

A routine to read a direct access file beginning at word DPT.

SUBROUTINE UNQFIL(FN)

A routine to find a unique name for file FN.

SUBROUTINE WDISK(LUN,BUF,BS,DPT)

A routine to write a direct access file starting at word DPT.

SUBROUTINE WRABSF(LUN,BUF,BS,DPT)

A routine to write a family member of the direct access file starting at word DPT.

D. General Routines

The following routines will have to be supplied by the user.

SUBROUTINE BLKCPY (BI,BO,N)

A routine to copy N words from BI to BO. For machines with a large core memory and a small core memory, BI and BO may be in either memory.

SUBROUTINE EXITA (FLG)

A routine to terminate a program. The image of the memory is retained in a disk file. FLG is a octal number that replaces the instruction counter as an error flag.

SUBROUTINE MSGFT

A routine to get a message from the execute line. The message is stored in the common block QMBUFFC variable MES.

FUNCTION SHIFT(I,J)

A function to shift I right or to do a left circular shift on I according to the following conditions:

J < 0 - Shift the bits of the word I to the right J places. Bits shifted out the right end of the word are lost. The empty positions on the left are filled with zeros.

J = 0 - I is unchanged.

J > 0 - Shift the bits of the word I to the left J places. Bits shifted out the left end of the word are inserted into the empty places created at the right end of the word.

APPENDIX H

T6P POSTPROCESSOR TEST PROBLEMS

There are three test problems, A, B, and C, for the T6P postprocessor. The input decks for the test problems A, B, and C are listed in Secs. I, II, and III respectively. The TAPE6s and TAPE36s for the test problems are generated by the two SIMMER-II test problems exported with the SIMMER-II code. The variables INTPLT and ISK36 on the second card of the SIMMER-II input decks must be set so the TAPE6s and TAPE36s will be written(INPLT = 1 and ISK36 = 0).

The T6P postprocessor test problems A and B are for the space-time neutronic test problem for SIMMER-II. The SIMMER-II update options FIXUP, AEOS, RHOMAP, and P1APRX are used to generated the SIMMER-II FORTRAN source deck. The T6P postprocessor test problems A and B do the same analysis. Test problem B is a restart problem using the T6P files generated by test problem A.

Test problem C is for the URANUS option of SIMMER-II. The variable ISPR on the second card of the input deck for SIMMER-II must be set to 2. The SIMMER-II update options URANUS and AEOS are used to generate the SIMMER-II FORTRAN source deck.

I) Test problem A for the space-time neutronic. The names of TAPE6 and TAPE36 have been changed to X4T6 and X4T36 respectively.

```
X4T6 CTSS 1
X4T36 CTSS 1
*
TYPE 1 VAR RHO AMP PINTG POW BETA GENT OM
TIMES 0.  .0001 .0114
MULTV (GENT BETA AMP) (-RHO -OM)
COLOR 0.  .5 1.5 2.  3.
RATE POW AMP
LABIV DERIVATIVE POWER TO AMPLITUDE
TYPE 2 VAR SMG1 SMG2 SML1 SML2 SML6 SML7 SMS1 SMS2 SMS3 SMS4
MULTV (SMG1 SML1 SML6 SMS1 SMS3) (SMG2 SML2 SML7 SMS2 SMS4)
RATE SMS1 SMS2
LABIV DERIVATIVE MASS FERTILE TO MASS FISSILE
REGION (0.  .005) ( .006 .009 ) (.01 .0113)
COLOR 0.  .5 1.5 2.  3.
GRID FHC .2 FPFH .6 FVC .2 FPFV .6
TYPE 4 VAR MESH4
MESH (-1 -4 -3 -7 2) (-5 -5 -1 -9 3) (-6 -7 -3 -7 2) &
(-8 -9 -1 -9 3) (-1 -4 -1 -2 4) (-6 -7 -1 -2 4) &
(-1 -4 -8 -9 5) (-6 -7 -8 -9 5)
COLOR 0.  .5 1.5 2.  3.
TYPE 5 VAR SUMALL PROMPT DELAYED SCATTER TOTAL LEAKAGE
TIMES 0.  .0001 .0114
SIDBAR 1 RHO 3
INTEG W RHO E 1.E-10 TYPE (2) REGIONS
NEWV +SUMALL +PROMPT +DELAYED +SCATTER +LEAKAGE
LABNV TOTAL FISSION
LABIV REACTIVITY AVERAGED
TYPE 6 VAR DEN1 DEN2 TEMP1 TEMP2 POW1 POW2 SPOW
TYPE 8 VAR FLUX1 FLUX2 FLUX3
REGION (1 4 4 7)
```

```
TYPE 9 VAR PFIS DFIS DSCAT RTOTL RLEAK TEBETA GENTIM REACT
MULTV (PFIS DFIS DSCAT -RTOTL RLEAK) (TEBETA REACT GENTIM -RTOTL)
COLOR 0.  .5 1.5 2.  3.
RATE PFIS RTOTL
LABIV DERIVATIVE PROMT FISSION TO TOTAL FISSION
TYPE 13 VAR MESH13
MESH (-1 -4 -3 -7 2) (-5 -5 -1 -9 3) (-6 -7 -3 -7 2) &
(-8 -9 -1 -9 3) (-1 -4 -1 -2 4) (-6 -7 -1 -2 4) &
(-1 -4 -8 -9 5) (-6 -7 -8 -9 5)
COLOR 0.  .5 1.5 2.  3.
TYPE 14 VAR ALPG ALPS ALPL P
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 15 VAR RSF1 RSF2 RLF1 RLF2 RGFB TRFU STEE TRSO TRCN
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 16 VAR RSB1 RSB2 RSB3 RSB4 RLB1 RLB2 RGB1 RGB2 RLB6 RLB7
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 17 VAR TS1 TS2 TS3 TS4 TS5 TSAF TSAS TSAN TSAC
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
```

```
TYPE 18 VAR RSB5 RSB6 RSB7 RLB3 RLB5 RLB8 RGB3 RGB5
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 19 VAR RSB8 RSB9 RLB4 RGB4 RGB6
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 20 VAR TL1 TL2 TL3 TL4 TL5 TL6 TG TVN
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 21 VAR DH RPFS RPSS RPLF RPLS RPLN RPLC
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 22 VAR QG EN1 EN2 EN3 EN4 EN5 KIJ FIJL FIJG
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 23 VAR FS1 FS2 FS3 FS4 FS5 FS6 FS7 FL1 FL2
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
```

```
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 24 VAR QL1 QL2 QL3 QL4 QL5 QL6
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 25 VAR QS1 QS2 QS3 QS4 QS5
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 26 VAR UG UL VG VL VLMG GAM1 GAM2 GAM3 GAM4 FAL3
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
CELL 4 0 3 4 2 6 0 4 1 6
COLOR .5 2.  1.5 4.5 2.  3.5 4.
```

II) Test problem B for the space-time neutronic. Please note that cards 1 and 2 are blank cards for this restart problem. If a continuation run of SIMMER-II had been made, then the names of the TAPE6 and TAPE36 would be named on these cards. Note also that the names of the BASE and BASET files have been change to X4TBASA and X4TBASTA, respectively. See card 4. For all types of variables that require data files from test problem A, the FILE card is used. The items listed on the FILE cards are the names given to the files generated by problem A. See cards 17, 30 and 51 as examples.

```
Blank card
Blank card
X4TBASA X4TBASTA 1
TYPE 1 VAR RHO AMP PINTG POW BETA GENT OM
TIMES 0.  .0001 .0114
MULTV (GENT BETA AMP) (-RHO -OM)
COLOR 0.  .5 1.5 2.  3.
RATE POW AMP
LABIV DERIVATIVE POWER TO AMPLITUDE
TYPE 2 VAR SMG1 SMG2 SML1 SML2 SML6 SML7 SMS1 SMS2 SMS3 SMS4
MULTV (SMG1 SML1 SML6 SMS1 SMS3) (SMG2 SML2 SML7 SMS2 SMS4)
RATE SMS1 SMS2
LABIV DERIVATIVE MASS FERTILE TO MASS FISSILE
REGION (0.  .005) ( .006 .009 ) (.01 .0113)
COLOR 0.  .5 1.5 2.  3.
GRID FHC .2 FPFH .6 FVC .2 FPFV .6
FILE HYDI1
TYPE 4 VAR MESH4
MESH (-1 -4 -3 -7 2) (-5 -5 -1 -9 3) (-6 -7 -3 -7 2) &
(-8 -9 -1 -9 3) (-1 -4 -1 -2 4) (-6 -7 -1 -2 4) &
(-1 -4 -8 -9 5) (-6 -7 -8 -9 5)
COLOR 0.  .5 1.5 2.  3.
TYPE 5 VAR SUMALL PROMPT DELAYED SCATTER TOTAL LEAKAGE
TIMES 0.  .0001 .0114
```

```
SIDBAR 1 RHO 3
INTEG W RHO E 1.E-10 TYPE (2) REGIONS
NEWV +SUMALL +PROMPT +DELAYED +SCATTER +LEAKAGE
LABNV TOTAL FISSION
LABIV REACTIVITY AVERAGED
FILE RACT11 RACT21 RACT31 RACT41 RACT51 RACT61
TYPE 6 VAR DEN1 DEN2 TEMP1 TEMP2 POW1 POW2 SPOW
FILE DEN11 DEN21 TEMP11 TEMP21 POW11 POW21 SPOW1
TYPE 8 VAR FLUX1 FLUX2 FLUX3
REGION (1 4 4 7)
FILE FLUX11 FLUX21 FLUX31
TYPE 9 VAR PFIS DFIS DSCAT RTOTL RLEAK TEBETA GENTIM REACT
MULTV (PFIS DFIS DSCAT -RTOTL RLEAK) (TEBETA REACT GENTIM -RTOTL)
COLOR 0.  .5 1.5 2.  3.
RATE PFIS RTOTL
LABIV DERIVATIVE PROMT FISSION TO TOTAL FISSION
TYPE 13 VAR MESH13
MESH (-1 -4 -3 -7 2) (-5 -5 -1 -9 3) (-6 -7 -3 -7 2) &
(-8 -9 -1 -9 3) (-1 -4 -1 -2 4) (-6 -7 -1 -2 4) &
(-1 -4 -8 -9 5) (-6 -7 -8 -9 5)
COLOR 0.  .5 1.5 2.  3.
TYPE 14 VAR ALPG ALPS ALPL P
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE ALGF1 ALSF1 ALLF1 PF1
TYPE 15 VAR RSF1 RSF2 RLF1 RLF2 RGFB TRFU STEE TRSO TRCN
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE RSF11 RSF21 RLF11 RLF21 RGFF1 UO2F1 FEF1 NAF1 CTRF1
TYPE 16 VAR RSB1 RSB2 RSB3 RSB4 RLB1 RLB2 RGB1 RGB2 RLB6 RLB7
```

```
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE RS1F1 RS2F1 RS3F1 RS4F1 RL1F1 RL2F1 RG1F1 RG2F1 RL6F1 RL7F1
TYPE 17 VAR TS1 TS2 TS3 TS4 TS5 TSAF TSAS TSAN TSAC
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 [illegible]
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE TS1F1 TS2F1 TS3F1 TS4F1 TS5F1 TSTF1 TSTS1 TSTN1 TSTC1
TYPE 18 VAR RSB5 RSB6 RSB7 RLB3 RLB5 RLB8 RGB3 RGB5
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE RS5F1 RS6F1 RS7F1 RL3F1 RL5F1 RL8F1 RG3F1 RG5F1
TYPE 19 VAR RSB8 RSB9 RLB4 RGB4 RGB6
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE RS8F1 RS9F1 RL4F1 RG4F1 RG6F1
TYPE 20 VAR TL1 TL2 TL3 TL4 TL5 TL6 TG TVN
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE TL1F1 TL2F1 TL3F1 TL4F1 TL5F1 TL6F1 TGF1 TVNF1
TYPE 21 VAR DH RPFS RPSS RPLF RPLS RPLN RPLC
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
```

```
FILE DHF1 RPSF1 RPSS1 RPLF1 RPLS1 RPLN1 RPLC1
TYPE 22 VAR QG EN1 EN2 EN3 EN4 EN5 KIJ FIJL FIJG
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE QGF1 EN1F1 EN2F1 EN3F1 EN4F1 EN5F1 KIJF1 FIJL1 FIJG1
TYPE 23 VAR FS1 FS2 FS3 FS4 FS5 FS6 FS7 FL1 FL2
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE FS1F1 FS2F1 FS3F1 FS4F1 FS5F1 FS6F1 FS7F1 FL1F1 FL2F1
TYPE 24 VAR QL1 QL2 QL3 QL4 QL5 QL6
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE QL1F1 QL2F1 QL3F1 QL4F1 QL5F1 QL6F1
TYPE 25 VAR QS1 QS2 QS3 QS4 QS5
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE QS1F1 QS2F1 QS3F1 QS4F1 QS5F1
TYPE 26 VAR UG UL VG VL VLMG GAM1 GAM2 GAM3 GAM4 FAL3
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35 &
IJC 3 1 4 1 2 3 1 4 8 9 3 5 5 1 9 3 6 7 1 2 3 6 7 8 9 &
3 8 9 1 9 6 1 4 3 7 6 6 7 3 7
COLOR .5 2.  1.5 4.5 2.  3.5 4.
FILE UGF1 ULF1 VGF1 VLF1 VLGN1 GC1F1 GC2F1 GC3F1 GC4F1 FAIL1
```

III) Test problem C for the URANUS option. The names of the files TAPE6 and TAPE36 have been changed to LWPT6 and LWPT36, respectively.

```
LWPT6 LTSS 1
LWPT36 LTSS 1
*
TYPE 2 VAR SMG1 SMG2 SML1 SML2 SML6 SML7 SMS1 SMS2 SMS3 SMS4
MULTV (SMG1 SML1 SML6 SMS1 SMS3) (SMG2 SML2 SML7 SMS2 SMS4)
TYPE 14 VAR ALPG ALPS ALPL P
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 15 VAR RSF1 RSF2 RLF1 RLF2 RGFB TRFU STEE TRSO TRCN
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 16 VAR RSB1 RSB2 RSB3 RSB4 RLB1 RLB2 RGB1 RGB2 RLB6 RLB7
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 17 VAR TS1 TS2 TS3 TS4 TS5 TSAF TSAS TSAN TSAC
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 18 VAR RSB5 RSB6 RSB7 RLB3 RLB5 RLB8 RGB3 RGB5
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 19 VAR RSB8 RSB9 RLB4 RGB4 RGB6
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 20 VAR TL1 TL2 TL3 TL4 TL5 TL6 TG TVN
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 21 VAR DH RPFS RPSS RPLF RPLS RPLN RPLC
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 22 VAR QG EN1 EN2 EN3 EN4 EN5 KIJ FIJL FIJG
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
```

```
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 23 VAR FS1 FS2 FS3 FS4 FS5 FS6 FS7 FL1 FL2
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 24 VAR QL1 QL2 QL3 QL4 QL5 QL6
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 25 VAR QS1 QS2 QS3 QS4 QS5
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
TYPE 26 VAR UG UL VG VL VLMG GAM1 GAM2 GAM3 GAM4 FAL3
VIEW HA 225 DHA 1 THA 225 VA 35 DVA 1 TVA 35
COLOR .5 2.  1.5 4.5 2.  3.5 4.
```

DISTRIBUTION

Nuclear Regulatory Commission, AN, Laurel, Maryland	105
Technical Information Center, Oak Ridge, Tennessee	2
Los Alamos National Laboratory, Los Alamos, New Mexico	50
	157

Available from

Superintendent of Documents
U.S. Government Printing Office
Post Office Box 37082
Washington, D. C. 20013-7982

and

National Technical Information Service
Springfield, VA 22161

Los Alamos