Y/M Lee

NUREG/CR-3046
PNL-4385
Vol. 5

COBRA/TRAC - A Thermal-Hydraulics Code for Transient Analysis of Nuclear Reactor Vessels and Primary Coolant Systems

Programmers' Manual

Prepared by A. S. Koontz, J. M. Cuta

Pacific Northwest Laboratory
Operated by
Battelle Memorial Institute

Prepared for
U.S. Nuclear Regulatory
Commission

Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, 1717 H Street, N.W.
   Washington, DC 20555
2. The NRC/GPO Sales Program, U.S. Nuclear Regulatory Commission,
   Washington, DC 20555
3. The National Technical Information Service, Springfield, VA 22161

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda; NRC Office of Inspection and Enforcement bulletins, circulars, information notices, inspection and investigation notices; Licensee Event Reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the NRC/GPO Sales Program: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, and NRC booklets and brochures. Also available are Regulatory Guides, NRC regulations in the *Code of Federal Regulations,* and *Nuclear Regulatory Commission Issuances.*

Documents available from the National Technical Information Service include NUREG series reports and technical reports prepared by other federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal and periodical articles, and transactions. *Federal Register* notices, federal and state legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free upon written request to the Division of Technical Information and Document Control, U.S. Nuclear Regulatory Commission, Washington, DC 20555.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at the NRC Library, 7920 Norfolk Avenue, Bethesda, Maryland, and are available there for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

GPO Printed copy price: $8.50

NUREG/CR-3046
PNL-4385
Vol. 5
R4

COBRA/TRAC - A Thermal-Hydraulics Code for Transient Analysis of Nuclear Reactor Vessels and Primary Coolant Systems

Programmers' Manual

Manuscript Completed: November 1982
Date Published: March 1983

Prepared by
A. S. Koontz, J. M. Cuta

Pacific Northwest Laboratory
Richland, WA 99352

Prepared for
Division of Accident Evaluation
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
NRC FIN B2391

ABSTRACT

The COBRA/TRAC computer program has been developed to predict the thermal-hydraulic response of nuclear reactor primary coolant systems to small and large break loss-of-coolant accidents and other anticipated transients. The code solves the compressible three-dimensional, two-fluid, three-field equations for two-phase flow in the reactor vessel. The three fields are the vapor field, the continuous liquid field, and the liquid drop field. A five-equation drift flux model is used to model fluid flow in the primary system piping, pressurizer, pumps, and accumulators. The heat generation rate of the core is specified by input and no reactor kinetics calculations are included in the solution. This volume explains the details of COBRA/TRAC's working parts from the programmer's viewpoint. The code's overlay structure is discussed. The memory management and COMMON block manipulation are explained, as are the restart/dump logic and the graphics logic. Suggestions for code conversion to "non-LANL" CDC computers and non-CDC computers are given.

CONTENTS

FIGURES

TABLES

ACKNOWLEDGEMENTS

COBRA/TRAC is the result of the efforts of a number of people. We wish to acknowledge the main contributors and to express our appreciation to those who have offered their advice and suggestions.

The main contributors to the program are listed below.

Fluid Dynamics:	M. J. Thurgood, T. L. George, and T. E. Guidotti
Heat Transfer:	J. M. Kelly and R. J. Kohrt
Turbulence Model:	K. R. Crowell
Graphics and Programming:	A. S. Koontz
Simulations:	K. L. Basehore, S. H. Bian, J. M. Cuta, R. J. Kohrt, G. A. Sly, and C. A. Wilkins
One-Dimensional Components and Code Architecture:	Members of the TRAC Code Development Group at LANL

We wish to thank Dr. S. Fabic of the U.S. Nuclear Regulatory Commission for his patience, support, and suggestions during this large undertaking. We also wish to thank Drs. Tong, Shotkin, Han and Zuber of the U.S. Nuclear Regulatory Commission and members of the Advanced Code Review Group for their many helpful suggestions. We also express our gratitude to our manager, Dr. D. S. Trent, for his support, and Peggy Snyder and Cathy Darby for their lead roles in typing this report.

COBRA/TRAC - A THERMAL-HYDRAULICS CODE FOR TRANSIENT ANALYSIS OF NUCLEAR RECTOR VESSELS AND PRIMARY COOLANT SYSTEMS
VOLUME 5: PROGRAMMERS' MANUAL

1.0 INTRODUCTION

TRAC-PD2 was developed at Los Alamos National Laboratory (LANL) to model an entire reactor primary system. This includes the reactor vessel, piping, pumps, steam generators, etc. COBRA-TF was developed at the Department of Energy's Pacific Northwest Laboratory, which is operated by Battelle Memorial Institute, to model only the reactor vessel component of the reactor system. Since TRAC-PD2's vessel component lacked the detailed analysis available in COBRA-TF, while COBRA-TF lacked the system solution capability (pipes, pumps, etc.), COBRA-TF and TRAC were merged into one code.

COBRA/TRAC is the result of this merger. Both TRAC-PD2 and COBRA-TF are complex, so COBRA/TRAC is even more complex. It is important for the programmer who must maintain it to know more than the average user about the code.

Basically, this connection involved replacing the TRAC-PD2 vessel component with the COBRA-TF vessel component and ensuring that the necessary connection logic worked. Both codes were written for the CDC-7600 class of computers and solution techniques were similar, so the actual connection of the two codes was comparatively easy. In fact, very little of the one-dimensional component logic is different from that found in TRAC-PD2.

The conservation equations and constitutive models for COBRA-TF are given in Volume 1. The numerical solution methods for the COBRA-TF part of COBRA/TRAC are presented in Volume 2. Input data is described in detail in Volume 3, the COBRA/TRAC Users' Manual. Developmental assessment of the COBRA/TRAC code's performance compared to experimental data is discussed in Volume 4.

This manual, Volume 5, the Programmers' Manual, explains the details of COBRA/TRAC's working parts from the programmer's viewpoint. The code's overlay structure and the mechanism by which the overlays are managed are discussed. The intricate memory management and COMMON block manipulation are explained, as are the restart/dump logic and the graphics logic. Suggestions for conversion of COBRA/TRAC to "non-LANL" CDC computers and non-CDC computers are given. Appendices give detailed descriptions of most COMMON block variables and listings of graphics and redimensioning codes provided as part of the COBRA/TRAC package.

2.0 COBRA/TRAC - HOW IT WORKS

COBRA/TRAC is a complicated code and, in many respects is machine dependent. This section explains these machine dependencies and describes how COBRA/TRAC works. COBRA/TRAC's overlay structure and the functions of each overlay are discussed. Details for the important vessel component subroutines are provided. Input and output files and their logical unit assignments are discussed. Small-core memory to large-core memory data transfers and the COMMON block structures are explained. Finally, the dump/restart and graphics logic are discussed.

2.1 OVERLAY

The released version of COBRA/TRAC runs on LANL's CDC-7600 computer system under the Livermore Time Sharing System (LTSS). It is a very large computer code and can easily demand more central memory than is available (approximately $560{,}000_8$ 64-bit words), even for relatively simple problems. So COBRA/TRAC must be overlaid.

Portions of the executable code for COBRA/TRAC share the available central memory, since they do not all need to be in memory at the same time. This process is simplified since COBRA/TRAC is very modular--each overlay performs its specific task without the help of other overlays. Overlays are maintained on some peripheral device (such as a disk) while not in use and are copied into memory when needed. Figure 2.1 shows a simple diagram demonstrating the advantage of using overlays.

Several rules govern the generation of overlays:

1. A "root" overlay must exist. (The root is always in memory and calls subordinate overlays as needed.)
2. Subordinate overlays cannot reference other overlays that would need to occupy the same memory.
3. Subordinate overlays can reference routines that are loaded in a higher memory level.
4. Data is communicated between overlays via COMMON blocks.

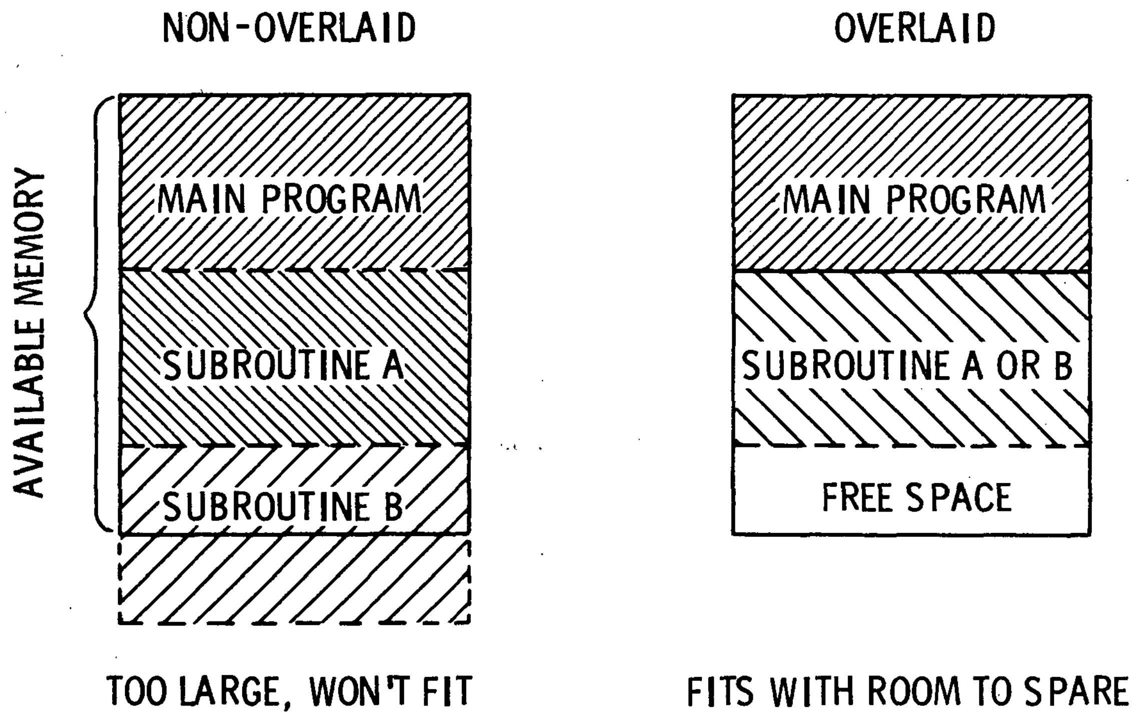

FIGURE 2.1. Example of Memory Allocation for Overlaid Coding

The actual overlay activity (transferring overlays between peripheral devices and memory) can be very complicated. Most systems provide a mechanism to automate this process. On the LANL computing facility the overlays are referenced by a statement of the form:

CALL OVERLAY (nHxxxxx, j, k)

where:

n = the number of characters in the Hollerith name for the overlay
xxxxx = Hollerith (alphanumeric) name of the overlay
j = major overlay identification number
k = secondary overlay identification number

The symbols j and k identify unique overlays; for the primary level of each overlay, k is 0; for overlays subordinate to this first level, k is some positive integer and j is the same as specified for the primary level.

The overlays are actually generated at load time. An input file, called a "load file", names the overlays and identifies their entry points. Each overlay is defined by a statement in the load file of the form:

*OVERLAY(j, k, EP=xxxxx)

where j, and k are the same values defined in the call to this overlay and xxxxx is the name of the subroutine that defines the beginning of this overlay.

The *FL cards force loading of the named subroutines into the basic overlay.

Figure 2.2 shows the load file used for COBRA/TRAC. A general view of COBRA/TRAC's overlay structure is shown in Figure 2.3. (More detailed flowcharts for each overlay are given in Figures 2.4 through 2.10.)

2.2 OVERLAY FUNCTIONS

COBRA/TRAC is modular in design and function. Each overlay accomplishes a specific function or set of functions.

```
*OVERLAY(0,0)
*FL AFROMH CURVEM CURVE GAUSS HGAS PROP SAT TGAS TRANSP
*FL VOLLIQ VOLVAP WARRAY SPLITIT ACCMBD BREAKX PUMPD SAVBD
*FL SCMOVE SETBD SETPRP MOVLEV FPROP SIGMA SPLIT ICMPR
*FL MIXPRP PUMPD MOVLEV
*FL GAPHTC GTHCON DEFORM
*OVERLAY(1,0,EP=INPUT)
*OVERLAY(1,1,EP=SETIN)
*OVERLAY(1,2,EP=SETUP)
*OVERLAY(1,3,EP=SETOUT)
*OVERLAY(1,4,EP=RPIPE)
*OVERLAY(1,5,EP=RPUMP)
*OVERLAY(1,6,EP=RTEE)
*OVERLAY(1,7,EP=RVLVE)
*OVERLAY(1,8,EP=RBREAK)
*OVERLAY(1,9,EP=RFILL)
*OVERLAY(1,10,EP=RPRIZR)
*OVERLAY(1,11,EP=RSTGEN)
*OVERLAY(1,12,EP=RACCUM)
*OVERLAY(2,0,EP=INIT)
*OVERLAY(3,0,EP=DMPIT)
*OVERLAY(4,0,EP=EDIT)
*OVERLAY(5,0,EP=GRAF)
*OVERLAY(6,0,EP=OUTER)
*OVERLAY(6,1,EP=OUT1DN)
*OVERLAY(6,2,EP=OUT3D)
*OVERLAY(11,0,EP=PREP)
*OVERLAY(12,0,EP=POST)
*OVERLAY(13,0,EP=BLKDAT)
```

FIGURE 2.2. Load File for COBRA/TRAC Overlay Structure

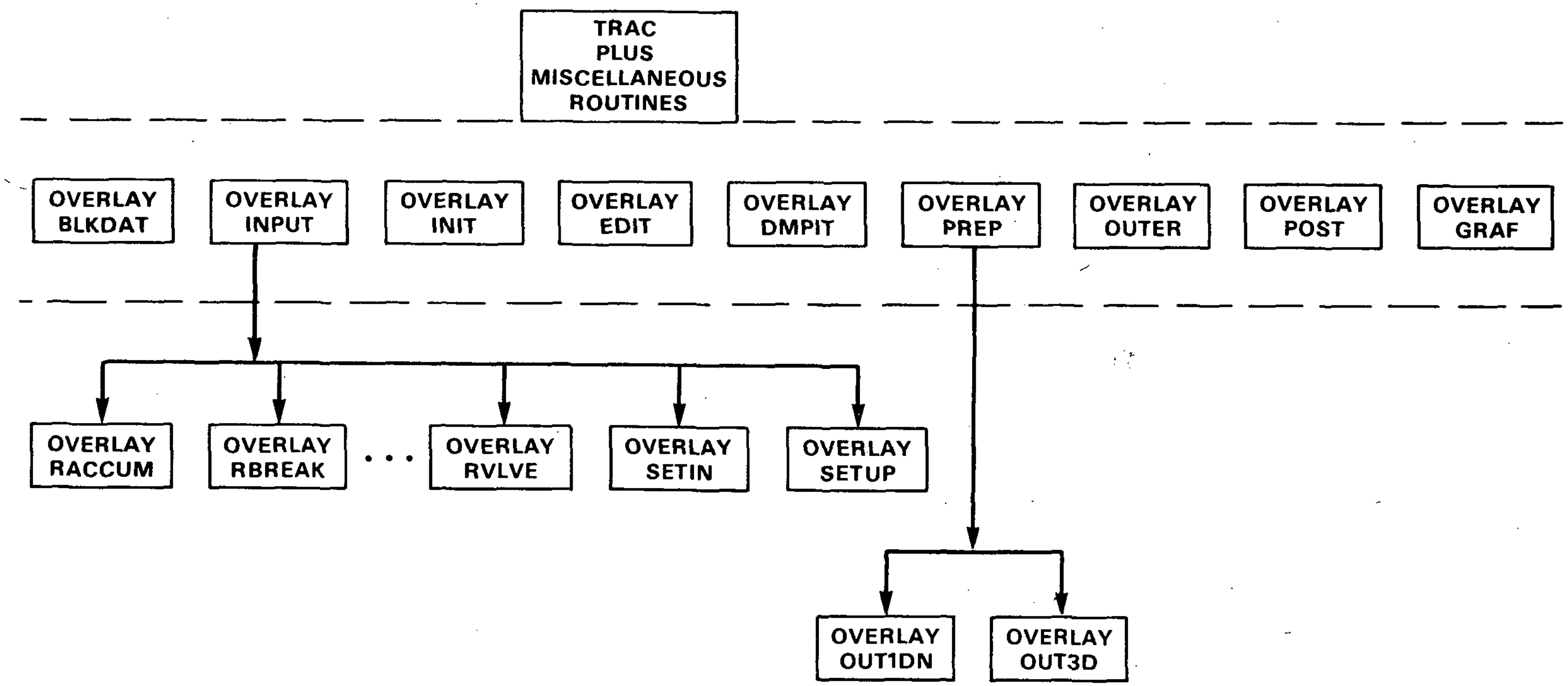

FIGURE 2.3. COBRA/TRAC Overlay Hierarchy

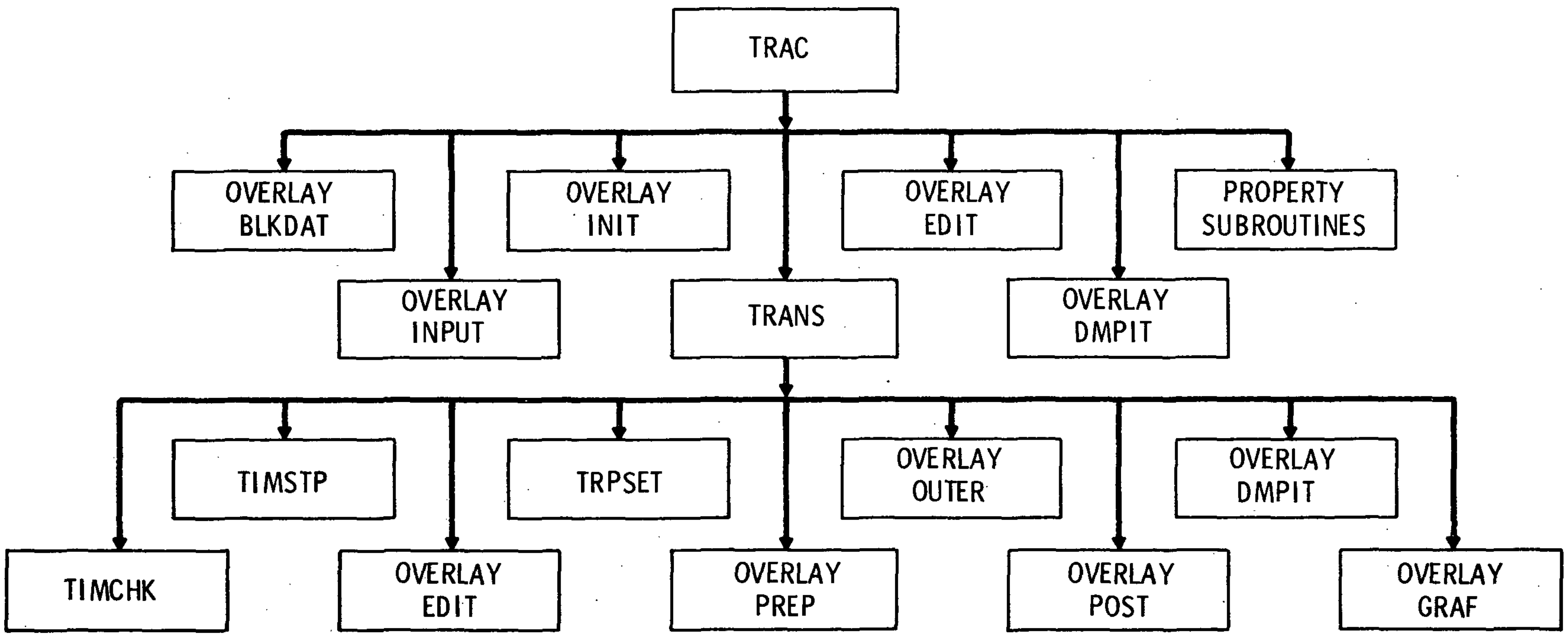

FIGURE 2.4. Overlay and Subroutine Calls from COBRA/TRAC Base

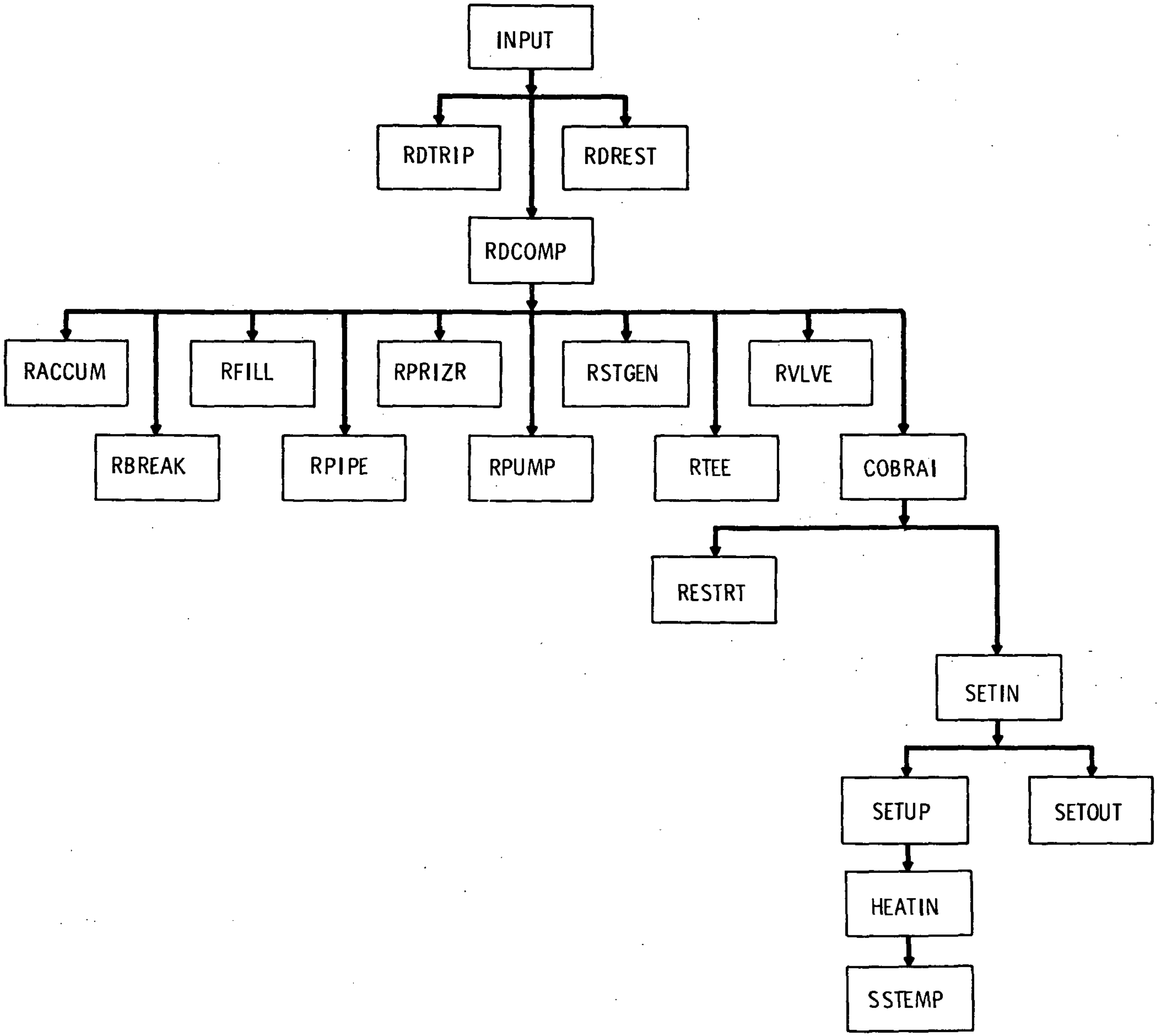

FIGURE 2.5. Subroutine Calling Hierarchy in Overlay INPUT

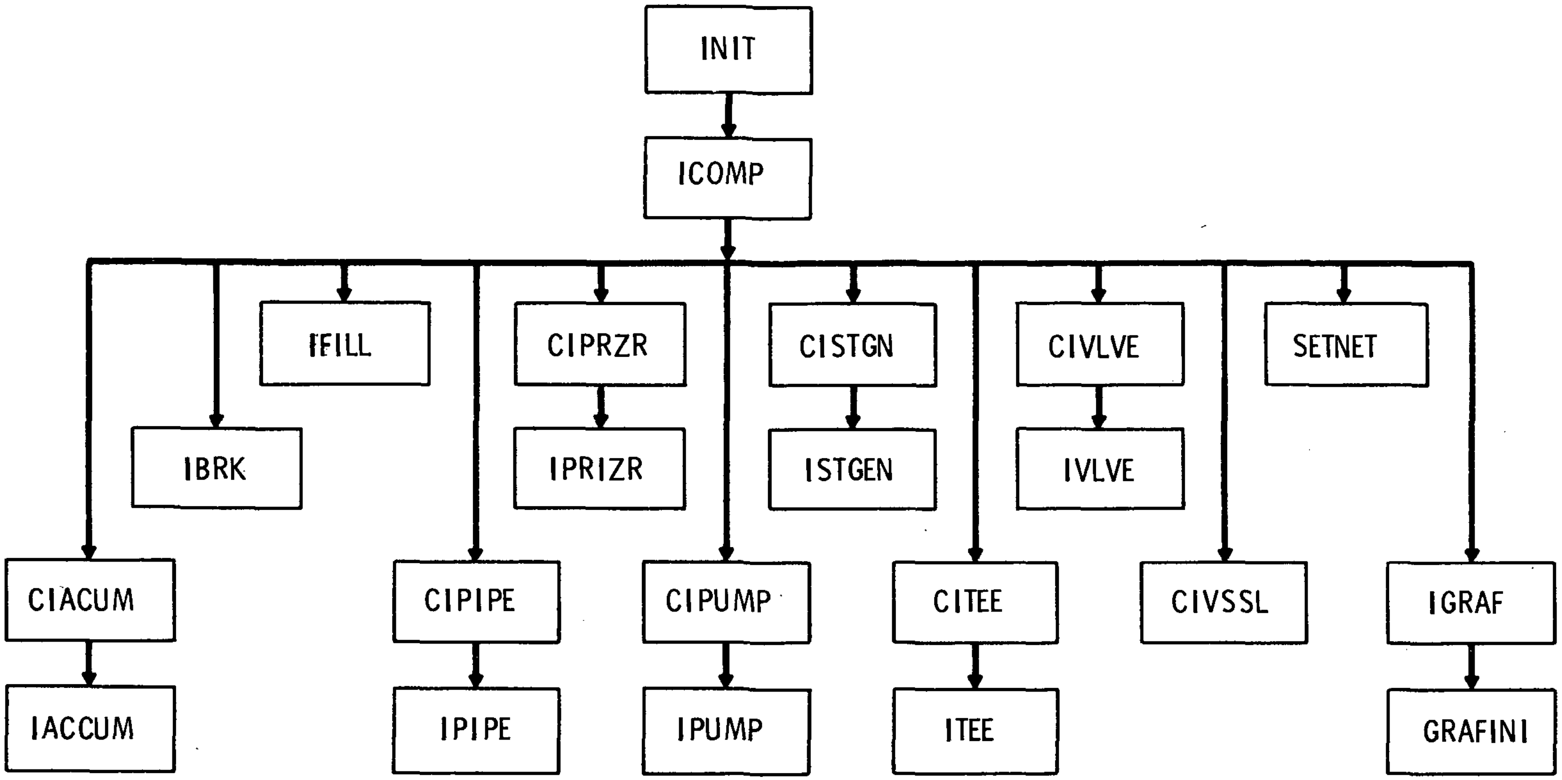

FIGURE 2.6. Subroutine Calling Hierarchy in Overlay INIT

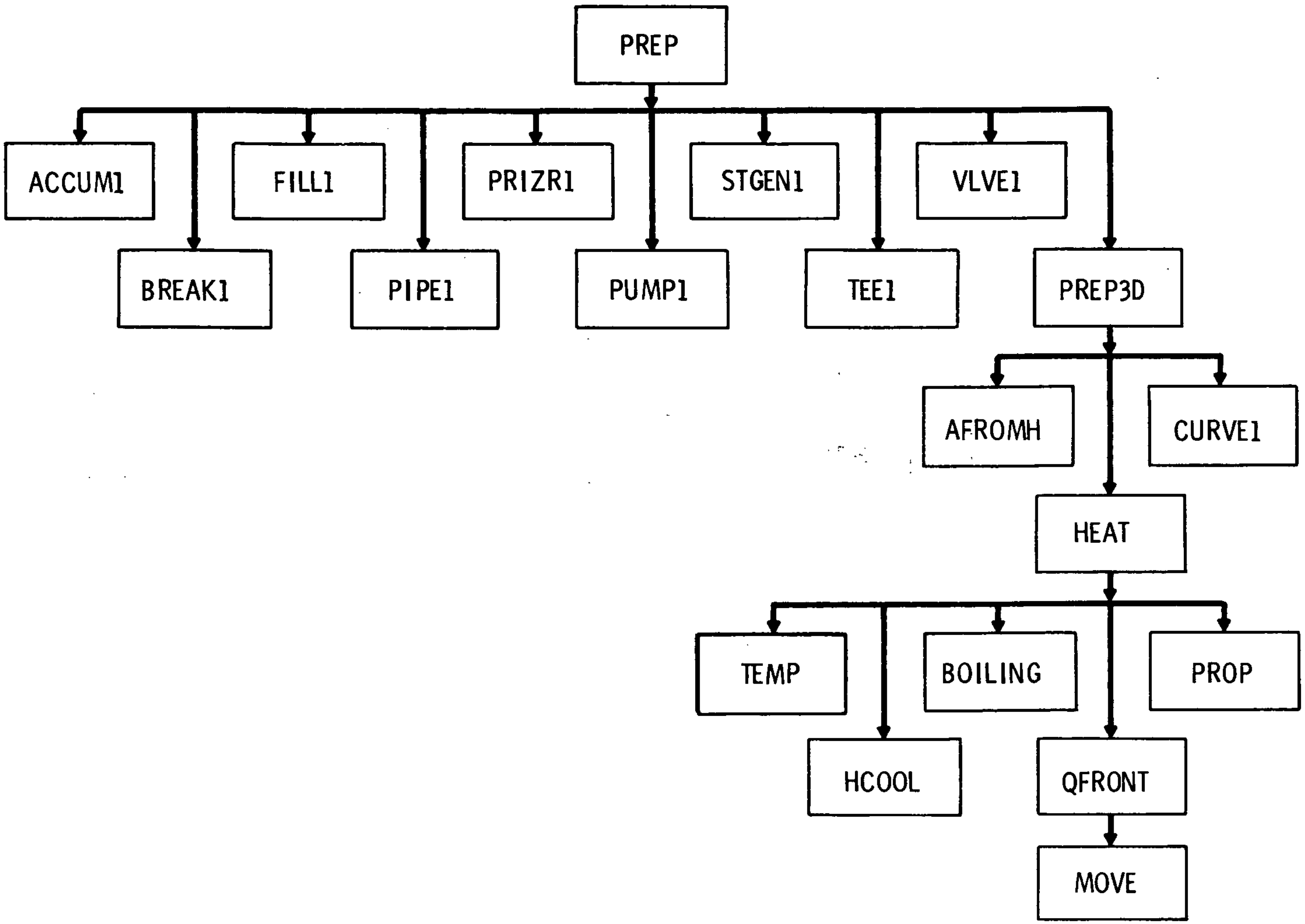

FIGURE 2.7. Subroutine Calling Hierarchy in Overlay PREP

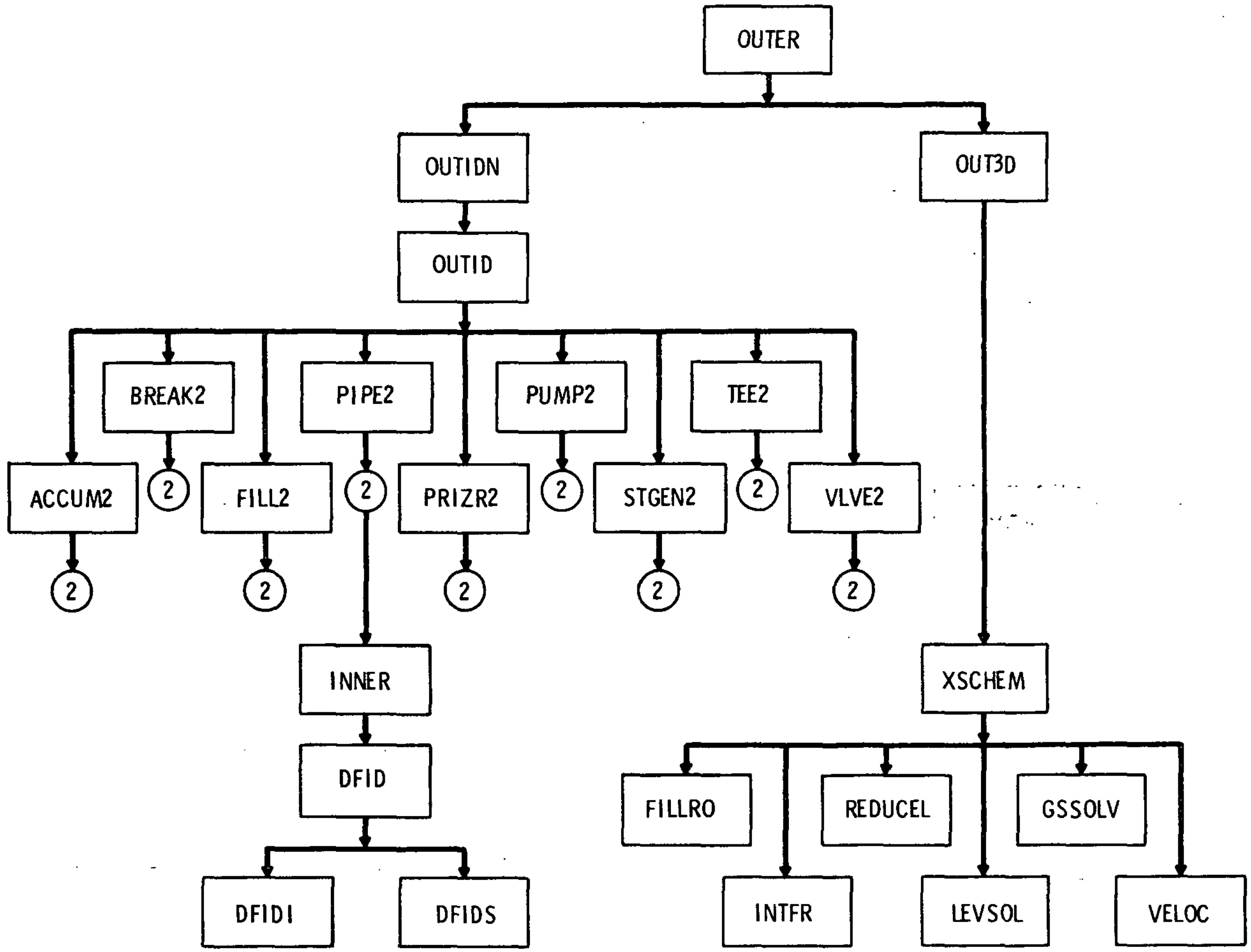

FIGURE 2.8. Subroutine Calling Hierarchy in Overlay OUTER

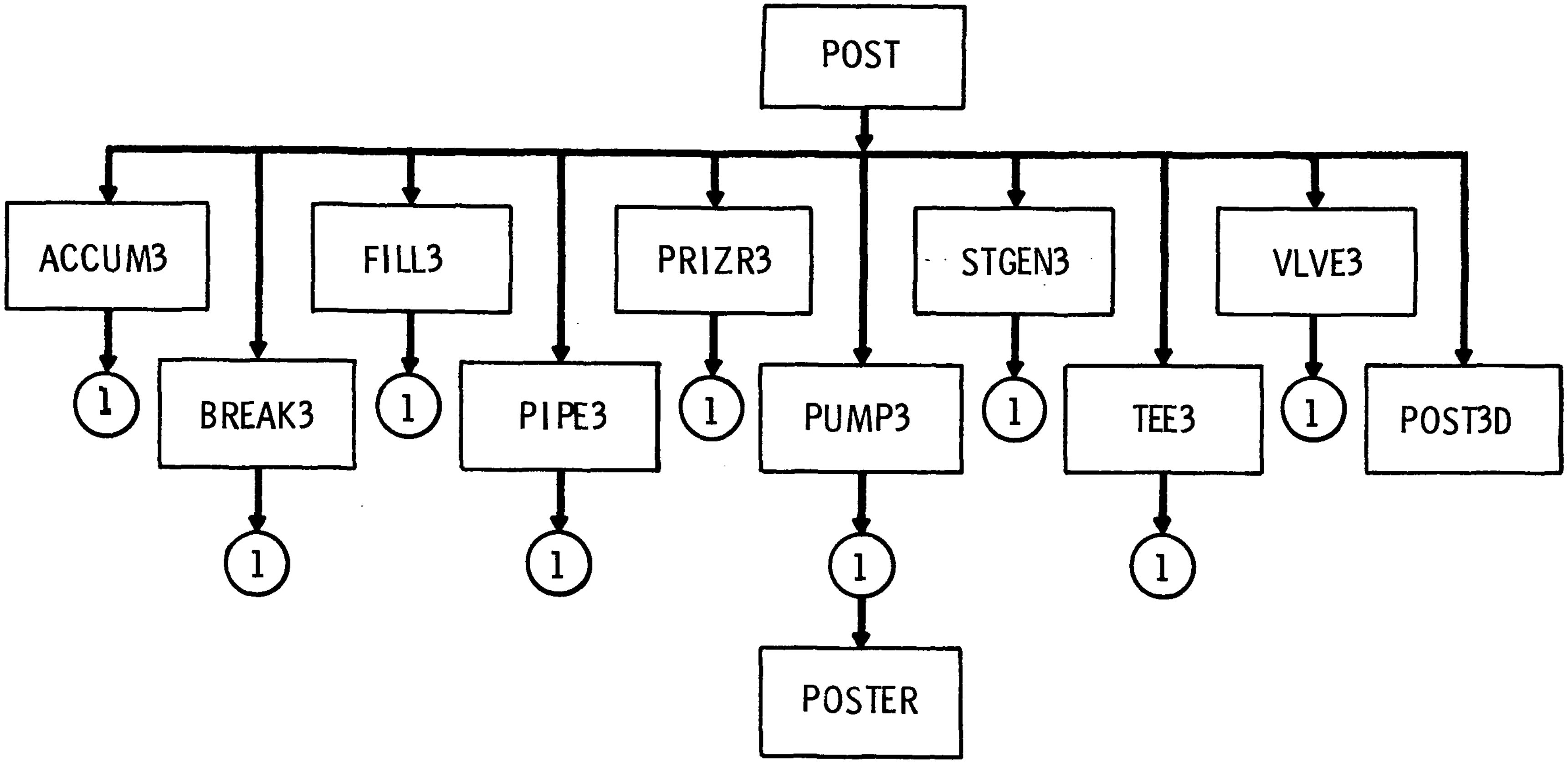

FIGURE 2.9. Subroutine Calling Hierarchy in Overlay POST

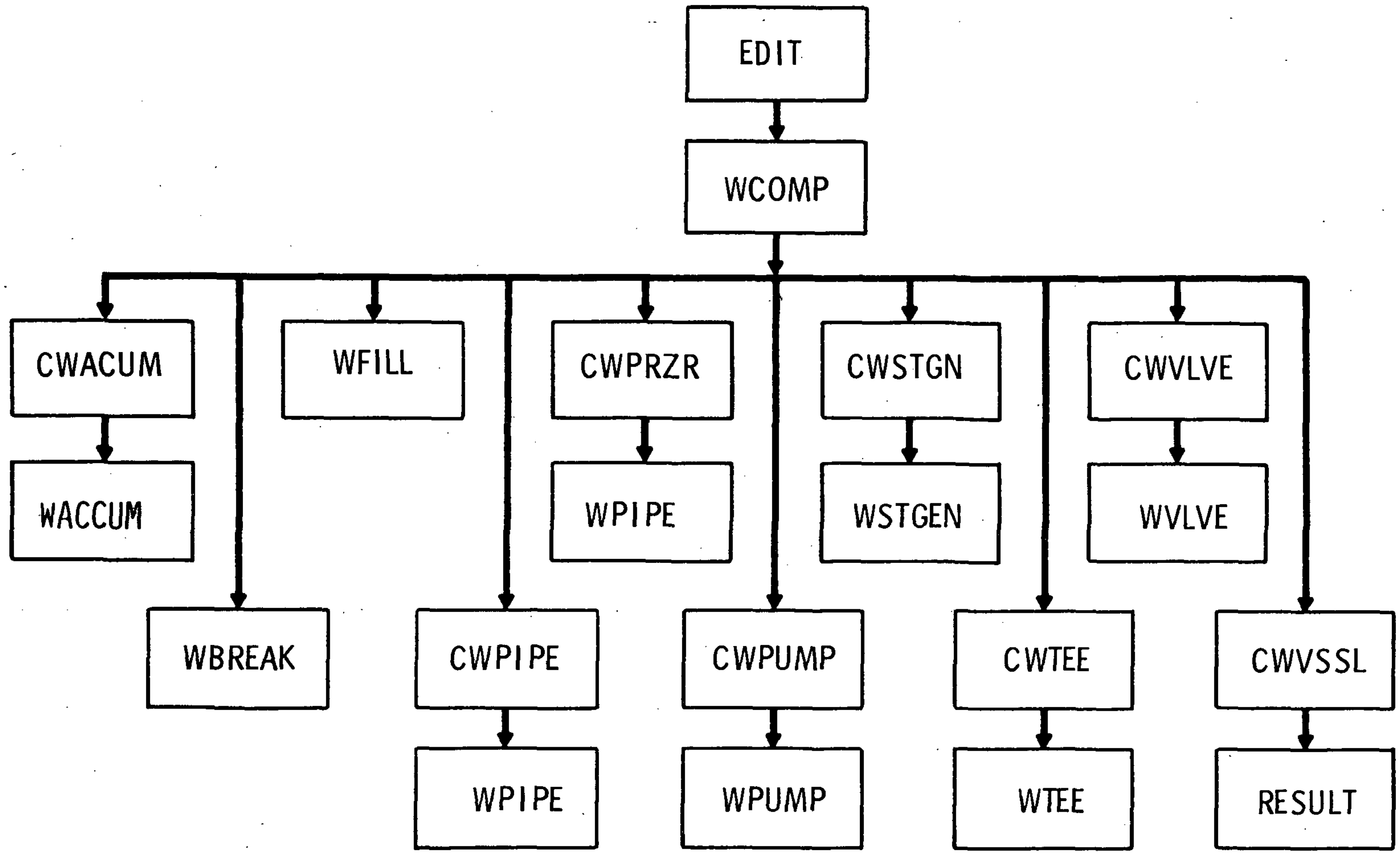

FIGURE 2.10. Subroutine Calling Hierarchy in Overlay EDIT

The INPUT overlay reads and processes most of the input data. Subroutine RDCOMP controls the input of all the component data from the input file via routines RACCUM, RBREAK, RFILL, RPIPE, RPRIZR, RPUMP, RSTGEN, RTEE, RVALVE and COBRAI. (Each of these routines reads data for a specific component type; e.g., RACCUM reads accumulator data, RPRIZR reads pressurizer data, and COBRAI reads the vessel data.) Subroutine RDREST reads the one-dimensional component restart file, if any, saved from a previous COBRA/TRAC simulation to continue a transient.

The INIT overlay defines variables that are not specified in the input data but are needed before the calculation begins. Junction input is checked to make sure all the components are connected logically. Subroutines are called to initialize arrays for each component type. For example, pump arrays are initialized in routine IPUMP; pipe arrays are initialized in IPIPE. Data to initialize the graphics dump process are read by subroutine IGRAF.

Subroutine TRANS drives the transient solution. Prepass calculations, the outer iteration and then postpass calculations are performed. Simulation control (new time domains, invocation of various output options, etc.) is monitored by calls to TIMCHK. Data controlling the time steps and various dump intervals are read by TIMSTP. (The steady-state solution driver in TRAC-PD2 was eliminated from COBRA/TRAC. Instead, a transient with constant boundary conditions is executed to arrive at a steady state. Discussion of this procedure can be found in Volume 3, the Users' Manual.)

Prior to performing the iterative solution for each time step, prepass calculations are made via overlay PREP. The prepass calculation uses the system state at the completion of the previous time step to evaluate numerous quantities to be used during the outer iteration. The prepass consists of a loop over all components in the system. Each component begins the prepass by moving the values calculated during the last time step into the storage area for old time values. Next the relative phase velocities are evaluated and used to calculate velocities of the liquid and vapor in all components except the vessel. These velocities are used to calculate the wall friction factors for each cell in the component. For components that require heat transfer calculations, the prepass also evaluates material properties and heat transfer coefficients. Finally, the prepass sets trip values and stores component data in the boundary data arrays. Prepass calculations for each one-dimensional component are done by calling appropriate routines depending upon component type, as was done for the input and initialization portions of COBRA/TRAC.

The prepass calculation for the vessel, performed by PREP3D, involves updating boundary conditions, obtaining heat transfer coefficients and heat fluxes, and solving the rod conduction equation. These functions are accomplished with calls to subroutines HEAT and TEMP.

The hydrodynamic state of the system is determined by a sequence of Newton-Raphson iterations that solve the linearized equations for each external loop and the vessel. Throughout the sequence of iterations that constitute a time step (called an outer iteration) the properties evaluated during the prepass and previous postpass remain fixed. These include wall and rod temperatures, heat transfer coefficients, wall friction factors and phase velocities. The remaining fluid properties are varied to solve the hydrodynamic models.

Each call to overlay OUTER completes a single outer (Newton-Raphson) iteration. Both the forward elimination and backward substitution sweeps through the external loops are performed by subroutine OUT1D and its associated routines. The calculations that these routines perform are controlled by the COMMON variable IBKS, which is set by subroutine OUTER. Subroutine OUT3D solves the hydrodynamic equations for the vessel component (IBKS = 0), or merely updates boundary data (IBKS = 1).

All one-dimensional components in a particular external loop are handled by a single call to subroutine OUT1D. OUT1D invokes the appropriate component outer iteration subroutine and returns the data to large-core memory. The outer iteration subroutines for one-dimensional components utilize subroutine INNER to perform common functions. INNER retrieves boundary information from the boundary arrays, tests other boundary information for consistency, calls subroutine DF1D to perform the appropriate hydrodynamic calculation, and resets the boundary data arrays by calling subroutine J1D. Subroutine DF1D will invoke subroutine DF1DI or DF1DS to perform fully or semi-implicit calculations, respectively.

Subroutine OUT3D solves the momentum, continuity and energy equations for the vessel component. Subroutines XSCHEM, INTFR, FILLRO and GSSOLV are the primary routines used by OUT3D to do this. XSCHEM linearizes the equations and INTFR computes the physical models. FILLRO and GSSOLV solve the linear system by direct inversion or Gauss-Seidel iteration. The boundary data arrays are updated by OUT3D.

Having evaluated the system hydrodynamic state by a sequence of outer iterations, COBRA/TRAC performs a postpass to unfold the hydrodynamic variables and update the boundary data array. This postpass is performed by overlay POST. The same overlay also implements the time step backup procedure when the outer iteration process fails to converge. When failure occurs, the outer iteration counter (OITNO) is set equal to -100 and overlay POST is invoked. Under these conditions, POST returns the component data arrays to their state at the beginning of the time step.

When the iteration converges successfully, POST calls the appropriate component postpass subroutines for the one-dimensional components and invokes subroutine POST3D for the vessel component. POST3D unfolds the values of the independent variables from the system matrix, updates the fluid densities and mass flow rates, and solves the drop interfacial area concentration equation.

When in the backup mode, the postpass routine for each one-dimensional component moves time-dependent data as needed. Subroutine THERMO reinitializes the thermodynamic arrays. The vessel variables are reset to the values they had at the beginning of the time step.

At user-specified time intervals during a transient, overlays GRAF, EDIT and DMPIT may be called. The GRAF overlay saves data for postprocess graphics (see Section 3.3). The EDIT overlay produces printed output of computed data. The DMPIT overlay saves data used to restart the simulation.

2.3 SUBROUTINE DETAILS

Programmers often find it helpful to know the purpose for particular subroutines, especially the ones that are frequently used. Since the one-dimensional component routines have not been changed appreciably, the reader is referred to the TRAC-PD2 Manual, Volume 1, (Ref. 1) for details on them. The following discussion describes the vessel subroutines briefly.

2.3.1 Input/Initialization Routines

- CLEARC sets necessary vessel COMMON variables to zero before processing the input.
- COBRAI drives the vessel input by calling SETIN, SETUP, SETOUT and HEATIN.
- HEATIN obtains a steady-state rod boundary solution and sets rod initial conditions.
- RESTRT reads a file saved from a previous simulation to reload the necessary vessel COMMON blocks for a restarted run.
- SETIN reads the vessel component input.
- SETOUT prints a summary of the vessel component input data.
- SETUP initializes additional data based on the data read by subroutine SETIN.
- SSTEMP calculates steady-state rod temperatures for subroutine HEATIN.

2.3.2 Property Routines

- AFROMH computes the equilibrium void fraction from the mixture enthalpy.
- PROP calculates fluid and vapor properties and groups of thermodynamic properties for use by various correlations.
- SAT calculates saturation enthalpy and density for liquid and vapor.
- TGAS computes the vapor temperature from pressure and enthalpy.
- VOLLIQ computes liquid specific volume at a specified enthalpy and pressure.
- VOLVAP computes vapor specific volume at a specified enthalpy and pressure.

2.3.3 Connector Routines

- BC2 sets up hydrodynamic cell connection logic for the vessel.

- CIVSSL initializes the vessel connection boundary data.

2.3.4 Heat Transfer Solution Routines

- BOILING creates the boiling curve for the fixed heat transfer nodes.
- CURVE1 performs linear interpolation in data tables.
- HCOOL selects heat transfer regimes and calculates heat transfer coefficients.
- HEAT drives the heat transfer solution; it calculates phasic heat input and explicit vapor generation for the hydrodynamic solution.
- MOVE adjusts the rod arrays after heat transfer node insertion or deletion.
- QFRONT inserts and deletes heat transfer nodes in the rods.
- TEMP solves the rod conduction equation for new time rod temperatures.

2.3.5 Fluid Solution Routines

- BACOUT unfolds the linear variation of the independent variables from the reduced linear equations for each cell.
- DVDHL computes the derivative of specific volume with respect to enthalpy for the liquid at constant pressure.
- DVDHV computes the derivative of specific volume with respect to enthalpy for the vapor at constant pressure.
- DVDPV computes the derivative of specific volume with respect to pressure for the vapor at constant enthalpy.
- FILLRO forms the system pressure matrix and saves the reduced linearized equations for later unfolding of independent variables.
- GSSOLV performs the Gauss-Seidel iteration or direct inversion on the pressure matrix.
- TPRESS computes the turbulent pressure and turbulent viscosity for each cell.

- STRESS computes the viscous and turbulent shear stress acting on each cell.
- QOXIDE computes the heat source from the Zircaloy metal-water reaction.
- INTFR computes all the hydrodynamic physical models and selects two-phase flow regimes.
- POST3D updates the independent and dependent variables and solves the drop interfacial area concentration equation.
- LEVSOL solves the rebalancing pressure matrix.
- PREP3D updates the vessel boundary conditions and performs prepass calculations.
- OUT3D drives the outer iteration for the vessel component.
- REDUCE performs Gaussian reduction of the linearized equations for each cell.
- REDUCEL performs Gaussian reduction of the linearized equations for each level (rebalancing).
- VELOC computes phase velocities.
- XSCHEM solves the explicit momentum equations and linearizes the scalar equations with respect to the independent variables.

2.3.6 Output Routines

- DUMPIT saves vessel COMMON data on a file to be saved for a subsequent restart.
- DMPMVY saves data for movie processing.
- CVSSL drives the vessel output routines.
- RESULT prints the vessel output.

2.3.7 Graphics Routines

- GRAF writes data to a direct access file for postprocess graphics (CALCOMP plots).
- FILLA fills an array with one-dimensional component data prior to writing the array to the postprocessing file.

2.4 INPUT/OUTPUT FILES AND LOGICAL UNITS

COBRA/TRAC reads and generates several files during execution. Some files are optional, depending upon the user's input. The following table describes the logical unit number currently associated with each file name and a brief description of the file.

TABLE 2.1. Input/Output Files for COBRA/TRAC

Logical Unit	Local File	Description
5	INPUT	COBRA/TRAC formatted input file; read-only
6	OUTPUT	COBRA/TRAC formatted output file; normally routed to a printer; write-only
7	TRCMSG	Brief printed messages are written to this file during COBRA/TRAC execution; may be routed to a teletype
8	TAPE8	Binary file containing vessel COMMON data saved by a previous run for restart purposes; read-only
9	TAPE9	Binary file containing vessel COMMON data to be saved for a subsequent restarted run; write-only
10	TAPE10	Formatted file for particle tracker movie data, write-only
11	TAPE11	Direct access file containing postprocess graphics data; read-write
12	TRCDMP	Binary file containing one-dimensional component data to be saved for a subsequent restart run; write-only
13	TRCRST	Binary file containing one-dimensional component data to be read for a restarted run; read-only
59	TAPE59	Formatted file for miscellaneous COBRA/TRAC timestep output; write-only

2.5 SMALL-CORE MEMORY TO LARGE-CORE MEMORY DATA TRANSFER

COBRA/TRAC was developed for CDC-7600 computers. Many of its unusual characteristics stem from this fact. Perhaps the most important (and also most potentially confusing) idiosyncrasy involves transfer of data between small-core memory and large-core memory.

CDC-7600 computers provide two classes of computer memory. Small-core memory (SCM) is in limited supply, but offers fast access to individual words of memory. Large-core memory (LCM) is available in much larger quantities, but access to individual items in LCM is considerably slower. Blocks of data can be transferred between SCM and LCM at speeds comparable to SCM access times, however. Manipulating this SCM versus LCM memory structure led to the current memory management logic in COBRA/TRAC.

All data for all one-dimensional components is stored sequentially in a single LCM array called ALCM. As each component is processed, data for it is transferred, in a block, from the ALCM array to an SCM array called A. This would not be too complicated except that there are many arrays within the block of data stored in the A array. Individual arrays are accessed by their relative locations in the A array. These relative locations are determined by pointers.

Pointers are integer words that contain the location in the A array at which the particular array is stored. The pointer variable LP, for example, is used to locate the pressure (P) array. If LP has the value 100, then the first word of the P array is located at A(100). Pointers are initialized for each component's arrays, based upon input parameters (number of cells, etc.) for that component. The pointers are used to pass the arrays within the A array to other subroutines. For example, to pass the pressure array to a ficticious subroutine SUB, the following call list could be used:

```
CALL SUB (A(LP))
```

When control reaches SUB, the pressure array is operated on as if it were the P array:

```
SUBROUTINE SUB(P)
DIMENSION P(1)
.
.
.
RETURN
END
```

Transfer between the ALCM array (in LCM) and the A array (in SCM) and back again is accomplished by two simple subroutines: RDLCM (to read LCM) and WRLCM (to write LCM). As each component is processed, the location of its data in the ALCM array is determined (again via a pointer) and the specified number of words of data is transferred, in a block, from ALCM to A.

This whole process makes the one-dimensional component portion of COBRA/TRAC very flexible since a wide variety of problems can be solved without changing the dimensions for either the ALCM or the A arrays. Unfortunately, there are also some inherent problems. It is difficult to find errors in the one-dimensional component logic (and also very easy to introduce such errors) since the pointer manipulations are quite complex. Also, there is some evidence that transferring long call lists to subroutines increases execution time significantly.

Variables for the vessel component are stored in arrays with specified dimensions, the majority of them in LCM. However, since LCM is word-addressable (individual data items are accessible without accessing a block), all of the vessel data transfers from LCM to SCM are transparent to the user or programmer.

2.6 COBRA/TRAC COMMON BLOCKS

There are only four (4) major COMMON blocks used by the one-dimensional components. Each COMMON block, however, is used for all of the one-dimensional components, via complex transfers of data between SCM and LCM. These data transfers are perhaps the most obscure and confusing part of the COBRA/TRAC logic.

There are four (4) basic sets of information required for each of the one-dimensional components. The "fixed-length table" contains basic information that does not depend on the component type. The "pointer table" contains integer words that define where the individual component arrays begin within the larger array in blank COMMON. The "variable length table" contains certain information that depends on the component type (junction numbers, geometry data, etc.). Finally, the A array in blank COMMON contains the variable arrays (pressure, enthalpy, void fraction, etc.) for the component under consideration. All four sets of data are maintained in the ALCM array for each component in the system. Since only one component is used at any one time, four (4) COMMON blocks are sufficient to transfer the component data between routines. As each component is processed, the COMMON blocks FIXLTAB (the fixed length table), PTAB (the pointer table) and VLTAB (the variable length table) are filled.

Programmers should note that the lengths of these COMMON blocks are not consistent throughout COBRA/TRAC. For example, the PTAB COMMON block is 300 words long in some places and fewer than 300 in others. Although it works very well on LANL's computer system, this is not a very good idea since it may generate some compiler problems on non-CDC computers.

The Fixed-Length Table

The fixed length table is filled first. An array, FLT, is equivalenced to the first word of the FIXLTAB COMMON block so that it can be filled by transferring one block of data from the ALCM array. Table A.1 in Appendix A shows the contents of the FIXLTAB COMMON block.

The Variable Length Table

The variable length table is the second common block filled. An array, VLT, is equivalenced to the first word of the VLTAB COMMON block so that it, too, can be filled by transferring one block of data from the ALCM array. The structure and size of the VLTAB common block depends upon the component type. This may be confusing, since each one-dimensional component redefines the structure of this COMMON block. An accumulator variable length table, for example, has little or no resemblance to a valve variable length table. This bit of sleight of hand is accomplished by redescribing the COMMON block for each component type in each of the routines that manipulate that component. Table A.2 in Appendix A shows the content of the variable length table for eacs ne-dimensional component type.

The Pointer Table

The pointer table is the third common block filled. An array, PT, is equivalenced to the first word of the PTAB COMMON block. One block of data is transferred from the ALCM array to fill the pointer table, just as were the fixed length and variable length tables. Four (4) sets of pointers are usually required to define the pointer table for a component: 1) pointers that refer to variables stored for both the previous and the current time step; 2) pointers that refer to hydrodynamic variables; 3) pointers that refer to integer variables; and 4) pointers that refer to variables used in wall heat transfer calculations. In addition, each component may define some pointers that are unique to itself.

The structure of the PTAB COMMON block is unusual. The pointer table COMMON block for a particular component is constructed by one or more references to the same COMMON block within the same subroutine. An example is shown in Figure 2.11. In this example, the first 40 locations of PTAB are filled with the old- and new-time variable pointers (DUALPT). The next 30 locations of PTAB are filled with hydrodynamic variable pointers (HYDROPT). Integer variable pointers (INTPT) occupy the next three locations. Then 12 locations are filled by the wall heat transfer variable pointers (HEATPT). A final 216 locations are occupied by the PTDUM array. (It is not typical FORTRAN syntax to see several definitions of the same COMMON block within a particular subroutine. Each successive reference to PTAB expands upon the previous definition of PTAB, which may cause problems for non-CDC FORTRAN compilers.) Table A.3 in Appendix A describes the pointer table for each of the one-dimensional components. After the pointer table has been read into COMMON block PTAB, each of the pointers is adjusted by the variable IFREE.

```
                                                            PIPEVLTAB     2
      COMMON/VLTAB/NODES,NCELLS,JUN1,JUN2,MAT,RADIN,TH,        PIPEVLTAB     3
     1 HOUTL,HOUTV,TOUTL,TOUTV,ICJ1,ICJ2,TYPE1,TYPE2,          PIPEVLTAB     4
     2 JS1,JS2,ISOLLB,ISOLRB,ICHF,IHYDRO,BSMASS,VLTDUM(128)    PIPEVLTAB     5
      INTEGER VLT(150)                                         PIPEVLTAB     6
      INTEGER TYPE1,TYPE2                                      PIPEVLTAB     7
      EQUIVALENCE (VLT(1),NODES)                               PIPEVLTAB     8
C                                                              PIPEVLTAB     9
C     PIPE POINTER TABLE                                       PIPEPT        2
       COMMON/PTAB/LALP,LALPD,LALV,LARA,LAREL,LAREV,LARL,LARV, DUALPT        2
     1LBIT,LEA,LEL,LEV,LHILO,LHIVO,LHLV,LP,LPA,                DUALPT        3
     1LROA,LROL,LROV,LTD,LTL,LTV,LVRD,LVM,LTW,LVL,LVV,LALPN,   DUALPT        4
     2LALPDN,LALVN,LARAN,LARELN,LAREVN,LARLN,LARVN,            DUALPT        5
     2LBITN,LEAN,LELN,LEVN,LHIL,LHIV,LHLVN,LPN,LPAN,LROAN,LROLN, DUALPT      6
     3LROVN,LTDN,LTLN,LTVN,LVRDN,LVMN,LTWN,LVLN,LVVN           DUALPT        7
      INTEGER PT(1)                                            DUALPT        8
      EQUIVALENCE (PT(1),LALP)                                 DUALPT        9
C                                                              DUALPT       10
      COMMON/PTAB/LB,LCFZ,LCL,LCPL,LCPV,LCV,LDRIV,LDX,         HYDROPT       2
     1LFA,LFRIC,LGRAV,LHD,LHFG,LQPPP,LRMEM,LRMVM,              HYDROPT       3
     2LROM,LRHS,LSIG,LTRID,LTSAT,LTSSN,LVISL,LVISV,LVOL,LVR,LWA HYDROPT      4
     3,LDFVDP,LDFLDP,LVLT,LVVT,LWFL,LWFV,LCIF                  HYDROPT       5
C                                                              HYDROPT       6
      COMMON/PTAB/LIDR,LMATID,LNFF                             INTPT         2
C                                                              INTPT         3
      COMMON/PTAB/LCPW,LCW,LDR,LEMIS,                          HEATPT        2
     1LHOL,LHOV,LQPPC,LRN,LRN2,LROW,LTOL,LTOV                  HEATPT        3
C                                                              HEATPT        4
      COMMON/PTAB/PTDUM(214)                                   PIPEPT        7
```

FIGURE 2.11. Illustration of Multiple References to the Pointer Table COMMON Block Within a Subroutine

The DO loop usually looks something like:

```
      DO 100 I = 1,LENPTR
      PT(I) = PT(I) + IFREE - 1
  100 CONTINUE
```

This is done so that the pointers refer to the actual locations of the arrays in the A array, not relative locations.

Data Arrays

The data arrays for the component are the fourth COMMON block transferred from the ALCM array to the unused portion of the A array. This unused portion of the A array begins at the subscript stored in the variable IFREE. (See Table A.4 in Appendix A for a description of the arrays for each component type.)

"ALCM" and "A" Array Structure

The ALCM array contains all four sets of data (the fixed length table, variable length table, and pointer table plus the data arrays) for all of the components. The data for each component is stored in a consecutive block, as illustrated in Figure 2.12. Data is stored for the components in the order read during the input portion of the execution.

The A array is used by COBRA/TRAC to store a multitude of variables. Many arrays are kept within the A array in addition to the component data arrays already discussed (see Table A.5 in Appendix A). Figure 2.13 shows the structure of the A array.

VESSEL COMMON Blocks

The COMMON blocks for the vessel component do not resemble the one-dimensional component COMMON blocks in any way. The vessel component does not use the ALCM and A arrays for data transfer. The vessel COMMON blocks contain arrays used in the vessel solution and problem geometry, and the sizes of the arrays are set by running the SPECSET dimensioning preprocessor (see Volume 3, Users' Manual, for a description of this process). Appendix B contains lists of the COMMON blocks used for the vessel component, with definitions of the variables.

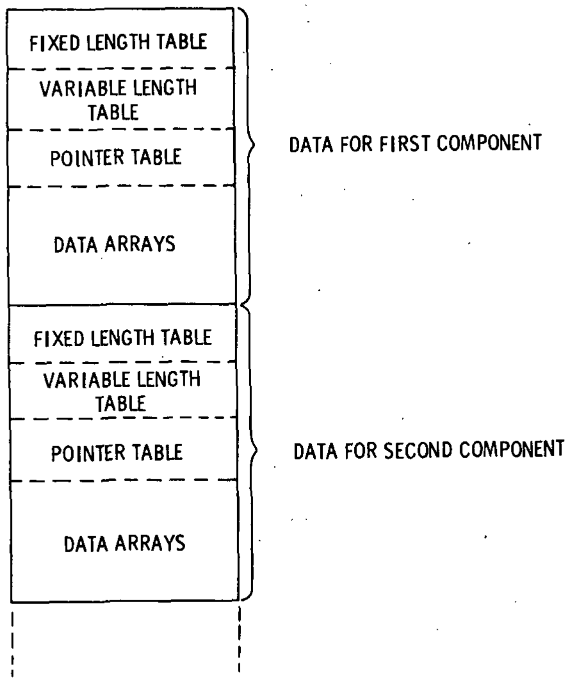

FIGURE 2.12. Sequence of Data Storage in ALCM Array

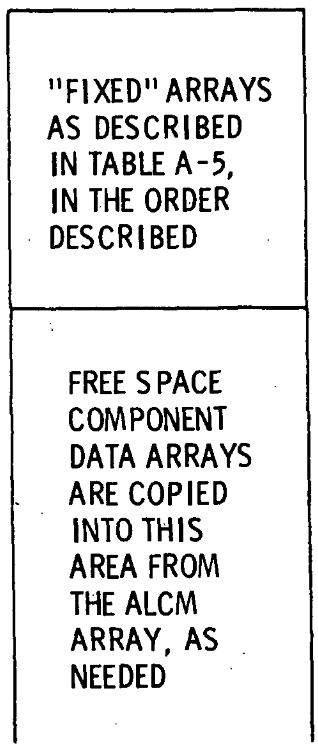

FIGURE 2.13. Sequence of Data Storage in A Array

Solution Control COMMON Blocks

The solution control COMMON blocks other than the one-dimensional COMMON and the vessel COMMON are used by COBRA/TRAC to control the solution. They contain the elapsed time, edit and graphics intervals, and so on. They are described in detail in the TRAC-PD2 Manual, Volume 1 (Ref. 1).

2.7 DUMP/RESTART FEATURE

COBRA/TRAC simulations are often lengthy and can be quite expensive. Therefore it is often prudent to perform a long simulation in several short segments to monitor results and minimize wasted computer time. COBRA/TRAC provides a dump/restart capability. Basically, this involves writing all the necessary information on two files that can be read into memory to fill the arrays with all relevant information at the time step at which the problem stopped. One-dimensional components can be added or removed to simulate changes in the system geometry occurring at a certain time in the transient, such as activating emergency core cooling system, etc.

The logic for dumping and restarting the one-dimensional components in COBRA/TRAC differs significantly from the corresponding logic in TRAC-PD2. The TRAC part of the restart dump was rewritten when it proved impossible to get consistent results on the restart with version TRAC/P1A. Specifically, a case run for 2 seconds without a restart did not yield exactly the same results as the same problem run for 1 second, then restarted and run for another second. The current restart logic meets this test without difficulty.

The one-dimensional component data is written to the dump file (TRCDMP) in the following manner. The locations of each component's arrays in LCM are determined. All of the data for that component (including the pointers and fixed and variable length tables) are written to the dump file.

When the run is restarted the user may add or replace components. The input deck for the restarted run includes only those components added or changed. COBRA/TRAC first reads the input file and generates a list of missing components (i.e., components that must be on the restart file, since they were not in the input file). Then the data from the restart file is read

(TRCRST), matching component numbers with those on the missing component list. If the component number is on the missing component list the code stores the data away in ALCM and removes its number from the missing list. If a component number is not on the missing component list, its data from the restart file is ignored.

The advantage of this method is that all the data for the one-dimensional components are saved, rather than selected arrays. The initialization logic is skipped for those components whose input comes from the restart file, and as a result, a case that has been run out with a series of restarts matches exactly the same case when it is run continuously from beginning to end without restarting.

The data for the vessel component is saved to a separate dump file (TAPE9). The pertinent COMMON blocks are written to the dump file and are read back when the run is restarted from the TAPE8 file.

2.8 GRAPHICS PROCESSING

Printed output from COBRA/TRAC can become quite voluminous. Simply storing all of its output is a problem, let alone understanding it. A means is provided to store data for subsequent postprocessing. The postprocessor program GRAFIX reads data saved by COBRA/TRAC and condenses it into graphical form so that the user can easily see the results of the simulation and show comparisons with experimental data.

The postprocessing data file is written to logical unit 11. This file is created by "direct-access" writes. Direct access means that the file can be written or read in a nonsequential manner. A data item can be accessed directly without having to read through all of the items that precede it on the file. (The direct access file can be processed from front-to-back, back-to-front, or in any other order the user desires.)

The chief advantage of this direct access file is that data can be retrieved very rapidly; this is handy when many different kinds of data must be retrieved from the same file many times. Another advantage is that, since direct access files can be both read and written during the same execution, it is possible to put the postprocessing results from many segments of a simulation into the same file. This eliminates the need to maintain (and merge together later) several separate postprocessing files from several segments of a simulation.

The postprocessing file contains the following information:

- heading records that contain the next available location on the file and the number of entries in the file
- a directory that contains the actual location for the data in the file for each component at each time step
- the data for each component at each time step.

The structure of the postprocessing file is very complex. Ordinarily the user need not concern himself with this structure, since both COBRA/TRAC and the graphics postprocessing program GRAFIX take care of all the details.

Before any given calculation begins, COBRA/TRAC either creates space for a directory on the postprocessing file or reads the directory from the existing postprocessing file (in the case of a restarted run). The postprocessing file must have read/write permission throughout COBRA/TRAC execution. (If the COBRA/TRAC execution is a restart of a previous execution, and if the postprocessing file is being written, the postprocessing file from the previous execution must be available for input and output for COBRA/TRAC.) As execution proceeds, subroutine GRAF is called at user-specified intervals. GRAF constructs a record for each component the user wants saved to the postprocessing file. The record is then written in the file and the header and directory records are updated accordingly.

The COMMON block /MVYDAT/ controls the graphics processing. Of particular interest is the variable NXTLOC. NXTLOC contains the location (word counter) for the next record to be written. The array INDCMP in COMMON block /MVYDT3/ contains an index. The value of INDCMP(5,94) is the location of the graphics record for the fifth component, the 94th graphics dump, for example.

A listing of the graphics program is provided in Appendix C. Also provided in Appendix C is a listing of the program used on a mini-computer (VAX11/780) to produce plots on a Calcomp plotter.

3.0 USING COBRA/TRAC

Because COBRA/TRAC is so complex, there are many steps involved in its use. This section is intended to guide the user past the pitfalls and to describe effective use of COBRA/TRAC. Examples will be given for typical CDC scope control statements that could be used to execute COBRA/TRAC. Refer to Figure 3.1 for an overview of COBRA/TRAC operation.

Included here are examples of the job control language (JCL) for both beginning and restarting a simulation with COBRA/TRAC. The use of the graphics and postprocessor program GRAFIX also is discussed.

3.1 BEGINNING A SIMULATION

Details for setting up data for a problem are not provided in this manual. The user is referred to the Users Manual (Volume 3) for that information. Instead, it is assumed that data are now ready for input to COBRA/TRAC.

CDC SCOPE control statements to accomplish the first step for this simulation are:

REQUEST(TRCDMP,*PF) REQUEST(TAPE9,*PF) REQUEST(TAPE11,*PF)	1. Request permanent file space for COBRA/TRAC output files.
ATTACH(BIN,COBRATRACB, etc.)	2. Get the current COBRA/TRAC executable file.
BIN.	3. Execute COBRA/TRAC.
CATALOG(TRCDMP,ONEDCOMP, etc.) CATALOG(TAPE9,VESSELDMP, etc.) CATALOG(TAPE11,GRFDMP, etc.)	4. Save COBRA/TRAC outputs on permanent file space for later access.
EXIT(S)	5. Abnormal exit option.

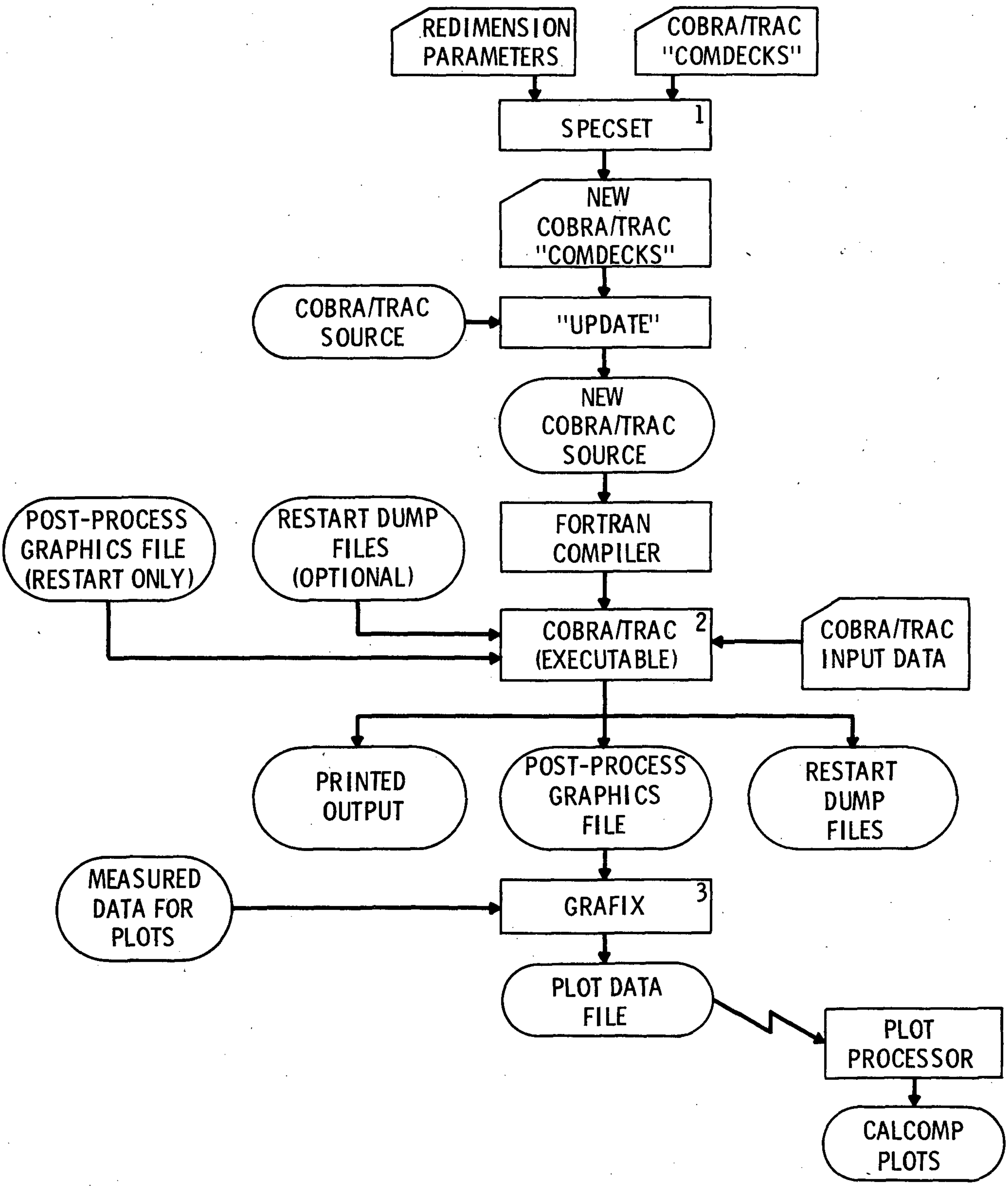

FIGURE 3.1. Flow Diagram of COBRA/TRAC Operation

CATALOG(TRCDMP,ONEDCOMP, etc.) CATALOG(TAPE9,VESSELDMP, etc.) CATALOG(TAPE11,GRFDMP, etc.)	6.	Save COBRA/TRAC output, no matter what happens (machine failure, etc.).

If there is no need to ever restart the simulation the REQUEST and CATALOG cards for the TRCDMP and TAPE9 files may be omitted. If no postprocess graphics data are being saved, the REQUEST and CATALOG cards for TAPE11 may be omitted.

The EXIT(S) card is included to provide an alternate route should the machine quit or the simulation terminate abnormally. The input data for COBRA/TRAC is assumed to come from a local file called INPUT. All of the printed results are written to a file called OUTPUT, which is routed to a printer or microfiche device.

3.2 RESTARTING A SIMULATION

The CDC control statements to restart the COBRA/TRAC simulation are:

ATTACH(TRCRST,ONEDCMP, etc.) ATTACH(TAPE8,VESSELDMP, etc.) ATTACH(TAPE11,GRFDMP, etc.)	1.	Find the files saved by the previous execution of COBRA/TRAC. TRCRST contains the one-dimensional component data; TAPE8 contains the data saved for the vessel; TAPE11 contains the data saved for postprocess graphics.
ATTACH(BIN,COBRATRACB, etc.)	2.	Find the current COBRA/TRAC executable file.
BIN.	3.	Execute COBRA/TRAC.
CATALOG(TRCDMP,ONEDCOMP, etc.) CATALOG(TAPE9,VESSELDMP, etc.) CATALOG(TAPE11,GRFDMP, etc.)	4.	Save COBRA/TRAC output files.
EXIT(S)	5.	Abnormal exit option.

CATALOG(TRCDMP,ONEDCOMP, etc.)	6.	Save COBRA/TRAC outputs in case of failure.
CATALOG(TAPE9,VESSELDMP, etc.)		
CATALOG(TAPE11,GRFDMP, etc.)		

Some of the REQUEST and CATALOG cards may be omitted for the various output files, depending on the options selected. The COBRA/TRAC input is assumed to be on a file called INPUT, and output is written to a file called OUTPUT. The INPUT file for a restarted run is usually much smaller than the fresh start deck, since most of the component data is not read in for a restart. (Instead this data comes from the TRCRST and TAPE8 files.) Only new or replaced components are entered. Details about adding, subtracting or replacing components may be found in Volume 3, the Users' Manual.

3.3 GRAFIX USE

GRAFIX is an interactive program used to produce CALCOMP plots of any variable saved by a COBRA/TRAC simulation. A variety of plot types may be produced, including variables versus time, variable versus axial distance and contours of velocity vectors. Measured data can be plotted against the COBRA/TRAC results if desired.

GRAFIX reads the postprocessing file in exactly the same manner that COBRA/TRAC writes it. If measured data is to be plotted, a separate measured data file must be provided for each plot to be produced. GRAFIX produces a file that consists of plot size and label information and a set of coordinates for the curve(s) to be plotted. This file is then sent to a minicomputer to produce the plots. (GRAFIX could easily be modified to produce the plots itself, but since COBRA/TRAC and GRAFIX are run at LANL and the plotter is at PNL, it was necessary to separate the actual plotting from GRAFIX.)

Instructions for setting up measured data files and a sample GRAFIX input are produced in Volume 3, the Users' Manual.

4.0 MODIFYING COBRA/TRAC

Unfortunately, the code has not yet been written that does not need occasional modification or correction. Changes to COBRA/TRAC hopefully will be minor, but if the need arises, a few words to the wise are in order. The UPDATE utility, the program libraries and recommendations for modifying RESPEC and GRAFIX are discussed. The programmer is cautioned not to undertake changes to COBRA/TRAC too lightly. Fully understanding the overall operation of COBRA/TRAC is vital before making any modifications. Specifically, changes should not be made to the one-dimensional component arrays. Most changes to COBRA/TRAC will be required to establish it on a computer system other than LANL's CDC computers. Specific suggestions for this conversion are provided in the next section of this manual.

4.1 CDC "UPDATE"

The codes in the COBRA/TRAC package are provided in CDC "UPDATE" source form. UPDATE is software available on most CDC computers to aid in maintenance of computer programs. It allows for a wide variety of changes to the source code and provides an "audit trail" of all code changes in chronological order.

All of the subroutines that make up a program are stored in a single file. In UPDATE terminology, this file is called a "program library". Each subroutine is stored in a unique element of the program library called a "deck". Every subroutine is preceded by a *DECK card. The *DECK card contains an alphanumeric name associated with that subroutine. A COMMON block (or a set of COMMON blocks) is stored in a unique element of the program library called a "COMDECK". The COMMON block (or set of COMMON blocks) is preceded by a *COMDECK card. The *COMDECK card, like the *DECK card, assigns an alphanumeric name to the COMMON block.

In any given subroutine, wherever a particular COMMON block is needed, a *CALL statement is inserted. The UPDATE facility can create a file for FORTRAN compilation. At the time the "compile" file is created, the corresponding COMMON blocks are inserted in place of the *CALL statements.

This makes changing COMMON blocks very easy since once a change to a COMMON block is made, it is carried throughout the code automatically.

4.2 COBRA/TRAC PROGRAM LIBRARIES

The UPDATE program libraries (in UPDATE source form) provided as the COBRA/TRAC package are COBRA/TRAC, RESPEC and GRAFIX.

COBRA/TRAC consists of many comdecks and subroutines, too many to list here. The COMMON blocks are the first elements of the COBRA/TRAC file; each COMMON block is preceded by a *COMDECK card. The main program and all of the subroutines follow, each preceded by a *DECK card.

RESPEC consists of two routines, the main program RESPEC and a simple subroutine BUFF. RESPEC performs all of the necessary redimensioning parameter calculations, reads the undimensioned vessel COMDECKS file, and produces a new vessel COMDECKS file with the user-specified dimensions in place. BUFF is called to write individual records to the new COMDECKS file.

4.3 CHANGING THE RESPEC PROGRAM

The RESPEC program is a simple code and therefore easy to modify. New RESPEC parameters can be incorporated either by adding them to the input list or having them calculated in RESPEC as functions of one or more input parameters. One important point to remember is that any new dimensioning parameter can have a name only <u>two</u> (2) characters in length; the first character can be I, J, K, L, M or N; the second character can be any of the letters A to Z or the digits 1 to 9. (It is important to determine that any new parameter defined by the programmer is really new--it cannot be in use elsewhere or the results will be predictably bad.)

Changes to the RESPEC procedure fall into one of two categories:
1) changes to the COMDECKS file that use existing RESPEC parameters, and
2) changes to the COMDECKS file that require new RESPEC parameters.

The first case, in which the user wants to add a variable or COMMON block to COBRA/TRAC that uses existing RESPEC parameters, is the easiest. In this event, the necessary changes to the COMDECKS file can be made and the RESPEC procedure executed as usual--no changes to the RESPEC program are needed.

The second case, where new RESPEC parameters are required, is somewhat more complicated. In this case, the programmer must select a RESPEC parameter that is not already in use and make the changes to the COMDECKS as required. Changes to RESPEC may or may not be required.

If the RESPEC parameter is not dependent on any other parameter for its value, no change to RESPEC is required. A new record is inserted in the RESPEC parameter input to assign the parameter a value (e.g., JA = 75), and the RESPEC procedure is executed as usual.

If, on the other hand, the value of the new parameter depends on an existing parameter's value, RESPEC will need to be changed to perform the calculation. Suppose, for example, that a new parameter, JA, must be computed as JA = 4*MC + 1. To perform the calculation in RESPEC, insert the FORTRAN statement JA = 4*MC + 1 in the appropriate position (generally immediately after the 270 CONTINUE statement). Then RESPEC must be recompiled and executed. (A listing of program RESPEC and the undimensioned COMMON is provided in Appendix D of this manual).

4.4 CHANGES TO GRAFIX

Of the codes provided with the COBRA/TRAC package, GRAFIX is the most likely candidate for code changes. While most contingencies have been provided for, someone will always want something a little different for his plot.

The programmer will have to become familiar with the structure of the direct-access file saved by COBRA/TRAC. Then GRAFIX will have to be modified to: 1) ask the user for any needed input information for the new type of plot, and 2) perform the necessary selection of the variable(s) from the postprocessing file.

Another likely source of GRAFIX modification arises when establishing it on a non-LANL CDC computer. Section 5 supplies suggestions for this conversion.

5.0 ESTABLISHING COBRA/TRAC ON A NON-LANL COMPUTER

COBRA/TRAC and its auxiliary codes have been designed for use on the CDC 7600 class of computers, specifically CDC 7600 computers that operate under the Livermore Time Sharing System (LTSS) at LANL. This does not mean, however, that COBRA/TRAC cannot be made to run elsewhere. It has been successfully converted to a CRAY computer, for example. COBRA/TRAC should be applicable quite easily to any CDC computer facility of the same class as the 7600 (CDC 175 or 176, for example) with relatively few modifications. Applying COBRA/TRAC to a computer such as the IBM/370 or UNIVAC 1100 will be considerably more difficult. This section provides suggestions for establishing COBRA/TRAC on any non-LANL computer system.

5.1 CDC CONVERSION

If a CDC computer system is available, the conversion of COBRA/TRAC will be fairly simple. This is especially true if the CDC UPDATE facility is available.

If UPDATE is not available, the COBRA/TRAC source must be changed to eliminate the UPDATE references (*COMDECK, *DECK and *CALL). A mechanism for inserting appropriate COMMON blocks at the location of *CALL records must be devised (many compilers use an INCLUDE statement for this purpose). Any further changes will have to be made by a text editor or by whatever similar processor is available.

If UPDATE is available, the COBRA/TRAC source file must be provided as input to UPDATE so that a program library can be created.

Once the program library has been established, a "compile" file for input to the FORTRAN compiler must be generated, again via UPDATE. This file is then compiled and errors, if any, detected. (There should be none.)

5.1.1 Overlay Modifications

Difficulties may be encountered with the overlay structure because the method used at LANL is not a standard CDC overlay system. (See Section 2.1 for a discussion of the overlay scheme.) The standard CDC overlay reference is of the form:

```
CALL OVERLAY(FNAME,I,J)
```

where:

FNAME	is an optional file name upon which the overlay is located.
I and J	are level numbers designating the overlay desired.

Each overlay must be preceded by an OVERLAY statement of the form:

```
OVERLAY(FNAME,I,J)
```

where:

FNAME	is an optional file name (except for the first overlay, which must be the 0,0 overlay) upon which the file name is required.
I and J	are level numbers designating the overlay desired.

This contrasts with the LANL system since the CALL OVERLAY statement looks like:

```
CALL OVERLAY(HOLL,I,J)
```

where:

HOLL	is not used.
I and J	have the same usage as the standard CDC reference.

On the LANL system, the OVERLAY statements are not required because an auxiliary file defines the entry point for each overlay.

Therefore, to overlay COBRA/TRAC on a more standard CDC operating system the following recommendations are made:

1. Insert OVERLAY statements as follows:

- OVERLAY(5HCTRAC,0,0) before the PROGRAM statement in TRAC, the main program
- OVERLAY(1,0) before the subroutine statement for INPUT
- OVERLAY(1,2) before the subroutine statement for SETUP
- OVERLAY(1,3) before the subroutine statement for SETOUT
- OVERLAY(1,4) before the subroutine statement for RPIPE
- OVERLAY(1,5) before the subroutine statement for RPUMP
- OVERLAY(1,6) before the subroutine statement for RTEE
- OVERLAY(1,7) before the subroutine statement for RVLVE
- OVERLAY(1,8) before the subroutine statement for RBREAK
- OVERLAY(1,9) before the subroutine statement for RFILL
- OVERLAY(1,10) before the subroutine statement for RPRIZR
- OVERLAY(1,11) before the subroutine statement for RSTGEN
- OVERLAY(1,12) before the subroutine statement for RACCUM
- OVERLAY(2,0) before the subroutine statement for INIT
- OVERLAY(3,0) before the subroutine statement for DMPIT
- OVERLAY(4,0) before the subroutine statement for EDIT
- OVERLAY(5,0) before the subroutine statement for GRAPH
- OVERLAY(6,0) before the subroutine statement for OUTER
- OVERLAY(6,1) before the subroutine statement for OUT1DN
- OVERLAY(6,2) before the subroutine statement for OUT3D
- OVERLAY(11,0) before the subroutine statement for PREP
- OVERLAY(12,0) before the subroutine statement for POST
- OVERLAY(13,0) before the subroutine statement for BLKDAT

2. Replace the CALL OVERLAY statements in COBRA/TRAC as follows:

- In TRAC

Replace	With
CALL OVERLAY(6HBLKDAT,13,0)	CALL OVERLAY(13,0)
CALL OVERLAY(6HCINPUT,1,0)	CALL OVERLAY(1,0)
CALL OVERLAY(4HINIT,2,0)	CALL OVERLAY(2,0)
CALL OVERLAY(5HDMPIT,3,0)	CALL OVERLAY(3,0)

- In ERROR and TIMCHK

Replace	With
CALL OVERLAY(4HEDIT,4,0)	CALL OVERLAY(4,0)
CALL OVERLAY(5HDMPIT,3,0)	CALL OVERLAY(3,0)

- IN TRANS

CALL OVERLAY(4HEDIT,4,0)	CALL OVERLAY(4,0)
CALL OVERLAY(4HGRAF,5,0)	CALL OVERLAY(5,0)
CALL OVERLAY(4HPREP,11,0)	CALL OVERLAY(11,0)
CALL OVERLAY(5HOUTER,6,0,6HRECALL)	CALL OVERLAY(6,0,6HRECALL)
CALL OVERLAY(4HPOST,12,0)	CALL OVERLAY(12,0)
CALL OVERLAY(5HDPMIT,3,0)	CALL OVERLAY(3,0)

- In COBRAI

Replace	With
CALL OVERLAY(5HSETIN,1,1)	CALL OVERLAY(1,1)
CALL OVERLAY(5HSETUP,1,2)	CALL OVERLAY(1,2)
CALL OVERLAY(6HSETOUT,1,3)	CALL OVERLAY(1,3)

- In OUTER

Replace	With
CALL OVERLAY(6HOUT1DN,6,1)	CALL OVERLAY(6,1)
CALL OVERLAY(5HOUT3D,6,2)	CALL OVERLAY(6,2)

The overlays in OUTER may help alleviate a memory shortage for very large problems. The programmer is cautioned that this may incur an input/output penalty, since the OUTER overlay is the most heavily used.

If memory is in ample supply, it is suggested that the following replacement be made.

Replace	With
CALL OVERLAY(6HOUT1DN,6,1)	CALL OUT1DN
CALL OVERLAY(5HOUT3D,6,2)	CALL OUT3D

The overlay statements for OUT1DN and OUT3D will need to be changed accordingly in this case.

3. The subroutine statements for the first routine in each overlay (INPUT,INIT,SETUP,SETOUT,RPIPE,RPUMP,RTEE,RVLVE,RBREAK,RFILL,RPRIZR, RSTGEN,RACCUM,DMPIT,EDIT,GRAF,PREP,OUTER,OUT1DN,OUT3D and POST) must be replaced by PROGRAM statements. For example, the statement

```
SUBROUTINE INPUT
```

must be changed to:

```
PROGRAM INPUT
```

4. Some routines may need to be physically moved around inside the program library. This is because all routines referenced by a particular overlay must be located between the OVERLAY statement for that level and the succeeding OVERLAY statement. The only exception to this rule is that an overlay may reference a routine loaded in a higher-level overlay.

5. If there is not enough memory available--approximately 150,000-octal words of SCM and 400,000-octal words of LCM--additional adjustments to the overlay structure may be required.

5.1.2 Direct-Access Input/Output Modifications

Direct-access input and output is used to generate the file TAPE11 used for the postprocess graphics. The mechanism for this direct-access input and output uses a set of LANL library routines. These calls will have to be modified for non-LANL CDC systems.

COBRA/TRAC uses RDABSF to read the direct-access file and WRABSF to write the direct-access file. The calls to these routines are of the form:

```
CALL RDABSF(11,ARRAY,NWDS,NREC)
```

and

```
CALL WRABSF(11,ARRAY,NWDS,NREC)
```

where: 11 is the logical unit number for the direct-access file

ARRAY is the array from which data is to be written or into which data is to be read

NWDS is the number of words to be read or written into or out of ARRAY

NREC is the <u>word</u> address in the direct-access file at which to begin the read or write.

CDC FORTRAN compilers usually provide routines to accomplish similar functions--OPENMS, WRITMS, READMS and CLOSMS. OPENMS and CLOSMS "open" (create) and "close" (complete) the direct-access file. READMS and WRITMS perform direct-access read and write operations. For establishing COBRA/TRAC on a more standard CDC system, the following recommendations are made:

1. In TRAC insert a CALL OPENMS statement immediately before the call to TRANS of the form:

   CALL OPENMS(11,INDEX,LENGTH,0)

   where: 11 is the logical unit number for the direct access file

   INDEX is an array used by the direct access routines and must be dimensioned (LENGTH+1)

   LENGTH is the number of records expected in the file

   0 indicates that the file is accessed by record numbers.

2. In TRAC insert a CALL CLOSMS statement immediately after the CALL TRANS statement, as follows:

   CALL CLOSMS(11)

3. In IGRAF and GRAF the following changes to the direct-access reads and writes will be necessary:

- Replace each call to RDABSF by a call to READMS of the form:

  CALL READMS(11,ARRAY,NWDS,NREC)

  ARRAY and NWDS have the same definition as in the call to RDABSF.

  NREC is the <u>record number</u> of the record to be read.

- Replace each call to WRABSF by a call to WRITMS of the form:

  CALL WRITMS(11,ARRAY,NWDS,NREC)

  ARRAY and NWDS have the same definition as in the call to WRABSF. NREC is the record number of the record to be written.

5.2 NON-CDC CONVERSION

Considerable difficulty is to be expected when establishing COBRA/TRAC on a non-CDC computer. Converting COBRA/TRAC to a non-CDC computer is not a simple task and should not be undertaken lightly, even by an expert. With that warning, some suggestions are provided.

The following types of problems are anticipated:

- CDC "UPDATE" references in the source file must be eliminated.
- The overlay mechanism must be modified.
- LCM memory identification (LEVEL 2 statements) will cause compilation errors.
- LCM to SCM data transfers will need modification.
- Direct-access file reads and writes will need modification.
- Various CDC FORTRAN statements will have to be modified. These include the PROGRAM statement, end-of-file checks, and I/O status checks.
- Some variables may require double precision to achieve the necessary accuracy during solution.

Since the solutions to these problems are very computer-dependent, only general suggestions can be given.

The CDC UPDATE directives mean nothing to a non-CDC computer. Therefore they must be removed before any progress can be made. All of the *COMDECK directives must be removed. The *CALL statements must be replaced by the COMMON blocks they reference throughout the code. (If an INCLUDE statement is allowed by the available FORTRAN compiler, physically merging the COMMON blocks into the source will not be necessary.) The *DECK cards must also be removed. The redimensioning process must be modified to work on the available system. One suggestion is to place the "undimensioned" COMMON blocks throughout the code and then RESPEC all of COBRA/TRAC, thus bypassing the UPDATE process. RESPEC should be modified so that it looks only at COMMON blocks and DIMENSION statements if this option is taken. COBRA/TRAC should be modified so that COMMON blocks and DIMENSION statements are clearly delineated from the rest of the code, by means of a special character or character set embedded in a comment line, for example.

The overlay structure may not be a problem at all if a "virtual memory" computer system is in use (available on some IBM computers). Otherwise, a suitable overlay mechanism must be implemented.

LCM memory identification means nothing to non-CDC FORTRAN compilers, so all the "LEVEL 2" statements must be deleted. In addition, LCM to SCM data transfer logic should be modified. One suggestion is to use the ALCM array as is and modify subroutines RDLCM and WRLCM to do simple data transfer between the ALCM and the A arrays.

The direct-access file reads and writes in subroutines IGRAF and GRAF should be modified. The process is similar to that outlined for the CDC conversion (see Section 5.1.2). The usual form for such direct-access reads and writes on non-CDC computers resembles:

```
READ (U'R)IOLIST
WRITE (U'R)IOLIST
```

where

U is the logical unit number.

R is the record number.

IOLIST is the data to be read or written.

An OPEN or DEFINE FILE statement (analogous to the OPENMS statement described in Section 5.1.2) will be needed to initialize the file. The resulting reads and writes in subroutines IGRAF and GRAF will resemble:

```
READ(11'NREC) (GRFX(I),I=1,NWDS)
WRITE(11'NREC) (GRFX(I),I=1,NWDS)
```

Since record numbers are used instead of word addresses for positioning the data in the direct-access file, modification to the NXTLOC and INDCMP assignments (as described in Section 5.2.8) will be necessary.

The CDC status check for end-of-file is of the form:

```
READ(LU) LIST
IF(ABS(EOF(LU)).NE.0.0) GO TO "label"
```

Control passes to "label" if an end-of-file is detected. This should be replaced by

```
READ(LU,END = "label")
```

for non-CDC computers. The I/O status check to ensure that a direct-access read or write has completed is of the form:

```
IF(UNIT(11))n,m,m
```

Control passes to statement m if the direct access I/O was successful; otherwise control passes to statement n. This logic should be replaced accordingly.

The PROGRAM statement on CDC computers serves to associate logical unit numbers with physical files. The PROGRAM statement should be removed and an appropriate method for assigning logical units to files implemented.

The LOCF function referenced by subroutines CLEARC and DUMPIT is used to determine the number of computer words between two variables. The typical usage of LOCF is:

```
IS = LOCF(IEND1) - LOCF(IBEG1)
```

When this instruction has been completed, IS contains the number of words between IBEG1 and IEND1. LOCF may not be available on non-CDC computers, but a similar function usually is, so all references to LOCF should be modified accordingly.

Routines used to determine the date, time and CPU time while COBRA/TRAC is in execution must also be replaced by appropriate routines.

5.3 CONVERTING RESPEC AND GRAFIX

RESPEC should present no particular problems on a non-LANL computer. There may be some minor FORTRAN differences, but these should be easily overcome.

GRAFIX may be a little more difficult, however. The same modifications to the direct access reads and writes are required for GRAFIX as for COBRA/TRAC. These changes are discussed in detail in Sections 5.1 and 5.2 of this manual.

REFERENCE

1. D. R. Liles and others, Los Alamos National Laboratory, "TRAC-PD2, An Advanced Best-Estimate Computer Program for Pressurized Water Reactor Loss-of-Coolant Analysis," USNRC Report NUREG/CR-2054, 1981. Available for purchase from National Technical Information Services, Springfield, Virginia, 22161.

APPENDIX A

ONE-DIMENSIONAL COMPONENT DATA STORAGE TABLES

TABLE A.1. Fixed Length Common Block (FIXLTAB)

Word	Variable	Description
1	NUM	Component number
2	TYPE	Component type
3	ID	Component identification number
4	NCELLT	Total number of cells for this component
5	LENVLT	Number of words in the variable length table for this component
6	LENPTR	Number of words in the pointer table for this component
7	LENARR	Number of words in the array data for this component
8	LFV	Relative position of fundamental variables from the previous time step
9	LFVN	Relative position of fundamental variables from the current time step
10	LENFV	Number of words in the fundamental variable arrays
11	LTDVO	Relative position of time-dependent variables from previous time step
12	LTDVN	Relative position of time-dependent variables from current time step
13	LENTDV	Number of words in the time-dependent variable arrays
14	IREST	Restart option for this component
15	LEXTRA	Not used
16	CTITLE(1)	Component identifying label, three words long

TABLE A.2. Variable Length Tables for One-Dimensional Components

Component Type	Word	Variable	Description
Accumulator	1	NODES	Number of heat transfer nodes (not used)
	2	NCELLS	Number of fluid cells
	3	JUN2	Junction number at discharge
	4	QINT	Initial water volume
	5	QOUT	Volume of discharged water
	6	TYPE2	Adjacent component type
	7	ICJ	Iteration index of adjacent component
	8	IUV1	Velocity update indicator at junction 1 (not used)
	9	IUV2	Velocity update indicator at junction 2
	10	JS2	Junction sequence number at the discharge
	11	Z	Water height above discharge
	12	FLOW	Volume flow rate at discharge
	13	ISTOP	Accumulator empty flag
	14	BSMASS	Time-integrated mass flow rate out of accumulator
Break	1	NODES	Number of heat transfer nodes (not used)
	2	JUN1	Junction number at discharge
	3	ICJ	Iteration index of adjacent component
	4	TYPE1	Adjacent component type
	5	JS1	Junction sequence number
	6	BXMASS	Current mass flow rate out of break
	7	BSMASS	Time-integrated mass flow rate out of break
	8	IBROP	Table option
	9	NBTB	Number of points in tables
	10	KPOINT	Pointer to last utilized entry in the velocity table
	11	ISAT	Option controlling break table use

TABLE A.2 (continued)

Component Type	Word	Variable	Description
	12	TIN	Fluid temperature
	13	INEXTI	Implicitness level of adjacent component
Fill	1	NODES	Number of heat transfer nodes (not used)
	2	JUN1	Junction number
	3	ICJ	Iteration index of adjacent components
	4	TYPE1	Adjacent component type
	5	JS1	Junction sequence number at first junction
	6	FXMASS	Current mass flow rate out
	7	FSMASS	Time-integrated mass flow rate out
	8	IFTY	Fill type option
	9	IFTR	Fill trip number
	10	NFTX	Number of points in the fill table
	11	KPOINT	Pointer to last used entry in the velocity table
	12	INEXTI	Implicitness level of adjacent component
	13	FLOWIN	Initial mass flow into or from adjacent component
Pipe	1	NODES	Number of heat tranfer nodes (not used)
	2	NCELLS	Number of fluid cells
	3	JUN1	Number of junction at low-numbered end of pipe
	4	JUN2	Number of junction at high-numbered end of pipe
	5	MAT	Material type
	6	RADIN	Inner radius of pipe wall
	7	TH	Thickness of pipe wall
	8	HOUTL	Heat transfer coefficient between wall and surrounding liquid
	9	HOUTV	Heat transfer coefficient between wall and surrounding vapor
	10	TOUTL	Liquid temperature outside of pipe

TABLE A.2 (continued)

Component Type	Word	Variable	Description
	11	TOUTV	Vapor temperature outside of pipe
	12	ICJ1	Iteration index of adjacent component at junction 1
	13	ICJ2	Iteration index of adjacent component at junction 2
	14	TYPE1	Adjacent component type at junction 1
	15	TYPE2	Adjacent component type at junction 2
	16	JS1	Junction sequence number at low-numbered end of pipe
	17	JS2	Junction sequence number at high-numbered end of pipe
	18	ISOLLB	Velocity update indicator at junction 1
	19	ISOLRB	Velocity update indicator at junction 2
	20	ICHF	CHF calculation option
	21	IHYDRO	Hydrodynamics option
	22	BSMASS	Time-integrated flowrate out of pipe
Pressurizer	1	NODES	Number of radial heat transfer nodes
	2	NCELLS	Number of fluid cells
	3	JUN2	Junction number
	4	QHEAT	Total heater power
	5	PSET	Pressurizer set point for heater-sprayer control
	6	DPMAX	Differential pressure at which heaters have maximum power
	7	QINT	Initial water volume
	8	ZHTR	Water height for heater cutoff
	9	QOUT	Volume of water discharged
	10	TYPE2	Adjacent component type
	11	ICJ	Iteration index of adjacent component
	12	IUV1	Indicator for velocity update at junction 1
	13	IUV2	Indicator for velocity update at junction 2

TABLE A.2 (continued)

Component Type	Word	Variable	Description
	14	JS2	Junction number at discharge
	15	Z	Water height above discharge
	16	QIN	Heater power being input to water
	17	FLOW	Volume flow rate at discharge
	18	BXMASS	Current mass flow rate out of pressurizer
	19	BSMASS	Time-integrated mass flow rate out of pressurizer
	20	BSMSSP	Total fluid mass
Pump	1	NODES	Number of radial heat transfer nodes
	2	NCELLS	Number of fluid cells
	3	JUN1	Number of the junction at the low-numbered end of the pump
	4	JUN2	Number of the junction at the high-numbered end of the pump
	5	IPMPTY	Pump type option
	6	IRP	Reverse speed option
	7	IPM	Two-phase option
	8	RHEAD	Rated head
	9	RTORK	Rated torque
	10	RFLOW	Rated flow
	11	RRHO	Rated density
	12	EFFMI	Moment of inertia
	13	TFR1	Frictional torque constant 1
	14	TRF2	Frictional torque constant 2
	15	ROMEGA	Rated angular velocity
	16	INDXHM	Heat degradation multiplier curve index
	17	INDXTM	Torque degradation multiplier curve index
	18	NHDM	Number of points in the head degradation multiplier curve
	19	NTDM	Number of points in the torque degradation multiplier curve

TABLE A.2 (continued)

Component Type	Word	Variable	Description
	20	ICJ1	Iteration index of adjacent component at junction 1
	21	ICJ2	Iteration index of adjacent component at junction 2
	22	TYPE1	Adjacent component type at junction 1
	23	TYPE2	Adjacent component type at junction 2
	24	ISOL1	Velocity update indicator at junction 1
	25	ISOL2	Velocity update indicator at junction 2
	26	OMEGA	Angular velocity at old time
	27	OMEGAN	Angular velocity at new time
	28	RHO	Mixture density
	29	FLOW	Volumetric flow rate
	30	ALPHA	Void fraction
	31	HEAD	Pump head
	32	TORQUE	Pump torque
	33	SMOM	Momentum source
	34	DELP	ΔP across the pump
	35	MAT	Material type
	36	RADIN	Inner radius of wall
	37	TH	Thickness of wall
	38	HOUTL	Heat transfer coefficient between wall and surrounding liquid
	39	HOUTV	Heat transfer coefficient between wall and surrounding vapor
	40	TOUTL	Liquid temperature surrounding wall
	41	TOUTV	Vapor temperature surrounding wall
	42	JS1	Junction sequence number at low-numbered end of pump
	43	JS2	Junction sequence number at high-numbered end of pump
	44	ICHF	CHF calculation option

TABLE A.2 (continued)

Component Type	Word	Variable	Description
	45	IHYDRO	Hydrodynamics option
	46	NDMAX	Size of scratch storage array
	47	MFLOW	Mass flow rate
	48	IPMPTR	Trip number
	49	NPMPTX	Number of entries in the pump speed table
	50	ISAVE	Index of pump speed table
	51	ICOND	Trip condition
	52	OPTION	Pump curve option
	53	BSMASS	Time-integrated mass flow rate out of pump
Steam	1	NODES	Number of radial heat transfer nodes
Generator	2	NCELL1	Number of fluid cells in primary side
	3	NCELL2	Number of fluid cells in secondary side
	4	JUN11	Junction number adjacent to cell 1 on the primary side
	5	JUN12	Junction number adjacent to cell NCELL1 on the primary side
	6	JUN21	Junction number adjacent to cell 1 on the secondary side
	7	JUN22	Junction number adjacent to cell NCELL2 on the secondary side
	8	MAT	Material type
	9	RADIN	Inner radius of wall
	10	TH	Wall thickness
	11	NFF1	Friction factor correlation option (primary side)
	12	NFF2	Friction factor correlation option (secondary side)
	13	ICJ11	Iteration index of component adjacent to JUN11
	14	ICJ12	Iteration index of component adjacent to JUN12

TABLE A.2 (continued)

Component Type	Word	Variable	Description
	15	ICJ21	Iteration index of component adjacent to JUN21
	16	ICJ22	Iteration index of component adjacent to JUN22
	17	TYPE11	Adjacent component type at JUN11
	18	TYPE12	Adjacent component type at JUN12
	19	TYPE21	Adjacent component type at JUN21
	20	TYPE22	Adjacent component type at JUN22
	21	ISVLB1	Velocity update indicator at JUN11
	22	ISVRB1	Velocity update indicator at JUN12
	23	ISVLB2	Velocity update indicator at JUN21
	24	ISVRB2	Velocity update indicator at JUN22
	25	KIND	Steam generator type
	26	JS11	Junction sequence number at primary side inlet
	27	JS12	Junction sequence number at primary side discharge
	28	JS21	Junction sequence number at secondary side inlet
	29	JS22	Junction sequence number at secondary side discharge
	30	IHYDRO	Hydrodynamics option
	31	ICHF1	CHF calculation option for primary side
	32	ICHF2	CHF calculation option for secondary side
	33	BSMSS1	Total fluid mass primary side
	34	BSMSS2	Total fluid mass secondary side
Tee	1	NODES	Number of radial heat transfer nodes
	2	NCELLS	Total number of fluid cells
	3	NCELL1	Number of fluid cells in primary tube
	4	NCELL2	Number of fluid cells in secondary tube

TABLE A.2 (continued)

Component Type	Word	Variable	Description
	5	JCELL	Cell number connecting primary and side tubes
	6	JUN1	Junction number at low-numbered end of primary tube
	7	JUN2	Junction number at high-numbered end of primary tube
	8	JUN3	Junction number at high-numbered end of secondary tube
	9	MATID	Material type
	10	COST	Cosine of the angle between the primary and side tubes
	11	RADIN1	Inner radius of the primary tube
	12	RADIN2	Inner radius of the secondary tube
	13	TH1	Primary tube wall thickness
	14	TH2	Secondary tube wall thickness
	15	HOUTL1	Heat transfer coefficient between primary tube wall and surrounding liquid
	16	HOUTV1	Heat transfer coefficient between primary tube wall and surrounding vapor
	17	HOUTL2	Heat transfer coefficient between secondary tube wall and surrounding liquid
	18	HOUTV2	Heat transfer coefficient between secondary tube wall and surrounding vapor
	19	TOUTL1	Liquid temperature outside of the primary tube
	20	TOUTV1	Vapor temperature outside of the primary tube
	21	TOUTL2	Liquid temperature outside of the secondary tube
	22	TOUTV2	Vapor temperature outside of the secondary tube

TABLE A.2 (continued)

Component Type	Word	Variable	Description
	23	ICJ1	Iteration index of component adjacent to JUN1
	24	ICJ2	Iteration index of component adjacent to JUN2
	25	ICJ3	Iteration index of component adjacent to JUN3
	26	TYPE1	Adjacent component type at JUN1
	27	TYPE2	Adjacent component type at JUN2
	28	TYPE3	Adjacent component type at JUN3
	29	JS1	Junction sequence number at JUN1
	30	JS2	Junction sequence number at JUN2
	31	JS3	Junction sequence number at JUN3
	32	ISOL1	Velocity update indicator at JUN1
	33	ISOL2	Velocity update indicator at JUN2
	34	ISOL3	Velocity update indicator at JUN3
	35	ICHF	CHF calculation option
	36	IHYD1	Primary tube hydrodynamics option
	37	IHYD2	Secondary tube hydrodynamics option
	38	ITRP	Trip number to start small break
	39	ALSEP	Phase separation void fraction
	40	ISEP	Phase separation option
	41	BSMASS	Time-integrated mass flow out of tee
Valve	1	NODES	Number of radial heat transfer nodes
	2	NCELLS	Number of fluid cells
	3	JUN1	Junction number at low-numbered end of valve
	4	JUN2	Junction number at high-numbered end of valve
	5	MAT	Material type
	6	RADIN	Inner radius of wall
	7	TH	Thickness of wall

TABLE A.2 (continued)

Component Type	Word	Variable	Description
	8	HOUTL	Heat transfer coefficient between valve wall and surrounding liquid
	9	HOUTV	Heat transfer coefficient between valve wall and surrounding vapor
	10	TOUTL	Liquid temperature outside the wall
	11	TOUTV	Vapor temperature outside of wall
	12	ICJ1	Iteration index of component adjacent to JUN1
	13	ICJ2	Iteration index of component adjacent to JUN2
	14	TYPE1	Adjacent component type at JUN1
	15	TYPE2	Adjacent component type at JUN2
	16	JS1	Junction sequence number at low-numbered valve end
	17	JS2	Junction sequence number at high-numbered valve end
	18	ISOLLB	Velocity update indicator at JUN1
	19	ISOLRB	Velocity update indicator at JUN2
	20	ICHF	CHF calculation option
	21	IHYDRO	Hydrodynamics option
	22	IVTY	Valve type
	23	IVTR	Valve trip number
	24	NVTX	Number of points in the valve table
	25	IVPG	Pressure gradient option
	26	PVC	Pressure gradient setpoint
	27	AVLVE	Open flow area
	28	HVLVE	Open hydraulic diameter
	29	IVPS	Valve position
	30	BSMASS	Time-integrated mass flow out of valve

TABLE A.3. Pointer Tables for One-Dimensional Components

Component Type	Position	Variable	Description
Accumulator	1	LALP	First word of old vapor fraction array
	2	LALPD	Not used
	3	LALV	First word of old interfacial surface area array
	4	LARA	Not used
	5	LAREL	Not used
	6	LAREV	Not used
	7	LARL	Not used
	8	LARV	Product of void fraction and microscopic vapor density
	9	LBIT	Bit flags (previous time step)
	10	LEA	Not used
	11	LEL	Old liquid internal energy
	12	LEV	Old vapor internal energy
	13	LHILO	Old heat transfer coefficient between inside wall and liquid
	14	LHIVO	Old heat transfer coefficient between inside wall and vapor
	15	LHLV	Old interfacial heat transfer coefficient
	16	LP	Old pressure
	17	LPA	Not used
	18	LROA1	Not used
	19	LROL	Old liquid density
	20	LROV	Old vapor density
	21	LTD	Not used
	22	LTL	Old liquid temperature
	23	LTV	Old vapor temperature
	24	LVRD	Not used
	25	LVM	Initial mixture velocities

TABLE 3 (continued)

Component Type	Position	Variable	Description
	26	LTW	Old wall temperature
	27	LVL	Old liquid velocities
	28	LVV	Old vapor velocities
	29	LALPN	New vapor fraction
	30	LALPDN	Not used
	31	LALVN	New interfacial surface area
	32	LARAN	Not used
	33	LARELN	Not used
	34	LAREVN	Not used
	35	LARLN	Not used
	36	LARVN	Not used
	37	LBITN	Bit flags for current time step
	38	LEAN	Not used
	39	LELN	New liquid internal energy
	40	LEVN	New vapor internal energy
	41	LHIL	New heat transfer coefficient between inside wall and liquid
	42	LHIV	New heat transfer coefficient between inside wall and vapor
	43	LHLVN	New interfacial heat transfer coefficient
	44	LPN	New pressure
	45	LPAN	Not used
	46	LROAN	Not used
	47	LROLN	New liquid density
	48	LROVN	New vapor density
	49	LTDN	Not used
	50	LTLN	New liquid temperature
	51	LTVN	New vapor temperature
	52	LVRDN	Not used
	53	LVMN	New mixture velocity

TABLE 3 (continued)

Component Type	Position	Variable	Description
	54	LTWN	New wall temperature
	55	LVLN	Not used
	56	LVVN	Not used
	57	LB	First word of a temporary array for storing implicit matrix solution
	58	LCFZ	First word of the total friction factor array
	59	LCL	First word of liquid conductivity array
	60	LCPL	First word of liquid specific heat array
	61	LCPV	First word of vapor specific heat array
	62	LCV	First word of vapor conductivity array
	63	LDRIV	First word of temporary array for thermodynamic derivatives
	64	LDX	First word of cell length array
	65	LFA	First word of cell flow area array
	66	LFRIC	First word of additive friction factor array
	67	LGRAV	First word of gravitational term array
	68	LHD	First word of hydraulic diameter array
	69	LHFG	First word of latent heat of vaporization array
	70	LQPPP	First word of wall heat source array
	71	LRMEM	First word of mixture internal energy array
	72	LRMVM	First word of array containing the product of ROM(I)*VM(I)
	73	LROM	First word of mixture density array
	74	LRHS	First word of array containing right-hand side for vapor continuity equations
	75	LSIG	First word of surface tension array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	76	LTRID	Not used
	77	LTSAT	First word of saturation temperature array
	78	LTSSN	Not used
	79	LVISL	First word of liquid viscosity array
	80	LVISV	First word of vapor viscosity array
	81	LVOL	First word of cell volume array
	82	LVR	First word of relative velocity array
	83	LWA	First word of wall area array
	84	LDFVDP	Not used
	85	LDFLDP	Not used
	86	LVLT	Not used
	87	LVVT	Not used
	88	LWFL	Not used
	89	LWFV	Not used
	90	LCIF	Not used
	91	LIDR	First word of heat transfer regime array
	92	LMATID	First word of material type array
	93	LNFF	First word of friction correlation option array
	94	LBD1	First word of temporary BD array
	95	LQPPL	First word of heat flux (wall to liquid) array
Break	1	LALP	First word of old vapor fraction array
	2	LALPD	Not used
	3	LALV	First word of old interfacial surface area array
	4	LARA	Not used
	5	LAREL	Not used
	6	LAREV	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	7	LARL	Not used
	8	LARV	Product of void fraction and microscopic vapor density
	9	LBIT	Bit flags (previous time step)
	10	LEA	Not used
	11	LEL	Old liquid internal energy
	12	LEV	Old vapor internal energy
	13	LHILO	Old heat transfer coefficient between inside wall and liquid
	14	LHIVO	Old heat transfer coefficient between inside wall and vapor
	15	LHLV	Old interfacial heat transfer coefficient
	16	LP	Old pressure
	17	LPA	Not used
	18	LROA1	Not used
	19	LROL	Old liquid density
	20	LROV	Old vapor density
	21	LTD	Not used
	22	LTL	Old liquid temperature
	23	LTV	Old vapor temperature
	24	LVRD	Not used
	25	LVM	Initial mixture velocities
	26	LTW	Old wall temperature
	27	LVL	Old liquid velocities
	28	LVV	Old vapor velocities
	29	LALPN	New vapor fraction
	30	LALPDN	Not used
	31	LALVN	New interfacial surface area
	32	LARAN	Not used
	33	LARELN	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	34	LAREVN	Not used
	35	LARLN	Not used
	36	LARVN	Not used
	37	LBITN	Bit flags for current time step
	38	LEAN	Not used
	39	LELN	New liquid internal energy
	40	LEVN	New vapor internal energy
	41	LHIL	New heat transfer coefficient between inside wall and liquid
	42	LHIV	New heat transfer coefficient between inside wall and vapor
	43	LHLVN	New interfacial heat transfer coefficient
	44	LPN	New pressure
	45	LPAN	Not used
	46	LROAN	Not used
	47	LROLN	New liquid density
	48	LROVN	New vapor density
	49	LTDN	Not used
	50	LTLN	New liquid temperature
	51	LTVN	New vapor temperature
	52	LVRDN	Not used
	53	LVMN	New mixture velocity
	54	LTWN	New wall temperature
	55	LVLN	Not used
	56	LVVN	Not used
	57	LB	First word of a temporary array for storing implicit matrix solution
	58	LCFZ	First word of the total friction factor array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	59	LCL	First word of liquid conductivity array
	60	LCPL	First word of liquid specific heat array
	61	LCPV	First word of vapor specific heat array
	62	LCV	First word of vapor conductivity array
	63	LDRIV	First word of temporary array for thermodynamic derivatives
	64	LDX	First word of cell length array
	65	LFA	First word of cell flow area array
	66	LFRIC	First word of additive friction factor array
	67	LGRAV	First word of gravitational term array
	68	LHD	First word of hydraulic diameter array
	69	LHFG	First word of latent heat of vaporization array
	70	LQPPP	First word of wall heat source array
	71	LRMEM	First word of mixture internal energy array
	72	LRMVM	First word of array containing the product of ROM(I)*VM(I)
	73	LROM	First word of mixture density array
	74	LRHS	First word of array containing right-hand side for vapor continuity equations
	75	LSIG	First word of surface tension array
	76	LTRID	Not used
	77	LTSAT	First word of saturation temperature array
	78	LTSSN	Not used
	79	LVISL	First word of liquid viscosity array
	80	LVISV	First word of vapor viscosity array
	81	LVOL	First word of cell volume array
	82	LVR	First word of relative velocity array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	83	LWA	First word of wall area array
	84	LDFVDP	Not used
	85	LDFLDP	Not used
	86	LVLT	Not used
	87	LVVT	Not used
	88	LWFL	Not used
	89	LWFV	Not used
	90	LCIF	Not used
	91	LBTAB	First word of break table array
Fill	1	LALP	First word of old vapor fraction array
	2	LALPD	Not used
	3	LALV	First word of old interfacial surface area array
	4	LARA	Not used
	5	LAREL	Not used
	6	LAREV	Not used
	7	LARL	Not used
	8	LARV	Product of void fraction and microscopic vapor density
	9	LBIT	Bit flags (previous time step)
	10	LEA	Not used
	11	LEL	Old liquid internal energy
	12	LEV	Old vapor internal energy
	13	LHILO	Old heat transfer coefficient between inside wall and liquid
	14	LHIVO	Old heat transfer coefficient between inside wall and vapor
	15	LHLV	Old interfacial heat transfer coefficient
	16	LP	Old pressure
	17	LPA	Not used
	18	LROA1	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	19	LROL	Old liquid density
	20	LROV	Old vapor density
	21	LTD	Not used
	22	LTL	Old liquid temperature
	23	LTV	Old vapor temperature
	24	LVRD	Not used
	25	LVM	Initial mixture velocities
	26	LTW	Old wall temperature
	27	LVL	Old liquid velocities
	28	LVV	Old vapor velocities
	29	LALPN	New vapor fraction
	30	LALPDN	Not used
	31	LALVN	New interfacial surface area
	32	LARAN	Not used
	33	LARELN	Not used
	34	LAREVN	Not used
	35	LARLN	Not used
	36	LARVN	Not used
	37	LBITN	Bit flags for current time step
	38	LEAN	Not used
	39	LELN	New liquid internal energy
	40	LEVN	New vapor internal energy
	41	LHIL	New heat transfer coefficient between inside wall and liquid
	42	LHIV	New heat transfer coefficient between inside wall and vapor
	43	LHLVN	New interfacial heat transfer coefficient
	44	LPN	New pressure
	45	LPAN	Not used
	46	LROAN	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	47	LROLN	New liquid density
	48	LROVN	New vapor density
	49	LTDN	Not used
	50	LTLN	New liquid temperature
	51	LTVN	New vapor temperature
	52	LVRDN	Not used
	53	LVMN	New mixture velocity
	54	LTWN	New wall temperature
	55	LVLN	Not used
	56	LVVN	Not used
	57	LB	First word of a temporary array for storing implicit matrix solution
	58	LCFZ	First word of the total friction factor array
	59	LCL	First word of liquid conductivity array
	60	LCPL	First word of liquid specific heat array
	61	LCPV	First word of vapor specific heat array
	62	LCV	First word of vapor conductivity array
	63	LDRIV	First word of temporary array for thermodynamic derivatives
	64	LDX	First word of cell length array
	65	LFA	First word of cell flow area array
	66	LFRIC	First word of additive friction factor array
	67	LGRAV	First word of gravitational term array
	68	LHD	First word of hydraulic diameter array
	69	LHFG	First word of latent heat of vaporization array
	70	LQPPP	First word of wall heat source array
	71	LRMEM	First word of mixture internal energy array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	72	LRMVM	First word of array containing the product of ROM(I)*VM(I)
	73	LROM	First word of mixture density array
	74	LRHS	First word of array containing right-hand side for vapor continuity equations
	75	LSIG	First word of surface tension array
	76	LTRID	Not used
	77	LTSAT	First word of saturation temperature array
	78	LTSSN	Not used
	79	LVISL	First word of liquid viscosity array
	80	LVISV	First word of vapor viscosity array
	81	LVOL	First word of cell volume array
	82	LVR	First word of relative velocity array
	83	LWA	First word of wall area array
	84	LDFVDP	Not used
	85	LDFLDP	Not used
	86	LVLT	Not used
	87	LVVT	Not used
	88	LWFL	Not used
	89	LWFV	Not used
	90	LCIF	Not used
	91	LFTAB	First word of velocity table array
Pipe	1	LALP	First word of old vapor fraction array
	2	LALPD	Not used
	3	LALV	First word of old interfacial surface area array
	4	LARA	Not used
	5	LAREL	Not used
	6	LAREV	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	7	LARL	Not used
	8	LARV	Product of void fraction and microscopic vapor density
	9	LBIT	Bit flags (previous time step)
	10	LEA	Not used
	11	LEL	Old liquid internal energy
	12	LEV	Old vapor internal energy
	13	LHILO	Old heat transfer coefficient between inside wall and liquid
	14	LHIVO	Old heat transfer coefficient between inside wall and vapor
	15	LHLV	Old interfacial heat transfer coefficient
	16	LP	Old pressure
	17	LPA	Not used
	18	LROA1	Not used
	19	LROL	Old liquid density
	20	LROV	Old vapor density
	21	LTD	Not used
	22	LTL	Old liquid temperature
	23	LTV	Old vapor temperature
	24	LVRD	Not used
	25	LVM	Initial mixture velocities
	26	LTW	Old wall temperature
	27	LVL	Old liquid velocities
	28	LVV	Old vapor velocities
	29	LALPN	New vapor fraction
	30	LALPDN	Not used
	31	LALVN	New interfacial surface area
	32	LARAN	Not used
	33	LARELN	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	34	LAREVN	Not used
	35	LARLN	Not used
	36	LARVN	Not used
	37	LBITN	Bit flags for current time step
	38	LEAN	Not used
	39	LELN	New liquid internal energy
	40	LEVN	New vapor internal energy
	41	LHIL	New heat transfer coefficient between inside wall and liquid
	42	LHIV	New heat transfer coefficient between inside wall and vapor
	43	LHLVN	New interfacial heat transfer coefficient
	44	LPN	New pressure
	45	LPAN	Not used
	46	LROAN	Not used
	47	LROLN	New liquid density
	48	LROVN	New vapor density
	49	LTDN	Not used
	50	LTLN	New liquid temperature
	51	LTVN	New vapor temperature
	52	LVRDN	Not used
	53	LVMN	New mixture velocity
	54	LTWN	New wall temperature
	55	LVLN	Not used
	56	LVVN	Not used
	57	LB	First word of a temporary array for storing implicit matrix solution
	58	LCFZ	First word of the total friction factor array
	59	LCL	First word of liquid conductivity array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	60	LCPL	First word of liquid specific heat array
	61	LCPV	First word of vapor specific heat array
	62	LCV	First word of vapor conductivity array
	63	LDRIV	First word of temporary array for thermodynamic derivatives
	64	LDX	First word of cell length array
	65	LFA	First word of cell flow area array
	66	LFRIC	First word of additive friction factor array
	67	LGRAV	First word of gravitational term array
	68	LHD	First word of hydraulic diameter array
	69	LHFG	First word of latent heat of vaporization array
	70	LQPPP	First word of wall heat source array
	71	LRMEM	First word of mixture internal energy array
	72	LRMVM	First word of array containing the product of ROM(I)*VM(I)
	73	LROM	First word of mixture density array
	74	LRHS	First word of array containing right-hand side for vapor continuity equations
	75	LSIG	First word of surface tension array
	76	LTRID	Not used
	77	LTSAT	First word of saturation temperature array
	78	LTSSN	Not used
	79	LVISL	First word of liquid viscosity array
	80	LVISV	First word of vapor viscosity array
	81	LVOL	First word of cell volume array
	82	LVR	First word of relative velocity array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	83	LWA	First word of wall area array
	84	LDFVDP	Not used
	85	LDFLDP	Not used
	86	LVLT	Not used
	87	LVVT	Not used
	88	LWFL	Not used
	89	LWFV	Not used
	90	LCIF	Not used
	91	LIDR	First word of heat transfer regime array
	92	LMATID	First word of material type array
	93	LNFF	First word of friction correlation option array
	94	LCPW	First word of wall specific heat array
	95	LCW	First word of wall conductivity array
	96	LDR	First word of radial mesh size array
	97	LEMIS	First word of wall emissivity array
	98	LHOL	First word of heat transfer coefficient array (outside wall to liquid)
	99	LHOV	First word of heat transfer coefficient array (outside wall to vapor)
	100	LQPPC	First word of critical heat flux array
	101	LRN	First word of node radius array
	102	LRN2	First word of node center radius array
	103	LROW	First word of wall density array
	104	LTOL	First word of liquid temperature outside of the wall array
	105	LTOV	First word of vapor temperature outside the wall array
Pressurizer	1	LALP	First word of old vapor fraction array
	2	LALPD	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	3	LALV	First word of old interfacial surface area array
	4	LARA	Not used
	5	LAREL	Not used
	6	LAREV	Not used
	7	LARL	Not used
	8	LARV	Product of void fraction and microscopic vapor density
	9	LBIT	Bit flags (previous time step)
	10	LEA	Not used
	11	LEL	Old liquid internal energy
	12	LEV	Old vapor internal energy
	13	LHILO	Old heat transfer coefficient between inside wall and liquid
	14	LHIVO	Old heat transfer coefficient between inside wall and vapor
	15	LHLV	Old interfacial heat transfer coefficient
	16	LP	Old pressure
	17	LPA	Not used
	18	LROA1	Not used
	19	LROL	Old liquid density
	20	LROV	Old vapor density
	21	LTD	Not used
	22	LTL	Old liquid temperature
	23	LTV	Old vapor temperature
	24	LVRD	Not used
	25	LVM	Initial mixture velocities
	26	LTW	Old wall temperature
	27	LVL	Old liquid velocities
	28	LVV	Old vapor velocities

TABLE 3 (continued)

Component Type	Position	Variable	Description
	29	LALPN	New vapor fraction
	30	LALPDN	Not used
	31	LALVN	New interfacial surface area
	32	LARAN	Not used
	33	LARELN	Not used
	34	LAREVN	Not used
	35	LARLN	Not used
	36	LARVN	Not used
	37	LBITN	Bit flags for current time step
	38	LEAN	Not used
	39	LELN	New liquid internal energy
	40	LEVN	New vapor internal energy
	41	LHIL	New heat transfer coefficient between inside wall and liquid
	42	LHIV	New heat transfer coefficient between inside wall and vapor
	43	LHLVN	New interfacial heat transfer coefficient
	44	LPN	New pressure
	45	LPAN	Not used
	46	LROAN	Not used
	47	LROLN	New liquid density
	48	LROVN	New vapor density
	49	LTDN	Not used
	50	LTLN	New liquid temperature
	51	LTVN	New vapor temperature
	52	LVRDN	Not used
	53	LVMN	New mixture velocity
	54	LTWN	New wall temperature
	55	LVLN	Not used
	56	LVVN	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	57	LB	First word of a temporary array for storing implicit matrix solution
	58	LCFZ	First word of the total friction factor array
	59	LCL	First word of liquid conductivity array
	60	LCPL	First word of liquid specific heat array
	61	LCPV	First word of vapor specific heat array
	62	LCV	First word of vapor conductivity array
	63	LDRIV	First word of temporary array for thermodynamic derivatives
	64	LDX	First word of cell length array
	65	LFA	First word of cell flow area array
	66	LFRIC	First word of additive friction factor array
	67	LGRAV	First word of gravitational term array
	68	LHD	First word of hydraulic diameter array
	69	LHFG	First word of latent heat of vaporization array
	70	LQPPP	First word of wall heat source array
	71	LRMEM	First word of mixture internal energy array
	72	LRMVM	First word of array containing the product of ROM(I)*VM(I)
	73	LROM	First word of mixture density array
	74	LRHS	First word of array containing right-hand side for vapor continuity equations
	75	LSIG	First word of surface tension array
	76	LTRID	Not used
	77	LTSAT	First word of saturation temperature array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	78	LTSSN	Not used
	79	LVISL	First word of liquid viscosity array
	80	LVISV	First word of vapor viscosity array
	81	LVOL	First word of cell volume array
	82	LVR	First word of relative velocity array
	83	LWA	First word of wall area array
	84	LDFVDP	Not used
	85	LDFLDP	Not used
	86	LVLT	Not used
	87	LVVT	Not used
	88	LWFL	Not used
	89	LWFV	Not used
	90	LCIF	Not used
	91	LIDR	First word of heat transfer regime array
	92	LMATID	First word of material type array
	93	LNFF	First word of friction correlation option array
	94	LBD1	First word of temporary BD array
	95	LQPPL	First word of heat flux (wall to liquid) array
Pump	1	LALP	First word of old vapor fraction array
	2	LALPD	Not used
	3	LALV	First word of old interfacial surface area array
	4	LARA	Not used
	5	LAREL	Not used
	6	LAREV	Not used
	7	LARL	Not used
	8	LARV	Product of void fraction and microscopic vapor density

TABLE 3 (continued)

Component Type	Position	Variable	Description
	9	LBIT	Bit flags (previous time step)
	10	LEA	Not used
	11	LEL	Old liquid internal energy
	12	LEV	Old vapor internal energy
	13	LHILO	Old heat transfer coefficient between inside wall and liquid
	14	LHIVO	Old heat transfer coefficient between inside wall and vapor
	15	LHLV	Old interfacial heat transfer coefficient
	16	LP	Old pressure
	17	LPA	Not used
	18	LROA1	Not used
	19	LROL	Old liquid density
	20	LROV	Old vapor density
	21	LTD	Not used
	22	LTL	Old liquid temperature
	23	LTV	Old vapor temperature
	24	LVRD	Not used
	25	LVM	Initial mixture velocities
	26	LTW	Old wall temperature
	27	LVL	Old liquid velocities
	28	LVV	Old vapor velocities
	29	LALPN	New vapor fraction
	30	LALPDN	Not used
	31	LALVN	New interfacial surface area
	32	LARAN	Not used
	33	LARELN	Not used
	34	LAREVN	Not used
	35	LARLN	Not used
	36	LARVN	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	37	LBITN	Bit flags for current time step
	38	LEAN	Not used
	39	LELN	New liquid internal energy
	40	LEVN	New vapor internal energy
	41	LHIL	New heat transfer coefficient between inside wall and liquid
	42	LHIV	New heat transfer coefficient between inside wall and vapor
	43	LHLVN	New interfacial heat transfer coefficient
	44	LPN	New pressure
	45	LPAN	Not used
	46	LROAN	Not used
	47	LROLN	New liquid density
	48	LROVN	New vapor density
	49	LTDN	Not used
	50	LTLN	New liquid temperature
	51	LTVN	New vapor temperature
	52	LVRDN	Not used
	53	LVMN	New mixture velocity
	54	LTWN	New wall temperature
	55	LVLN	Not used
	56	LVVN	Not used
	57	LB	First word of a temporary array for storing implicit matrix solution
	58	LCFZ	First word of the total friction factor array
	59	LCL	First word of liquid conductivity array
	60	LCPL	First word of liquid specific heat array
	61	LCPV	First word of vapor specific heat array
	62	LCV	First word of vapor conductivity array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	63	LDRIV	First word of temporary array for thermodynamic derivatives
	64	LDX	First word of cell length array
	65	LFA	First word of cell flow area array
	66	LFRIC	First word of additive friction factor array
	67	LGRAV	First word of gravitational term array
	68	LHD	First word of hydraulic diameter array
	69	LHFG	First word of latent heat of vaporization array
	70	LQPPP	First word of wall heat source array
	71	LRMEM	First word of mixture internal energy array
	72	LRMVM	First word of array containing the product of ROM(I)*VM(I)
	73	LROM	First word of mixture density array
	74	LRHS	First word of array containing right-hand side for vapor continuity equations
	75	LSIG	First word of surface tension array
	76	LTRID	Not used
	77	LTSAT	First word of saturation temperature array
	78	LTSSN	Not used
	79	LVISL	First word of liquid viscosity array
	80	LVISV	First word of vapor viscosity array
	81	LVOL	First word of cell volume array
	82	LVR	First word of relative velocity array
	83	LWA	First word of wall area array
	84	LDFVDP	Not used
	85	LDFLDP	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	86	LVLT	Not used
	87	LVVT	Not used
	88	LWFL	Not used
	89	LWFV	Not used
	90	LCIF	Not used
	91	LIDR	First word of heat transfer regime array
	92	LMATID	First word of material type array
	93	LNFF	First word of friction correlation option array
	94	LCPW	First word of wall specific heat array
	95	LCW	First word of wall conductivity array
	96	LDR	First word of radial mesh size array
	97	LEMIS	First word of wall emissivity array
	98	LHOL	First word of heat transfer coefficient array (outside wall to liquid)
	99	LHOV	First word of heat transfer coefficient array (outside wall to vapor)
	100	LQPPC	First word of critical heat flux array
	101	LRN	First word of node radius array
	102	LRN2	First word of node center radius array
	103	LROW	First word of wall density array
	104	LTOL	First word of liquid temperature outside the wall array
	105	LTOV	First word of vapor temperature outside the wall array
	106	LSPTBL	First word of pump speed table array
	107	LNDATA	First word of array containing the number of points in the pump torque curves
	108	LHSP1	First word of single-phase head curve 1 array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	109	LHSP2	First word of single-phase head curve 2 array
	110	LHSP3	First word of single-phase head curve 3 array
	111	LHSP4	First word of single-phase head curve 4 array
	112	LHTP1	First word of two-phase head curve 1 array
	113	LHTP2	First word of two-phase head curve 2 array
	114	LHTP3	First word of two-phase head curve 3 array
	115	LHTP4	First word of two-phase head curve 4 array
	116	LTSP1	First word of single-phase torque curve 1 array
	117	LTSP2	First word of single-phase torque curve 2 array
	118	LTSP3	First word of single-phase torque curve 3 array
	119	LTSP4	First word of single-phase torque curve 4 array
	120	LTTP1	First word of two-phase torque curve 1 array
	121	LTTP2	First word of two-phase torque curve 2 array
	122	LTTP3	First word of two-phase torque curve 3 array
	123	LTTP4	First word of two-phase torque curve 4 array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	124	LHDM	First word of head degradation multiplier array
	125	LTDM	First word of torque degradation multiplier curve array
	126	LIDXCS	First word of curve set index array
	127	LBD4	Not used
Steam	1	LALP	First word of old vapor fraction array
Generator	2	LALPD	Not used
	3	LALV	First word of old interfacial surface area array
	4	LARA	Not used
	5	LAREL	Not used
	6	LAREV	Not used
	7	LARL	Not used
	8	LARV	Product of void fraction and microscopic vapor density
	9	LBIT	Bit flags (previous time step)
	10	LEA	Not used
	11	LEL	Old liquid internal energy
	12	LEV	Old vapor internal energy
	13	LHILO	Old heat transfer coefficient between inside wall and liquid
	14	LHIVO	Old heat transfer coefficient between inside wall and vapor
	15	LHLV	Old interfacial heat transfer coefficient
	16	LP	Old pressure
	17	LPA	Not used
	18	LROA1	Not used
	19	LROL	Old liquid density
	20	LROV	Old vapor density

TABLE 3 (continued)

Component Type	Position	Variable	Description
	21	LTD	Not used
	22	LTL	Old liquid temperature
	23	LTV	Old vapor temperature
	24	LVRD	Not used
	25	LVM	Initial mixture velocities
	26	LTW	Old wall temperature
	27	LVL	Old liquid velocities
	28	LVV	Old vapor velocities
	29	LALPN	New vapor fraction
	30	LALPDN	Not used
	31	LALVN	New interfacial surface area
	32	LARAN	Not used
	33	LARELN	Not used
	34	LAREVN	Not used
	35	LARLN	Not used
	36	LARVN	Not used
	37	LBITN	Bit flags for current time step
	38	LEAN	Not used
	39	LELN	New liquid internal energy
	40	LEVN	New vapor internal energy
	41	LHIL	New heat transfer coefficient between inside wall and liquid
	42	LHIV	New heat transfer coefficient between inside wall and vapor
	43	LHLVN	New interfacial heat transfer coefficient
	44	LPN	New pressure
	45	LPAN	Not used
	46	LROAN	Not used
	47	LROLN	New liquid density
	48	LROVN	New vapor density

TABLE 3 (continued)

Component Type	Position	Variable	Description
	49	LTDN	Not used
	50	LTLN	New liquid temperature
	51	LTVN	New vapor temperature
	52	LVRDN	Not used
	53	LVMN	New mixture velocity
	54	LTWN	New wall temperature
	55	LVLN	Not used
	56	LVVN	Not used
	57	LB	First word of a temporary array for storing implicit matrix solution
	58	LCFZ	First word of the total friction factor array
	59	LCL	First word of liquid conductivity array
	60	LCPL	First word of liquid specific heat array
	61	LCPV	First word of vapor specific heat array
	62	LCV	First word of vapor conductivity array
	63	LDRIV	First word of temporary array for thermodynamic derivatives
	64	LDX	First word of cell length array
	65	LFA	First word of cell flow area array
	66	LFRIC	First word of additive friction factor array
	67	LGRAV	First word of gravitational term array
	68	LHD	First word of hydraulic diameter array
	69	LHFG	First word of latent heat of vaporization array
	70	LQPPP	First word of wall heat source array
	71	LRMEM	First word of mixture internal energy array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	72	LRMVM	First word of array containing the product of ROM(I)*VM(I)
	73	LROM	First word of mixture density array
	74	LRHS	First word of array containing right-hand side for vapor continuity equations
	75	LSIG	First word of surface tension array
	76	LTRID	Not used
	77	LTSAT	First word of saturation temperature array
	78	LTSSN	Not used
	79	LVISL	First word of liquid viscosity array
	80	LVISV	First word of vapor viscosity array
	81	LVOL	First word of cell volume array
	82	LVR	First word of relative velocity array
	83	LWA	First word of wall area array
	84	LDFVDP	Not used
	85	LDFLDP	Not used
	86	LVLT	Not used
	87	LVVT	Not used
	88	LWFL	Not used
	89	LWFV	Not used
	90	LCIF	Not used
	91	LIDR	First word of heat transfer regime array
	92	LMATID	First word of material type array
	93	LNFF	First word of friction correlation option array
	94	LCPW	First word of wall specific heat array
	95	LCW	First word of wall conductivity array
	96	LDR	First word of radial mesh size array
	97	LEMIS	First word of wall emissivity array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	98	LHOL	First word of heat transfer coefficient array (outside wall to liquid)
	99	LHOV	First word of heat transfer coefficient array (outside wall to vapor)
	100	LQPPC	First word of critical heat flux array
	101	LRN	First word of node radius array
	102	LRN2	First word of node center radius array
	103	LROW	First word of wall density array
	104	LTOL	First word of liquid temperature outside the wall array
	105	LTOV	First word of vapor temperature outside the wall array
	106	LHLEFF	First word of effective wall to liquid heat transfer coefficient array (stored within HIL)
	107	LHVEFF	First word of effective wall to vapor heat transfer coefficient array (stored within HIV)
	108	LTWEFF	First word of effective wall temperature array (stored within TW)
	109	LHLO	First word of wall to liquid heat transfer coefficient array (secondary side)
	110	LHVO	First word of wall to vapor heat transfer coefficient array (secondary side)
	111	LTLO	First word of liquid temperature array (secondary side)
	112	LTVO	First word of vapor temperature array (secondary side)
	113	LQPPL	First word of heat flux array (wall to liquid)

TABLE 3 (continued)

Component Type	Position	Variable	Description
	114	LQPPV	First word of heat flux array (wall to vapor)
Tee	1	LALP	First word of old vapor fraction array
	2	LALPD	Not used
	3	LALV	First word of old interfacial surface area array
	4	LARA	Not used
	5	LAREL	Not used
	6	LAREV	Not used
	7	LARL	Not used
	8	LARV	Product of void fraction and microscopic vapor density
	9	LBIT	Bit flags (previous time step)
	10	LEA	Not used
	11	LEL	Old liquid internal energy
	12	LEV	Old vapor internal energy
	13	LHILO	Old heat transfer coefficient between inside wall and liquid
	14	LHIVO	Old heat transfer coefficient between inside wall and vapor
	15	LHLV	Old interfacial heat transfer coefficient
	16	LP	Old pressure
	17	LPA	Not used
	18	LROA1	Not used
	19	LROL	Old liquid density
	20	LROV	Old vapor density
	21	LTD	Not used
	22	LTL	Old liquid temperature
	23	LTV	Old vapor temperature
	24	LVRD	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	25	LVM	Initial mixture velocities
	26	LTW	Old wall temperature
	27	LVL	Old liquid velocities
	28	LVV	Old vapor velocities
	29	LALPN	New vapor fraction
	30	LALPDN	Not used
	31	LALVN	New interfacial surface area
	32	LARAN	Not used
	33	LARELN	Not used
	34	LAREVN	Not used
	35	LARLN	Not used
	36	LARVN	Not used
	37	LBITN	Bit flags for current time step
	38	LEAN	Not used
	39	LELN	New liquid internal energy
	40	LEVN	New vapor internal energy
	41	LHIL	New heat transfer coefficient between inside wall and liquid
	42	LHIV	New heat transfer coefficient between inside wall and vapor
	43	LHLVN	New interfacial heat transfer coefficient
	44	LPN	New pressure
	45	LPAN	Not used
	46	LROAN	Not used
	47	LROLN	New liquid density
	48	LROVN	New vapor density
	49	LTDN	Not used
	50	LTLN	New liquid temperature
	51	LTVN	New vapor temperature
	52	LVRDN	Not used

TABLE 3 (continued)

Component Type	Position	Variable	Description
	53	LVMN	New mixture velocity
	54	LTWN	New wall temperature
	55	LVLN	Not used
	56	LVVN	Not used
	57	LB	First word of a temporary array for storing implicit matrix solution
	58	LCFZ	First word of the total friction factor array
	59	LCL	First word of liquid conductivity array
	60	LCPL	First word of liquid specific heat array
	61	LCPV	First word of vapor specific heat array
	62	LCV	First word of vapor conductivity array
	63	LDRIV	First word of temporary array for thermodynamic derivatives
	64	LDX	First word of cell length array
	65	LFA	First word of cell flow area array
	66	LFRIC	First word of additive friction factor array
	67	LGRAV	First word of gravitational term array
	68	LHD	First word of hydraulic diameter array
	69	LHFG	First word of latent heat of vaporization array
	70	LQPPP	First word of wall heat source array
	71	LRMEM	First word of mixture internal energy array
	72	LRMVM	First word of array containing the product of ROM(I)*VM(I)
	73	LROM	First word of mixture density array
	74	LRHS	First word of array containing right-hand side for vapor continuity equations
	75	LSIG	First word of surface tension array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	76	LTRID	Not used
	77	LTSAT	First word of saturation temperature array
	78	LTSSN	Not used
	79	LVISL	First word of liquid viscosity array
	80	LVISV	First word of vapor viscosity array
	81	LVOL	First word of cell volume array
	82	LVR	First word of relative velocity array
	83	LWA	First word of wall area array
	84	LDFVDP	Not used
	85	LDFLDP	Not used
	86	LVLT	Not used
	87	LVVT	Not used
	88	LWFL	Not used
	89	LWFV	Not used
	90	LCIF	Not used
	91	LIDR	First word of heat transfer regime array
	92	LMATID	First word of material type array
	93	LNFF	First word of friction correlation option array
	94	LCPW	First word of wall specific heat array
	95	LCW	First word of wall conductivity array
	96	LDR	First word of radial mesh size array
	97	LEMIS	First word of wall emissivity array
	98	LHOL	First word of heat transfer coefficient array (outside wall to liquid)
	99	LHOV	First word of heat transfer coefficient array (outside wall to vapor)
	100	LQPPC	First word of critical heat flux array
	101	LRN	First word of node radius array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	102	LRN2	First word of node center radius array
	103	LROW	First word of wall density array
	104	LTOL	First word of liquid temperature outside the wall array
	105	LTOV	First word of vapor temperature outside the wall array
	106	LBD4	First word of temporary BD array
Valve	1	LALP	First word of old vapor fraction array
	2	LALPD	Not used
	3	LALV	First word of old interfacial surface area array
	4	LARA	Not used
	5	LAREL	Not used
	6	LAREV	Not used
	7	LARL	Not used
	8	LARV	Product of void fraction and microscopic vapor density
	9	LBIT	Bit flags (previous time step)
	10	LEA	Not used
	11	LEL	Old liquid internal energy
	12	LEV	Old vapor internal energy
	13	LHILO	Old heat transfer coefficient between inside wall and liquid
	14	LHIVO	Old heat transfer coefficient between inside wall and vapor
	15	LHLV	Old interfacial heat transfer coefficient
	16	LP	Old pressure
	17	LPA	Not used
	18	LROA1	Not used
	19	LROL	Old liquid density

TABLE 3 (continued)

Component Type	Position	Variable	Description
	20	LROV	Old vapor density
	21	LTD	Not used
	22	LTL	Old liquid temperature
	23	LTV	Old vapor temperature
	24	LVRD	Not used
	25	LVM	Initial mixture velocities
	26	LTW	Old wall temperature
	27	LVL	Old liquid velocities
	28	LVV	Old vapor velocities
	29	LALPN	New vapor fraction
	30	LALPDN	Not used
	31	LALVN	New interfacial surface area
	32	LARAN	Not used
	33	LARELN	Not used
	34	LAREVN	Not used
	35	LARLN	Not used
	36	LARVN	Not used
	37	LBITN	Bit flags for current time step
	38	LEAN	Not used
	39	LELN	New liquid internal energy
	40	LEVN	New vapor internal energy
	41	LHIL	New heat transfer coefficient between inside wall and liquid
	42	LHIV	New heat transfer coefficient between inside wall and vapor
	43	LHLVN	New interfacial heat transfer coefficient
	44	LPN	New pressure
	45	LPAN	Not used
	46	LROAN	Not used
	47	LROLN	New liquid density

TABLE 3 (continued)

Component Type	Position	Variable	Description
	48	LROVN	New vapor density
	49	LTDN	Not used
	50	LTLN	New liquid temperature
	51	LTVN	New vapor temperature
	52	LVRDN	Not used
	53	LVMN	New mixture velocity
	54	LTWN	New wall temperature
	55	LVLN	Not used
	56	LVVN	Not used
	57	LB	First word of a temporary array for storing implicit matrix solution
	58	LCFZ	First word of the total friction factor array
	59	LCL	First word of liquid conductivity array
	60	LCPL	First word of liquid specific heat array
	61	LCPV	First word of vapor specific heat array
	62	LCV	First word of vapor conductivity array
	63	LDRIV	First word of temporary array for thermodynamic derivatives
	64	LDX	First word of cell length array
	65	LFA	First word of cell flow area array
	66	LFRIC	First word of additive friction factor array
	67	LGRAV	First word of gravitational term array
	68	LHD	First word of hydraulic diameter array
	69	LHFG	First word of latent heat of vaporization array
	70	LQPPP	First word of wall heat source array
	71	LRMEM	First word of mixture internal energy array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	72	LRMVM	First word of array containing the product of ROM(I)*VM(I)
	73	LROM	First word of mixture density array
	74	LRHS	First word of array containing right-hand side for vapor continuity equations
	75	LSIG	First word of surface tension array
	76	LTRID	Not used
	77	LTSAT	First word of saturation temperature array
	78	LTSSN	Not used
	79	LVISL	First word of liquid viscosity array
	80	LVISV	First word of vapor viscosity array
	81	LVOL	First word of cell volume array
	82	LVR	First word of relative velocity array
	83	LWA	First word of wall area array
	84	LDFVDP	Not used
	85	LDFLDP	Not used
	86	LVLT	Not used
	87	LVVT	Not used
	88	LWFL	Not used
	89	LWFV	Not used
	90	LCIF	Not used
	91	LIDR	First word of heat transfer regime array
	92	LMATID	First word of material type array
	93	LNFF	First word of friction correlation option array
	94	LCPW	First word of wall specific heat array
	95	LCW	First word of wall conductivity array
	96	LDR	First word of radial mesh size array
	97	LEMIS	First word of wall emissivity array

TABLE 3 (continued)

Component Type	Position	Variable	Description
	98	LHOL	First word of heat transfer coefficient array (outside wall to liquid)
	99	LHOV	First word of heat transfer coefficient array (outside wall to vapor)
	100	LQPPC	First word of critical heat flux array
	101	LRN	First word of node radius array
	102	LRN2	First word of node center radius array
	103	LROW	First word of wall density array
	104	LTOL	First word of liquid temperature outside the wall array
	105	LTOV	First word of vapor temperature outside the wall array
	106	LVLTB	First word of valve table array

TABLE A.4. One-Dimensional Component Arrays

Component Type	Array	Size	Description	Pointer
See 1-D components	ALP	NCELLS	Old vapor fractions	LALP
	ALV	NCELLS	Old interfacial surface area	LALV
	ARV	NCELLS	Product of ALPN and macroscopic vapor density	LARV
	BIT	(NCELLS+1)	Bit flags from previous time step	LBIT
	EL	NCELLS	Old liquid internal energy	LEL
	EV	NCELLS	Old vapor internal energy	LEV
	HILO	NCELLS	Old heat transfer coefficient between inside wall and liquid	LHILO
	HIVO	NCELLS	Old heat transfer coefficient inside wall and vapor	LHIVO
	HLV	NCELLS	Old interfacial heat transfer coefficient	LHLV
	P	NCELLS	Old pressure	LP
	ROL	NCELLS	Old liquid density	LROL
	ROV	NCELLS	Old vapor density	LROV
	TL	NCELLS	Old liquid temperature	LTL
	TV	NCELLS	Old vapor temperature	LTV
	VM	(NCELLS+1)	Initial mixture velocity	LVM
	TW	(NODES* NCELLS)	Old wall temperature	LTW
	VL	(NCELLS+1)	Old liquid velocity	LVL
	VV	(NCELLS+1)	Old vapor velocity	LVV
	ALPN	NCELLS	New vapor fraction	LALPN
	ALVN	NCELLS	New interfacial surface area	LALVN
	BITN	(NCELLS+1)	Bit flags from current time step	LBITN
	ELN	NCELLS	New liquid internal energy	LELN
	EVN	NCELLS	New vapor internal energy	LEVN
	HIL	NCELLS	New heat transfer coefficient between inside wall and liquid	LHIL

TABLE A.4 (continued)

Component Type	Array	Size	Description	Pointer
	HIV	NCELLS	New heat transfer coefficient between inside wall and vapor	LHIV
	HLVN	NCELLS	New interfacial heat transfer coefficient	LHLVN
	PN	NCELLS	New pressure	LPN
	ROLN	NCELLS	New liquid density	LROLN
	ROVN	NCELLS	New vapor density	LROVN
	TLN	NCELLS	New liquid temperature	LTLN
	TVN	NCELLS	New vapor temperature	LTVN
	VMN	(NCELLS+1)	New mixture velocity	LVMN
	TWN	(NODES* NCELLS)	New wall temperature	LTWN
	B	(IHYDRO* 30*CELLS)	Temporary storage for implicit matrix solution	LB
	CFZ	(NCELLS+1)	Total friction factor	LCFZ
	CL	NCELLS	Liquid conductivity	LCL
	CPL	NCELLS	Liquid specific heat at constant pressure	LCPL
	CPV	NCELLS	Vapor specific heat at constant pressure	PCPV
	CV	NCELLS	Vapor conductivity	LCV
	DRIV	(NCELLS*15)	Storage for thermodynamic derivatives and enthalpies	LDRIV
	DX	NCELLS	Cell length in flow direction	LDX
	FA	(NCELLS+1)	Cell flow area	LFA
	FRIC	(NCELLS+1)	Additive friction factor	LFRIC
	GRAV	(NCELLS+1)	Gravitational terms	LGRAV
	HD	(NCELLS+1)	Hydraulic diameter	LHD
	HFG	NCELLS	Latent heat of vaporization	LHFG
	QPPP	NCELLS	Wall heat source	LQPPP

TABLE A.4 (continued)

Component Type	Array	Size	Description	Pointer
	RMEM	NCELLS	Mixture internal energy	LRMEM
	RMVM	(NCELLS+1)	Product of ROM array and VM array	LRMVM
	ROM	NCELLS	Mixture density	LROM
	RHS	(3*NCELLS)	Right hand side for vapor continuity and energy equations	LRHS
	SIG	NCELLS	Surface tension	LSIG
	TSAT	NCELLS	Saturation temperature	LTSAT
	TSSN	NCELLS	Not used	LTSSN
	VISL	NCELLS	Liquid viscosity	LVISL
	VISV	NCELLS	Vapor viscosity	LVISV
	VOL	NCELLS	Cell volume	LVOL
	VR	(NCELLS+1)	Relative velocity	LVR
	WA	NCELLS	Wall area	LWA
All 1-D components except Breaks and Fills	IDR	NCELLS	Heat transfer regime	LIDR
	MATID	(NODES-1)	Material types	LMATID
	NFF	(NCELLS+1)	Friction correlation options	LNFF
Accumulator only	BD1	LENBD	Temporary BD array	LBD1
	QPPL	NCELLS	Heat flux from wall to liquid	LQPPL
Break	BTAB	(2*NBTB* IBROP)	Break table (temperature/pressure, void fraction)	LBTAB
Fill	FTAB	(2*NFTX)	Fill velocity table	LFTAB

TABLE A.4 (continued)

Component Type	Array	Size	Description	Pointer
Pipes Pumps Component	CPW	(NODES-1* NCLHT)	Wall specific heat	LCPW
Tees, Valves, and Steam Generators	CW	(NODES-1* NCLHT)	Wall conductivity	LCW
	DR	(NODES-1)	Radial mesh size	LDR
	EMIS	NCLHT	Wall emissivity	LEMIS
	HOL	NCLHT	Heat transfer coefficient between outside wall and liquid	LHOL
	HOV	NCHLT	Heat transfer coefficient between outside wall and vapor	LHOV
	QPPC	NCELLS	Critical heat flux	LQPPC
	RN	NODES	Node radii	LRN
	RN2	(NODES-1)	Node center radii	LRN2
	ROW	(NODES-1)* NCLHT	Wall density	LROW
	TOL	NCLHT	Liquid temperature outside wall	LTOL
	TOV	NCLHT	Vapor temperature outside wall	LTOV
Pressurizer	BD1	LENBD	Temporary BD array	LBD1
	QPPL	NCELLS	Heat flux from wall to liquid	LQPPL
Pumps	SPTBL	(2*NPMPTX)	Pump speed table	LSPTBL
	NDATA	16	Number of sets of points in pump curves	LNDATA
	HSP1	(2*NDATA(1))	Single-phase head curve 1	LHSP1

TABLE A.4 (continued)

Component Type	Array	Size	Description	Pointer
	HSP2	(2*NDATA(2))	Single-phase head curve 2	LHSP2
	HSP3	(2*NDATA(3))	Single-phase head curve 3	LHSP3
	HSP4	(2*NDATA(4))	Single-phase head curve 4	LHSP4
	HTP1	(2*NDATA(5))	Two-phase head curve 1	LHTP1
	HTP2	(2*NDATA(6))	Two-phase head curve 2	LHTP2
	HTP3	(2*NDATA(7))	Two-phase head curve 3	LHTP3
	HTP4	(2*NDATA(8))	Two-phase head curve 4	LHTP4
	TSP1	(2*NDATA(9))	Single-phase torque curve 1	LTSP1
	TSP2	(2*NDATA(10))	Single-phase torque curve 2	LTSP2
	TSP3	(2*NDATA(11))	Single-phase torque curve 3	LTSP3
	TSP4	(2*NDATA(12))	Single-phase torque curve 4	LTSP4
	TTP1	(2*NDATA(13))	Two-phase torque curve 1	LTTP1
	TTP2	(2*NDATA(14))	Two-phase torque curve 2	LTTP2
	TTP3	(2*NDATA(15))	Two-phase torque curve 3	LTTP3
	TTP4	(2*NDATA(16))	Two-phase torque curve 4	LTTP4
	HDM	2*NHDM	Head degradation multiplier curve	LHDM
	TDM	2*NTDM	Torque degradation multiplier curve	LTDM
	IDXCS	15	Curve set indices	LIDXCS
Steam Generators	HLEFF	NCELL2	Effective wall-to-liquid heat transfer coefficient	LHLEFF
	HVEFF	NCELL2	Effective wall-to-vapor heat transfer coefficient. (This array is stored within the HIV array.)	LHVEFF

TABLE A.4 (continued)

Component Type	Array	Size	Description	Pointer
	TWEFF	(NODES* NCELL2)	Effective wall temperature. (This array is stored within the TW array.)	LTWEFF
	HLO	NCELL1	Wall-to-liquid heat transfer coefficient (secondary side)	LHLO
	HVO	NCELL1	Wall-to-vapor heat transfer coefficient (secondary side)	LHVO
	TLO	NCELL1	Liquid temperature (secondary side)	LTLO
	TVO	NCELL1	Vapor temperature (secondary side)	LTVO
	QPPL	NCELLT	Heat flux from wall to liquid	LQPPL
	QPPV	NCELLT	Heat flux from wall to vapor	LQPPV
Tees	BD4	LENBD	Temporary BD array	LBD4
Valves	VLTB	(2*NVTX)	Valve table	LVLTB

TABLE A.5. Composition of the A Array

Pointer Variable	Array Name	Description	Length
LTITLE	TITLE	Hollerith data describing the simulation	LENTTL
LORDER	ORDER	Array containing component numbers stored in the order used for iteration	NCOMP
LILCMP	ILCMP	Component LCM pointers stored in the order in which components were read	NCOMP
LNBR	NBR	Component numbers stored in the order in which components were read	NCOMP
LCOMPT	COMPT	Component LCM pointers stored in the order used for iteration	NCOMP
LIITNO	LIITNO	Number of inner iterations during the last outer iteration for each component (in the order used for iteration)	NCOMP
LLCON	LCON	Array containing the number of times each component was the last to converge since printout of results	NCOMP
LJUN	JUN	Junction-component pair array JUN(1,I) = the junction number JUN(2,I) = the component number connected at this junction JUN(3,I) = the component type connected at this junction JUN(4,I) = the junction direction flag 0 = positive flow into this component 1 = positive flow out of this component	4*NJUN*2

TABLE A.5 (continued)

Pointer Variable	Array Name	Description	Length
		Each junction requires two sets of this data to describe the components on either side of the junction.	
LJSEQ	JSEQ	Junction numbers in the order in which junctions occur in the junction-component array	NJUN
LVSI	VSI	Array containing the junction flow reversal indicators	NJUN
LBD	BD	Boundary data array, as follows: BD(1,I) = adjacent mesh cell width BD(2,I) = adjacent mesh cell volume Note: The remaining BD array variables are defined at one of three possible locations in the mesh cell. These are: (1) = edge of mesh cell closest to the junction (2) = mesh cell midpoint (3) = edge of mesh cell farthest from the junction BD(3,I) = old mixture density at position (2) BD(4,I) = old vapor density times old void fraction at position (2) BD(5,I) = old energy density at position (2) BD(6,I) = old mixture mass flow at position (1) BD(7,I) = old void fraction at position (2) BD(8,I) = old vapor density at position (2) BD(9,I) = old liquid density at position (2)	38*NJUN

TABLE A.5 (continued)

Pointer Variable	Array Name	Description	Length
		BD(10,I) = old mixture velocity at position (3)	
		BD(11,I) = old relative velocity at position (3)	
		BD(12,I) = old specific vapor energy at position (2)	
		BD(13,I) = old specific liquid energy at position (2)	
		BD(14,I) = old pressure at position (2)	
		BD(15,I) = new void fraction at position (2)	
		BD(16,I) = new vapor density at position (2)	
		BD(17,I) = new liquid density at position (2)	
		BD(18,I) = new mixture velocity at position (3)	
		BD(19,I) = flow area at position (3)	
		BD(20,I) = new vapor energy density at position (2)	
		BD(21,I) = new liquid energy density at position (2)	
		BD(22,I) = new pressure at position (2)	
		BD(23,I) = new mixture velocity at position (1)	
		BD(24,I) = new relative velocity at position (1)	
		BD(25,I) = surface tension at position (2)	
		BD(26,I) = new vapor velocity at position (1)	
		BD(27,I) = new liquid velocity at position (1)	
		BD(28,I) = new vapor velocity at position (2)	
		BD(29,I) = new liquid velocity at position (2)	
		BD(30,I) = vapor viscosity at position (2)	
		BD(31,I) = liquid viscosity at position (2)	
		BD(32,I) = flow area at position (1)	
		BD(33,I) = hydraulic diameter at position (1)	
		BD(34,I) = friction factor at position (2)	
		BD(35,I) = old mixture velocity at position (1)	
		BD(36,I) = last component type to enter data into this array	
		BD(37,I) = last component number to enter data into this array	

TABLE A.5 (continued)

Pointer Variable	Array Name	Description	Length
		BD(38,I) = bit flags for donor cell logic and detecting crossings of the saturation line	
LTRIP	TRIP	Array containing trip data, as follows: TRIP(1,I) = trip identification number TRIP(2,I) = trip signal index TRIP(3,I) = trip qualifier 1 TRIP(4,I) = trip qualifier 2 TRIP(5,I) = trip qualifier 3 TRIP(6,I) = trip qualifier 4 TRIP(7,I) = trip condition flag: 0 = off 1 = on TRIP(8,I) = trip set point TRIP(9,I) = trip delay time TRIP(10,I) = time at which the trip turned on TRIP(11,I) = set point reached flag: 0 = no 1 = yes TRIP(12,I) = time at which set point was reached	12*NTRX
LDMPTR	DMPTR	Array containing numbers of the trips that will cause a restart dump to be saved when they are activated	NDMPTR
LIVCON	IVCON	Array containing junction numbers with vessel connections	NVCON+1
LJOUT	JOUT	Array containing loop start points (component numbers)	IT+1*
LNVCNL	NVCNL	IA(LNVCNL+IL-1) points to the elements of IVCON AND IVLJN that begin the ILth loop	IT+1

TABLE A.5 (continued)

Pointer Variable	Array Name	Description	Length
LLOOPN	LOOPN	IA(LLOOPN+IL-1) gives the element of IORDER array that begins the ILth loop	IT+1
LNSIGP	NSIGP	Array containing the total number of components in each loop, not counting fills or breaks and counting each steam generator twice	IT
LNSIG	NSIG	Array containing the total number of components in each loop, not counting fills or breaks	IT
LNJN	NJN	Array containing the number of junctions solved for each loop	IT
LIVLJN	IVLJN	IVLJN(I) is the vessel junction number corresponding to the network junction number given by IVCON(I)	NVCON+1
LIOU	IOU	Array for junction numbers for the junctions of all components excluding breaks and fills as follows: IOU(1,I) = junction number for left boundary of one-dimensional segment IOU(2,I) = junction number for right boundary of one-dimensional segment IOU(3,I) = junction number for tee source	IT2**
LVRH	VRH	Array storage for junction momentum equation solution	NJNT
LDVB	DVB	For forward elimination: contains evaluation of junction momentum equation. For back substitution: contains changes to junction velocity	NJNT

TABLE A.5 (continued)

Pointer Variable	Array Name	Description	Length
LIDPCV	NVCON	Array containing pointers to elements in the DPVC array	NVCON
LDPVC	DPVC	Array containing coefficients for evaluating junction velocity changes as functions of pressure change in the vessel	J***
LAOU	AOU	Array containing the Jacobian for the vessel junction equation	JC****

* IT = (NVCON + the number of BREAKS + the number of FILLS)/2 + 2

** IT2 = MAX {3,3* [A(LJOUT+NLOOPS)-1]}

*** J = product of number of junctions and the number of vessel connections in each loop, summed over all the loops

**** JC = MAX [NVCON*2*(JNVSSL+1),NJNMX*(NJNMX+2)]

APPENDIX B

TABLE B.1. COBRA/TRAC Vessel Common Block List

Variable	Description	Common Block
A(M1)	Off-diagonal element in conduction matrix	ABCDT
AAB(NB,NA)	Pressure coefficients for surrounding cells	SIMSOL
AABL(NZ,3)	Pressure coefficients for levels above and below current level	REBAL
AAC(NE,NI)	Banded system pressure coefficient matrix	SIMSOL
ABOT(MC)	Momentum flow area at the bottom of the channel	MOMEN
ABSCIS(MV,NF)	Time at which boundary condition forcing function factor is applied	ABSORD
ACONT(MC,MX)	Continuity flow area	MCMX
AE(MC,MX)	New time entrained liquid volume fraction	TWOPHAS
AEOLD(MC,MX)	Old time entrained liquid volume fraction	TWOPHAS
AESINK(MS)	Entrained liquid volume fraction at pressure sink	SINK
AFACT(ML,MA)	Variation table factor	SETUPD
AFLUX	Average linear heat rate	EXTRA
AGFACT(NF,MF)	User-input cold gap width or variable gap conductance	GAPHTC
AHF(9,3)	Constants used in saturation enthalpy calculation	VARS
AHG(9,3)	Constants used in saturation enthalpy calculation	VARS
AINJT(MS)	Injection boundary condition flow area	INJECT
AINTF(MC,MX)	Cell interfacial area	GAPDAT2
AIRS(5)	Cell residual error array	SIMS1
AIRSL(5)	Level residual error array	REBAL
AJ(ME)	Coefficients for linear variation in junction velocity	REBAL
AL(MC,MX)	New time vapor volume fraction	TWOPHAS
ALAT(MC,MY)	Area for vertical convection of transverse momentum between sections	GAPDAT1
ALEX	Not used	EXTRA

TABLE B.1. (continued)

Variable	Description	Common Block
ALFA	Interpolated void fraction value for heat transfer calculation	QUEN
ALFA1	0.5*(AL(I,J-1) + AL(I,J))	QUEN
ALFA2	0.5*(AL(I,J) + AL(I,J+1))	QUEN
ALFAF	Not used	QUEN
ALFAJ	Cell-centered void fraction value for heat transfer calculation	QUEN
ALFL	Interpolated value of continuous liquid fraction	QUEN
ALFL1	0.5*(ALIQ(I,J-1) + ALIQ(I,J)	QUEN
ALFL2	0.5*(ALIQ(I,J) + ALIQ(I,J+1)	QUEN
ALFLF	Not used	QUEN
ALFLJ	Not used	QUEN
ALFJ	Cell-centered value of continuous liquid fraction	QUEN
ALIQ(MC,MX)	New time liquid volume fraction	TWOPHAS
ALIQS(MS)	Liquid volume fraction at pressure sink	SINK
ALMAX	Maximum vapor volume fraction (1.0 - ALMIN)	EXTRA
ALMIN	Minimum vapor volume fraction (1.0×10^{-8})	EXTRA
ALOLD(MC,MX)	Old time vapor volume fraction	TWOPHAS
ALSINK(MS)	Vapor volume fraction at pressure sink	SINK
AMOM(MC,MX)	Momentum flow area	MOMEN
AN(MC)	Nominal channel area	SETUPD
AP(M6)	Temporary storage for transfer of TRAC arrays between overlay levels	OVS3
ASINK(MS)	Pressure sink flow area	SINK
ATCAVG(NZ)	Cladding temperature (°K) averaged radially and axially over one fluid cell	TAXAVG
ATCIS(NZ)	Cladding inside surface temperature (°F) averaged over one fluid cell	TAXAVG

TABLE B.1. (continued)

Variable	Description	Common Block
ATFS(NZ)	Fuel outside surface temperature (°F) averaged over one fluid cell	TAXAVG
ATOP(MC)	Momentum flow area at the top of the channel	MOMEN
AXCON(NN,NH)	Axial conductance (Btu/hr-°F)	HTGEN
AXIAL(N3,NF)	Power profile tables	EXTRA
AXIALP(N3,NX)	Integrated and normalized power profile tables	RODS
AXIALT(N4,N5)	Table of axial location for rod temperature initialization	EXTRA
AXJ(NF,MF)	Table of axial location for cold gap width	GAPHTC
B(M1)	Diagonal element of conduction matrix	ABCDT
BD(38,M8)	Temporary storage for transfer of TRAC arrays between overlay levels	OVS3
BJ(ME)	Coefficients for linear variation in junction velocity	REBAL
C(M1)	Off-diagonal element of conduction matrix	ABCDT
CAEXP(N8)	Clad axial expansion	GAPHTC
CAREA(N1,MF)	Area of conduction node	RODS
CCX(5,3)	Constants in equation of state	VARS
CCXX(5,3)	Constants in equation of state	VARS
CD(MC,MX)	Form loss coefficient (velocity head)	GRDDAT
CDEXP(NZ,N8)	Clad radial expansion	GAPHTC
CELLNO(MC,MX)	Solution array cell numbers	SIMSOL
CHEN1	Term in Chen nucleate boiling correlation	QUEN
CHEN2	Term in Chen nucleate boiling correlation	QUEN
COEFD	Drop deposition coefficient	QUEN
COEFP	Drop repulsion coefficient	QUEN
COLDGP(NZ,N8)	Cold gap widths	GAPHTC
COND(N1,NX)	Thermal conductivity	HTGEN
CONS(NA)	Factor to nondimensionalize vessel error	SIMSOL
CPF	Liquid specific heat (saturated)	PROP1
CPF1(NP,MT)	Table of material specific heat	MATPRO

TABLE B.1. (continued)

Variable	Description	Common Block
CPFF(90)	Liquid specific heat table	PROPS
CPFILM	Vapor specific heat	PROP1
CPG(90)	Vapor specific heat table	PROPS
CPGAS1	Coefficient for vapor specific heat correlation	PROP1
CPGAS2	Coefficient for vapor specific heat correlation	PROP1
CPL	Subcooled liquid specific heat	PROP1
D12L(M3,M4)	$(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})_\ell$	TURBVAR
D12V(M3,M4)	$(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})_v$	TURBVAR
D32L(M2,M4)	$(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y})_\ell$	TURBVAR
D32V(M2,M4)	$(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y})_v$	TURBVAR
DATE(2)	Current date for result headings	EXTRA
DAX(NX,MR)	Axial distance between fine mesh nodes	RODS
DAXMIN(MR)	Minimum distance between fine mesh nodes	RODS
DD(M1)	Right hand side of conduction matrix	ABCDT
DDELP(NA)	Change in pressure variation from last iteration	SIMSOL
DDROP	Droplet diameter	QUEN
DE	Hydraulic diameter	QUEN
DEDVS(5)	Linear variation of continuity and energy equations with respect to the vessel connection mixture velocity	LOOP
DEDVSL(5,ME)	Level loop source derivatives	REBAL
DELP(NA)	Linear pressure variation	SIMSOL
DELPL(NZ)	Level linear pressure variation	REBAL
DFEMDP(MC,MX)	Derivative of vertical entrained liquid axial flow with respect to pressure	XTEMP

TABLE B.1. (continued)

Variable	Description	Common Block
DFGMDP(MC,MX)	Derivative of vertical vapor axial flow with respect to pressure	XTEMP
DFLMDP(MC,MX)	Derivative of vertical liquid axial flow with respect to pressure	XTEMP
DFUEL(MF)	Diameter of fuel pellet	RODS
DHFDP	Derivative of saturated liquid enthalpy with respect to pressure	PROP1
DHGDP	Derivative of saturated vapor enthalpy with respect to pressure	PROP1
DHTCDT(NX,NR)	Derivative of liquid heat transfer coefficient	HTCS
DHTSDT(MX,NV)	Derivative of liquid heat transfer coefficient for unheated conductors	HTCS
DHYD(MC,MX)	Hydraulic diameter	MCMX
DHYDN(MC)	Nominal hydraulic diameter	SETUPD
DN(MC,MX)	New time drop interfacial area density	DROP
DNO(MC,MX)	Old time drop interfacial area density	DROP
DPDTC	DP/DT term for Chen correlation	QUEN
DRIA(NC,MR)	Parameter used in conductance solution	RODS
DT	Current hydrodynamic time step size	EXTRA
DVIDH	Inverse derivative of vapor specific volume with respect to enthalpy	PROP1
DWEMDP(MG,MX)	Derivative of transverse entrained liquid flow with respect to pressure	XTEMP
DWESINK(MS)	Derivative of sink entrainment flow with respect to pressure	SINK
DWGMDP(MG,MX)	Derivative of transverse vapor flow with respect to pressure	XTEMP
DWGSINK(MS)	Derivative of sink vapor flow with respect to pressure	SINK
DWLMDP(MG,MX)	Derivative of transverse liquid flow with respect to pressure	XTEMP

TABLE B.1. (continued)

Variable	Description	Common Block
DWLSINK(MS)	Derivative of sink liquid flow with respect to pressure	SINK
DX	Axial mesh node increment	EXTRA
DX1	Axial length increment of bottom half of rod heat transfer node	EXTRA
DX2	Axial length increment of top half of rod heat transfer node	EXTRA
DXI	1/DX	EXTRA
DXS(NQ)	Vertical mesh increment for section	DROP
DXSINK(MS)	Mesh cell length for pressure sink	SINK
DXT(NX,MR)	Length of fine mesh conduction nodes	RODS
E1X(6)	Constants in equation of state	VARS
ECOUR	Not used	EXTRA
EOLDT	Not used	EXTRA
ERRIN	Maximum normalized error	SIMSOL
ETA(MC,MX)	Fraction of vapor generation coming from drops	DROP
ETAEN(MC,MX)	Not used	DROP
ETAENP(MC,MX)	Not used	DROP
ETANR(MG)	Rod bundle crossflow de-entrainment efficiency	DROP
ETIME	Current elapsed simulation time	EXTRA
F11(M2)	First element of anisotropy tensor	TURBDAT
F22(M2)	Second element of anisotropy tensor	TURBDAT
FACTOR(MG)	Sign convention normalization factor	GAPDAT1
FAPS(NS)	Flow area associated with vessel connection	LOOP
FATHX(N8)	Fuel axial thermal expansion	GAPHTC
FCOUR	Not used	EXTRA
FDEXP(NZ,N8)	Fuel diametral expansion	GAPHTC
FDT	Not used	EXTRA
FDTM	Not used	EXTRA
FEM(MC,MX)	New time vertical entrained liquid momentum flow rate	TWOPHAS

TABLE B.1. (continued)

Variable	Description	Common Block
FEMN(MC,MX)	Old time vertical entrained liquid momentum flow rate	TWOPHAS
FGAMA(NX,NR)	Fraction of heat flux that causes subcooled boiling	HTCS
FGAMS(MX,NV)	Fraction of heat flux that causes subcooled boiling for unheated conductors	HTCS
FGM(MC,MX)	New time vertical vapor momentum flow rate	TWOPHAS
FGMN(MC,MX)	Old time vertical vapor momentum flow rate	TWOPHAS
FGPFF(NF)	Temporal forcing function for gap conductance	EXTRA
FIHLB(MG,MX)	Horizontal intercell interfacial drag on liquid	XKVARS
FIHLJ(MG)	Array to accumulate horizontal intercell drag on the liquid side of the interface at the J axial level	INTFRD
FIHLJP(MG)	Array to accumulate horizontal intercell drag on the liquid side of the interface at the J+1 axial level	INTFRD
FIHVB(MG,MX)	Horizontal intercell interfacial drag on vapor	XKVARS
FIHVJ(MG)	Array to accumulate horizontal intercell drag on the vapor side of the interface at the J axial level	INTFRD
FIHVJP(MG)	Array to accumulate horizontal intercell drag on the vapor side of the interface at the J+1 axial level	INTFRD
FINLET(MC)	Initial channel flow rate	SETUPD
FIVL(MC)	Array to accumulate vertical intercell drag on the liquid side of the interface	INTFRD
FIVLB(MC,MX)	Vertical intercell interfacial drag on liquid	XKVARS
FIVV(MC)	Array to accumulate vertical intercell drag on the vapor side of the interface	INTFRD
FIVVB(MC,MX)	Vertical intercell interfacial drag on vapor	XKVARS
FLM(MC,MX)	New time vertical liquid momentum flow rate	TWOPHAS

TABLE B.1. (continued)

Variable	Description	Common Block
FLMN(MC,MX)	Old time vertical liquid momentum flow rate	TWOPHAS
FQ(NF)	Axial power factor table	EXTRA
FRACS	Interpolation constant for properties	PROP1
FRACT	Interpolation constant for properties	PROP1
FRACTW	Fraction of surface in contact with drops	QUEN
FRDREL(NZ,N8)	Fuel radial displacement due to relocation	GAPHTC
FTDENS(MF)	Fuel theoretical density	RODS
FTYPE(MF)	Alphanumeric flag for rod type	ALPHN
FWALL(MG)	Transverse wall friction flag	SPLIT
FWD	View factor--wall to drops	QUEN
FWG	View factor--wall to steam	QUEN
G	Total mass flux	QUEN
GAMA(MC,MX)	Vapor generation rate	MCMX
GAMAO(MC,MX)	Old time vapor generation rate	MCMX
GAMEXP(MC,MX)	Heat flux which causes subcooled boiling vapor generation rate	MCMX
GAMEXPO(MC,MX)	Old time value of subcooled boiling vapor generation rate	MCMX
GAP(MG,MX)	Width of transverse connection between channels	GAPDAT2
GAPH(NX,MR)	Local value of gap conductance	GAPHTC
GAPN(MG)	Nominal transverse connection width	SETUPD
GAPTH(NZ,N8)	Gap thickness	GAPHTC
GASMOL(MF)	Number of moles of gas present	GAPHTC
GC	Gravitational constant	EXTRA
GG	Vapor mass flux	QUEN
GG1	Mass flux of vapor at bottom of cell	QUEN
GG2	Mass flux of vapor at top of cell	QUEN
GIN	Initial mass flux	SETUPD
GLIQ	Liquid mass flux	QUEN
GLIQ1	Mass flux of liquid at bottom of cell	QUEN
GLIQ2	Mass flux of liquid at top of cell	QUEN

TABLE B.1. (continued)

Variable	Description	Common Block
GMULT(MG)	Number of true gaps in mesh cell	DROP
GPCON(NZ,MR)	Gap conductance at each axial level	GAPHTC
GRFN(M5)	Array used for short vessel dump option	MVYDT3
GRFX(7500)	Array used as a buffer for the post-process plot file	MVYDT3
GRID	Not used	GRDDAT
GSFRAC(6,MF)	Mole fractions of fill gas for nuclear fuel rods	EXTRA
GVAPC	Mass flux of vapor at cell center	QUEN
H(MC,MX)	Old time pressure	MCMX
H11	Local value of HH1	PROP1
H22	Local value of HH2	PROP1
H33	Local value of HH3	PROP1
H44	Local value of HH4	PROP1
H55	Local value of HH5	PROP1
H66	Local value of HH6	PROP1
HASCL(MC,MX)	Product of interfacial area and heat transfer coefficient for subcooled liquid	INTFHT
HASCV(MC,MX)	Product of interfacial area and heat transfer coefficient for subcooled vapor	INTFHT
HASHL(MC,MX)	Product of interfacial area and heat transfer coefficient for superheated liquid	INTFHT
HASHV(MC,MX)	Product of interfacial area and heat transfer coefficient for superheated vapor	INTFHT
HEATPI(MF)	Heated perimeter for rod interior surface	RODS
HEATP(MF)	Heated perimeter for rods.	RODS
HF	Saturated liquid enthalpy	PROP1
HFG	Heat of vaporization	PROP1
HG	Saturated vapor enthalpy	PROP1
HGAP(MR)	Gap conductance (constant value)	GAPHTC
HGINJ(MS)	Donor cell vapor enthalpy of injection boundary condition	INJECT

TABLE B.1. (continued)

Variable	Description	Common Block
HGSINK(MS)	Sink donor cell vapor enthalpy	SINK
HH1(90)	Saturation properties used in Chen nucleate boiling correlation	PROPS
HH2(90)	Properties used in Berenson film boiling correlation	PROPS
HH3(90)	Properties used in Bromley film boiling correlation	PROPS
HH4(90)	Properties used in Forsland-Rohsenow correlation	PROPS
HH5(90)	Properties used in Zuber CHF correlation	PROPS
HH6(90)	Properties used in Henry contact temperature factor	PROPS
HHF(90)	Saturated liquid enthalpy table	PROPS
HHG(90)	Saturated vapor enthalpy table	PROPS
HIN	Initial enthalpy	SETUPD
HINJ(MS)	Mixture enthalpy of injection boundary condition	INJECT
HL(MC,MX)	New time liquid enthalpy	TWOPHAS
HLINJ(MS)	Donor cell liquid enthalpy of injection boundary condition	INJECT
HLN(MC,MX)	Old time liquid enthalpy	TWOPHAS
HLSINK(MS)	Sink donor cell liquid enthalpy	SINK
HPERIM(NT)	Wetted perimeter for unheated conductor	RODS
HPERIMI(NT)	Wetted perimeter on interior of unheated conductor	RODS
HROD	Heater rod type flag	ALPHN
HSINK(MS)	Sink mixture enthalpy	SINK
HSPL	Dittus-Boelter correlation for liquid	QUEN
HSPV	Dittus-Boelter correlation for vapor	QUEN
HSPVC	Vapor convective heat transfer coefficient at cell center	QUEN
HSURF(NR,NZ)	Not used	RODS
HTCL(NX,NR)	Heat transfer coefficient to the liquid	RODS

TABLE B.1. (continued)

Variable	Description	Common Block
HTCLS(MX,NV)	Heat transfer coefficient to liquid for unheated conductor	RODS
HTCV(NX,NR)	Heat transfer coefficient to vapor	RODS
HTCVS(MX,NV)	Heat transfer coefficient to vapor for unheated conductor	RODS
HV(MC,MX)	New time vapor enthalpy	TWOPHAS
HVALUE(MU)	Enthalpy used for boundary conditions	ABSORD
HVN(MC,MX)	Old time vapor enthalpy	TWOPHAS
I2	Input file logical unit	EXTRA
I3	Output file logical unit	EXTRA
I8	Vessel restart logical unit	EXTRA
I9	Vessel dump logical unit	EXTRA
IACTAB(MC)	Continuity area variation table number	SETUPD
IAMTAB(MC)	Momentum area variation table number	SETUPD
IASPEC(MC,MX)	Cell boundary condition type flag	MCMX
IAXP(MR)	Flag for rod axial power table	RODS
IBEG1	First word of /ABSORD/	ABSORD
IBEG10	First word of /MATPRO/	MATPRO
IBEG11	First word of /MCMX/	MCMX
IBEG12	First word of /MOMENT/	MOMENT
IBEG13	First word of /MOMEN/	MOMEN
IBEG14	First word of /PROP1/	PROP1
IBEG14	First word of /PROPS/	PROPS
IBEG15	First word of /QUEN/	QUEN
IBEG16	First word of /RODS/	RODS
IBEG17	First word of /SIMSOL/	SIMSOL
IBEG18	First word of /SINK/	SINK
IBEG19	First word of /SPLIT/	SPLIT
IBEG2	First word of /ENTRDAT/	ENTRDAT
IBEG20	First word of /TWOPHAS/	TWOPHAS
IBEG21	First word of /VARS/	VARS

TABLE B.1. (continued)

Variable	Description	Common Block
IBEG22	First word of /VEL/	VEL
IBEG23	First word of /VELS/	VELS
IBEG24	First word of /XKVARS/	XKVARS
IBEG25	First word of /XTEMP/	XTEMP
IBEG26	First word of /EXTRA/	EXTRA
IBEG27	First word of /DROP/	DROP
IBEG28	First word of /INTFHT/	INTFHT
IBEG29	First word of /GAPHTC/	GAPHTC
IBEG30	First word of /ALPHN/	ALPHN
IBEG31	First word of /TURBDAT/	TURBDAT
IBEG32	First word of /TURBVAR/	TURBVAR
IBEG4	First word of /GAPDAT1/	GAPDAT1
IBEG5	First word of /GAPDAT2/	GAPDAT2
IBEG6	First word of /GRDDAT/	GRDDAT
IBEG7	First word of /HTGEN/	HTGEN
IBEG8	First word of /INJECT/	INJECT
IBEG9	First word of /LOOP/	LOOP
IBOUND(2,MU)	Channel and axial node numbers at which boundary conditions will be applied	ABSORD
ICENT	Center of AAC array (diagonal element of pressure array)	SIMSOL
ICHAN(2,6,6)	Not used	SPLIT
ICLL(MB,MZ)	Channels to be included in calculating the liquid level	MVYDT2
ICONF	Fuel conductivity degradation flag	GAPHTC
IDCHAN(MC)	Channel identification numbers in the order input	SPLIT
IDGAP(MG)	Gap identification numbers in the order input	SPLIT
IDGPC(MF)	Gap conductance model flag	GAPHTC
IEND1	Last word of /ABSORD/	ABSORD
IEND10	Last word of /MATPRO/	MATPRO

TABLE B.1. (continued)

Variable	Description	Common Block
IEND11	Last word of /MCMX/	MCMX
IEND12	Last word of /MOMENT/	MOMENT
IEND13	Last word of /MOMEN/	MOMEN
IEND14	Last word of /PROPS/	PROPS
IEND15	Last word of /QUEN/	QUEN
IEND16	Last word of /RODS/	RODS
IEND17	Last word of /SIMSOL/	SIMSOL
IEND18	Last word of /SINK/	SINK
IEND19	Last word of /SPLIT/	SPLIT
IEND2	Last word of /ENTRDAT/	ENTRDAT
IEND20	Last word of /TWOPHAS/	TWOPHAS
IEND21	Last word of /VARS/	VARS
IEND22	Last word of /VEL/	VEL
IEND23	Last word of /VELS/	VELS
IEND24	Last word of /XKVARS/	XKVARS
IEND25	Last word of /XTEMP/	XTEMP
IEND26	Last word of /EXTRA/	EXTRA
IEND27	Last word of /DROP/	DROP
IEND28	Last word of /INTFHT/	INTFHT
IEND29	Last word of /GAPHTC/	GAPHTC
IEND30	Last word of /ALPHN/	ALPHN
IEND31	Last word of /TURBDAT/	TURBDAT
IEND32	Last word of /TURBVAR/	TURBVAR
IEND4	Last word of /GAPDAT1/	GAPDAT1
IEND5	Last word of /GAPDAT2/	GAPDAT2
IEND6	Last word of /GRDDAT/	GRDDAT
IEND7	Last word of /HTGEN/	HTGEN
IEND8	Last word of /INJECT/	INJECT
IEND9	Last word of /LOOP/	LOOP
IERROR	Error flag	EXTRA
IFOUTH	Not used	EXTRA

TABLE B.1. (continued)

Variable	Description	Common Block
IFTURB	Flag to turn turbulence calculation on or off for the entire mesh	TURBDAT
IFTYP(MR)	Fuel type flag	RODS
IGAP(MG,3)	Number of gaps that face the II side of gap	GAPDAT1
IGAPA(MG)	Number of gaps in section above to which gap connects	GAPDAT1
IGAPB(MG)	Number of gaps in section below to which gap connects	GAPDAT1
IGAPC(MR)	Gap conductance pointer array	GAPHTC
IGATAB(MG)	Transverse connection width variation table number	SETUPD
IGDUM(15)	Miscellaneous plot file information	MVYDT3
IGRF(M5,2)	Array used for short vessel dump option	MVYDT3
IGRFIT(MH)	Component numbers to be saved for plotting	MVYDT3
IGRFOP	Vessel post-process plot dump option	MVYDAT
IK(MG)	Channel on II side of gap	GAPDAT2
ILOCS(8,MC)	Gap numbers that make transverse connections to channel MC	SPLIT
ILVL	Number of levels for rebalancing	REBAL
IMATIX(MF)	Flag for surface oxide material properties on inner heat transfer surface	RODS
IMATOX(MF)	Flag for surface oxide material properties on outer heat transfer surface	RODS
IMATYP(N1,MF)	Material type flag	RODS
IND14	Last word of /PROP1/	PROP1
INDCMP(MH,MI)	Directory array with record numbers for each component	MVYDT3
INODE(MC,MY)	Axial level of channel used to convect lateral momentum between sections	GAPDAT1
IOPT	Output option	EXTRA
IPROP	Interpolation index	PROP1

TABLE B.1. (continued)

Variable	Description	Common Block
IPROPP	Property table index	PROP1
IPROPP1	Interpolation index	PROP1
IPROPS	Interpolation index	PROP1
IPWTAB(MC)	Wetted perimeter variation table number	SETUPD
IREBAL	Rebalancing flag	SIMSOL
IRELF	Fuel radial relocation flag	GAPHTC
IRSTRT	Restart option	EXTRA
IRTAB(N4,MR)	Rods using temperature initialization table	EXTRA
ISECT	Section number	SPLIT
ISECTS(NQ,6)	General channel splitting data array	SPLIT
ISIJ(MC,MX)	Flow regime flag	MCMX
ISPEC(MU)	Boundary condition type	ABSORD
ISRC(NS)	Node number associated with vessel connection	LOOP
ISRF(NS)	Gap number associated with vessel connection	LOOP
ISRL(NS)	Channel number associated with vessel connection	LOOP
ISTAB(N4,NT)	Unheated conductors using temperature initialization table	EXTRA
ISTYP(NT)	Geometry type for unheated conductor	SIMSOL
ITERAT	Not used	EXTRA
ITIN	Number of vessel iterations	SIMSOL
ITURB(NQ)	Flag to turn turbulent shear stress calculation on or off in a section	TURBDAT
IVIS(NQ)	Flag to turn viscous shear stress calculation on or off in a section	TURBDAT
J7	Not used	EXTRA
JAXL(ML,MA)	Node associated with axial variation	SETUPD
JCELL(MZ)	Continuity cell where the liquid level ends	MVYDT2
JFLEND(MR)	Fluid node at end of rod	RODS
JFLST(MR)	Fluid node at start of rod	RODS
JFLUID(NZ,MR)	Hydrodynamics node associated with heat transfer node	RODS

TABLE B.1. (continued)

Variable	Description	Common Block
JGAP(MG,3)	Number of gaps that face the JJ side of the gap	GAPDAT1
JHT(NZ,MR)	Heat transfer node pointer array	RODS
JHTEND(MR)	Node at end of rod	RODS
JK(MG)	Channel on JJ side of the gap	GAPDAT2
JNOFLO	Not used	EXTRA
JSLL(MB,MZ)	Continuity cell where the liquid level starts	MVYDT2
JSN(NS)	Junction sequence numbers associated with vessel connections	LOOP
JSTEND(NT)	Fluid node at start of conductor	RODS
JUNS(NS)	Junction numbers associated with vessel connections	LOOP
K11	Not used	EXTRA
KA	Not used	SIMSOL
KASE	Not used	EXTRA
KBND(2,NL)	Index numbers of gaps with specified crossflows	ABSORD
KCHANA(MC,11)	Channel numbers connecting to the top of Channel I	SPLIT
KCHANB(MC,11)	Channel numbers connecting to the bottom of Channel I	SPLIT
KCHEK(MG)	Flag that indicates whether or not the intercell drag at this gap has been accumulated in the FIVL or FIVV arrays	INTFRD
KD	Not used	SIMSOL
KF	Saturated liquid thermal conductivity	PROP1
KFILM	Vapor thermal conductivity at film temperature	PROP1
KG	Saturated vapor thermal conductivity	PROP1
KGAP1(MD)	Number of the gap convecting orthogonal transverse momentum	VW
KGAP2(MD)	Gap number to which transverse momentum is convected	VW

TABLE B.1. (continued)

Variable	Description	Common Block
KGAP3(MD)	Gap number from which transverse momentum is convected	VW
KGAPA(MC,MY)	Gap number in section above to or from which the axial velocity of channel I convects momentum	GAPDAT1
KGAPB(MC,MY)	Gap number in section below to or from which the axial velocity of channel convects momentum	GAPDAT1
KKF(90)	Liquid thermal conductivity table	PROPS
KKG(90)	Vapor thermal conductivity table	PROPS
KL	Liquid thermal conductivity	PROP1
KNOFLO	Not used	EXTRA
LBDT	Temporary storage for transfer of TRAC arrays between overlay levels	OVS3
LCHAN(MC)	Section number channel I is in	SPLIT
LCS(NV)	Channel connected to conductor	RODS
LENBDT	Temporary storage for transfer of TRAC arrays between overlay levels	OVS3
LENGTH(MG)	Mesh length increment in transverse direction	GAPDAT2
LPVCT	Temporary storage for transfer of TRAC arrays between overlay levels	OVS3
LR(MC,6)	Rod numbers connecting to channel MC	RODS
LRC(NZ,MR)	Channel connected to rod	GAPHTC
LS(MC,6)	Unheated conductor numbers connecting to channel MC	RODS
LT(MC)	Number of rods in channel	RODS
LVSIT	Temporary storage for transfer of TRAC arrays between overlay levels	OVS3
MATR(N1)	Material index for subregion	EXTRA
MAXNDS	Maximum number of vessel nodes in any axial level	MVYDAT

TABLE B.1. (continued)

Variable	Description	Common Block
MCON(NA,NJ)	Contains hydrodynamic cell connection data for the vessel	SIMSOL
MODE(NX,NR)	Heat transfer mode on rods	RODS
MODES(MX,NV)	Heat transfer mode on unheated conductors	RODS
MOVIE	Option to save data for post-process movies	MVYDAT
MSIM(NA)	Array containing last cell number in each simultaneous solution group	SIMSOL
MSS(NS)	Vessel cell number associated with vessel connection	LOOP
MTSAVE	Counter on time step number	EXTRA
MXGDMP	Maximum number of post-process plot dumps	MVYDAT
N1DVAR	Number of variables being dumped to plot file for 1-D components	MVYDAT
NAAH	Constant = 8	EXTRA
NAFACT	Number of axial variation tables	SETUPD
NAMGAP(MC)	Number of gaps for which a channel convects lateral momentum between sections	GAPDAT1
NAX	Total number of axial nodes (all levels)	EXTRA
NAXL(ML)	Number of entries in each axial variation table	SETUPD
NAXN(N3)	Number of entries in axial power table	EXTRA
NAXP	Number of axial power tables	EXTRA
NC	Conduction flag	EXTRA
NCD	Number of form loss coefficients to be read	GRDDAT
NCELL	Hydrodynamic solution mesh cell number	SIMSOL
NCELLS	Number of hydrodynamic mesh cells	SIMSOL
NCGRAF	Number of components that may have data saved for post-process plots	MVYDAT
NCHANL	Total number of channels	EXTRA
NCHLL(MZ)	Number of channels included in the current liquid level plot	MVYDT2

TABLE B.1. (continued)

Variable	Description	Common Block
NCLHT	Total number of axial heat transfer mesh cells in the one-dimensional components. Equal to NCELL1 in steam generators, otherwise equal to NCELLS.	
NCOLS	Column number in system pressure coefficient array	SIMSOL
NCSEC(MB,NQ)	Channel numbers contained in axial section ISECT in the order read	SPLIT
NCSR	Number of vessel connections	LOOP
NCVAR	Number of vessel channel variables being dumped to plot file	MVYDAT
NDT	Time step number at dump	EXTRA
NDX	Total number of levels in vessel component	EXTRA
NDXP1	Total number of axial levels in channel, plus 1	EXTRA
NFUELT	Number of fuel types	EXTRA
NFUNCT	Number of forcing functions	ABSORD
NGAPS	Not used	SETUPD
NGDUMP	Current number of graphics dumps that have been taken	MVYDAT
NGPAX(MF)	Number of entries in cold gap width table	GAPHTC
NGPFF	Number of entries in gap conductance forcing function	EXTRA
NGRPS	Number of simultaneous solution groups	SIMSOL
NGVAR	Number of vessel gap variables being dumped to plot file	MVYDAT
NHFN(MU)	Number of forcing functions to be applied to axial boundary condition enthalpy	ABSORD
NHTC	Not used	EXTRA
NIBND	Number of axial and/or injection boundary conditions	ABSORD

TABLE B.1. (continued)

Variable	Description	Common Block
NK	Total number of transverse connections (gaps) between channels	GAPDAT2
NKBND	Total number of specified crossflows	ABSORD
NLEV	Not used	GRF
NLLR	Number of liquid level plots to be saved for graphics	MVYDT2
NLMGAP	Number of gaps that convect orthogonal transverse momentum	VW
NMAT	Number of materials	EXTRA
NMSK	Number of sinks	EXTRA
NNODES(MF)	Total number of nodes in conductor	RODS
NODER(N1)	Number of conduction nodes in region	EXTRA
NOUT1	Vessel output option	EXTRA
NOUT2	Vessel output option	EXTRA
NOUT3	Vessel output option	EXTRA
NOUT4	Vessel output option	EXTRA
NPCHAN(KC)	Not used	GRF
NPFN(MU)	Number of forcing function to be applied to axial boundary condition pressure or mass flow	ABSORD
NPROP	Number of entries in property tables	PROP1
NPTS(MV)	Number of points in each forcing function table	ABSORD
NQ	Number of entries in axial power factor table	EXTRA
NRAX(N4)	Number of entries in temperature initialization table	EXTRA
NRENODE(MR)	Number of time steps between rezoning	RODS
NROD	Total number of fuel rods	RODS
NRODIN(MR)	Index of heat transfer surface on inside of conductor	RODS
NRT(N4)	Number of rods using initialization tables	EXTRA
NRTAB	Number of temperature initialization tables	EXTRA

TABLE B.1. (continued)

Variable	Description	Common Block
NRVAR	Number of vessel rod variables being dumped to plot file	MVYDAT
NSECTS	Number of axial sections	SPLIT
NSKIPX	Number of axial nodes to skip between lines of printed output	EXTRA
NSROD	Number of unheated conductors	RODS
NST(N4)	Number of unheated conductors using initialization tables	EXTRA
NSTR(NT)	Slab geometry index	RODS
NTDP(MT)	Number of entries in properties versus temperature table	MATPRO
NTDP(MT)	Not used	MATPRO
NTINT	Not used	EXTRA
NTRYX	Not used	EXTRA
NTSTRT	Not used	EXTRA
NUCL	Nuclear rod flag	ALPHN
NVCT	Temporary storage for transfer of TRAC arrays between overlay levels	OVS3
NWFN(NL)	Number of forcing functions to be applied to crossflow boundary condition	ABSORD
NXTLOC	Location of next available record on the post-process plot file	MVYDAT
ORDINT(MV,MF)	Boundary condition forcing function factor	ABSORD
OXIDET	Initial clad oxide thickness	RODS
P(MC,MX)	New time pressure	MCMX
PATM	Pressure	PROP1
PATM12	$PATM^{12}$	PROP1
PATM13	$PATM^{13}$	PROP1
PATM2	$PATM^{2}$	PROP1
PATM3	$PATM^{3}$	PROP1
PATM4	$PATM^{4}$	PROP1

TABLE B.1. (continued)

Variable	Description	Common Block
PGAP(MF)	Input gas pressure	GAPHTC
PGAS(MR)	Pressure in rod N	GAPHTC
PI	Constant, π	EXTRA
PINJ(MS)	Pressure for pressure source boundary condition	INJECT
PINT(NZ,N8)	Contact pressure (fuel-clad)	GAPHTC
PL	Local pressure	QUEN
PLOG	Log (pressure)	PROP1
PLOYF(MF)	Factor in Hermite interpolation	RODS
POUT	Not used	EXTRA
POWER	Power applied to rod	RODS
POWR(NC)	Radial power profile	EXTRA
PP(90)	Pressure table	PROP1
PREF	System reference pressure	PROP1
PREFOL	Not used	PROP1
PRFILM	Vapor Prandtl number	PROP1
PRINHS(NT)	Unheated conductors to print	SPLIT
PRINTC(MC)	Channel numbers to print	SPLIT
PRINTG(ME)	Gap numbers to print	SPLIT
PRINTR(NR)	Rod number to print	SPLIT
PRL	Liquid Prandtl number	PROP1
PRLL(90)	Liquid Prandtl number table	PROPS
PSINK(MS)	Sink pressure	SINK
PSNEW(NS)	Linear variation in mixture velocity associated with vessel connection	LOOP
PSOLD(NS)	Old source pressure for vessel connection	LOOP
PTURBL(M2,M4)	Liquid turbulent pressure	TURBVAR
PTURBV(M2,M4)	Vapor turbulent pressure	TURBVAR
PVALUE(MU)	Pressure or flow used for boundary conditions	ABSORD
PW(MC)	Nominal wetted perimeter	SETUPD
QAX	Rod power	EXTRA

TABLE B.1. (continued)

Variable	Description	Common Block
QCHF	Interpolated value for critical heat flux	QUEN
QCHFF(MC,MX)	Cell critical heat flux	MCMX
QCTL(M2)	Turbulent heat flux to the liquid	TURBVAR
QCTV(M2)	Turbulent heat flux to the vapor	TURBVAR
QD	Not used	QUEN
QDM1	Not used	QUEN
QFRACT(N1,MF)	Fraction of heat generated in node	RODS
QG	Not used	QUEN
QGM1	Not used	QUEN
QHN	Near wall condensation (Hancox-Nicoll)	QUEN
QLIQ(MC,MX)	Heat transfer rate to liquid (Btu/sec)	MCMX
QLIQI(NX)	Heat transfer rate to liquid on inner surface of tube or wall	HTGEN
QLIQO(NX)	Heat transfer rate to liquid on outer surface of tube or wall	HTGEN
QNB	Not used	QUEN
QREG(N1)	Power fraction for region	EXTRA
QROD(NX,NR)	Heat flux (Btu/hr-ft^2)	RODS
QSPL	Not used	QUEN
QVAP(MC,MX)	Heat transfer rate to vapor (Btu/sec)	MCMX
RAD(N1,MF)	Radial location of node center	RODS
RADB(N6,MF)	Radial location of node boundary	RODS
RADCON(N6,N7)	Radial conductance (Btu/hr-ft^2)	HTGEN
RADIAL(MR)	Radial power factor	RODTAB
RADP(NC)	Radial location for power profile table	EXTRA
RBUBB(MC)	Small-bubble radius	DROP
RBUBBL(MC)	Large-bubble radius	DROP
RCOLD(MT)	Cold-state density for material properties table	MATPRO
RDROPP(MC)	Drop radius	DROP
REG	Vapor Reynolds number	QUEN
REGU	REG/μ	QUEN

TABLE B.1. (continued)

Variable	Description	Common Block
RFILM	Vapor density at film temperature	PROP1
RHOF	Saturated liquid enthalpy	PROP1
RHOFF(90)	Saturated liquid density table	PROPS
RHOG	Saturated vapor enthalpy	PROP1
RHOGG(90)	Saturated vapor density table	PROPS
RHOVOL(N1,MF)	Material mass in conduction node	RODS
RHS(NN,NX)	Source terms for the conduction equation	HTGEN
RI144	1/144	EXTRA
RJAC(5,NK)	Cell Jacobian matrix	SIMS1
RJACL(5,7)	Level Jacobian matrix	REBAL
RL(MC,MX)	New time liquid density	TWOPHAS
RLSINK(MS)	Sink donor cell liquid density	SINK
RMULS(NV)	Number of unheated conductors represented by a single unheated conductor	RODS
RMULT(NR)	Number of rods a single rod represents	RODS
RODQ	Power * axial profile	RODS
ROUFC(MF)	Clad surface roughness	GAPHTC
ROUFF(MF)	Fuel surface roughness	GAPHTC
ROX(NZ,MR)	Radius of the clad-oxide boundary	RODS
RUVAB(MG,3)	Lateral momentum convected by axial velocity at the bottom of a section	GAPDAT1
RUVAT(MG,3)	Lateral momentum convected by axial velocity at the top of a section	GAPDAT1
RV(MC,MX)	New time vapor density	TWOPHAS
RVSINK(MS)	Sink donor cell vapor density	SINK
SAVE(MC)	Not used	ENTRDAT
SCBMOD	Subcooled boiling fraction	QUEN
SCRS(NS)	Reduced variation of error with respect to vessel connection	LOOP
SCRSL(ME)	Level loop source error array	REBAL
SDENT(MC,MX)	Drop mass de-entrainment rate	DROP

TABLE B.1. (continued)

Variable	Description	Common Block
SDETB	Transition boiling heat flux due to drop deposition	QUEN
SENT(MC,MX)	Drop mass entrainment rate	DROP
SIGMA	Surface tension	PROP1
SINKK(MS)	Sink form loss coefficient	SINK
SLC(NS)	Vessel connection explicit liquid continuity source	LOOP
SLCV(NS)	Vessel connection implicit liquid continuity source	LOOP
SLE(NS)	Vessel connection explicit liquid energy source	LOOP
SLEV(NS)	Vessel connection implicit liquid energy source	LOOP
SLV(NS)	Not used	LOOP
SND(MC,MX)	Drop interfacial area source	DROP
SOURS(NA)	Cell reduced error array	SMIS1
SOURSL(NZ)	System rebalancing error array	REBAL
SOURST(NI)	System reduced error array	SIMS1
SPHTS(N1,NX)	Specific heat of conductor node	HTGEN
SSIGMA(90)	Surface tension table	PROPS
STORJ(NA,ND)	Storage for reduced system Jacobian used for back substitution to unfold linear variation of independent variables	SIMSOL
SUPF	Suppression factor in Chen correlation	QUEN
SURF	Surface tension	VEL
SVC(NS)	Vessel connection explicit vapor continuity source	LOOP
SVCV(NS)	Vessel connection implicit vapor continuity source	LOOP
SVE(NS)	Vessel connection explicit vapor energy source	LOOP
SVEV(NS)	Vessel connection implicit vapor energy source	LOOP

TABLE B.1. (continued)

Variable	Description	Common Block
SXXL(M2)	Liquid normal stress on bottom of vertical momentum cell $(\alpha_\ell P_\ell^T)A_{I,J}$	TURBVAR
SXXPL(M2)	Liquid normal stress on top of vertical momentum cell $(\alpha_\ell P_\ell^T)A_{I,J+1}$	TURBVAR
SXXPV(M2)	Vapor normal stress on top of vertical momentum cell	TURBVAR
SXXV(M2)	Vapor normal stress on bottom of vertical momentum cell	TURBVAR
SXYL(M3)	Liquid transverse shear on bottom of transverse momentum cell	TURBVAR
SXYPL(M3)	Liquid transverse shear on top of transverse momentum cell	TURBDAT
SXYPV(M3)	Vapor transverse shear on top of transverse momentum cell	TURBDAT
SXYV(M3)	Vapor transverse shear on bottom of transverse momentum cell	TURBVAR
SYXL(M2)	Vertical liquid shear stress on the side of vertical momentum cell	TURBVAR
SYXV(M2)	Vertical vapor shear stress on the side of vertical momentum cell	TURBVAR
SYYIIV(M3)	Vapor turbulent normal stress on II side of transverse momentum cell	TURBVAR
SYYIIL(M3)	Liquid turbulent normal stress on II side of transverse momentum cell	TURBDAT
SYYJJL(M3)	Liquid turbulent normal stress on JJ side of transverse momentum cell	TURBVAR
SYYJJV(M3)	Vapor turbulent normal stress on JJ side of transverse momentum cell	TURBVAR
SZYL(M3)	Liquid shear stress on sides of transverse momentum cell	TURBVAR
SZYV(M3)	Vapor shear stress on sides of transverse cell	TURBVAR

TABLE B.1. (continued)

Variable	Description	Common Block
TCHF	CHF temperature	QUEN
TCHF1	CHF temperature at bottom of cell	QUEN
TCHF2	CHF temperature at top of cell	QUEN
TCHFF(MC,MX)	CHF temperature	MCMX
TEXT(17)	Title information for vessel printout	EXTRA
TF	Saturation temperature (°F)	PROP1
TFAVG(NZ,N8)	Fuel temperature (°F) averaged radially over pellet and axially over one fluid node	RODS
TFLUID(NR,NX)	Not used	RODTABS
TG	Vapor temperature (°F)	QUEN
TGASP1	Term used in TGAS	PROP1
TGASP2	Term used in TGAS	PROP1
TGDMP(MI)	Simulation times at which plot data has been saved	MVYDT3
TGINT	Not used	GAPHTC
TGLST	Not used	GAPHTC
THCF(NP,MT)	Thermal conductivity of node (Btu/hr-ft-°F)	MATPRO
THN	Homogeneous nucleation temperature	PROP1
TI	Not used	EXTRA
TIME(2)	Current time of day	EXTRA
TINT	Not used	EXTRA
TL	Liquid temperature	QUEN
TLIQ(NX,NR)	Liquid temperature seen by rod	HTCS
TLIQS(MX,NV)	Liquid temperature seen by unheated conductor	HTCS
TMBOIL	Cell merger criterion in nucleate boiling	QUEN
TMERGE	Cell merger criterion for nonnucleate boiling	QUEN
TMIN	Minimum film boiling temperature	QUEN
TMINN(MC,MX)	Not used	MCMX
TMOVIE	Time interval at which to save data for movies	MVYDAT
TOLDJ(NN)	Old time rod temperature at J-1 level	HTGEN
TOLDR(NG,NH)	Old time rod temperature	HTGEN

TABLE B.1. (continued)

Variable	Description	Common Block
TPRNOL(M2)	Turbulent Prandtl number for the liquid	TURBDAT
TPRNOV(M2)	Turbulent Prandtl number for the vapor	TURBDAT
TPROP(NP,MT)	Temperatures corresponding to entries in fuel property table	MATPRO
TREG(N1)	Thickness of material region	EXTRA
TRINIT(N4,N5)	Temperature in rod initialization table	EXTRA
TROD(NN,NX,MR)	Rod temperature for each axial node and each radial node	RODS
TSBOIL	Cell split criterion for nucleate boiling	QUEN
TSPLIT	Cell split criterion for nonnucleate boiling	QUEN
TSTR(NY,MX,NT)	Temperature of unheated conductor	RODS
TT(90)	Temperature table	PROP1
TT(M1)	New rod temperature	ABCDT
TTIME	Total transient length	EXTRA
TUBE	Tube type flag	ALPHN
TVAP(NX,NR)	Vapor temperature seen by rod	HTCS
TVAPS(MX,NV)	Vapor temperature seen by unheated conductor	HTCS
UEBOT(MC)	Entrained liquid velocity of first node in section for which vertical momentum equation is solved	MOMENT
UEJ(MC)	Entrained liquid velocity, channel I, node J	VELS
UEJM(MC)	Entrained liquid velocity, channel I, node J-1	VELS
UEJP(MC)	Entrained liquid velocity, channel I, node J+1	VELS
UF	Saturated liquid viscosity	PROP1
UFEM(MC)	Vertical drop momentum into bottom of cell	MOMENT
UFGM(MC)	Vertical vapor momentum into bottom of cell	MOMENT
UFILM	Vapor viscosity	PROP1
UFLM(MC)	Vertical liquid momentum into bottom of cell	MOMENT
UFOLD(1,MG)	Lateral momentum convected by vertical velocity into bottom of cell	MOMENT
UG	Saturated vapor viscosity	PROP1

TABLE B.1. (continued)

Variable	Description	Common Block
UL	Liquid viscosity	PROP1
ULBOT(MC)	Liquid velocity of first node in section for which the vertical momentum equation is solved	MOMENT
UVJ(MC)	Vapor velocity, channel I, node J	VELS
UVJM(MC)	Vapor velocity, channel I, node J-1	VELS
UVJP(MC)	Vapor velocity, channel I, node J+1	VELS
UZL(M2)	Vertical liquid velocity	TURBVAR
UZV(M2)	Vertical vapor velocity	TURBVAR
VECTUV	Vapor vector velocity	VEL
VEINJ(MS)	Entrainment velocity at injection boundary	INJECT
VEJ(MG)	Entrained liquid velocity for gap K, node J	VELS
VEJM	Entrained liquid velocity at J-1 level	QUEN
VEJP(MG)	Entrained liquid velocity, gap K, node J+1	VELS
VELSL(NS)	Mixture velocity associated with vessel connection at beginning of time step	LOOP
VESINK(MS)	Sink drop velocity	SINK
VFACE(MG)	Flag that indicates the presence of a vertical intercell interface	INTFRD
VFEM(MC)	Vertical drop momentum convected by transverse velocities	MOMENT
VFG	Specific volume of vapor minus liquid	PROP1
VFGM(MC)	Vertical vapor momentum convected by transverse velocities	MOMENT
VFLM(MC)	Vertical liquid momentum convected by transverse velocities	MOMENT
VGJM	Vapor velocity at J-1 level	QUEN
VIHVB(MG,MX)	Horizontal intercell interfacial drag on vapor	XKVARS
VINJ(MS)	Mixture velocity at injection boundary	INJECT
VISL	Liquid viscosity	VEL
VISM	Mixture viscosity	
VISV	Vapor viscosity	VEL

TABLE B.1. (continued)

Variable	Description	Common Block
VL(NS)	Liquid volumetric flux associated with vessel connection (V_ℓ*A)	LOOP
VLFS	Liquid macroscopic density associated with vessel connection $(\alpha_\ell \rho_\ell)^*$	LOOP
VLFS(NS)	Not used	LOOP
VLINJ(MS)	Liquid velocity at injection boundary	INJECT
VLJ(MG)	Liquid velocity, gap K, node J	VELS
VLJM	Liquid velocity at J-1 level	QUEN
VLJP(MG)	Liquid velocity, gap K, node J+1	VELS
VLN(MC,MX)	Old time liquid density	TWOPHAS
VLSINK(MS)	Sink liquid velocity	SINK
VMS	Mixture velocity associated with vessel connection	LOOP
VMS(NS)	Not used	LOOP
VPLEN(MF)	Volume of rod plenum	GAPHIC
VSAVE	Not used	ENTRDAT
VSI(M8)	Temporary storage for transfer of TRAC arrays between overlay levels	OVS3
VSIG(NS)	Sign convention multiplier between one-dimensional component and vessel connection	LOOP
VV(NS)	Vapor volumetric flow associated with vessel connection (V_v*A)	LOOP
VVFS(NS)	Vapor macroscopic density associated with vessel connection source $(\alpha_v \rho_v)^*$	LOOP
VVINJ(MS)	Vapor velocity at injection boundary	INJECT
VVJ(MG)	Vapor velocity, gap K, node J	VELS
VVJP(MG)	Vapor velocity, gap K, node J+1	VELS
VVN(MC,MX)	Old time vapor density	TWOPHAS
VVSINK(MS)	Sink vapor velocity	SINK
VWEJ(MC)	Entrained liquid vertical momentum associated with a vessel connection	MOMENT

TABLE B.1. (continued)

Variable	Description	Common Block
VWEM(MG)	Transverse entrained liquid momentum convected by orthogonal transverse entrained liquid velocity	VW
VWGJ(MC)	Vapor vertical momentum convected by transverse velocities	MOMENT
VWGM(MG)	Transverse vapor momentum convected by orthogonal transverse vapor velocity	VW
VWLJ(MC)	Liquid vertical momentum convected by transverse velocities	MOMENT
VWLM(MG)	Transverse liquid momentum convected by orthogonal transverse liquid velocity	VW
VZL(M3)	Transverse liquid velocity cell j	TURBVAR
VZPL(M3)	Transverse liquid velocity cell j+1	TURBVAR
VZPV(M3)	Transverse vapor velocity cell j+1	TURBVAR
VZV(M3)	Transverse vapor velocity cell j	TURBVAR
WALL	Wall type flag	ALPHN
WEINJ(MS)	Entrainment massflow rate at injection boundary	INJECT
WEM(MG,MX)	Entrain liquid mass flow rate in transverse momentum cell	TWOPHAS
WEMO(MG,MX)	Old time liquid mass flow rate in transverse momentum cell	TWOPHAS
WESINK(MS)	Entrained droplet flow at pressure sink boundary	SINK
WESINKO(MS)	Old time entrained droplet flow at pressure sink boundary	SINK
WGINJ(MS)	Vapor mass flow rate at injection boundary	INJECT
WGM(MG,MX)	New time transverse vapor momentum flow rate	TWOPHAS
WGMO(MG,MX)	Old time transverse vapor momentum flow rate	TWOPHAS
WGSINK(MS)	New time sink vapor flow rate	SINK
WGSINKO(MS)	Old time sink vapor flow rate	SINK
WINJ(MS)	Mixture mass flow rate at injection boundary	INJECT
WKR(MG)	Lateral form drag coefficient	SPLIT

TABLE B.1. (continued)

Variable	Description	Common Block
WLINJ(MS)	Liquid mass flow rate at injection boundary	INJECT
WLM(MG,MX)	New time transverse liquid momentum flow rate	TWOPHAS
WLMO(MG,MX)	Old time transverse liquid momentum flow rate	TWOPHAS
WLSINK(MS)	New time sink liquid flow rate	SINK
WLSINKO(MS)	Old time liquid flow at pressure sink boundary	SINK
WPERIM	Wetted perimeter	VEL
WSINK(MS)	Sink mixture flow rate	SINK
WVALUE(50)	Specified transverse flow value	ABSORD
X(NZ)	Axial node coordinates	EXTRA
XA	Quality	QUEN
XC(NX,MR)	Axial node elevations for each rod	RODS
XIL(M2,M4)	$\left(\frac{U_\ell}{Pr_\ell} + \frac{U_\ell^T}{Pr_\ell^T}\right)$	TURBVAR
XIV(M2,M4)	$\left(\frac{U_v}{Pr_v} + \frac{U_v^T}{Pr_v^T}\right)$	TURBVAR
XK(MC,MX)	Vertical interfacial drag coefficient between the continuous liquid and vapor phases	XVARS
XKES(MS)	Sink interfacial drag coefficient between the liquid and vapor phases	XKVARS
XKGE(MC,MX)	Vertical interfacial drag coefficient between the entrained liquid and vapor phases	XKVARS
XKL(MG,MX)	Transverse interfacial drag coefficient between the continuous liquid and vapor phases	XKVARS
XKLE(MG,MX)	Transverse interfacial drag coefficient between the entrained liquid and vapor phases	XKVARS
XKVLS(MS)	Sink interfacial drag coefficient between the continuous liquid and vapor phases	XKVARS

TABLE B.1. (continued)

Variable	Description	Common Block
XKWEW(MG)	Transverse entrained liquid form loss coefficient, K_e	XKVARS
XKWEX(MC)	Vertical entrained liquid form loss coefficient, K_e	XKVARS
XKWLS	Sink liquid wall drag coefficient, K_ℓ	XKVARS
XKWLW(MG)	Transverse liquid wall drag coefficient, K_ℓ	XKVARS
XKWLX(MC)	Vertical liquid wall drag coefficient, K_ℓ	XKVARS
XKWVS	Sink vapor wall drag coefficient, K_v	XKVARS
XKWVW(MG)	Transverse vapor wall drag coefficient, K_v	XKVARS
XKWVX(MC)	Vertical vapor wall drag coefficient, K_v	XKVARS
XLIQ	Liquid heat flux ramp	EXTRA
XLIQ1	Liquid heat flux ramp for node above	EXTRA
XMPLUS	Not used	XKVARS
XQ(MC)	Hot wall flow regime flag	XKVARS
XTTI	Inverse Lockhart-Martinelli parameter	QUEN
XVAP	Not used	EXTRA
XVAP1	Not used	EXTRA
XZERR	Not used	EXTRA
Y(N3,NF)	Axial location for power profile	EXTRA
YGPFF(NF)	Time for gap conductance forcing function	EXTRA
YQ(NF)	Power profile table	EXTRA
Z	Total axial length	EXTRA
ZLMXXL(M2)	Liquid turbulent mixing length	TURBDAT
ZLMXXV(M2)	Vapor turbulent mixing length	TURBDAT

TABLE B.2. Important COBRA/TRAC Computational Variables

Variable	Description
AEP	Entrained liquid volume fraction in momentum cell
AERL	$(\alpha_e \rho_\ell)$ in momentum cell
ALA	Minimum void fraction for transition between churn turbulent and annular flow regimes
ALB	Void fraction for transition between small bubble and large bubble flow regimes
ALCRIT	Critical void fraction at which unstable film develops
ALIQP	Liquid volume volume fraction in momentum cell
ALP	Vapor volume fraction in momentum cell
ALRL	$(\alpha_\ell \rho_\ell)$ in momentum cell
ALRV	$(\alpha_v \rho_v)$ in momentum cell
ALSA	Void fraction for transition between large bubble and film flow regimes
ALSV	Void fraction at which single-phase vapor is assumed
CCD	Drag coefficient on drop
CDCB	Bubble drag coefficient in capped-bubble regime
CDCT	Bubble drag coefficient in distorted-bubble regime
CDNT	Bubble drag coefficient in Newton's regime
CDVR	Bubble drag coefficient in viscous regime
CONFC1	Constant used in conductivity degradation calculation
CONREL	Volume available for cracking in conductivity degradation calculation
DALDX	Change in void fraction between adjacent mesh cells

TABLE B.2. (continued)

Variable	Description
DMAX	Maximum drop size which can be lifted by vapor velocity
DSAUT	Size at which drops are created
F	F factor in Hanratty film friction factor
FI	Interfacial friction factor
FIU	Hanratty film friction factor
FL	Liquid wall friction factor
FV	Vapor wall friction factor
GA	G factor in Hanratty film friction factor
GLL	Liquid mass flux
GVL	Vapor mass flux
I	Channel number
II	Channel number on upstream side of gap (positive flow out of channel II)
J	Axial hydrodynamic node number within a section
JABS1	Absolute hydrodynamic node number from the bottom of the hydrodynamic mesh for the flow at the bottom of the continuity mesh cell
JABS2	Absolute hydrodynamic node number from the bottom of the hydrodynamic mesh for the flow at the top of the continuity mesh cell
JH1	First axial heat transfer node in hydrodynamic mesh cell
JH2	Last axial heat transfer node in hydrodynamic mesh cell
JJ	Channel number on downstream side of gap (positive flow into JJ)
JNODES	Number of hydrodynamic nodes in section
K	Gap number
PRV	Vapor Prandtl number
QGAM	Total heat flux to liquid; includes subcooled boiling

TABLE B.2. (continued)

Variable	Description
REB	Bubble Reynolds number
REL	Liquid Reynolds number
REV	Vapor Reynolds number
RFACTOR	Conductivity degradation factor
STRATE	Force on entrained liquid phase due to transverse void fraction gradients in vertically one-dimensional regions
STRATL	Force on liquid phase due to transverse void fraction gradients in vertically one-dimensional regions
STRATV	Force on vapor phase due to transverse void fraction gradients in vertically one-dimensional regions
TSURF	Maximum rod surface temperature within hydrodynamic mesh cell
VDENT	Momentum transfer to the liquid phase because of de-entrainment
VECTUR	Vector relative velocity between the vapor and liquid phases
VENT	Momentum transfer from the liquid phase as a result of entrainment
VGAM	Velocity created by vapor generation rate alone
VGAME	Momentum transfer from the entrained liquid phase because of phase change
VGAML	Momentum transfer from the liquid phase because of phase change
XKD	Mass transfer coefficient for drop deposition
XKF	Liquid thermal conductivity
XKWEC	Entrained liquid form loss coefficient
XKWLC	Liquid form loss coefficient (K/2 U)
XKWLTP	Liquid wall friction coefficient $(\frac{f}{2} \frac{M}{D_H \alpha})$

TABLE B.2. (continued)

Variable	Description
YYBOT	Void fraction at which condensation ramp begins
YYTOP	Void fraction at which evaporation ramp begins
ZZBOT	Void fraction at which condensation is shut off
ZZTOP	Void fraction at which evaporation is shut off

APPENDIX C

POST-PROCESS GRAPHICS PROGRAMS

C.1 PROGRAM GRAFIX TO READ DATA SAVED BY COBRA/TRAC

```
*COMDECK LOGIC
      COMMON /LOGIC/ TIME,AXIAL,RADIAL,STOPP
      LOGICAL TIME,AXIAL,RADIAL,STOPP
C
*COMDECK GEN
      COMMON /GEN/ NGDUMP,NCGRAF,NCVAR,NGVAR,NRVAR,
     1             IGRFOP,MXGDMP,NUM(10),
     2             LOC,IVSSL,NVSSL,IDONE
*COMDECK SPACE
      COMMON /SPACE/ ALCM(10000),X(999,5),Y(999,5),INDCMP(10,999),
     1               TGDMP(999),XM(100),YM(100)
C
      DIMENSION ILCM(10000)
C
      LEVEL 2, ALCM,ILCM,X,Y,INDCMP,TGDMP
      EQUIVALENCE (ILCM,ALCM)
C
*COMDECK VSLSPC
      COMMON /VSLSPC/ ICHOIC,IAVG,NCURVE,IA,IB,C1,C2,FACTOR,
     1                ICHA(5),INA(5),ELEVA(5),ICHB(5),
     2                INB(5),ELEVB(5),IA1,IA2,
     3                EA1,EA2,C1N(5),C2N(5),FAC(5),
     4                ICELA(5),ICELB(5),
     5                NSEC,NSS(10),NCHN(10),NCHS(10,10),
     6                IBS(10),IQS(10),PLTINT,PTIME,SHIFT,
     7                IANSWR
*COMDECK VSLDAT
      COMMON /VSLDAT/ NAX,NCHANL,NK,NROD,NFUELT,MAXNDS,MAXC,
     1                IK(10),JK(10),IDGAP(10),GAP(10,15),
     2                NCSEC(8,7),ISECTS(7,6),KCHANA(10,11),
     3                KCHANB(10,11),DXS(7),LCHAN(10),
     4                IDCHAN(10),IFTYP(11),NRODIN(11),
```

```
     5                 NNODES(8),AMOM(10,15),XX(31),RAD(14,8),
     6                 NSTR(8),FTYPE(8),NSROD,NSECTS
C
      INTEGER FTYPE
C
C
*COMDECK PLTNFO
      COMMON /PLTNFO/ DEF,NLABL,LABL(4,10),XSIZE,YSIZE,
     1                 XMIN,YMIN,XMAX,YMAX,
     2                 XL(10),YL(10),SIZE(10),ROT(10)
C
      LOGICAL DEF
*DECK GRAFIX
      PROGRAM GRAFIX (TTY,TAPE5=TTY,TAPE6=TTY,TAPE11=0,
     1                 TAPE2,TAPE3=0,TAPE4)
C
C   GRAPHIX'S OBJECTIVE IS TO PRESENT
C   THE RESULTS OF COBRA/TRAC SIMULATIONS
C   IN PICTORIAL FORM.
C
C   GRAFIX IS USER-ORIENTED (HOPEFULLY),
C   ALLOWING THE USER TO INTERACTIVELY
C   DESCRIBE THE PLOT.  THE USER MAY ALSO
C   INSTRUCT GRAFIX TO USE, AS INPUT,
C   DATA SAVED FROM A PREVIOUS INTERACTIVE
C   SESSION
C
C   GRAFIX PRODUCES 5 TYPES OF PLOTS:
C
C      1. VARIABLE VS TIME (UP TO 5 CURVES ON ONE PLOT +
C                              MEASURED DATA).
C
C      2. VARIABLE VS AXIAL DISTANCE (UP TO 5 CURVES ON ONE
C                                        PLOT + MEASURED DATA).
```

```
C
C       3. VARIABLE VS RADIAL DISTANCE--ROD TEMPERATURE ONLY
C                                       (UP TO 5 CURVES ON ONE
C                                        PLOT + MEASURED DATA).
C
C       4. VELOCITY VECTORS PLOTS--VESSEL COMPONENT ONLY
C
C       5. CONTOUR PLOTS--VESSEL COMPONENT ONLY (NOT FUNCTIONAL YET:!)
C
C    GRAFIX REQUIRES THE FOLLOWING INFORMATION
C
C       1. PLOT SIZE AND SCALING INFORMATION (OR USER CAN
C                                             SELECT DEFAULTS)
C
C       2. PLOT LABEL INFORMATION (OR USER CAN SELECT DEFAULTS)
C
C       3. PLOT TYPE
C
C       4. COMPONENT NUMBER TO BE PLOTTED
C
C       5. "FORMULA" FOR CALCULATING THE VARIABLE TO BE PLOTTED
C
C
C
C
C              A. "A" * CONVERSION FACTOR
C
C              B. ("A" + "B") * CONVERSION FACTOR
C
C              C. ("A" - "B") * CONVERSION
C
C              D. ("A" + CONSTANT 1) * CONVERSION FACTOR + CONSTANT 2
C
C
```

```
C
C       6. SELECTION FOR "A","B",CONVERSION FACTORS AND CONSTANTS
C
C          NOTE: "A" AND "B" ARE SPECIFIED BY ENTERING THE
C                TYPE OF VARIABLE TO BE PLOTTED, CHANNEL/GAP/ROD
C                NUMBER AND AXIAL NODE OR ELEVATION
C
C
C   GIVEN THIS INFORMATION, GRAFIX SELECTS THE DATA FROM
C   TAPE11 FILE AND DOES ALL THE NECESSARY WORK
C   TO GET IT READY TO PLOT.  A FILE IS WRITTEN CONTAINING
C   PLOT SIZE AND LABEL INFORMATION, PLUS THE COORDINATES
C   TO BE PLOTTED.  THIS FILE IS SENT TO THE VAX FOR PLOTTING.
C
C   ENOUGH SAID, GET ON WITH IT.
C
*CALL LOGIC
C
*CALL SPACE
C
      INTEGER VSSLH,STGENH
      DATA STGENH /10HSTGEN     /
      DATA VSSLH /10HVESSEL    /
C
C
      IDIDIT=0
C
C   IDIDIT=0, FIRST CALL TO TALK
C
C   IDIDIT>0, SUBSEQUENT CALL TO TALK
C
100   CONTINUE
C
      CALL TALK(IDIDIT)
```

```
      IF(STOPP) GO TO 500
C
      IDIDIT=IDIDIT + 1
C
C   DECIDE WHAT TYPE OF PLOT IS BEING
C   DONE AND CALL ROUTINE TO DO IT
C
      IF(TIME) CALL TIMPLT
      IF(AXIAL) CALL AXPLT
      IF(RADIAL) CALL RADPLT
C
      GO TO 100
C
500   CONTINUE
C
      STOP
      END
*DECK TALK
      SUBROUTINE TALK (IDIDIT)
C
C   THIS SUBROUTINE CONVERSES WITH THE USER
C   TO CONVEY WHAT IT KNOWS ABOUT THE TAPE11
C   FILE.
C   THE USER'S INSTRUCTIONS AS TO WHAT TO DO
C   WITH THIS DATA ARE OBTAINED.
C
C
*CALL GEN
C
*CALL SPACE
C
      INTEGER VSSLH,DAT,TIM
      INTEGER STGENH
      DATA STGENH/10HSTGEN     /
```

```
      DATA VSSLH/10HVESSEL    /
C
      IF(IDIDIT.GT.0) GO TO 1000
C
C
      IDONE=0
C
C   IDONE IS USED LATER TO SEE IF VESSEL HEADER
C   RECORD HAS BEEN READ.
C
C   FIRST, READER HEADER RECORDS FROM TAPE11 FILE
C   AND ECHO FOR USER
C
      CALL RDABSF(11,ILCM(1),15,1)
      IF(UNIT(11)) 150,125,125
C
125   CONTINUE
C
      WRITE(6,10001)
10001 FORMAT(//"***** ERROR ON TAPE11 FILE ***** RUN ABORTED")
      STOP
C
150   CONTINUE
C
      NGDUMP=ILCM(2)
C
C   NGDUMP IS THE NUMBER OF TIME STEPS FOR WHICH DATA WAS SAVED
C
      NCGRAF=ILCM(3)
C
C   NCGRAF IS THE NUMBER OF COMPONENTS FOR WHICH DATA WAS SAVED
C
      NCVAR=ILCM(4)
      NGVAR=ILCM(5)
```

```
      NRVAR=ILCM(6)
C
C   NCVAR IS THE NUMBER OF VARIABLES DUMPED FOR
C         EACH CHANNEL AND AXIAL NODE
C   NGVAR IS THE NUMBER OF VARIABLES DUMPED FOR
C         EACH GAP AND AXIAL NODE
C   NRVAR IS THE NUMBER OF VARIABLES DUMPED FOR
C         EACH ROD AND AXIAL NODE
C
C
      IGRFOP=ILCM(8)
      MXGDMP=ILCM(9)
C
C   MXGDMP IS MAXIMUM NUMBER OF GRAPHICS DUMPS
C
C
C   IGRFOP IS THE VESSEL DUMP OPTION  0 IS NORMAL  N=NO. OF ITEMS DUMPED
C
      DAT=ILCM(10)
      TIM=ILCM(11)
C
C   DAT AND TIM ARE THE DATE AND TIME THAT TAPE11 WAS LAST
C   WRITTEN TO.
C
C
C
C
C
      CALL RDABSF(11,ILCM(1),MXGDMP * NCGRAF,15 + NCGRAF + 1)
      IF(UNIT(11)) 250,225,225
C
225   CONTINUE
C
      WRITE(6,10001)
```

```
      STOP
C
250   CONTINUE
C
      KNT=1
      DO 350 J=1,NGDUMP
C
      DO 300 I=1,NCGRAF
      INDCMP(I,J)=ILCM(KNT)
C
      KNT=KNT + 1
C   INDCMP(I,J) IS THE LOCATION OF THE DATA FOR THE I'TH
C   COMPONENT, DUMP J
C
300   CONTINUE
C
350   CONTINUE
C
C
      CALL RDABSF(11,TGDMP(1),NGDUMP,15 + NCGRAF + MXGDMP *
     1    NCGRAF + 1)
C
      IF(UNIT(11)) 375,360,360
C
360   CONTINUE
C
      WRITE(6,10001)
C
      STOP
C
375   CONTINUE
C
      WRITE(6,10010) DAT,TIM,NGDUMP,NCGRAF
10010 FORMAT(//5X,"HELLO THERE! GRAFIX IS AT YOUR SERVICE"//
```

```
     1    10X,"THE DATA FILE WAS LAST WRITTEN ON ",2A10/
     2    10X,"THE FILE CONTAINS",I6," DUMPS FOR",I6,
     3    " COMPONENTS"/10X,"THE COMPONENTS ARE:"/)
C
      NVSSL=0
C
      WRITE(6,10015)
10015 FORMAT(/15X,"NO.",4X,"TYPE",10X,"TIME FIRST DATA WAS SAVED"/)
C
      DO 500 I=1,NCGRAF
C
      DO 475 NRC=1,NGDUMP
C
      NREC=INDCMP(I,NRC)
C
      IF(NREC.LE.0) GO TO 475
C
      CALL RDABSF(11,ALCM(1),10,NREC)
      IF(UNIT(11)) 450,400,400
C
400   CONTINUE
C
      WRITE(6,10001)
      STOP
C
450   CONTINUE
C
      TFIRST=ALCM(5)
      IF(ILCM(2).EQ.VSSLH) TFIRST=ALCM(3)
      IF(ILCM(2).EQ.STGENH) TFIRST=ALCM(6)
C
      NUM(I)=ILCM(6)
      IF(ILCM(2).EQ.STGENH) NUM(I)=ILCM(7)
      IF(ILCM(2).EQ.VSSLH) NUM(I)=ILCM(8)
```

```
C
C
C
      WRITE(6,10020) NUM(I),ILCM(2),TFIRST
10020 FORMAT(13X,I4,5X,A10,4X,F10.4)
C
      IF(ILCM(2).EQ.VSSLH) NVSSL=I
C
      GO TO 500
C
475   CONTINUE
C
500   CONTINUE
C
      IF(NVSSL.NE.0) IVSSL=1
C
      WRITE(6,10060)
10060 FORMAT(//5X,"IF YOU WISH TO ENTER OPTIONS INTERACTIVELY"/
     1    5X,"ENTER 0 (ZERO). IF YOU WANT TO USE INTERACTIVE"/
     2    5X,"DATA SAVED FROM AN EARLIER RUN (TAPE3) ENTER 1")
C
      READ(5,10070) INTER
10070 FORMAT(I5)
C
1000  CONTINUE
C
      IF(INTER.EQ.0) CALL NORMAL
      IF(INTER.NE.0) CALL BYPASS
C
      RETURN
C
C
      END
*DECK NORMAL
```

```
      SUBROUTINE NORMAL
C
C   THIS SUBROUTINE INTERACTS WITH
C   THE USER TO OBTAIN INFORMATION
C   ON WHAT TO PRODUCE. IT
C   CALLS "FREEF" TO INTERPRET FREE-FIELD
C   DATA.
C
C   CALLS "QUE1DN" TO FIND OUT ABOUT 1-D COMPONENTS
C
C   CALLS "QUEVSL" TO FIND OUT ABOUT VESSEL PLOTS
C
C   CALLS "RESTN" TO FIND OUT ABOUT PLOT SIZES, LABELS, ETC.
C
C
*CALL LOGIC
C
*CALL GEN
C
      STOPP=.FALSE.
C
100   CONTINUE
C
      WRITE(6,10010)
10010 FORMAT(5X,"ENTER THE COMPONENT NUMBER TO BE PLOTTED"/
     1    5X,"ENTER 0 (ZERO) TO TERMINATE THIS RUN"/)
      CALL FREEF
      READ(3,10020) ICOMP
10020 FORMAT(I5)
      IF(ICOMP.EQ.0) GO TO 500
C
      IVSSL=0
      DO 150 I=1,NCGRAF
      IF(ICOMP.NE.NUM(I)) GO TO 150
```

```
C
      IF(I.EQ.NVSSL) IVSSL=1
      LOC=I
      GO TO 200
C
150   CONTINUE
C
      WRITE(6,10030) ICOMP
10030 FORMAT(10X,"COMPONENT",I5," INVALID--TRY AGAIN")
      GO TO 100
C
200   CONTINUE
C
      IF(IVSSL.EQ.0) CALL QUE1DN
      IF(IVSSL.EQ.1) CALL QUEVN
C
      CALL RESTN
C
      RETURN
C
500   CONTINUE
C
      STOPP=.TRUE.
      RETURN
      END
*DECK QUEVN
      SUBROUTINE QUEVN
C
C   THIS SUBROUTINE OBTAINS INFORMATION
C   ABOUT THE VESSEL (WHAT TYPE OF PLOT, ETC.)
C
*CALL LOGIC
C
*CALL GEN
```

```
C
*CALL VSLSPC
C
      TIME=.FALSE.
      AXIAL=.FALSE.
      RADIAL=.FALSE.
C
      WRITE(6,10010)
10010 FORMAT(5X,"YOU HAVE SELECTED THE VESSEL")
C
      IF(IGRFOP.NE.0) GO TO 2000
C
C   IN CASE YOU CAN'T REMEMBER, IGRFOP IS THE
C   VESSEL DUMP OPTION (DEFINED IN MAIN PROGRAM)
C
C   NORMAL VESSEL DUMP
C
      WRITE(6,10020)
10020 FORMAT(/10X,"ENTER 1 FOR TIME PLOTS"/
     1        10X,"ENTER 2 FOR AXIAL PLOTS"/
     2        10X,"ENTER 3 FOR RADIAL PLOTS"/)
C
      CALL FREEF
      READ(3,10030) IOPT
10030 FORMAT(I5)
C
      GO TO (100,500,1000) IOPT
C
100   CONTINUE
C
C   USER WANTS A TIME PLOT
C
      TIME=.TRUE.
C
```

```
      WRITE(6,10040)
10040 FORMAT(5X,"YOU HAVE SELECTED A TIME PLOT."/
     1    5X,"THE FOLLOWING CALCULATIONS OR CONVERSIONS ARE"/
     2    5X,"CURRENTLY AVAILABLE:"//
     3    10X,"1 = A * FACTOR"/
     4    10X,"2 = (A + B) * FACTOR"/
     5    10X,"3 = (A - B) * FACTOR"/
     6    10X,"4 = (A + C1) * C2 + FACTOR"//
     *    5X,"ENTER YOUR CHOICE")
C
      CALL FREEF
      READ(3,10030) ICHOIC
C
      IF(ICHOIC.LT.1) ICHOIC=1
C
      IAVG=0
C
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) WRITE(6,10050)
10050 FORMAT(5X,"CHOICES 1 OR 4 MAY BE AVERAGED OVER"/
     1       5X,"AXIAL RANGE. ENTER 1 TO AVERAGE, 0 OTHERWISE")
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) CALL FREEF
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) READ(3,10030) IAVG
C
      IF(IAVG.EQ.1) NCURVE=1
C
      IF(IAVG.EQ.0) WRITE(6,10060)
10060 FORMAT(5X,"ENTER THE NUMBER OF CURVES FOR THIS PLOT."/
     1        5X,"(MIN.=1,MAX.=5)")
      IF(IAVG.EQ.0) CALL FREEF
      IF(IAVG.EQ.0) READ(3,10030) NCURVE
C
      IF(NCURVE.LT.1) NCURVE=1
      IF(NCURVE.GT.5) NCURVE=5
C
```

```
      WRITE(6,10070)
10070 FORMAT(5X,"THE FOLLOWING VARIABLES ARE POSSIBLE SELECTIONS"/
     1    5X,"FOR 'A' OR 'B'"//)
      WRITE(6,10075)
10075 FORMAT(9X,"1=LIQUID FRACTION",19X,"21=XKGE(I,J)"/
     1       9X,"2=VAPOR FRACTION",20X,"22=SDENT(I,J)"/
     2       9X,"3=ENTRAINED LIQUID FRACTION",9X,"23=SENT(I,J)"/
     3       9X,"4=LIQUID FLOW (CONTINUITY)",10X,"24=ISIJ(I,J)"/
     4       9X,"5=VAPOR FLOW (CONTINUITY)",11X,"25=HTCL(JX,N)"/
     5       9X,"6=ENTRAINED FLOW (CONTINUITY)",7X,"26=HTCV(JX,N)"/
     6       9X,"7=LIQUID FLOW (MOMENTUM)",12X,"27=QROD(JX,N)"/
     7       9X,"8=VAPOR FLOW (MOMENTUM)",13X,"28=MODE(JX,N)"/
     8       9X,"9=ENTRAINED FLOW (MOMENTUM)",9X,
     8          "29=ROD TEMPERATURE")
      WRITE(6,10080)
10080 FORMAT(8X,"10=PRESSURE",26X,
     1          "30=LIQUID CROSSFLOW (CONTINUITY)"/
     2       8X,"11=LIQUID TEMPERATURE",16X,
     3          "31=VAPOR CROSSFLOW (CONTINUITY)"/
     4       8X,"12=VAPOR TEMPERATURE",17X,
     5          "32=ENTRAINED CROSSFLOW (CONTINUITY)"/
     6       8X,"13=LIQUID DENSITY",20X,"33=HTCLS(J,N)"/
     7       8X,"14=VAPOR DENSITY",21X,"34=HTCVS(J,N)"/
     8       8X,"15=HASHL(I,J)",24X,"35=MODES(J,N)"/
     9       8X,"16=HASHV(I,J)",24X,"36=TLIQS(J,N)"/
     *       8X,"17=GAMA(I,J)",25X,"37=TVAPS(J,N)"/
     *       8X,"18=HASCL(I,J)",24X,"38=TSTR(I,J,N)"/
     *       8X,"19=HASCV(I,J)"/
     *       8X,"20=XK(I,J)"//)
      WRITE(6,10090)
10090 FORMAT(5X,"ENTER SELECTIONS FOR A,B,C1,C2,FACTOR"/
     1       5X,"-- 0 IF NOT USED  FORMAT IS (2I5,3F10)")
      CALL FREEF
      READ(3,10100) IA,IB,C1,C2,FACTOR
```

```
10100 FORMAT(2I5,3F10.0)
C
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) IB=0
      IF(ICHOIC.LT.4.AND.FACTOR.EQ.0.0) FACTOR=1.0
C
      WRITE(6,10105)
10105 FORMAT(5X,"ENTER SHIFT FOR TIME AXIS, 0 IF NOT NEEDED")
      CALL FREEF
C
      READ(3,10140) SHIFT
C
C
      IF(IAVG.EQ.1) GO TO 300
C
      DO 250 NC=1,NCURVE
      WRITE(6,10110) NC
10110 FORMAT(5X,"ENTER CHANNEL, ROD OR GAP NUMBER FOR 'A' - CURVE",I3)
      CALL FREEF
      READ(3,10030) ICHA(NC)
C
      WRITE(6,10120)
10120 FORMAT(5X,"ENTER NODE NO. (AXIAL FOR CHANNELS OR GAPS)"/
     1       5X,"RADIAL FOR ROD TEMPERATURES, 0 OTHERWISE)")
      CALL FREEF
      READ(3,10030) INA(NC)
C
      IF(IA.GT.24.AND.IA.LT.30) WRITE(6,10130)
10130 FORMAT(5X,"FOR ROD VARIABLES ENTER ELEVATION (INCHES)")
      IF(IA.GT.24.AND.IA.LT.30) CALL FREEF
      IF(IA.GT.24.AND.IA.LT.30) READ(3,10140) ELEVA(NC)
10140 FORMAT(F10.0)
C
      IF(IA.GT.32) WRITE(6,10145)
10145 FORMAT(5X,"FOR SLAB RODS ENTER AXIAL NODE AS A FLOATING",
```

```
     1    " POINT VALUE")
      IF(IA.GT.32) CALL FREEF
      IF(IA.GT.32) READ(3,10140) ELEVA(NC)
C
      IF(IB.LE.0) GO TO 250
C
C   B HAS BEEN SELECTED, TOO.
C
      WRITE(6,10150) NC
10150 FORMAT(5X,"ENTER CHANNEL, ROD OR GAP NUMBER FOR 'B' - CURVE",I3)
      CALL FREEF
      READ(3,10030) ICHB(NC)
C
      WRITE(6,10120)
      CALL FREEF
      READ(3,10030) INB(NC)
C
      IF(IB.GT.24.AND.IB.LT.30) WRITE(6,10130)
      IF(IB.GT.24.AND.IB.LT.30) CALL FREEF
      IF(IB.GT.24.AND.IB.LT.30) READ(3,10140) ELEVB(NC)
C
250   CONTINUE
C
      GO TO 400
C
C
C
300   CONTINUE
C
C   FIND OUT WHAT AXIAL RANGE WILL
C   BE USED, SINCE USER WANTS TO AVERAGE
C
C
      IF(IA.GT.24.AND.IA.LT.30) GO TO 375
```

```
C
      IF(IA.GT.32) GO TO 375
C
C
      WRITE(6,10200)
10200 FORMAT(5X,"YOU ARE AVERAGING OVER AXIAL RANGE."/
     1       5X,"ENTER THE NUMBER OF AXIAL SECTIONS"/
     2       5X,"OVER WHICH TO AVERAGE (10 MAX.)")
      CALL FREEF
      READ(3,10030) NSEC
C
      DO 350 NS=1,NSEC
C
      WRITE(6,10220) NS
10220 FORMAT(10X,"ENTER THE NUMBER OF CHANNELS TO USE"/
     1       10X,"IN SECTION",I5," (10 MAX.)")
      CALL FREEF
      READ(3,10030) NCHN(NS)
      NCHN(NS)=MINO(NCHN(NS),10)
C
C
      NCH=NCHN(NS)
C
      WRITE(6,10230)
10230 FORMAT(15X,"ENTER THE CHANNEL NUMBERS")
C
      CALL FREEF
      READ(3,10240) (NCHS(I,NS),I=1,NCH)
10240 FORMAT(10I5)
C
      WRITE(6,10250)
10250 FORMAT(20X,"ENTER FIRST AND LAST AXIAL NODES (INCLUSIVE)"/
     1       20X,"(ALL CHANNELS IN THIS SECTION USE THESE NODES)")
      CALL FREEF
```

```
      READ(3,10240) IBS(NS),IQS(NS)
C
350   CONTINUE
C
      GO TO 400
C
375   CONTINUE
C
C   GET INFORMATION FOR AVERAGING RODS
C
      WRITE(6,10260)
10260 FORMAT(5X,"ENTER ROD NO. OVER WHICH TO COMPUTE AVERAGE")
      CALL FREEF
      READ(3,10030) ICHA(1)
C
      IF(IA.EQ.29) WRITE(6,10265)
10265 FORMAT(10X,"ENTER RADIAL NODE FOR ROD TEMPERATURE")
      IF(IA.EQ.29) CALL FREEF
      IF(IA.EQ.29) READ(3,10030) INA(1)
C
      IF(IA.GT.24.AND.IA.LT.30) WRITE(6,10270)
10270 FORMAT(15X,"ENTER FIRST AND LAST AXIAL ELEVATION",
     1    " FOR ROD VARIABLES")
C
      IF(IA.GT.32) WRITE(6,10285)
10285 FORMAT(5X,"FOR SLAB RODS, ENTER FIRST AND LAST NODE",
     1    " NOS. AS FLOATING POINT VALUES")
C
      CALL FREEF
      READ(3,10280) EA1,EA2
10280 FORMAT(2F10.0)
C
400   CONTINUE
C
```

```
      WRITE(6,10300)
10300 FORMAT(5X,"DO YOU WANT MEASURED DATA ON THIS PLOT?")
      CALL FREEF
      READ(3,10320) IANSWR
10320 FORMAT(A3)
      IF(IANSWR.NE.3HYES) GO TO 475
C
      WRITE(6,10340)
10340 FORMAT(5X,"ENTER FILE NAME (5 CHAR. MAX.) FOR",
     1    " MEASURED DATA --"/5X,"MUST BE A LOCAL FILE")
      CALL FREEF
      READ(3,10360) NAME
10360 FORMAT(A5)
      CALL ASSIGN (4,NAME,-1)
C
475   CONTINUE
C
      RETURN
C
C
500   CONTINUE
C
C   USER WANTS AN AXIAL PLOT
C
      AXIAL=.TRUE.
C
      WRITE(6,10500)
10500 FORMAT(5X,"YOU HAVE SELECTED AN AXIAL PLOT."/
     1       5X,"THE FOLLOWING CALCULATIONS OR CONVERSIONS ARE"/
     2       5X,"CURRENTLY AVAILABLE"//
     3       10X,"1=A * FACTOR"/
     4       10X,"2=(A + B) * FACTOR"/
     5       10X,"3=(A - B) * FACTOR"/
     6       10X,"4=(A + C1) * C2 + FACTOR"//
```

```
     *        5X,"ENTER YOUR CHOICE")
C
      CALL FREEF
      READ(3,10030) ICHOIC
C
      IF(ICHOIC.LT.1) ICHOIC=1
C
      WRITE(6,10060)
      CALL FREEF
      READ(3,10030) NCURVE
      IF(NCURVE.LE.0) NCURVE=1
      IF(NCURVE.GT.5) NCURVE=5
C
      WRITE(6,10070)
      WRITE(6,10075)
      WRITE(6,10080)
      WRITE(6,10090)
C
      CALL FREEF
      READ(3,10100) IA,IB,C1,C2,FACTOR
C
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) IB=0
      IF(ICHOIC.LT.4.AND.FACTOR.EQ.0.0) FACTOR=1.0
C
C
      IF(IA.LT.25.OR.IA.GT.29) WRITE(6,10501)
10501 FORMAT(5X,"ENTER FIRST AND LAST AXIAL NODE NOS. (2I5)")
      IF(IA.GT.24.AND.IA.LT.30) WRITE(6,10502)
10502 FORMAT(5X,"ENTER FIRST AND LAST AXIAL ELEVATIONS (2F10)")
C
      CALL FREEF
C
      IF(IA.LT.25.OR.IA.GT.29) READ(3,10503) IA1,IA2
10503 FORMAT(2I5)
```

```
C
      IF(IA.GT.24.AND.IA.LT.30) READ(3,10504) EA1,EA2
10504 FORMAT(2F10.0)
C
      DO 550 NC=1,NCURVE
      WRITE(6,10110) NC
      CALL FREEF
      READ(3,10030) ICHA(NC)
C
      IF(IA.EQ.29.OR.IA.EQ.38) WRITE(6,10505)
10505 FORMAT(5X,"ENTER RADIAL NODE FOR ROD",
     1    " TEMPERATURES")
      IF(IA.EQ.29.OR.IA.EQ.38) CALL FREEF
      IF(IA.EQ.29.OR.IA.EQ.38) READ(3,10030) INA(NC)
C
C
      IF(IB.LE.0) GO TO 550
C
      WRITE(6,10150)
      CALL FREEF
C
      READ(3,10030) ICHB(NC)
      IF(IB.EQ.29.OR.IB.EQ.38) WRITE(6,10505)
      IF(IB.EQ.29.OR.IB.EQ.38) CALL FREEF
      IF(IB.EQ.29.OR.IB.EQ.38) READ(3,10030) INB(NC)
C
C
550   CONTINUE
C
575   CONTINUE
C
      WRITE(6,10550)
10550 FORMAT(5X,"ENTER THE TIME INTERVAL (SECONDS)"/
     1       5X,"AT WHICH TO PRODUCE PLOTS")
```

```
      CALL FREEF
      READ(3,10600) PLTINT
10600 FORMAT(F10.0)
C
      PTIME=PLTINT
C
      WRITE(6,10300)
      CALL FREEF
      READ(3,10320) IANSWR
      IF(IANSWR.NE.3HYES) GO TO 600
C
      WRITE(6,10620)
10620 FORMAT(5X,"ENTER FILE NAME (5 CHAR. MAX.) FOR",
     1    " MEASURED DATA --"/5X,"MUST BE A LOCAL AND",
     2    " MUST HAVE ALL DATA FOR ALL PLOTS TO BE"/
     3    5X,"PRODUCED  (SEE USERS MANUAL)")
      CALL FREEF
      READ(3,10360) NAME
      CALL ASSIGN (4,NAME,-1)
C
600   CONTINUE
C
C
      RETURN
C
1000  CONTINUE
C
C   USER WANTS A RADIAL PLOT
C
      RADIAL=.TRUE.
C
      WRITE(6,10700)
10700 FORMAT(5X,"YOU HAVE SELECTED A RADIAL PLOT OF ROD TEMPERATURE"/
     1       5X,"THE FOLLOWING CALCULATIONS OR CONVERSIONS ARE"/
```

```
     2        5X,"CURRENTLY AVAILABLE"//
     3        10X,"1=A * FACTOR"/
     4        10X,"2=(A + C1) * C2 + FACTOR"//
     8        5X,"ENTER YOUR CHOICE")
C
      CALL FREEF
      READ(3,10030) ICHOIC
C
      IF(ICHOIC.LT.1) ICHOIC=1
C
      WRITE(6,10705)
10705 FORMAT(5X,"ENTER 0 IF NORMAL ROD, 1 IF SLAB ROD")
      CALL FREEF
      READ(3,10030) IOPTN
      IF(IOPTN.EQ.0) IA=29
      IF(IOPTN.EQ.1) IA=38
C
C
      WRITE(6,10710)
10710 FORMAT(5X,"ENTER C1,C2 AND FACTOR (0 IF NOT USED)"/
     1        5X,"FORMAT IS (3F10)")
C
      CALL FREEF
      READ(3,10720) C1,C2,FACTOR
10720 FORMAT(3F10.0)
C
      IF(ICHOIC.EQ.1.AND.FACTOR.EQ.0.0) FACTOR=1.0
C
      WRITE(6,10060)
      CALL FREEF
      READ(3,10030) NCURVE
C
      IF(NCURVE.LE.1) NCURVE=1
      IF(NCURVE.GT.5) NCURVE=5
```

```
C
      DO 1100 NC=1,NCURVE
C
      WRITE(6,10725) NC
10725 FORMAT(5X,"ENTER ROD NUMBER - CURVE",I5)
      CALL FREEF
      READ(3,10030) ICHA(NC)
C
      IF(IA.EQ.29) WRITE(6,10730)
10730 FORMAT(10X,"ENTER AXIAL ELEVATION (INCHES)")
C
      IF(IA.EQ.38) WRITE(6,10735)
10735 FORMAT(5X,"FOR SLAB RODS, ENTER NODE NO. AS",
     1    " A FLOATING POINT VALUE")
      CALL FREEF
      READ(3,10140) ELEVA(NC)
1100  CONTINUE
C
C
      WRITE(6,10550)
      CALL FREEF
      READ(3,10600) PLTINT
C
C
      PTIME=PLTINT
C
      WRITE(6,10300)
      CALL FREEF
      READ(3,10320) IANSWR
      IF(IANSWR.NE.3HYES) GO TO 1200
C
      WRITE(6,10620)
      CALL FREEF
      READ(3,10360) NAME
```

```
      CALL ASSIGN (4,NAME,-1)
C
1200  CONTINUE
C
      RETURN
C
2000  CONTINUE
C
C   GET USER'S INPUT TO USE FOR "SHORT" DUMPS.
C
      WRITE(6,10900)
10900 FORMAT(5X,"SINCE ONLY MINIMAL DATA WAS SAVED, ONLY"/
     2       5X,"TIME PLOTS ARE AVAILABLE."//
     3       5X,"THE FOLLOWING CALCULATIONS OR CONVERSIONS ARE"/
     4       5X,"CURRENTLY AVAILABLE"//
     5       10X,"1=A * FACTOR"/
     6       10X,"2=(A + B) * FACTOR"/
     7       10X,"3=(A - B) * FACTOR"/
     8       10X,"4=(A + C1) * C2 + FACTOR"//
     *       5X,"ENTER YOUR CHOICE")
C
      CALL FREEF
      READ(3,10030) ICHOIC
C
      IF(ICHOIC.LT.1) ICHOIC=1
C
      TIME=.TRUE.
C
      WRITE(6,10060)
      CALL FREEF
      READ(3,10030) NCURVE
      IF(NCURVE.LE.0) NCURVE=1
      IF(NCURVE.GT.5) NCURVE=5
C
```

```
      DO 2100 NC=1,NCURVE
C
      WRITE(6,10910) NC
10910 FORMAT(5X,"A AND B ARE SELECTED BY ENTERING AN INTEGER"/
     1       5X,"THAT DEFINES ITS RELATIVE LOCATION"/
     2       5X,"IE, ENTER 5 TO SELECT THE FIFTH DUMPED VARIABLE"//
     3       5X,"ENTER A,B,C1,C2,FACTOR (0 IF NOT USED) - CURVE",I3/
     4       5X,"FORMAT IS (2I5,3F10)")
C
      CALL FREEF
      READ(3,10100) ICHA(NC),ICHB(NC),C1N(NC),C2N(NC),FAC(NC)
C
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) ICHB(NC)=0
      IF(ICHOIC.LT.4.AND.FACTOR.EQ.0.0) FACTOR=1.0
C
C
2100  CONTINUE
C
      WRITE(6,10105)
      CALL FREEF
      READ(3,10140) SHIFT
C
      WRITE(6,10300)
      CALL FREEF
      READ(3,10320) IANSWR
      IF(IANSWR.NE.3HYES) GO TO 2150
      WRITE(6,10340)
      CALL FREEF
      READ(3,10360) NAME
      CALL ASSIGN (4,NAME,-1)
C
2150  CONTINUE
C
C
```

```
      RETURN
      END
*DECK QUE1DN
      SUBROUTINE QUE1DN
C
C   THIS SUBRUTINE OBTAINS INFORMATION
C   ABOUT THE ONE-D COMPONENT PLOTS
C   INTERACTIVELY.
C
*CALL VSLSPC
C
*CALL LOGIC
C
      TIME=.TRUE.
C
      WRITE(6,10010)
10010 FORMAT(5X,"YOU HAVE SELECTED A ONE-D COMPONENT."/
     1       5X,"CURRENTLY ONLY TIME PLOTS ARE AVAILABLE."/
     2       5X,"THE FOLLOWING COMPUTATIONS OR CONVERSIONS"/
     3       5X,"MAY BE SELECTED"//
     4       10X,"1=A * FACTOR"/
     5       10X,"2=(A + B) * FACTOR"/
     6       10X,"3=(A - B) * FACTOR"/
     7       10X,"4=(A + C1) * C2 + FACTOR"//
     *       5X,"ENTER YOUR CHOICE")
C
      CALL FREEF
      READ(3,10020) ICHOIC
10020 FORMAT(I5)
C
      WRITE(6,10030)
10030 FORMAT(5X,"THE FOLLOWING VARIABLES ARE POSSIBLE SELECTIONS"/
     1       5X,"FOR A AND B"//
     2       10X,"1=VAPOR FRACTION",13X,"11=VAPOR FLOW"/
```

```
     3       10X,"2=PRESSURE"/
     4       10X,"3=LIQUID DENSITY"/
     5       10X,"4=VAPOR DENSITY"/
     6       10X,"5=LIQUID TEMPERATURE"/
     7       10X,"6=VAPOR TEMPERATURE"/
     8       10X,"7=LIQUID VELOCITY"/
     9       10X,"8=VAPOR VELOCITY"/
     *       10X,"9=MIXTURE VELOCITY"/
     *       9X,"10=LIQUID FLOW"/)
C
      WRITE(6,10040)
10040 FORMAT(5X,"ENTER SELECTIONS FOR A,B,C1,C2,FACTOR"/
     1       5X,"(0 IF NOT USED). FORMAT IS (2I5,3F10)")
C
      CALL FREEF
      READ(3,10050) IA,IB,C1,C2,FACTOR
10050 FORMAT(2I5,3F10.0)
C
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) IB=0
      IF(ICHOIC.LT.4.AND.FACTOR.EQ.0.0) FACTOR=1.0
C
      WRITE(6,10055)
10055 FORMAT(5X,"ENTER SHIFT FOR TIME AXIS, 0 IF NOT NEEDED")
      CALL FREEF
C
      READ(3,10057) SHIFT
10057 FORMAT(F10.0)
C
C
      WRITE(6,10060)
10060 FORMAT(5X,"ENTER THE NUMBER OF CURVES FOR THIS PLOT"/
     1       5X,"(MIN.=1, MAX.=5)"/)
      CALL FREEF
      READ(3,10020) NCURVE
```

```
      DO 150 NC=1,NCURVE
      WRITE(6,10070) NC
10070 FORMAT(5X,"ENTER CELL NUMBER FOR A - CURVE",I3)
      CALL FREEF
      READ(3,10020) ICELA(NC)
C
      IF(IB.NE.0) WRITE(6,10080)
10080 FORMAT(5X,"ENTER CELL NUMBER FOR B")
      IF(IB.NE.0) CALL FREEF
      IF(IB.NE.0) READ(3,10020) ICELB(NC)
C
150   CONTINUE
C
      WRITE(6,10100)
10100 FORMAT(5X,"DO YOU WANT MEASURED DATA ON THIS PLOT?")
      CALL FREEF
      READ(3,10110) IANSWR
10110 FORMAT(A3)
      IF(IANSWR.NE.3HYES) GO TO 300
C
      WRITE(6,10120)
10120 FORMAT(5X,"ENTER FILE NAME (5 CHAR. MAX.) FOR",
     1    " MEASURED DATA --"/5X,"MUST BE A LOCAL FILE")
      CALL FREEF
      READ(3,10130) NAME
10130 FORMAT(A5)
      CALL ASSIGN (4,NAME,-1)
C
300   CONTINUE
      RETURN
      END
*DECK RESTN
      SUBROUTINE RESTN
C
```

```
C   THIS SUBROUTINE OBTAINS BASIC INFORMAT
C   ABOUT THE PLOT (SIZE, SCALING, LABELS, ETC.)
C
*CALL GEN
C
*CALL VSLSPC
C
*CALL PLTNFO
C
*CALL LOGIC
C
      DO 200 J=1,10
      DO 100 I=1,4
      LABL(I,J)="            "
100   CONTINUE
C
200   CONTINUE
C
      DEF=.FALSE.
C
      WRITE(6,10010)
10010 FORMAT(5X,"NOW DESCRIBE THE PHYSICAL CHARACTERISTICS",
     2          " OF THE PLOT"//
     3       10X,"DEFAULT SIZE IS 6.0 IN (X) BY 5.0 IN (Y)"/
     4       10X,"DEFAULT LABELS ARE MINIMAL DESCRIPTIONS",
     5           " OF THE AXES"/
     6       10X,"DEFAULT SCALING USES MINS. AND MAXS.",
     7           " ENCOUNTERED IN THE DATA"//
     8       5X,"ENTER 0 TO USE THESE DEFAULTS, 1 TO DESCRIBE",
     9          " PLOT FULLY")
C
      CALL FREEF
      READ(3,10020) IOPT
10020 FORMAT(I5)
```

```
C
      IF(IOPT.EQ.1) GO TO 1000
C
C    USING DEFAULTS
C
      DEF=.TRUE.
C
      XSIZE=6.0
      YSIZE=5.0
      XMIN=1.0E+30
      XMAX=-1.0E+30
      YMIN=1.0E+30
      YMAX=-1.0E+30
C
      NLABL=2
C
      XL(1)=2.40
      YL(1)=-1.0
      SIZE(1)=0.14
      ROT(1)=0.0
      XL(2)=-1.0
      YL(2)=2.10
      SIZE(2)=0.14
      ROT(2)=90.0
C
      IF(.NOT.TIME) GO TO 300
C
C    SET UP LABELS FOR TIME PLOTS
C
      LABL(1,1)=" TIME, SEC"
C
      CALL GETLAB(IA,IVSSL,IGRFOP,LABEL)
C
      LABL(1,2)=LABEL
```

```
C
      RETURN
C
300   CONTINUE
C
      IF(.NOT.AXIAL) GO TO 400
C
C   SET UP AXIAL PLOT LABELS
C
      LABL(1,2)="AXIAL DIST"
C
      CALL GETLAB(IA,IVSSL,IGRFOP,LABEL)
C
      LABL(1,1)=LABEL
C
      RETURN
C
400   CONTINUE
C
C   SET UP RADIAL PLOT LABELS
C
      LABL(1,1)="RADIAL DIS"
C
      CALL GETLAB(IA,IVSSL,IGRFOP,LABEL)
C
      LABL(1,2)=LABEL
C
      RETURN
C
1000  CONTINUE
C
C   USER WILL DESCRIBE PLOT EXPLICITLY
C
      WRITE(6,10050)
```

```
10050 FORMAT(5X,"ENTER XSIZE,YSIZE,XMIN,XMAX,YMIN,YMAX"//
     1       11X,"WHERE XSIZE IS LENGTH OF X (HORIZONTAL) AXIS"/
     2       17X,"YSIZE IS LENGTH OF Y (VERTICAL) AXIS"/
     3       17X,"XMIN AND YMIN ARE MINIMUM DATA VALUES"/
     4       17X,"XMAX AND YMAX ARE MAXIMUM DATA VALUES"//
     5       11X,"FORMAT IS (6F10)")
C
      CALL FREEF
      READ(3,10060) XSIZE,YSIZE,XMIN,XMAX,YMIN,YMAX
10060 FORMAT(6F10.0)
C
      WRITE(6,10070)
10070 FORMAT(5X,"ENTER THE NUMBER OF LABELS (AXIAL LABELS,TITLES,ETC.)"/
     1       5X,"MAXIMUM IS 10"/)
      CALL FREEF
      READ(3,10020) NLABL
C
      IF(NLABL.LT.1) RETURN
C
      IF(NLABL.GT.10) NLABL=10
C
C   READ LABEL INFO.
C
      WRITE(6,10080)
10080 FORMAT(5X,"FOR EACH LABEL, ENTER XL,YL (IN INCHES ON THE PLOT)"/
     1       27X,"SIZE (INCHES HIGH), ROTATION (IN DEG.), AND THE LABEL"
     2      /5X,"FORMAT IS (4F10,4A10)")
C
      DO 1100 NL=1,NLABL
C
      WRITE(6,10090) NL
10090 FORMAT(5X,"LABEL",I3/)
      CALL FREEF
      READ(3,10100) XL(NL),YL(NL),SIZE(NL),ROT(NL),(LABL(I,NL),I=1,4)
```

```
10100 FORMAT(4F10.0,4A10)
C
1100  CONTINUE
C
      RETURN
C
      END
*DECK BYPASS
      SUBROUTINE BYPASS
C
C   THIS SUBROUTINE SHORT-CIRCUITS ALL OF
C   THE USER INTERACTION.  IT READS THE USER'S
C   INPUT FROM TAPE3, WHICH WAS SAVED
C   BY A PREVIOUS RUN.
C
C
C   CALLS "QUE1DB" TO FIND OUT ABOUT 1-D COMPONENTS
C
C   CALLS "QUEVB" TO FIND OUT ABOUT VESSEL PLOTS
C
C   CALLS "RESTB" TO FIND OUT ABOUT PLOT SIZES, LABELS, ETC.
C
C
*CALL LOGIC
C
*CALL GEN
C
      STOPP=.FALSE.
C
100   CONTINUE
C
      READ(4,10020) ICOMP
10020 FORMAT(I5)
      IF(ICOMP.EQ.0) GO TO 500
```

```
C
      IVSSL=0
      DO 150 I=1,NCGRAF
      IF(ICOMP.NE.NUM(I)) GO TO 150
C
      IF(I.EQ.NVSSL) IVSSL=1
      LOC=I
      GO TO 200
C
150   CONTINUE
C
      WRITE(6,10030) ICOMP
10030 FORMAT(10X,"COMPONENT",I5," INVALID--TRY AGAIN")
      GO TO 100
C
200   CONTINUE
C
      IF(IVSSL.EQ.0) CALL QUE1DB
      IF(IVSSL.EQ.1) CALL QUEVB
C
      CALL RESTB
C
      RETURN
C
500   CONTINUE
C
      STOPP=.TRUE.
      RETURN
      END
*DECK QUEVB
      SUBROUTINE QUEVB
C
C   THIS SUBROUTINE OBTAINS INFORMATION
C   ABOUT THE VESSEL (WHAT TYPE OF PLOT, ETC.)
```

```
C
*CALL LOGIC
C
*CALL GEN
C
*CALL VSLSPC
C
      TIME=.FALSE.
      AXIAL=.FALSE.
      RADIAL=.FALSE.
C
C
      IF(IGRFOP.NE.0) GO TO 2000
C
C   IN CASE YOU CAN'T REMEMBER, IGRFOP IS THE
C   VESSEL DUMP OPTION (DEFINED IN MAIN PROGRAM)
C
C   NORMAL VESSEL DUMP
C
      READ(4,10030) IOPT
10030 FORMAT(I5)
C
      GO TO (100,500,1000) IOPT
C
100   CONTINUE
C
C   USER WANTS A TIME PLOT
C
      TIME=.TRUE.
C
      READ(4,10030) ICHOIC
      IAVG=0
C
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) READ(4,10030) IAVG
```

```
C
      IF(IAVG.EQ.1) NCURVE=1
C
      IF(IAVG.EQ.0) READ(4,10030) NCURVE
C
      IF(NCURVE.LE.0) NCURVE=1
      IF(NCURVE.GT.5) NCURVE=5
C
      READ(4,10100) IA,IB,C1,C2,FACTOR
10100 FORMAT(2I5,3F10.0)
C
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) IB=0
      IF(ICHOIC.LT.4.AND.FACTOR.EQ.0.0) FACTOR=1.0
C
      READ(4,10140) SHIFT
C
C
      IF(IAVG.EQ.1) GO TO 300
C
      DO 250 NC=1,NCURVE
      READ(4,10030) ICHA(NC)
C
      READ(4,10030) INA(NC)
C
      IF(IA.GT.24.AND.IA.LT.30) READ(4,10140) ELEVA(NC)
10140 FORMAT(F10.0)
C
      IF(IA.GT.32) READ(4,10140) ELEVA(NC)
C
      IF(IB.LE.0) GO TO 250
C
C   B HAS BEEN SELECTED, TOO.
C
      READ(4,10030) ICHB(NC)
```

```
C
      READ(4,10030) INB(NC)
C
      IF(IB.GT.24.AND.IB.LT.30) READ(4,10140) ELEVB(NC)
      IF(IB.GT.32) READ(4,10140) ELEVB(NC)
C
C
250   CONTINUE
C
      GO TO 400
C
C
300   CONTINUE
C
C   FIND OUT WHAT AXIAL RANGE WILL
C   BE USED, SINCE USER WANTS TO AVERAGE
C
C
      IF(IA.GT.24.AND.IA.LT.30) GO TO 375
C
      IF(IA.GT.32) GO TO 375
C
C
      READ(4,10030) NSEC
C
      DO 350 NS=1,NSEC
C
      CALL FREEF
      READ(4,10030) NCHN(NS)
      NCHN(NS)=MINO(NCHN(NS),10)
C
C
      NCH=NCHN(NS)
C
```

```
C
      READ(4,10240) (NCHS(I,NS),I=1,NCH)
10240 FORMAT(10I5)
C
      READ(4,10240) IBS(NS),IQS(NS)
C
350   CONTINUE
C
      GO TO 400
C
375   CONTINUE
C
C   GET INFORMATION FOR AVERAGING RODS
C
      READ(4,10030) ICHA(1)
C
      IF(IA.EQ.29) READ(4,10030) INA(1)
C
      READ(4,10280) EA1,EA2
10280 FORMAT(2F10.0)
C
400   CONTINUE
C
      READ(4,10320) IANSWR
10320 FORMAT(A3)
      IF(IANSWR.NE.3HYES) GO TO 475
C
      READ(4,10360) NAME
10360 FORMAT(A5)
      CALL ASSIGN (4,NAME,-1)
C
475   CONTINUE
      RETURN
C
```

```
C
500   CONTINUE
C
C   USER WANTS AN AXIAL PLOT
C
      AXIAL=.TRUE.
C
      READ(4,10030) ICHOIC
      READ(4,10030) NCURVE
      IF(NCURVE.LE.0) NCURVE=1
      IF(NCURVE.GT.5) NCURVE=5
C
      READ(4,10100) IA,IB,C1,C2,FACTOR
C
C
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) IB=0.0
      IF(ICHOIC.LT.4.AND.FACTOR.EQ.0.0) FACTOR=1.0
C
C
C
      IF(IA.LT.25.OR.IA.GT.29) READ(4,10503) IA1,IA2
10503 FORMAT(2I5)
C
      IF(IA.GT.24.AND.IA.LT.30) READ(4,10504) EA1,EA2
10504 FORMAT(2F10.0)
C
      DO 550 NC=1,NCURVE
      READ(4,10030) ICHA(NC)
C
      IF(IA.EQ.29.OR.IA.EQ.38) READ(4,10030) INA(NC)
C
C
C
C
```

```
      IF(IB.LE.0) GO TO 550
C
C
      READ(4,10030) ICHB(NC)
      IF(IB.EQ.29.OR.IB.EQ.38) READ(4,10030) INB(NC)
C
C
550   CONTINUE
C
575   CONTINUE
C
      READ(4,10600) PLTINT
10600 FORMAT(F10.0)
C
      PTIME=PLTINT
C
      READ(4,10320) IANSWR
      IF(IANSWR.NE.3HYES) GO TO 600
C
      READ(4,10360) NAME
      CALL ASSIGN (4,NAME,-1)
C
600   CONTINUE
      RETURN
C
1000  CONTINUE
C
C   USER WANTS A RADIAL PLOT
C
      RADIAL=.TRUE.
C
      READ(4,10030) ICHOIC
C
      READ(4,10030) IOPTN
```

```
      IF(IOPTN.EQ.0) IA=29
      IF(IOPTN.EQ.1) IA=38
C
C
      READ(4,10720) C1,C2,FACTOR
10720 FORMAT(3F10.0)
C
      IF(ICHOIC.EQ.1.AND.FACTOR.EQ.0.0) FACTOR=1.0
C
      READ(4,10030) NCURVE
C
      IF(NCURVE.LE.1) NCURVE=1
      IF(NCURVE.GT.5) NCURVE=5
C
      DO 1100 NC=1,NCURVE
C
      READ(4,10030) ICHA(NC)
C
      READ(4,10140) ELEVA(NC)
1100  CONTINUE
C
C
      READ(4,10600) PLTINT
C
      PTIME=PLTINT
C
      READ(4,10320) IANSWR
      IF(IANSWR.NE.3HYES) GO TO 1200
      READ(4,10360) NAME
      CALL ASSIGN (4,NAME,-1)
C
1200  CONTINUE
C
      RETURN
```

```
C
2000  CONTINUE
C
C   GET USER'S INPUT TO USE FOR "SHORT" DUMPS.
C
      READ(4,10030) ICHOIC
C
      TIME=.TRUE.
C
      READ(4,10030) NCURVE
      IF(NCURVE.LE.0) NCURVE=1
      IF(NCURVE.GT.5) NCURVE=5
C
      DO 2100 NC=1,NCURVE
C
      CALL FREEF
      READ(4,10100) ICHA(NC),ICHB(NC),C1N(NC),C2N(NC),FAC(NC)
      IF(ICHOIC.EQ.1.OR.ICHOIC.EQ.4) ICHB(NC)=0
      IF(ICHOIC.LT.4.AND.FACTOR.EQ.0.0) FACTOR=0.0
C
2100  CONTINUE
C
      READ(4,10140) SHIFT
      READ(4,10320) IANSWR
      IF(IANSWR.NE.3HYES) GO TO 2150
      READ(4,10360) NAME
      CALL ASSIGN (4,NAME,-1)
C
2150  CONTINUE
C
      RETURN
      END
*DECK QUE1DB
      SUBROUTINE QUE1DB
```

```
C
C   THIS SUBRUTINE OBTAINS INFROMATION
C   ABOUT THE ONE-D COMPONENT PLOTS
C   INTERACTIVELY.
C
*CALL VSLSPC
C
*CALL LOGIC
C
      TIME=.TRUE.
C
      READ(4,10020) ICHOIC
10020 FORMAT(I5)
C
C
      READ(4,10050) IA,IB,C1,C2,FACTOR
10050 FORMAT(2I5,3F10.0)
C
      READ(4,10057) SHIFT
10057 FORMAT(F10.0)
C
      READ(4,10020) NCURVE
C
      DO 150 NC=1,NCURVE
      READ(4,10020) ICELA(NC)
C
      IF(IB.GT.0) READ(4,10020) ICELB(NC)
C
150   CONTINUE
C
      READ(4,10110) IANSWR
10110 FORMAT(A3)
      IF(IANSWR.NE.3HYES) GO TO 300
C
```

```
      READ(4,10130) NAME
10130 FORMAT(A5)
C
      CALL ASSIGN (4,NAME,-1)
C
300   CONTINUE
C
      RETURN
      END
*DECK RESTB
      SUBROUTINE RESTB
C
C   THIS SUBROUTINE OBTAINS BASIC INFORMAT
C   ABOUT THE PLOT (SIZE, SCALING, LABELS, ETC.)
C
*CALL GEN
C
*CALL VSLSPC
C
*CALL PLTNFO
C
*CALL LOGIC
C
      DO 200 J=1,10
      DO 100 I=1,4
      LABL(I,J)="          "
100   CONTINUE
C
200   CONTINUE
C
      DEF=.FALSE.
C
      READ(4,10020) IOPT
10020 FORMAT(I5)
```

```
C
      IF(IOPT.EQ.1) GO TO 1000
C
C    USING DEFAULTS
C
      DEF=.TRUE.
C
      XSIZE=6.0
      YSIZE=5.0
      XMIN=1.0E+30
      XMAX=-1.0E+30
      YMIN=1.0E+30
      YMAX=-1.0E+30
C
      NLABL=2
C
      XL(1)=2.40
      YL(1)=-1.0
      SIZE(1)=0.14
      ROT(1)=0.0
      XL(2)=-1.0
      YL(2)=2.10
      SIZE(2)=0.14
      ROT(2)=90.0
C
      IF(.NOT.TIME) GO TO 300
C
C    SET UP LABELS FOR TIME PLOTS
C
      LABL(1,1)=" TIME, SEC"
C
      CALL GETLAB(IA,IVSSL,IGRFOP,LABEL)
C
      LABL(1,2)=LABEL
```

```
C
      RETURN
C
300   CONTINUE
C
      IF(.NOT.AXIAL) GO TO 400
C
C   SET UP AXIAL PLOT LABELS
C
      LABL(1,2)="AXIAL DIST"
C
      CALL GETLAB(IA,IVSSL,IGRFOP,LABEL)
C
      LABL(1,1)=LABEL
C
      RETURN
C
400   CONTINUE
C
C   SET UP RADIAL PLOT LABELS
C
      LABL(1,1)="RADIAL DIS"
C
      CALL GETLAB(IA,IVSSL,IGRFOP,LABEL)
C
      LABL(1,2)=LABEL
C
      RETURN
C
1000  CONTINUE
C
C   USER WILL DESCRIBE PLOT EXPLICITLY
C
      READ(4,10060) XSIZE,YSIZE,XMIN,XMAX,YMIN,YMAX
```

```
10060 FORMAT(6F10.0)
C
      READ(4,10020) NLABL
C
      IF(NLABL.LT.1) RETURN
C
      IF(NLABL.GT.10) NLABL=10
C
C   READ LABEL INFO.
C
C
      DO 1100 NL=1,NLABL
C
      READ(4,10100) XL(NL),YL(NL),SIZE(NL),ROT(NL),(LABL(I,NL),I=1,4)
10100 FORMAT(4F10.0,4A10)
C
1100  CONTINUE
C
      RETURN
C
      END
*DECK GETLAB
      SUBROUTINE GETLAB(IA,IVSSL,IGRFOP,LABEL)
C
C   THIS SUBROUTINE FINDS DEFAULT LABELS
C
      DIMENSION LVSSL(38),LONED(11)
C
      DATA LBLANK /"          "/
C
      DATA (LVSSL(I),I=1,38)/"ALIQ(I,J) ","AL(I,J)   ","AE(I,J)   ",
     1                       "FLIQ CONT.","FVAP CONT.","FENT CONT.",
     2                       "FLIQ MOM. ","FVAP MOM. ","FENT MOM. ",
     3                       "P(I,J)    ","TLIQ      ","TVAP      ",
```

```
     4                       "LIQ. DENS.","VAP. DENS.","HASHL(I,J)",
     5                       "HASHV(I,J)","GAMA(I,J) ","HASCL(I,J)",
     6                       "HASCV(I,J)","XK(I,J)   ","XKGE(I,J) ",
     7                       "SDENT(I,J)","SENT(I,J) ","ISIJ(I,J) ",
     8                       "HTCL(JX,N)","HTCV(JX,N)","QROD(JX,N)",
     9                       "MODE(JX,N)","TROD      ","WL(K,J)   ",
     *                       "WV(K,J)   ","WE(K,J)   ","HTCLS(J,N)",
     *                       "HTCVS(J,N)","MODES(J,N)","TLIQS(J,N)",
     *                       "TVAPS(J,N)","TSTR      "/
C
      DATA (LONED(I),I=1,11)/"AL(NC)    ","P(NC)     ","LIQ. DENS.",
     1                       "VAP. DENS.","TLIQ(NC)  ","TVAP(NC)  ",
     2                       "LIQ. VEL. ","VAP. VEL. ","MIX. VEL. ",
     3                       "LIQ. FLOW ","VAPOR FLOW"/
C
      IF(IVSSL.LE.0) GO TO 100
C
C   VESSEL COMPONENT
C
      IF(IGRFOP.EQ.0) LABEL=LVSSL(IA)
      IF(IGRFOP.NE.0) LABEL=LBLANK
C
      RETURN
C
100   CONTINUE
C
C   ONE DIMENSIONAL COMPONENT
C
      LABEL=LONED(IA)
C
      RETURN
      END
*DECK TIMPLT
      SUBROUTINE TIMPLT
```

```
C
C   THIS SUBROUTINE DOES EVERYTHING TO
C   SAVE DATA FOR TIME PLOTS
C
*CALL VSLSPC
C
*CALL PLTNFO
C
*CALL GEN
C
*CALL SPACE
C
      DO 3000 NPTS=1,NGDUMP
C
      IF(NPTS.EQ.1.AND.IDONE.EQ.0) CALL VESSL1
C
      NREC=INDCMP(LOC,NPTS)
      IF(NREC.LE.0) GO TO 3000
C
      CALL RDABSF(11,ALCM(1),1,NREC)
C
      IF(UNIT(11)) 125,100,100
C
100   CONTINUE
C
      WRITE(6,10010)
10010 FORMAT(//" ***** ERROR ON TAPE11 ***** -- ABORTING"//)
      STOP
C
125   CONTINUE
C
C
      DO 2000 NC=1,NCURVE
C
```

```
      NWDS=ILCM(1)
C
C   NWDS IS THE NUMBER OF WORDS ON THIS RECORD
C
      CALL RDABSF(11,ALCM(1),NWDS,NREC)
      IF(UNIT(11)) 175,150,150
C
150   CONTINUE
C
      WRITE(6,10010)
      STOP
C
175   CONTINUE
C
C   OK, HAVE READ THE RECORD.
C
C   EXTRACT DATA AND DO CALCULATIONS
C
C
      IF(IAVG.NE.0) CALL AVGVSL (A,TIM)
C
      IF(IAVG.NE.0) GO TO 250
C
      IF(IVSSL.EQ.0) CALL GET1D(IA,IB,ICELA(NC),ICELB(NC),A,B,TIM)
C
C
      IF(IVSSL.EQ.1) CALL GETVSL(IA,IB,ICHA(NC),ICHB(NC),
     1                           INA(NC),INB(NC),ELEVA(NC),
     2                           ELEVB(NC),A,B,TIM)
C
250   CONTINUE
C
C
C   NOW WE HAVE A, B, AND TIME
```

```
C
      GO TO (300,400,500,600) ICHOIC
C
300   CONTINUE
C
C   COMPUTE (A * FACTOR) STORE IN Y
C
      Y(NPTS,NC)=A * FACTOR
      GO TO 900
C
400   CONTINUE
C
C   COMPUTE (A + B) * FACTOR  STORE IN Y
C
      Y(NPTS,NC)=(A + B) * FACTOR
      GO TO 900
C
500   CONTINUE
C
C   COMPUTE (A - B) * FACTOR  STORE IN Y
C
      Y(NPTS,NC)=(A - B) * FACTOR
      GO TO 900
C
600   CONTINUE
C
C   COMPUTE (A + C1) * C2 + FACTOR
C
      Y(NPTS,NC)=(A + C1) * C2 + FACTOR
C
900   CONTINUE
C
C   STORE TIME IN X
C
```

```
      X(NPTS,1)=TIM + SHIFT
C
      IF(.NOT.DEF) GO TO 2000
C
C   FIND MINS. AND MAXS.
C
      IF(X(NPTS,1).LT.XMIN) XMIN=X(NPTS,1)
      IF(X(NPTS,1).GT.XMAX) XMAX=X(NPTS,1)
      IF(Y(NPTS,NC).LT.YMIN) YMIN=Y(NPTS,NC)
      IF(Y(NPTS,NC).GT.YMAX) YMAX=Y(NPTS,NC)
C
2000  CONTINUE
C
3000  CONTINUE
C
      IF(IANSWR.NE.3HYES) GO TO 3030
C
C   READ MEASURED DATA FROM LOGICAL UNIT 4
C
      READ(4,10020) NMEAS
10020 FORMAT(I5)
C
C
      READ(4,10040) (XM(I),YM(I),I=1,NMEAS)
10040 FORMAT(8E10.0)
C
      IF(.NOT.DEF) GO TO 3030
C
      DO 3025 I=1,NMEAS
C
C   FIND MINS. AND MAXS.
C
      IF(XM(I).LT.XMIN) XMIN=XM(I)
      IF(XM(I).GT.XMAX) XMAX=XM(I)
```

```
      IF(YM(I).LT.YMIN) YMIN=YM(I)
      IF(YM(I).GT.YMAX) YMAX=YM(I)
C
3025  CONTINUE
C
3030  CONTINUE
C
C   HAVE ALL THE DATA NOW,  WRITE TO FILE
C
C
      IF(XMIN.EQ.XMAX) XMAX=XMAX + 1.0E-4
      IF(YMIN.EQ.YMAX) YMAX=YMAX + 1.0E-4
C
C
      WRITE(2,10100) XSIZE,YSIZE,XMIN,XMAX,YMIN,YMAX
10100 FORMAT(8E10.4)
      WRITE(2,10200) NLABL,NCURVE,NGDUMP
10200 FORMAT(3I5)
      IF(NLABL.LE.0) GO TO 3150
      WRITE(2,10300) (XL(N),YL(N),ROT(N),SIZE(N),
     1                (LABL(I,N),I=1,4),N=1,NLABL)
10300 FORMAT(4F10.3,4A10)
C
3150  CONTINUE
C
C
      DO 3200 NC=1,NCURVE
      WRITE(2,10400) (X(NPTS,1),Y(NPTS,NC),NPTS=1,NGDUMP)
10400 FORMAT(8F10.3)
3200  CONTINUE
C
      WRITE(2,10420) IANSWR
10420 FORMAT(A3)
C
```

```
      IF(IANSWR.NE.3HYES) GO TO 3500
C
C
      WRITE(2,10200) NMEAS
C
      WRITE(2,10400) (XM(I),YM(I),I=1,NMEAS)
C
      CALL CLOSE(4)
C
3500  CONTINUE
      RETURN
      END
*DECK AVGVSL
      SUBROUTINE AVGVSL (A,TIM)
C
C   THIS SUBROUTINE PERFORMS AVERAGING
C   OVER AXIAL SECTIONS FOR THE VESSEL.
C   IT RETURNS A AND TIME.
C
*CALL VSLDAT
C
*CALL VSLSPC
C
*CALL GEN
C
*CALL SPACE
C
      SUM=0.0
      KNT=0
      IF(IA.GT.24.AND.IA.LT.30) GO TO 500
C
C
      DO 300 NS=1,NSEC
C
```

```
      NCH=NCHN(NS)
      IB=IBS(NS)
      IQ=IQS(NS)
C
      DO 200 II=1,NCH
      ICH=NCHS(II)
C
      DO 100 J=IB,IQ
C
      CALL GETVSL (IA,IB,ICH,ICH,J,J,E,E,A,B,TIM,0)
C
      SUM=SUM + A
      KNT=KNT + 1
C
100   CONTINUE
C
200   CONTINUE
C
300   CONTINUE
C
      A=SUM / FLOAT(KNT)
C
      RETURN
C
500   CONTINUE
C
C   COMPUTE AVERAGE FOR ROD VARIABLE
C
      IPT=MAXNDS * (NCHANL * NCVAR + NK * NGVAR) + 7
      IROD=ICHA(1)
      IRM1=IROD - 1
      IF(IRM1.LT.1) GO TO 525
C
      DO 510 N=1,IRM1
```

```
      JJ=ILCM(IPT + 2)
      NN=ILCM(IPT + 3)
      NIN=ILCM(IPT + 4)
C
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
      IPT=IPT + 4 + JJ * (NRVAR + NN + IPLUS)
C
510   CONTINUE
C
525   CONTINUE
C
      JJ=ILCM(IPT + 2)
      NN=ILCM(IPT + 3)
      NIN=ILCM(IPT + 4)
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
C
      IB=0
      IQ=0
C
      DO 550 J=1,JJ
C
C   SEARCH FOR USER REQUESTED ELEVATIONS
C
      IPTX=4 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 2
      XC=ALCM(IPTX) * 12.0
      IF(XC.GE.EA1.AND.IB.EQ.0) IB=J
      IF(XC.LE.EA2) IQ=J
C
550   CONTINUE
C
      DO 600 NPTS=IB,IQ
      IPTX=4 + IPT + (NPTS - 1) * (NN + NRVAR + IPLUS) + 2
```

```
      ELA=ALCM(IPTX) * 12.0
C
      CALL GETVSL (IA,0,ICHA(1),ICHA(1),INA(1),INA(1),ELA,ELA,
     1    A,B,TIM,0)
C
      SUM=SUM + A
      KNT=KNT + 1
C
600   CONTINUE
C
      A=SUM / FLOAT(KNT)
      RETURN
C
      END
*DECK AXPLT
      SUBROUTINE AXPLT
C
C   THIS SUBROUTINE DOES EVERYTHING
C   TO SAVE DATA FOR AXIAL PLOTS.
C
*CALL GEN
C
*CALL VSLSPC
C
*CALL SPACE
C
*CALL PLTNFO
C
*CALL VSLDAT
C
      DO 5000 NRCRDS=1,NGDUMP
C
      IF(NRCRDS.EQ.1.AND.IDONE.EQ.0) CALL VESSL1
C
```

```
C   LOC IS THE LOCATION OF THE VESSEL COMPONENT
C   IN THE INDCMP ARRAY -- DEFINED BY "NORMAL" OR "BYPASS"
C
      NREC=INDCMP(LOC,NRCRDS)
      IF(NREC.LE.0) GO TO 5000
C
C   IVSSL MUST BE ONE (1)
C
C   NREC IS THE LOCATION OF VESSEL RECORD
C   FOR THIS DUMP
C
      CALL RDABSF(11,ALCM(1),1,NREC)
      IF(UNIT(11)) 125,100,100
C
100   CONTINUE
C
      WRITE(6,10010)
10010 FORMAT(//"***** ERROR ON TAPE11 ***** - ABORTING"//)
      STOP
C
125   CONTINUE
C
      NWDS=ILCM(1)
C
C   NWDS IS THE NO. OF WORDS ON THIS RECORD
C
      CALL RDABSF(11,ALCM(1),NWDS,NREC)
      IF(UNIT(11)) 175,150,150
C
150   CONTINUE
C
      WRITE(6,10010)
      STOP
C
```

```
175   CONTINUE
C
C   OK, HAVE READ THE RECORD
C
      TIM=ALCM(3)
      DT=ALCM(7)
C
      IF(ABS(TIM - PTIME).GT.DT) GO TO 5000
C
C   HAVE FOUND PROPER TIME -- WITHIN TIME STEP
C
500   CONTINUE
      IF(IA.LT.25.OR.IA.GT.29) GO TO 550
C
C   SET UP IBEGA,ISTPA FOR ROD VARIABLES
C
      IPT=MAXNDS * (NCHANL * NCVAR + NK * NGVAR) + 7
      IROD=ICHA(1)
      IRM1=IROD - 1
      IF(IRM1.LE.0) GO TO 510
C
      DO 505 N=1,IRM1
      JJ=ILCM(IPT + 2)
      NN=ILCM(IPT + 3)
      NIN=ILCM(IPT + 4)
C
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
      IPT=IPT + 4 + JJ * (NRVAR + NN + IPLUS)
505   CONTINUE
C
510   CONTINUE
C
      JJ=ILCM(IPT + 2)
```

```
      NN=ILCM(IPT + 3)
      NIN=ILCM(IPT + 4)
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
C
      IBEGA=0
      ISTPA=0
C
      DO 525 J=1,JJ
C
C   SEARCH FOR STARTING AND ENDING NODE NUMBERS
C   THAT CORRESPOND TO THE USERS REQUESTED ELEVATIONS
C
C
      IPTX=4 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 2
      IF((ALCM(IPTX) * 12.0).GE.EA1.AND.IBEGA.EQ.0) IBEGA=J
      IF((ALCM(IPTX) * 12.0) .LE.EA2) ISTPA=J
C
C
525   CONTINUE
C
      GO TO 575
C
550   CONTINUE
C
      IBEGA=IA1
      ISTPA=IA2
C
575   CONTINUE
C
C
C   CALL GETVSL TO RETURN A,B AND AXIAL NODE COORDINATES
C
      DO 3000 NC=1,NCURVE
```

```
C
C
      DO 2000 NPTS=IBEGA,ISTPA
C
      IPTX=4 + IPT + (NPTS - 1) * (NN + NRVAR + IPLUS) + 2
      ELA=ALCM(IPTX) * 12.0
      ELB=ELA
C
C
      IF((IA.GT.14.AND.IA.LT.30).OR.(IA.GT.32)) CALL GETVSL
     1    (IA,IB,ICHA(NC),ICHB(NC),INA(NC),INB(NC),
     2     ELA,ELB,A,B,XXX,1)
C
      IF((IA.LT.25).OR.(IA.GT.29.AND.IA.LT.33)) CALL GETVSL
     1    (IA,IB,ICHA(NC),ICHB(NC),NPTS,NPTS,ELA,ELB,A,B,XXX,1)
C
C
      GO TO (600,700,800,900), ICHOIC
C
600   CONTINUE
C
C   COMPUTE (A * FACTOR), STORE IN X
C
      X(NPTS,NC)=A * FACTOR
      GO TO 1200
C
700   CONTINUE
C
C   COMPUTE (A + B) * FACTOR
C
      X(NPTS,NC)=(A + B) * FACTOR
      GO TO 1200
C
800   CONTINUE
```

```
C
C   COMPUTE (A - B) * FACTOR
C
      X(NPTS,NC)=(A - B) * FACTOR
      GO TO 1200
C
900   CONTINUE
C
C   COMPUTE (A + C1) * C2 + FACTOR
C
      X(NPTS,NC)=(A + C1) * C2 + FACTOR
C
1200  CONTINUE
C
C   STORE AXIAL LOCATION IN Y
C
      Y(NPTS,NC)=XXX
C
      IF(.NOT.DEF) GO TO 2000
C
      IF(X(NPTS,NC).LT.XMIN) XMIN=X(NPTS,NC)
      IF(X(NPTS,NC).GT.XMAX) XMAX=X(NPTS,NC)
      IF(Y(NPTS,NC).LT.YMIN) YMIN=Y(NPTS,NC)
      IF(Y(NPTS,NC).GT.YMAX) YMAX=Y(NPTS,NC)
C
2000  CONTINUE
C
3000  CONTINUE
C
      IF(IANSWR.NE.3HYES) GO TO 3030
C
C   READ(MEASURED DATA FROM LUN 4
C
      READ(4,10020) NMEAS
```

```
10020 FORMAT(I5)
C
      IF(NMEAS.LE.0) GO TO 3030
C
      READ(4,10040) (XM(I),YM(I),I=1,NMEAS)
10040 FORMAT(8E10.0)
C
      IF(.NOT. DEF) GO TO 3030
C
      DO 3025 I=1,NMEAS
C
C   FIND MINS. AND MAXS.
C
      IF(XM(I).LT.XMIN) XMIN=XM(I)
      IF(XM(I).GT.XMAX) XMAX=XM(I)
      IF(YM(I).LT.YMIN) YMIN=YM(I)
      IF(YM(I).GT.YMAX) YMAX=YM(I)
C
3025  CONTINUE
C
3030  CONTINUE
C
C   HAVE ALL THE DATA NOW,  WRITE TO FILE
C
C
      WRITE(2,10100) XSIZE,YSIZE,XMIN,XMAX,YMIN,YMAX,TIM
10100 FORMAT(8E10.4)
      NPTS=ISTPA - IBEGA + 1
      WRITE(2,10200) NLABL,NCURVE,NPTS
10200 FORMAT(3I5)
      IF(NLABL.LE.0) GO TO 3150
C
      WRITE(2,10300) (XL(N),YL(N),ROT(N),SIZE(N),
     1               (LABL(I,N),I=1,4),N=1,NLABL)
```

```
10300 FORMAT(4F10.3,4A10)
C
3150  CONTINUE
C
      DO 3200 NC=1,NCURVE
      WRITE(2,10400) (Y(NPTS,1),X(NPTS,NC),NPTS=IBEGA,ISTPA)
10400 FORMAT(8F10.3)
3200  CONTINUE
C
      WRITE(2,10420) IANSWR
10420 FORMAT(A3)
C
      IF(IANSWR.NE.3HYES) GO TO 3500
C
      WRITE(2,10200) NMEAS
C
      IF(NMEAS.LE.0) GO TO 3300
C
C
C
      IF(NMEAS.GT.0) WRITE(2,10400) (XM(I),YM(I),I=1,NMEAS)
C
3300  CONTINUE
C
3500  CONTINUE
C
      PTIME=PTIME + PLTINT
C
5000  CONTINUE
C
      RETURN
      END
*DECK RADPLT
      SUBROUTINE RADPLT
```

```
C
C   THIS SUBROUTINE DOES EVERYTINHG
C   TO SAVE DATA FOR RADIAL PLOTS.
C
*CALL GEN
C
*CALL VSLSPC
C
*CALL SPACE
C
*CALL VSLDAT
C
*CALL PLTNFO
C
      DO 5000 NRCRDS=1,NGDUMP
C
      IF(NRCRDS.EQ.1.AND.IDONE.EQ.0) CALL VESSL1
C
C   LOC IS THE LOCATION OF THE VESSEL COMPONENT
C   IN THE INDCMP ARRAY -- DEFINDE BY "NORMAL" OR "BYPASS"
C
      NREC=INDCMP(LOC,NRCRDS)
      IF(NREC.LE.0) GO TO 5000
C
C   IVSSL MUST BE ONE (1) FOR RADIAL PLOTS
C
C   NREC IS THE LOCATION OF THE VESSEL RECORD FOR
C   THIS DUMP.
C
      CALL RDABSF(11,ALCM(1),1,NREC)
C
      IF(UNIT(11)) 125,100,100
C
100   CONTINUE
```

```
C
      WRITE(6,10010)
10010 FORMAT(//"***** ERROR ON TAPE11 ***** - RUN ABORTED"//)
      STOP
C
125   CONTINUE
C
      NWDS=ILCM(1)
C
C   NWDS IS THE NO. OF WORDS ON THIS RECORD
C
      CALL RDABSF(11,ALCM(1),NWDS,NREC)
      IF(UNIT(11)) 175,150,150
C
150   CONTINUE
C
      WRITE(6,10010)
      STOP
C
175   CONTINUE
C
C   OK, HAVE READ THE RECORD
C
      TIM=ALCM(3)
      DT=ALCM(7)
C
      IF(ABS(TIM - PTIME).GT. DT) GO TO 5000
C
      IAA=IA

      IBB=0
C
      DO 3000 NC=1,NCURVE
C
```

```
      NPT=NNODES(ICHA(NC))
C
      DO 2000 NPTS=1,NPT
C
      CALL GETVSL(IAA,IBB,ICHA(NC),ICHA(NC),NPTS,NPTS,ELEVA(NC),
     1            ELEVA(NC),A,B,XXX,2)
C
      GO TO (500,1000) ,ICHOIC
C
500   CONTINUE
C
C   COMPUTE (A * FACTOR)
C
      Y(NPTS,NC)=A * FACTOR
      GO TO 1250
C
1000  CONTINUE
C
C   COMPUTE (A + C1) * C2 + FACTOR
C
      Y(NPTS,NC)=(A + C1) * C2 + FACTOR
C
1250  CONTINUE
C   STORE RADIAL LOCATION IN X
C
      X(NPTS,NC)=XXX
C
      IF(.NOT.DEF) GO TO 2000
C
      IF(X(NPTS,NC).LT.XMIN) XMIN=X(NPTS,NC)
      IF(X(NPTS,NC).GT.XMAX) XMAX=X(NPTS,NC)
      IF(Y(NPTS,NC).LT.YMIN) YMIN=Y(NPTS,NC)
      IF(Y(NPTS,NC).GT.YMAX) YMAX=Y(NPTS,NC)
C
```

```
2000  CONTINUE
C
3000  CONTINUE
C
      IF(IANSWR.NE.3HYES) GO TO 3030
C
C   READ MEASURED DATA FROM LOGICAL UNIT 4
      READ(4,10020) NMEAS
10020 FORMAT(A3)
C
      IF(NMEAS.LE.0) GO TO 3030
C
      READ(4,10040) (XM(I),YM(I),I=1,NMEAS)
10040 FORMAT(8E10.0)
C
      IF(.NOT.DEF) GO TO 3030
C
      DO 3025 I=1,NMEAS
C
C   FIND MINS. AND MAXS
C
      IF(XM(I).LT.XMIN) XMIN=XM(I)
      IF(XM(I).GT.XMAX) XMAX=XM(I)
      IF(YM(I).LT.YMIN) YMIN=YM(I)
      IF(YM(I).GT.YMAX) YMAX=YM(I)
C
3025  CONTINUE
C
3030  CONTINUE
C
C   HAVE ALL THE DATA NOW, WRITE IT TO A FILE
C
C
      WRITE(2,10100) XSIZE,YSIZE,XMIN,XMAX,YMIN,YMAX,TIM
```

```
10100 FORMAT(8E10.4)
      WRITE(2,10200) NLABL,NCURVE,NPT
10200 FORMAT(3I5)
      IF(NLABL.LE.0) GO TO 3150
C
      WRITE(2,10300) (XL(N),YL(N),ROT(N),SIZE(N),
     1               (LABL(I,N),I=1,4),N=1,NLABL)
10300 FORMAT(4F10.3,4A10)
C
3150  CONTINUE
C
      DO 3200 NC=1,NCURVE
      WRITE(2,10400) (X(NPTS,NC),Y(NPTS,NC),NPTS=1,NPT)
10400 FORMAT(8F10.3)
3200  CONTINUE
C
      WRITE(2,10420) IANSWR
10420 FORMAT(A3)
C
      IF(IANSWR.NE.3HYES) GO TO 3500
C
      WRITE(2,10200) NMEAS
C
      IF(NMEAS.LE.0) GO TO 3500
C
C
      IF(NMEAS.GT.0) WRITE(2,10400) (XM(I),YM(I),I=1,NMEAS)
C
3500  CONTINUE
C
C
      PTIME=PTIME + PLTINT
C
5000  CONTINUE
```

```
C
      RETURN
      END
*DECK VESSL1
      SUBROUTINE VESSL1
C
C   THIS SUBROUTINE READS THE FIRST
C   RECORD OF VESSEL INFORMATION (STUFF THAT DOES NOT
C   CHANGE WITH TIME)
C
C
*CALL SPACE
C
*CALL GEN
C
*CALL VSLDAT
C
      CALL RDABSF(11,ALCM(1),1,15 + NCGRAF +
     1    NCGRAF * MXGDMP + MXGDMP + 1)
      IF(UNIT(11)) 150,125,125
C
125   CONTINUE
C
      WRITE(6,10010)
10010 FORMAT(//5X,"ERROR ON DIRECT ACCESS FILE --",
     1    " RUN ABORTED IN SUBROUTINE VESSL1"//)
C
      STOP
C
150   CONTINUE
C
      IWDS=ILCM(1)
      CALL RDABSF(11,ALCM(1),IWDS,15 + NCGRAF
     1    + MXGDMP * NCGRAF + MXGDMP + 1)
```

```
      IF(UNIT(11)) 200,175,175
C
175   CONTINUE
C
      WRITE(6,10010)
      STOP
C
200   CONTINUE
C
      NAX=ILCM(2)
      NCHANL=ILCM(3)
      NK=ILCM(4)
      NFUELT=ILCM(5)
      NROD=ILCM(6)
      NSROD=ILCM(7)
      NSECTS=ILCM(8)
      MAXNDS=ILCM(9)
      MAXC=ILCM(10)
      IPTR=11
C
      IF(NK.LE.0) GO TO 372
      DO 370 K=1,NK
      IK(K)=ILCM(IPTR)
      IPTR=IPTR + 1
      JK(K)=ILCM(IPTR)
      IPTR=IPTR + 1
      IDGAP(K)=ILCM(IPTR)
      IPTR=IPTR + 1
370   CONTINUE
C
372   CONTINUE
C
      DO 400 I=1,NSECTS
      DO 375 J=1,MAXC
```

```
      NCSEC(J,I)=ILCM(IPTR)
      IPTR=IPTR + 1
375   CONTINUE
      DO 380 J=1,6
      ISECTS(I,J)=ILCM(IPTR)
      IPTR=IPTR + 1
380   CONTINUE
C
      DO 390 J=1,NCHANL
      KCHANA(J,I)=ILCM(IPTR)
      IPTR=IPTR + 1
      KCHANB(J,I)=ILCM(IPTR)
      IPTR=IPTR + 1
390   CONTINUE
      DXS(I)=ALCM(IPTR)
      IPTR=IPTR + 1
400   CONTINUE
C
      DO 410 I=1,NCHANL
      LCHAN(I)=ILCM(IPTR)
      IPTR=IPTR + 1
      IDCHAN(I)=ILCM(IPTR)
      IPTR=IPTR + 1
410   CONTINUE
C
      IF(NFUELT.LE.0) GO TO 440
C
      DO 430 N=1,NFUELT
      FTYPE(N)=ILCM(IPTR)
      IPTR=IPTR + 1
      NNODES(N)=ILCM(IPTR)
      IPTR=IPTR + 1
      NN=NNODES(N)
C
```

```
      DO 420 NNN=1,NN
      RAD(NNN,N)=ALCM(IPTR)
      IPTR=IPTR + 1
420   CONTINUE
C
430   CONTINUE
C
440   CONTINUE
C
      IF(NROD.LE.0) GO TO 455
C
      DO 450 N=1,NROD
      NRODIN(N)=ILCM(IPTR)
      IPTR=IPTR + 1
450   CONTINUE
C
455   CONTINUE
C
      IF(NSROD.LE.0) GO TO 470
C
      DO 460 N=1,NSROD
      NSTR(N)=ILCM(IPTR)
      IPTR=IPTR + 1
460   CONTINUE
C
470   CONTINUE
C
      DO 500 J=1,MAXNDS
      DO 480 I=1,NCHANL
      AMOM(I,J)=ALCM(IPTR)
      IPTR=IPTR + 1
480   CONTINUE
C
      IF(NK.LE.0) GO TO 495
```

```
      DO 490 K=1,NK
      GAP(K,J)=ALCM(IPTR)
      IPTR=IPTR + 1
490   CONTINUE
C
495   CONTINUE
C
500   CONTINUE
C
      DO 510 J=1,NAX
      XX(J)=ALCM(IPTR)
      IPTR=IPTR + 1
510   CONTINUE
C
      IDONE=1
C
      RETURN
C
      END
*DECK GETVSL
      SUBROUTINE GETVSL (IA,IB,ICHNA,ICHNB,INODA,INODB,
     1                   ELVA,ELVB,A,B,OTHER,IOPT)
C
C    THIS SUBROUTINE ASSIGNS VALUES TO
C    'A' AND 'B'.  IT ALSO RETURNS 'TIME' OR AXIAL
C    DEPENDING UPON THE TYPE PLOT
C
C    IF THIS IS A SHORT VESSEL DUMP, THIS FAIRLY EASY
C
C    IF THIS IS A NORMAL VESSEL DUMP, IT CAN BE A BIT
C    TRICKIER.
C
C      BASICALLY, THE DATA IS STORED IN A CHANNEL GROUP,
C      A GAP LOOP AND A ROD LOOP
```

```
C
C     FOR ALL OF THESE A VALUES IS STORED FOR EACH
C     (CHANNEL, GAP OR ROD) AND EACH AXIAL NODE
C
C     FOR ROD VARIABLES THE AXIAL NODING IS VARIABLE
C
C     SO, YOU CAN SEE THAT FINDING ANY PARTICULAR
C     VARIABLE IS A BIT TEDIOUS
C
C   THE STORAGE ARRANGEMENT IS DESCRIBED IN THE
C   PROGRAMMERS MANUAL OF THE COBRA/TRAC DOCUMENTATION.
C   THE USER IS ALSO URGED TO LOOK AT SUBROUTINE
C   GRAPH OF COBRA/TRAC.
C
*CALL SPACE
C
*CALL VSLDAT
C
*CALL GEN
C
      IF(IGRFOP.GT.0) GO TO 3000
C
C   NORMAL VESSEL DUMP
C
C     DECIDE IF THE VARIABLE REQUESTED IS
C     A CHANNEL, GAP OR ROD VARIABLE
C
      IF(IA.GT.24.AND.IA.LT.30) GO TO 1000
C
      IF(IA.GT.29.AND.IA.LT.32) GO TO 2000
      IF(IA.GE.32) GO TO 2250
C
C
C   CHANNEL VARIABLE
```

```
C
      IF(IOPT.NE.1) OTHER=ALCM(3)
C
C   COMPUTE 'IPT' = NO. OF WORDS BEFORE THIS CHANNELS VARIABLES
C
      IPTA=8 + (INODA - 1) * (NCHANL * NCVAR  + NK * NGVAR) +
     1     (ICHNA - 1) * NCVAR
C
      IPTB=8 + (INODB - 1) * (NCHANL * NCVAR  + NK * NGVAR) +
     1     (ICHNB - 1) * NCVAR
C
      IF(IOPT.EQ.1) OTHER=XX(INODA)
C
      GO TO (210,220,230,240,250,260,
     1       270,280,290,300,310,320,
     2       330,340,350,360,370,380,
     3       390,400,410,420,430,440), IA
C
210   CONTINUE
C
C   LIQUID FRACTION
C
      A=ALCM(IPTA + 4)
      IF(IB.GT.0) B=ALCM(IPTB + 4)
      RETURN
C
220   CONTINUE
C
C   VAPOR FRACTION
C
      A=ALCM(IPTA + 2)
      IF(IB.GT.0) B=ALCM(IPTB + 2)
      RETURN
C
```

```
230   CONTINUE
C
C   ENTRAINED FRACTION
C
      A=ALCM(IPTA + 3)
      IF(IB.GT.0) B=ALCM(IPTB + 3)
      RETURN
C
240   CONTINUE
C
C   LIQUID FLOW (CONTINUITY)
C
      IPTAP=IPTA + NCHANL * NCVAR + NK * NGVAR
      A=ALCM(IPTA + 9) * ALCM(IPTA + 4) * ALCM(IPTA + 7) /
     1    (0.5 * (ALCM(IPTA + 4) * ALCM(IPTA + 7) + ALCM(IPTAP + 4)
     2    * ALCM(IPTAP + 7)))
C
      IF(IB.GT.0) IPTBP=IPTB * NCHANL * NCVAR + NK * NGVAR
      IF(IB.GT.0) B=ALCM(IPTB + 9) * ALCM(IPTB + 4) * ALCM(IPTB + 7) /
     1    (0.5 * (ALCM(IPTB + 4) * ALCM(IPTB + 7) + ALCM(IPTBP + 4)
     2    * ALCM(IPTBP + 7)))
C
      RETURN
C
250   CONTINUE
C
C   VAPOR FLOW (CONTINUITY)
C
      IPTAP=IPTA + NCHANL * NCVAR + NK * NGVAR
C
      A=ALCM(IPTA + 10) * ALCM(IPTA + 2) * ALCM(IPTA + 8) /
     1    (0.5 * (ALCM(IPTA + 2) * ALCM(IPTA + 8) + ALCM(IPTAP + 2)
     2    * ALCM(IPTAP + 8)))
C
```

```
      IF(IB.GT.0) IPTBP=IPTB + NCHANL * NCVAR + NK * NGVAR
      IF(IB.GT.0) B=ALCM(IPTB + 10) * ALCM(IPTB + 2) * ALCM(IPTB + 8) /
     1    (0.5 * (ALCM(IPTB + 2) * ALCM(IPTB + 8) + ALCM(IPTBP + 2)
     2    * ALCM(IPTBP + 8)))
C
      RETURN
C
260   CONTINUE
C
C   ENTRAINED FLOW (CONTINUITY)
C
      IPTAP=IPTA + NCHANL * NCVAR + NK * NGVAR
C
      A=ALCM(IPTA + 11) * ALCM(IPTA + 3) * ALCM(IPTA + 7) /
     1    (0.5 * (ALCM(IPTA + 3) * ALCM(IPTA + 7) + ALCM(IPTAP + 3)
     2    * ALCM(IPTAP + 7)))
C
      IF(IB.GT.0) IPTBP=IPTB * NCHANL * NCVAR + NK * NGVAR
C
      IF(IB.GT.0) B=ALCM(IPTB + 11) * ALCM(IPTB + 3) * ALCM(IPTB + 7) /
     1    (0.5 * (ALCM(IPTB + 3) * ALCM(IPTB + 7) + ALCM(IPTBP + 3)
     2    * ALCM(IPTBP + 7)))
C
      RETURN
C
270   CONTINUE
C
C   LIQUID FLOW (MOMENTUM)
C
      A=ALCM(IPTA + 9)
      IF(IB.GT.0) B=ALCM(IPTB + 9)
      RETURN
C
280   CONTINUE
```

```
C
C   VAPOR FLOW (MOMENTUM)
C
      A=ALCM(IPTA + 10)
      IF(IB.GT.0) B=ALCM(IPTB + 10)
      RETURN
C
290   CONTINUE
C
C   ENTRAINED FLOW (MOMENTUM)
C
      A=ALCM(IPTA + 11)
      IF(IB.GT.0) B=ALCM(IPTB + 11)
      RETURN
C
300   CONTINUE
C
C   PRESSURE
C
      A=(ALCM(IPTA+1) / 144.0) + ALCM(4)
      IF(IB.GT.0) B=(ALCM(IPTB + 1) / 144.0) + ALCM(4)
      RETURN
C
310   CONTINUE
C
C   LIQUID TEMPERATURE
C
      A=ALCM(IPTA + 5)
      IF(IB.GT.0) B=ALCM(IPTB + 5)
      RETURN
C
320   CONTINUE
C
C   VAPOR TEMPERATURE
```

```
C
      A=ALCM(IPTA + 6)
      IF(IB.GT.0) B=ALCM(IPTB + 6)
      RETURN
C
330   CONTINUE
C
C   LIQUID DENSITY
C
      A=ALCM(IPTA + 7)
      IF(IB.GT.0) B=ALCM(IPTB + 7)
      RETURN
C
340   CONTINUE
C
C   VAPOR DENSITY
C
      A=ALCM(IPTA + 8)
      IF(IB.GT.0) B=ALCM(IPTB + 8)
      RETURN
C
350   CONTINUE
C
C   HASHL
C
      A=ALCM(IPTA + 20)
      IF(IB.GT.0) B=ALCM(IPTB + 20)
      RETURN
C
360   CONTINUE
C
C   HASHV
C
      A=ALCM(IPTA + 21)
```

```
      IF(IB.GT.0) B=ALCM(IPTB + 21)
      RETURN
C
370   CONTINUE
C
C   GAMA
C
      A=ALCM(IPTA + 13)
      IF(IB.GT.0) B=ALCM(IPTB + 13)
      RETURN
C
380   CONTINUE
C
C   HASCL
C
      A=ALCM(IPTA + 22)
      IF(IB.GT.0) B=ALCM(IPTB + 22)
      RETURN
C
390   CONTINUE
C
C   HASCV
C
      A=ALCM(IPTA + 23)
      IF(IB.GT.0) B=ALCM(IPTB + 23)
      RETURN
C
400   CONTINUE
C
C   XK
C
      A=ALCM(IPTA + 26)
      IF(IB.GT.0) B=ALCM(IPTB + 26)
      RETURN
```

```
C
410   CONTINUE
C
C   XKGE
C
      A=ALCM(IPTA + 27)
      IF(IB.GT.0) B=ALCM(IPTB + 27)
      RETURN
C
420   CONTINUE
C
C   SDENT
C
      A=ALCM(IPTA + 17)
      IF(IB.GT.0) B=ALCM(IPTB + 17)
      RETURN
C
430   CONTINUE
C
C   SENT
C
      A=ALCM(IPTA + 16)
      IF(IB.GT.0) B=ALCM(IPTB + 16)
      RETURN
C
440   CONTINUE
C
C   ISIJ
C
      A=FLOAT(ILCM(IPTA + 24))
      IF(IB.GT.0) B=FLOAT(ILCM(IPTB + 24))
      RETURN
C
1000  CONTINUE
```

```
C
C   ROD VARIABLES...SEARCH FOR ELEVATION
C
      IF(IOPT.NE.1) OTHER=ALCM(3)
      IPT=MAXNDS * (NCHANL * NCVAR + NK * NGVAR) + 8
C
C   MAXNDS = MAX. NO. OF AXIAL NODES IN ANY CHANNEL
C
C   (THE ABOVE CALCULATION MOVES THE POINTER BEYOND
C   ALL CHANNEL AND GAP VARIBLES)
C
C   NOW, MOVE POINTER TO ROD USER WANTS
C
      IROD=ICHNA
      IRM1=IROD - 1
      IF(IRM1.LE.0) GO TO 1100
C
      DO 1050 N=1,IRM1
      JJ=ILCM(IPT + 1)
      NN=ILCM(IPT + 2)
      NIN=ILCM(IPT + 3)
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
      IPT=IPT + 3 + JJ * (NRVAR + NN + IPLUS)
1050  CONTINUE
C
1100  CONTINUE
C
      JJ=ILCM(IPT + 1)
      NN=ILCM(IPT + 2)
      NIN=ILCM(IPT + 3)
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
C
```

```
      IGOTO=IA - 24
      ELEV=ELVA
C
      DO 1375 J=1,JJ
      IPTX=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 2
      XC=ALCM(IPTX) * 12.0
C
      IF(ABS(XC - ELEV).GT.0.10) GO TO 1375
C
C   FOUND THE ELEVATION
C
1150  CONTINUE
C
      IF(IOPT.EQ.1) OTHER=XC
      GO TO (1200,1210,1220,1230,1240), IGOTO
C
1200  CONTINUE
C
C   HTCL
C
      IPTA=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 5
      A=ALCM(IPTA)
      GO TO 1400
C
1210  CONTINUE
C
C   HTCV
C
      IPTA=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 6
      A=ALCM(IPTA)
      GO TO 1400
C
1220  CONTINUE
C
```

```
C   QROD
C
      IPTA=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 7
      A=ALCM(IPTA)
      GO TO 1400
C
1230  CONTINUE
C
C   MODE
C
      IPTA=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 1
      A=FLOAT(ILCM(IPTA))
      GO TO 1400
C
1240  CONTINUE
C
C   ROD TEMPERTURE
C
      INODE=INODA
      IPTA=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + NRVAR + IPLUS
     1    + INODE
      A=ALCM(IPTA)
      IF(IOPT.EQ.2) OTHER=RAD(INODE,ICHNA) * 12.0
      GO TO 1400
C
1375  CONTINUE
C
C   SEE IF WE NEED B
C
C
1400  CONTINUE
C
      IF(IB.LE.0) RETURN
C
```

```
C   YES...DO SAME THING FOR B
C
      IPT=MAXNDS * (NCHANL * NCVAR + NK * NGVAR) + 8
C
      IROD=ICHNB
      IRM1=IROD - 1
      IF(IRM1.LE.0) GO TO 1500
C
      DO 1450 N=1,IRM1
      JJ=ILCM(IPT + 1)
      NN=ILCM(IPT + 2)
      NIN=ILCM(IPT + 3)
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
      IPT=IPT + 3 + JJ * (NRVAR + NN + IPLUS)
1450  CONTINUE
C
1500  CONTINUE
C
      JJ=ILCM(IPT + 1)
      NN=ILCM(IPT + 2)
      NIN=ILCM(IPT + 3)
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
      IGOTO=IB - 24
      ELEV=ELVB
C
      DO 1675 J=1,JJ
      IPTX=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 2
      XC=ALCM(IPTX) * 12.0
C
      IF(ABS(XC - ELEV).GT.0.10) GO TO 1675
C
      IF(IOPT.EQ.1) OTHER=XC
```

```
C
C   FOUND THE ELEVATION
C
      GO TO (1510,1520,1530,1540,1550), IGOTO
C
1510  CONTINUE
C
C   HTCL
C
      IPTB=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 5
      B=ALCM(IPTB)
      GO TO 1700
C
1520  CONTINUE
C
C   HTCV
C
      IPTB=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 6
      B=ALCM(IPTB)
      GO TO 1700
C
1530  CONTINUE
C
C   QROD
C
      IPTB=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 7
      B=ALCM(IPTB)
      GO TO 1700
C
1540  CONTINUE
C
C   MODE
C
      IPTB=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + 1
```

```
      B=FLOAT(ILCM(IPTB))
      GO TO 1700
C
1550  CONTINUE
C
C   ROD TEMPERATURE
C
      INODE=INODB
      IPTB=3 + IPT + (J - 1) * (NN + NRVAR + IPLUS) + NRVAR + IPLUS
     1    + INODE
      B=ALCM(IPTB)
      IF(IOPT.EQ.2) OTHER=RAD(INODE,ICHNB) * 12.0
      GO TO 1700
C
1675  CONTINUE
C
1700  CONTINUE
C
      RETURN
C
2000  CONTINUE
C
C   GAP VARIABLES
C
      IF(IOPT.NE.1) OTHER=ALCM(3)
      IF(IOPT.EQ.1) OTHER=XX(INODA)
C
C   IPT=FIRST LOCATION AFTER CHANNEL VARIABLES
C
      KA=ICHNA
      JA=INODA
      IIA=IK(KA)
      JJA=JK(KA)
      ISECA=LCHAN(IIA)
```

```
      DXA=DXS(ISECA)
      IAA=IA - 29
C
      IF(IB.LE.0) GO TO 2020
C
      KB=ICHNB
      JB=INODB
      IIB=IK(KB)
      JJB=JK(KB)
      ISECB=LCHAN(IIB)
      DXB=DXS(ISECB)
      IBB=IB - 29
C
2020  CONTINUE
C
C   FIND CROSSFLOWS
C
      IPTA=8 +JA * (NCHANL * NCVAR + NK * NGVAR) + NCHANL * NCVAR +
     1    (KA - 1) * NGVAR + IAA
      IF(IB.GT.0) IPTB= 8 +JB * (NCHANL * NCVAR + NK * NGVAR) +
     1    NCHANL * NCVAR + (KB - 1) * NGVAR + IBB
C
      WA=ALCM(IPTA)
      IF(IB.GT.0) WB=ALCM(IPTB)
C
C
C   FIND VOID FRACTION, DENSITY FOR (II,J) AND (JJ,J)
C
      IPT1A=8 + (JA - 1) * (NCHANL * NCVAR + NK * NGVAR) + (IIA - 1)
     1    * NCVAR + 1 + IAA
      IPT2A=8 + (JA - 1) * (NCHANL * NCVAR + NK * NGVAR) + (IIA - 1)
     1    * NCVAR + 6 + IAA
      IPT3A=8 + (JA - 1) * (NCHANL * NCVAR + NK * NGVAR) + (JJA - 1)
     1    * NCVAR + 1 + IAA
```

```
      IPT4A=8 + (JA - 1) * (NCHANL * NCVAR + NK * NGVAR) + (JJA - 1)
     1    * NCVAR + 6 + IAA
      IF(IAA.EQ.3) IPT2A=IPT2A - 1
      IF(IAA.EQ.3) IPT4A=IPT4A - 1
C
      IF(IB.LE.0) GO TO 2040
      IPT1B=8 + (JB - 1) * (NCHANL * NCVAR + NK * NGVAR) + (IIB - 1)
     1    * NCVAR + 1 + IBB
      IPT2B=8 + (JB - 1) * (NCHANL * NCVAR + NK * NGVAR) + (IIB - 1)
     1    * NCVAR + 6 + IBB
      IPT3B=8 + (JB - 1) * (NCHANL * NCVAR + NK * NGVAR) + (JJB - 1)
     1    * NCVAR + 1 + IBB
      IPT4B=8 + (JB - 1) * (NCHANL * NCVAR + NK * NGVAR) + (JJB - 1)
     6    * NCVAR + 1 + IBB
      IF(IBB.EQ.3) IPT2B=IPT2B - 1
      IF(IBB.EQ.3) IPT2B=IPT2B - 1
C
2040  CONTINUE
C
      A=WA / (GAP(KA,J) * DXA * ((ALCM(IPT1A) * ALCM(IPT2A) +
     1                            ALCM(IPT3A) * ALCM(IPT4A)) * 0.5))
C
C   GET B IF NEEDED
C
      B=WB / (GAP(KB,J) * DXB * ((ALCM(IPT1B) * ALCM(IPT2B) +
     1                            ALCM(IPT3B) * ALCM(IPT4B)) * 0.5))
C
      RETURN
C
C
2250  CONTINUE
C
C   SLAB ROD...
C
```

```
C     SLIDE POINTER PAST CHANNELS, GAPS AND NORMAL RODS
C
      IPT=8 + (MAXNDS * (NCHANL * NCVAR + NK * NGVAR))
C
      IF(NROD.LE.0) GO TO 2270
C
      DO 2260 N=1,NROD
      JJ=ILCM(IPT+1)
      NN=ILCM(IPT+2)
      NIN=ILCM(IPT+3)
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
      IPT=3 + JJ * (NRVAR + NN + IPLUS)
2260  CONTINUE
C
2270  CONTINUE
C
      IF(IOPT.NE.1) OTHER=ALCM(3)
C
C   MOVE POINTER TO SLAB ROD USER WANTS
C
      IROD=ICHNA
      IRM1=IROD - 1
      IF(IRM1.LE.0) GO TO 2290
      INDA=IFIX(ELVA)
      INDB=IFIX(ELVB)
C
      DO 2280 N=1,IRM1
      JJ=ILCM(IPT + 1)
      NN=ILCM(IPT + 2)
      NIN=ILCM(IPT + 3)
      IPLUS=0
      IF(NIN.GT.NN) IPLUS=1
      IPT=IPT + 3 + JJ * (NRVAR + NN + IPLUS)
```

```
2280  CONTINUE
C
2290  CONTINUE
C
      JJ=ILCM(IPT + 1)
      NN=ILCM(IPT + 2)
      NIN=ILCM(IPT + 3)
      IPLUS=0
      IF(IOPT.EQ.1) OTHER=ALCM(3 + IPT * (INDA - 1) *
     1    (NN + NRVAR + IPLUS) + 2)
C
      IGOTO=IA - 32
C

      GO TO (2350,2360,2370,2380,2390,2400), IGOTO
C
2350  CONTINUE
C
C   SLAB ROD HEAT TRNASFER COEF. (LIQUID)
C
      IPTA=3 + IPT + (INDA - 1) * (NN + NRVAR + IPLUS) + 5
      A=ALCM(IPTA)
      IF(IB.GT.0) IPTB=3 + IPT + (INDB - 1) * (NN + NRVAR + IPLUS) + 5
      IF(IB.GT.0) B=ALCM(IPTB)
C
      RETURN
C
2360  CONTINUE
C
C   SLAB ROD HEAT TRANSFER COEFICIENT (VAPOR)
C
      IPTA=3 + IPT + (INDA - 1) * (NN + NRVAR + IPLUS) + 6
      A=ALCM(IPTA)
C
```

```
      IF(IB.GT.0) IPTB=3 + IPT + (INDB - 1) * (NN +NRVAR + IPLUS) + 6
      IF(IB.GT.0) B=ALCM(IPTB)
C
      RETURN
C
2370  CONTINUE
C
C   SLAB ROD HEAT TRANSFER MODE
C
      IPTA=3 + IPT + (INDA - 1) * (NN + NRVAR + IPLUS) + 1
      A=ALCM(IPTA)
      IF(IB.GT.0) IPTB=3 + (INDB - 1) * (NN + NRVAR + IPLUS) + 1
      IF(IB.GT.0) B=ALCM(IPTB)
C
      RETURN
C
2380  CONTINUE
C
C   SLAB ROD LIQUID TEMPERATURE
C
      IPTA=3 + IPT + (INDA - 1) * (NN + NRVAR + IPLUS) + 3
      A=ALCM(IPTA)
      IF(IB.GT.0) IPTB=3 + IPT + (INDB - 1) * (NN + NRVAR + IPLUS) + 3
      IF(IB.GT.0) B=ALCM(IPTB)
C
      RETURN
C
2390  CONTINUE
C
C
C   SLAB ROD VAPOR TEMPERATURE
C
      IPTA=3 + IPT + (INDA - 1) * (NN + NRVAR + IPLUS) + 4
      A=ALCM(IPTA)
```

```
      IF(IB.GT.0) IPTB=3 + IPT + (INDB - 1) * (NN + NRVAR + IPLUS)
     1    + 4
      IF(IB.GT.0) B=ALCM(IPTB)
C
      RETURN
C
C
2400  CONTINUE
C
C   SLAB ROD TEMPERATURE
C
      IPTA=3 + IPT + (INDA - 1) * (NN + NRVAR + IPLUS) + NRVAR
     1    + IPLUS + INODA
C
      A=ALCM(IPTB)
C
      IF(IB.GT.0) IPTB=3 + (INDB - 1) * (NN + NRVAR + IPLUS) + NRVAR
     1    + IPLUS + INODB
      IF(IB.GT.0) B=ALCM(IPTB)
C
      RETURN
C

3000  CONTINUE
C
C   GET DATA FOR SHORT DUMP
C
      OTHER=ALCM(3)
C
      IPTA=8 + ICHNA
      IPTB=8 + ICHNB
C
      A=ALCM(IPTA)
      IF(IB.GT.0) B=ALCM(IPTB)
```

```
C
      RETURN
C
      END
*DECK GET1D
      SUBROUTINE GET1D (IA,IB,NCA,NCB,A,B,TIME)
C
C   THIS SUBROUTINE ASSIGNS VALUES
C   TO A AND B FOR 1-D COMPONENTS
C   IT ALSO RETURNS TIME.
C
C   IA=USER ENTERED CHOICE FOR A
C   IB=USER ENTERED CHOICE FOR B
C
C   NCA,NCB = CELL NUMBERS FOR A AND B, RESPECTIVELY
C
*CALL SPACE
C
      INTEGER TYPE
C
      TYPE=ILCM(2)
C
C   FOR EACH COMPONENT, DATA MAY BE STORED
C   IN A SLIGHTLY DIFFERENT MANNER, SO WE
C   MUST BRANCH ON TYPE.
C
C   ALSO NOTE:  THIS SUBROUTINE IS CURRENTLY
C               CODED TO OBTAIN ONLY A FRACTION
C               OF THE AVAILABLE DATA FOR 1-D
C               COMPONENTS.  TO ADD CAPABILITIES,
C               REFER TO THE GRAPHICS SECTION OF THE
C               COBRA/TRAC PROGRAMMERS MANUAL AND TO
C               COBRA/TRAC SUBROUTINE GRAPH.
C
```

```
C
      IF(TYPE.EQ.10HSTGEN     ) GO TO 200
C
C   ALL COMPONENTS EXCEPT STEAM GENERATORS USE THE FOLLOWING
C
      ISKIP=6
C
C   ISKIP WILL ADJUST THE POINTER TO THE ALCM
C   ARRAY BY THE NUMBER OF MISCELLANEOUS DATA
C   WORDS (TYPE, COMPONENT NUMBER, ETC.)
C   WRITTEN AT THE FRONT OF EACH COMPONENT DATA RECORD.
C
      NTIME=5
C
C   NTIME IS THE LOCATION OF THE VALUE FOR TIME.
C
      GO TO 400
C
200   CONTINUE
C
C   STEAM GENERATOR
C
      ISKIP=7
      NTIME=6
C
400   CONTINUE
C
C   OK, NOW FIND VALUE FOR A (THERE MUST BE SOMETHING FOR A)
C   AND B, IF REQUIRED
C
C   TO DO THIS, BRANCH ON THE VALUE FOR IA SPECIFIED
C   BY THE USER IN EITHER "QUE1DN" OR "QUE1D"
C
      GO TO (410,420,430,440,450,460,
```

```
     1        470,480,490,500,510), IA
C
410   CONTINUE
C
C   VAPOR FRACTION
C
      IMAGIC=3
C
C   IMAGIC IS THE NUMBER INDICATING
C   THE RELATIVE LOCATION IN THE DATA FOR VAPOR
C   FRACTION
C
      GO TO 1000
C
420   CONTINUE
C
C   PRESSURE
C
      IMAGIC=8
      GO TO 1000
C
430   CONTINUE
C
C   LIQUID DENSITY
C
      IMAGIC=9
      GO TO 1000
C
440   CONTINUE
C
C   VAPOR DENSITY
C
      IMAGIC=10
      GO TO 1000
```

```
C
450   CONTINUE
C
C   LIQUID TEMPERATURE
C
      IMAGIC=11
      GO TO 1000
C
460   CONTINUE
C
C   VAPOR TEMPERATURE
C
      IMAGIC=12
      GO TO 1000
C
470   CONTINUE
C
C   LIQUID VELOCITY
C
      IMAGIC=13
      GO TO 1000
C
480   CONTINUE
C
C   VAPOR VELOCITY
C
      IMAGIC=1
      GO TO 1000
C
490   CONTINUE
C
C   MIXTURE VELOCITY
C
      IMAGIC=13
```

```
      GO TO 1000
C
500   CONTINUE
C
C   LIQUID FLOW
C
C     IMAGIC=-----
      GO TO 1000
C
510   CONTINUE
C
C   VAPOR FLOW
C
C     IMAGIC=-----
C
1000  CONTINUE
C
      A=ALCM(ISKIP + (NCA - 1) * 35 + IMAGIC)
      IF(IB.GT.0) B=ALCM(ISKIP + (NCB - 1) * 35 + IMAGIC)
      TIME=ALCM(NTIME)
C
      RETURN
      END
*DECK FREEF
      SUBROUTINE FREEF
C
C   THIS SUBROTINE TRANSLATES
C   FREE FIELD INPUT AND WRITES
C   IT TO LOGICAL UNIT 3
C
      LOGICAL DEC,NUM,ALPHA
      DIMENSION IN(80),ICOM(80)
C
      INTEGER COMMA,DOT,DIG(10),OUT(80),SPACE,ASC(26)
```

```
C
      DATA COMMA/1H,/, DOT/1H./, SPACE/1H /
C
      DATA DIG/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
      DATA ASC/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,
     1         1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,
     2         1HT,1HU,1HV,1HW,1HX,1HY,1HZ/
C
      DO 100 I=1,80
      IN(I)=SPACE
      OUT(I)=SPACE
100   CONTINUE
C
      READ(5,10015) IN
10015 FORMAT(80A1)
C
C
C   PUT A COMMA AT END OF INPUT LINE IF THERE'S
C   NOT ALREADY ONE THERE.
C
      DO 115 I=1,80
      II=80 - I + 1
      IF(IN(II).EQ.SPACE) GO TO 115
C
      IF(IN(II).EQ.COMMA) GO TO 120
      IF(IN(II).EQ.DOT) IN(II + 1)=COMMA
      IF(IN(II).EQ.DOT) GO TO 120
C
      DO 110 J=1,10
      IF(IN(II).NE.DIG(J)) GO TO 110
C
      IN(II + 1)=COMMA
      GO TO 120
C
```

```
110   CONTINUE
C
115   CONTINUE
C
120   CONTINUE
C
      IO=0
C
C   LOOK FOR COMMAS AND REMEMBER
C   WHERE THEY ARE
C
      NCOM=0
      DO 125 I=1,80
      IF(IN(I).NE.COMMA) GO TO 125
      NCOM=NCOM + 1
      ICOM(NCOM)=I
125   CONTINUE
C
      IF(NCOM.LE.0) WRITE(3,10020) (IN(I),I=1,80)
10020 FORMAT(80A1)
      IF(UNIT(3)) 140,130,130
C
130   CONTINUE
C
      STOP
C
140   CONTINUE
C
      IF(NCOM.LE.0) GO TO 500
C
C   PROCESS DATA BETWEEN COMMAS
C
      DO 350 N=1,NCOM
      IF(N.EQ.1) IFRST=1
```

```
      IF(N.GT.1) IFRST=ICOM(N - 1) + 1
C
      ILAST=ICOM(N) - 1
C
      NUM=.FALSE.
      DEC=.FALSE.
      ALPHA=.FALSE.
C
      DO 200 I=IFRST,ILAST
C
      DO 145 II=1,26
      IF(IN(I).NE.ASC(II)) GO TO 145
      ALPHA=.TRUE.
      GO TO 210
C
145   CONTINUE
C
      DO 150 II=1,10
      IF(IN(I).NE.DIG(II)) GO TO 150
      NUM=.TRUE.
      GO TO 175
150   CONTINUE
C
175   CONTINUE
C
      IF(IN(I).EQ.DOT) DEC=.TRUE.
C
200   CONTINUE
C
210   CONTINUE
C
      IF(.NOT.NUM) GO TO 300
C
      IF(.NOT.DEC) GO TO 250
```

```
C
C   THIS IS A REAL VARIABLE,
C   MOVE IT INTO A 10 POSITION FIELD
C
      IOP=IO + 10
      DO 225 I=IFRST,ILAST
      INN=ILAST - I + IFRST
      OUT(IOP)=IN(INN)
      IOP=IOP - 1
225   CONTINUE
C
      IO=IO + 10
      GO TO 350
C
250   CONTINUE
C
C   THIS MUST BE AN INTEGER,
C   MOVE IT INTO A 5 POSITION FIELD
C
      IOP=IO + 5
      DO 275 I=IFRST,ILAST
      INN=ILAST - I + IFRST
      OUT(IOP)=IN(INN)
      IOP=IOP - 1
275   CONTINUE
C
      IO=IO + 5
      GO TO 350
C
300   CONTINUE
C
C   NOT NUMERIC DATA, LEAVE IT AS IT IS.
C
      ILST=ILAST + 1
```

```
      IKN=1
      DO 325 I=IFRST,ILST
      OUT(IO + IKN)=IN(I)
      IKN=IKN + 1
325   CONTINUE
C
      IF(IN(ILST).EQ.COMMA) OUT(IO + IKN - 1)=SPACE
      IO=IO + ILST - IFRST + 1
C
350   CONTINUE
C
C   PROCESS DATA AFTER LAST COMMA
C
      IFRST=ICOM(NCOM) + 1
      DO 375 I=IFRST,80
      IO=IO + 1
      IF(IO.GT.80) GO TO 380
      OUT(IO)=IN(I)
375   CONTINUE
C
380   CONTINUE
C
      WRITE(3,10020) (OUT(I),I=1,80)
C
      IF(UNIT(3)) 400,390,390
C
390   CONTINUE
C
      STOP
C
400   CONTINUE
C
500   CONTINUE
C
```

```
      BACKSPACE 3
C
      RETURN
      END

$
```

C.2 PROGRAM PLOTIT

```
C      PLOTIT.FOR
C
C    THIS PROGRAM PLOTS INFORMATION
C    SAVED FROM COBRA/TRAC SIMULATIONS
C
C    ALL PROCESSING OF GRAPHICS DUMPS
C    IS DONE AT LASL. A FILE CONTAINING
C    ONLY THE DATA TO BE PLOTTED
C    IS SENT FROM LASL TO THE VAX.
C    THIS PROGRAM READS THAT DATA AND
C    PLOTS IT
C
      CHARACTER*40 FNAME,LABL(10),LAB
      CHARACTER*3 ANSWER,YES
      BYTE STRNG(40)
      DIMENSION XPOS(10),YPOS(10),ROT(10),X(1500),Y(1500)
      DIMENSION SIZ(10)
      DIMENSION XM(200),YM(200)
C
      DATA YES/'YES'/
      CALL PLOTS(0.0,0.0,7)
C
      NPL=0
C
100   CONTINUE
C
C    BEGIN HERE
C
      WRITE(6,10010)
10010 FORMAT(' ENTER NAME OF FILE TO BE PLOTTED.'/
     1    1X,'ENTER  Z TO QUIT PLOTTING.'//)
      READ(5,10020,END=2000) FNAME
```

```
C
10020 FORMAT(A40)
C
      OPEN(UNIT=1,NAME=FNAME,TYPE='OLD',FORM='FORMATTED',
     1    READONLY)
C
C   TELL USER TO CHECK PLOTTER OUT.
C
      WRITE(6,10030)
10030 FORMAT(' CHECK OUT THE PLOTTER:'//
     1    5X,'1. MAKE SURE PEN IS AT ORIGIN OF AXES'/
     2    5X,'2. MAKE SURE THERE IS ENOUGH PAPER IN IT.'/
     3    5X,'3. MAKE SURE IT IS TURNED ON.'/
     4    5X,'4. MAKE SURE ROTATE SWITCH IS UP.'/
     5    5X,'5. WHEN YOU ARE READY TYPE <RETURN>.'//)
C
      READ(5,10040) IDUM
10040 FORMAT(3I5)
C
C   READ DATA HERE
C
200   CONTINUE
C
      READ(1,10050,END=1500) XAXIS,YAXIS,XMIN,XMAX,YMIN,YMAX
10050 FORMAT(8E10.4)
      READ(1,10040) NLABL,NCURVE,NPTS
C
      IF(NLABL.LE.0) GO TO 205
      READ(1,10060) (XPOS(N),YPOS(N),ROT(N),SIZ(N),
     1    LABL(N),N=1,NLABL)
C
10060 FORMAT(4F10.3,A40)
C
205   CONTINUE
```

```
C
C   BEGIN PLOTTING
C
C
C   CENTER PLOT ON 8.5 X 11 INCH PAGE (ROTATED)
C
      IF(NPL.EQ.1) GO TO 210
C
      XR=(11.0 / 2.0) - (XAXIS / 2.0)
      YR=(8.65 / 2.0) - (YAXIS / 2.0)
      CALL PLOT(XR,YR,-3)
C
      NPL=1
C
210   CONTINUE
C
      CALL PLOT(0.0,0.0,3)
C
      NTICX=XAXIS
      NTICY=YAXIS
C
      XNUM=XMIN
      XADD=(XMAX - XMIN) / NTICX
      YNUM=YMIN
      YADD=(YMAX - YMIN) / NTICY
C
      XDIF=XMAX - XMIN
      YDIF=YMAX - YMIN
C
      ALOGX=ALOG10(XDIF)
      ALOGY=ALOG10(YDIF)
      IF(ALOGX.LE.0.0) NDIGX=2
      IF(ALOGX.GT.0.0.AND.ALOGX.LE.1.0) NDIGX=2
      IF(ALOGX.GT.1.0.AND.ALOGX.LE.3.0) NDIGX=-1
```

```
      IF(ALOGX.GT.4.0) NDIGX=-999
      IF(ALOGY.LE.0.0) NDIGY=2
      IF(ALOGY.GT.0.0.AND.ALOGY.LE.1.0) NDIGY=2
      IF(ALOGY.GT.1.0.AND.ALOGY.LE.3.0) NDIGY=-1
      IF(ALOGY.GT.4.0) NDIGY=-999
C
      IF(NDIGX.EQ.-999) GO TO 227
C
      CALL NUMCHR(0.0,-0.25,0.14,XNUM,0.0,NDIGX)
      CALL PLOT(0.0,0.0,3)
      XNUM=XNUM + XADD
C
      DO 225 I=1,NTICX
      XX=I
      CALL PLOT(XX,0.0,2)
      CALL PLOT(XX,0.05,2)
      CALL PLOT(XX,0.0,2)
      CALL NUMCHR(XX-0.17,-0.25,0.14,XNUM,0.0,NDIGX)
      CALL PLOT(XX,0.0,3)
      XNUM=XNUM + XADD
225   CONTINUE
C
      GO TO 235
C
227   CONTINUE
C
C   X-AXIS LABELS IN POWER OF 10 FORM
C
      NPOW=0
      IF(XNUM.NE.0.0) NPOW=ALOG10(XNUM)
      AMULT=10.0 ** NPOW
      XN=XNUM / AMULT
      CALL NUMCHR(0.0,-0.25,0.14,XN,0.0,1)
      CALL CHRPLT(0.24,-0.25,1,0.14,0.14,0.,0.,'X 10',4)
```

```
      CALL NUMCHR(0.50,-0.21,0.14,FLOAT(NPOW),0.0,-1)
      CALL PLOT(0.0,0.0,3)
      XNUM=XNUM + XADD
C
      DO 230 I=1,NTICX
      XX=I
      CALL PLOT(XX,0.0,2)
      CALL PLOT(XX,0.05,2)
      CALL PLOT(XX,0.0,2)
      NPOW=0
      IF(XNUM.NE.0.0) NPOW=ALOG10(XNUM)
      AMULT=10.0 ** NPOW
      XN=XNUM / AMULT
      CALL NUMCHR(XX,-0.25,0.14,XN,0.0,1)
      CALL CHRPLT(XX + 0.24,-0.25,1,0.14,0.14,0.,0.,'X 10',4)
      CALL NUMCHR(XX + 0.50,-0.21,0.14,FLOAT(NPOW),0.0,-1)
      XNUM=XNUM + XADD
      CALL PLOT(XX,0.0,3)
230   CONTINUE
C
235   CONTINUE
C
C
      CALL PLOT(0.0,0.0,3)
C
      IF(NDIGY.EQ.-999) GO TO 252
C
C
      CALL NUMCHR(-0.4,-0.05,0.14,YNUM,0.0,NDIGY)
      CALL PLOT(0.0,0.0,3)
C
      YNUM=YNUM + YADD
C
      DO 250 I=1,NTICY
```

```
      YY=I
      CALL PLOT(0.0,YY,2)
      CALL PLOT(0.05,YY,2)
      CALL PLOT(0.0,YY,2)
      CALL NUMCHR(-0.4,YY-0.05,0.14,YNUM,0.0,NDIGY)
      CALL PLOT(0.0,YY,3)
      YNUM=YNUM + YADD
250   CONTINUE
C
      GO TO 260
C
252   CONTINUE
C
C   Y-AXIS LABELS IN POWER OF 10 FORM
C
      NPOW=0
      IF(YNUM.NE.0.0) NPOW=ALOG10(YNUM)
      AMULT=10.0 ** NPOW
      YN=YNUM / AMULT
      CALL NUMCHR(-0.60,-0.05,0.14,YN,0.0,1)
      CALL CHRPLT(-0.36,-0.05,1,0.14,0.14,0.,0.,'X 10',4)
      CALL NUMCHR(-0.10,0.0,0.14,FLOAT(NPOW),0.0,-1)
      YNUM=YNUM + YADD
      CALL PLOT(0.0,0.0,3)
C
      DO 255 I=1,NTICY
      YY=I
      CALL PLOT(0.0,YY,2)
      CALL PLOT(0.05,YY,2)
      CALL PLOT(0.0,YY,2)
      NPOW=0
      IF(YNUM.NE.0.0) NPOW=ALOG10(YNUM)
      AMULT=10.0 ** NPOW
      YN=YNUM / AMULT
```

```
      CALL NUMCHR(-0.60,YY - 0.05,0.14,YN,0.0,1)
      CALL CHRPLT(-0.36,-0.05,1,0.14,0.14,0.,0.,'X 10',4)
      CALL NUMCHR(-0.10,YY,0.14,FLOAT(NPOW),0.0,-1)
      YNUM=YNUM + YADD
      CALL PLOT(0.0,YY,3)
C
255   CONTINUE
C
260   CONTINUE
C
      CALL PLOT(0.0,YAXIS,3)
      CALL PLOT(XAXIS,YAXIS,2)
      CALL PLOT(XAXIS,0.0,2)
C
      IF(YMIN.LT.0.0.AND.YMAX.GT.0.0) GO TO 270
      IF(YMIN.GE.0.0) GO TO 290
C
270   CONTINUE
C
C   FIND WHERE ZERO IS
C
      YFACT=YAXIS / (YMAX - YMIN)
      YZERO=(0.0 - YMIN) * YFACT
      CALL PLOT(0.0,YZERO,3)
      CALL PLOT(0.0,YZERO,2)
      CALL PLOT(XAXIS,YZERO,2)
      CALL PLOT(XAXIS,YZERO,3)
C
290   CONTINUE
C
      IF(TIME.LE.0.0) GO TO 292
C
      CALL CHRPLT(XSIZE + 0.50,-1.0,1,0.07,0.07,0.0,0.0,'TIME IS',
     1    7)
```

```
      CALL NUMCHR(XSIZE + 1.00,-1.0,0.07,TIME,0.0,3)
C
292   CONTINUE
C
C   LABELS
C
      IF(NLABL.LE.0) GO TO 350
C
      DO 300 N=1,NLABL
C
      LAB=LABL(N)
C
      DO 295 I=1,40
      STRNG(I)=ICHAR(LAB(I:I))
295   CONTINUE
C
      CALL CHRPLT(XPOS(N),YPOS(N),1,SIZ(N),SIZ(N),0.,ROT(N),
     1    STRNG,40)
300   CONTINUE
C
C
350   CONTINUE
C
      NSYM=5
      DO 1000 NCRV=1,NCURVE
C
      READ(1,10050) (X(I),Y(I),I=1,NPTS)
C
C   SCALE THE DATA
C
      XFACT=XAXIS / (XMAX - XMIN)
      YFACT=YAXIS / (YMAX - YMIN)
C
      DO 375 I=1,NPTS
```

```
      X(I)=(X(I) - XMIN) * XFACT
      IF(X(I).LT.0.0) X(I)=0.0
      IF(X(I).GT.XAXIS) X(I)=XAXIS
      Y(I)=(Y(I) - YMIN) * YFACT
      IF(Y(I).LT.0.0) Y(I)=0.0
      IF(Y(I).GT.YAXIS) Y(I)=YAXIS
375   CONTINUE
C
C
C   PLOT THE POINTS
C
      CALL PLOT(X(1),Y(1),3)
C
      DO 400 I=1,NPTS
      CALL PLOT(X(I),Y(I),2)
      IF(MOD(I,NSYM).EQ.0)
     1    CALL NUMCHR(X(I),Y(I) - 0.035,0.07,FLOAT(NCRV),0.0,-1)
      IF(MOD(I,NSYM).EQ.0) CALL PLOT(X(I),Y(I),3)
400   CONTINUE
C
      NSYM=NSYM + 1
C
1000  CONTINUE
C
      READ(1,10070) ANSWER
10070 FORMAT(A3)
C
      IF(ANSWER.NE.YES) GO TO 1200
C
C   PLOT MEASURED DATA
C
      READ(1,10040) NPTS
      IF(NPTS.GT.0) READ(1,10050) (XM(I),YM(I),I=1,NPTS)
C
```

```
      IF(NPTS.LE.0) GO TO 1200
      DO 1050 I=1,NPTS
      XM(I)=(XM(I) - XMIN) * XFACT
      IF(XM(I).LT.0.0) XM(I)=0.0
      IF(XM(I).GT.XAXIS) XM(I)=XAXIS
      YM(I)=(YM(I) - YMIN) * YFACT
      IF(YM(I).LT.0.0) YM(I)=0.0
      IF(YM(I).GT.YAXIS) YM(I)=YAXIS
1050  CONTINUE
      DO 1100 I=1,NPTS
      CALL SYMBOL(XM(I),YM(I),0.07,11,0.0,-1)
1100  CONTINUE
C
1200  CONTINUE
C
C   MOVE PEN TO NEW ORIGIN
C
      CALL PLOT(0.0,8.65,-3)
C
      GO TO 200
C
1500  CONTINUE
C
      CLOSE(UNIT=1)
      GO TO 100
C
2000  CONTINUE
C
      CALL PLOTND
      STOP
      END

$
```

APPENDIX D

PROGRAM RESPEC

D.1 PROGRAM RESPEC TO ALTER SIZE OF VESSEL ARRAYS

```
*DECK RESPEC
      PROGRAM RESPEC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1    TAPE3,TAPE4)
C
C   THIS PROGRAM IS A SIMPLIFIED
C   VERSION OF XSPECSET.  IT IS TO
C   BE USED TO REDIMENSION THE COBRA
C   COMMON DECKS WITHIN COBRA+TRAC.
C
C   THE COBRA COMMON DECKS WITHIN
C   COBRA+TRAC DO NOT REQUIRE THE
C   THE COMPLICATED EQUIVALENCING
C   OF XCOBRA SO FAR FEWER RESPEC
C   PARAMETERS ARE REQUIRED.
C
C   LOGICAL UNIT 3 IS FOR UNDIMENSION COMMON DECKS
C
C   LOGICAL UNIT 4 IS FOR REDIMENSIONED COMMON DECKS
C
C   LOGICAL UNIT 5 IS FOR CARD INPUT (REDIMENSIONING
C   INSTRUCTIONS.
C
C   LOGICAL UNIT 6 IS FOR PRINTER OUTPUT (ERROR MESSAGES, ETC.)
C
      INTEGER BUFIN(72),BUFOUT(100)
      INTEGER LCM
C
      DIMENSION IVALU(3,35),ICHK(35),ICHKI(3),NCH(3,35)
      DIMENSION IV(5)
C
      EQUIVALENCE (IVALU(3,1),K1),(IVALU(3,2),K2),(IVALU(3,3),K3),
     1    (IVALU(3,4),K4),(IVALU(3,5),K5),(IVALU(3,6),K6),
```

```
     2    (IVALU(3,7),K7),(IVALU(3,8),K8),(IVALU(3,9),K9),
     3    (IVALU(3,10),KA),(IVALU(3,11),KB),(IVALU(3,12),KC),
     4    (IVALU(3,13),KD),(IVALU(3,14),KE),(IVALU(3,15),KF),
     5    (IVALU(3,16),KG),(IVALU(3,17),KH),(IVALU(3,18),KI),
     6    (IVALU(3,19),KJ),(IVALU(3,20),KK),(IVALU(3,21),KL),
     7    (IVALU(3,22),KM),(IVALU(3,23),KN),(IVALU(3,24),KO),
     8    (IVALU(3,25),KP),(IVALU(3,26),KQ),(IVALU(3,27),KR),
     9    (IVALU(3,28),KS),(IVALU(3,29),KT),(IVALU(3,30),KU),
     +    (IVALU(3,31),KV),(IVALU(3,32),KW),(IVALU(3,33),KX),
     *    (IVALU(3,34),KY),(IVALU(3,35),KZ)
C
      EQUIVALENCE (IVALU(1,1),M1),(IVALU(1,2),M2),(IVALU(1,3),M3),
     1    (IVALU(1,4),M4),(IVALU(1,5),M5),(IVALU(1,6),M6),
     2    (IVALU(1,7),M7),(IVALU(1,8),M8),(IVALU(1,9),M9),
     3    (IVALU(1,10),MA),(IVALU(1,11),MB),(IVALU(1,12),MC),
     4    (IVALU(1,13),MD),(IVALU(1,14),ME),(IVALU(1,15),MF),
     5    (IVALU(1,16),MG),(IVALU(1,17),MH),(IVALU(1,18),MI),
     6    (IVALU(1,19),MJ),(IVALU(1,20),MK),(IVALU(1,21),ML),
     7    (IVALU(1,22),MM),(IVALU(1,23),MN),(IVALU(1,24),MO),
     8    (IVALU(1,25),MP),(IVALU(1,26),MQ),(IVALU(1,27),MR),
     9    (IVALU(1,28),MS),(IVALU(1,29),MT),(IVALU(1,30),MU),
     +    (IVALU(1,31),MV),(IVALU(1,32),MW),(IVALU(1,33),MX),
     *    (IVALU(1,34),MY),(IVALU(1,35),MZ)
C
      EQUIVALENCE (IVALU(2,1),N1),(IVALU(2,2),N2),(IVALU(2,3),N3),
     1    (IVALU(2,4),N4),(IVALU(2,5),N5),(IVALU(2,6),N6),
     2    (IVALU(2,7),N7),(IVALU(2,8),N8),(IVALU(2,9),N9),
     3    (IVALU(2,10),NA),(IVALU(2,11),NB),(IVALU(2,12),NC),
     4    (IVALU(2,13),ND),(IVALU(2,14),NE),(IVALU(2,15),NF),
     5    (IVALU(2,16),NG),(IVALU(2,17),NH),(IVALU(2,18),NI),
     6    (IVALU(2,19),NJ),(IVALU(2,20),NK),(IVALU(2,21),NL),
     7    (IVALU(2,22),NM),(IVALU(2,23),NN),(IVALU(2,24),NO),
     8    (IVALU(2,25),NP),(IVALU(2,26),NQ),(IVALU(2,27),NR),
     9    (IVALU(2,28),NS),(IVALU(2,29),NT),(IVALU(2,30),NU),
```

```
     +    (IVALU(2,31),NV),(IVALU(2,32),NW),(IVALU(2,33),NX),
     *    (IVALU(2,34),NY),(IVALU(2,35),NZ)
C
      DATA ICHK/"1","2","3","4","5","6","7","8","9","A","B",
     $      "C","D","E","F","G","H","I","J","K","L","M","N",
     $       "O","P","Q","R","S","T","U","V","W","X","Y","Z"/
C
      DATA ICHKI/"M","N","K"/
      DATA IBLANK/" "/,ISTAR/"*"/,IPL/"("/,IPR/")"/,ICOMMA/","/,
     $    IEQUAL/"="/
C
      DATA LCM/"LCM"/
C
C    SET UP DEFAULT VALUES
C
C    IVALU(1,X) CONTAINS VALUES FOR PARAMETER NAMES
C    STARTING WITH THE LETER @M@. PARAMETERS ARE
C    STORED AS FOLLOWS
C
C      X=1 THRU X=9, SECOND CHARACTER IS NUMERIC
C      X=10 THRU X=35, SECOND CHARACTER IS ALPHA
C
C      EXAMPLE...VALUE OF INDEX NAME @IC@ IS STORED
C      IN IVALU(1,12)
C
C    IVALU(2,X) CONTAINS VALUES FOR PARAMETER NAMES
C    STARTING WITH THE LETER @N@.
C
      DO 150 I=1,3
      DO 125 J=1,35
      IVALU(I,J)=0
125   CONTINUE
150   CONTINUE
C
```

```
C   SET UP DEFAULTS HERE
C
C
C   GET STARTED...READ OPTION FOR LARGE CORE
C   MEMORY...IF LCMV=@LCM@, SET UP LARGE CORE
C   MEMORY ALLOCATION (FOR CDC7600).
C
      READ(5,10) LCMV
10    FORMAT(A3)
C
C      READ TURBULENCE OPTION FLAG
C
       READ(5,25) IFTURB
   25  FORMAT(I5)
C
C
200   CONTINUE
C
      READ(5,20) IC1,IC2,(IV(N),N=1,5)
20    FORMAT(2A1,1X,5A1)
C
      IF(ABS(EOF(5)).NE.0) GO TO 270
C
      DO 225 I=1,3
      IF(IC1.NE.ICHKI(I)) GO TO 225
      INDX=I
      GO TO 240
225   CONTINUE
C
C   ERROR+++INVALID SPECIFICATION
C
      WRITE(6,55) IC1,IC2,IVAL
   55 FORMAT(1H0,15X,"RESPEC ERROR...INVALID INPUT SPECIFICATION...",
     $    5X,2A1,"=",I4," RUN ABORTED.")
```

```
C
      STOP
C
240   CONTINUE
C
      DO 250 J=1,35
      IF(IC2.NE.ICHK(J)) GO TO 250
      JNDX=J
      GO TO 260
250   CONTINUE
C
C   ERROR++ INVALID SPECIFICATION
C
      WRITE(6,55) IC1,IC2,IVAL
C
      STOP
C
260   CONTINUE
C
C   COMPUTE IVAL
C
      IVAL=0
      NDG=0
      DO 268 N=1,5
      NNN=5 - N + 1
      IF(IV(NNN).EQ.1H ) GO TO 268
      NDG=NDG + 1
      NUM=0
C
      DO 265 NKK=1,9
      IF(IV(NNN).NE.ICHK(NKK)) GO TO 265
      NUM=NKK
      GO TO 266
C
```

```
265   CONTINUE
C
266   CONTINUE
C
      IVAL=IVAL + (10 ** (NDG - 1) * NUM)
C
268   CONTINUE
C
C
C   STORE PARAMETER VALUE
C
      IVALU(INDX,JNDX)=IVAL
      GO TO 200
C
270   CONTINUE
C
C   ALL PARAMETERS HAVE BEEN READ...
C
C   PERFORM ANY NECESSARY CALCULATIONS.
C
      N1=MAX0(NN,NY)
      NC=2 * (N1 + 1)
      N6=N1 + 2
      N7=MAX0(MX,NX) + 2
      KD=KC + 1
      KK=KE + 1
      ND=25 + 5 * NB
      NG=NN + 2
      NH=NX + 2
      NJ=NB + 1
      NK=NB + 5
      NZ=NZ + 1
      M1=MAX0(NX,NN,NY) + 1
      M2=1
```

```
      M3=1
      M4=1
      IF(IFTURB.LE.0) GO TO 280
      M2=MC
      M3=MG
      M4=MX
  280 CONTINUE
       M7=2 * M8
       M9=8 * M8
C
C   SET UP ARRAY WITH NUMBER OF DIGITS FOR EACH IVALU
C
      DO 350 J=1,35
      DO 325 I=1,3
      NCH(I,J)=0
      IF(IVALU(I,J).LE.0) GO TO 325
      NCH(I,J)=ALOG10(FLOAT(IVALU(I,J)) + 1.0E-3) + 1
325   CONTINUE
350   CONTINUE
C
C   READ FIXED SPECS INPUT
C   FROM LUN3
C
500   CONTINUE
C
      DO 510 I=1,100
      BUFOUT(I)=IBLANK
510   CONTINUE
C
      READ(3,30) (BUFIN(I),I=1,72)
30    FORMAT(72A1)
      IF(ABS(EOF(3)).NE.0.0) GO TO 2000
C
      IF(BUFIN(1).EQ.ICHK(21)) GO TO 2000
```

```
C
C   SCAN INPUT BUFFER FOR SPECIAL
C   CHARACTERS @=@,@(@,@)@.
C
      KNTO=1
      KNTI=1
C
525   CONTINUE
C
      IF(BUFIN(KNTI).EQ.IEQUAL) GO TO 600
      IF(BUFIN(KNTI).EQ.IPL) GO TO 700
C
      BUFOUT(KNTO)=BUFIN(KNTI)
      KNTO=KNTO+1
      KNTI=KNTI+1
      IF(KNTI.LT.73) GO TO 525
      KNTI=KNTI+1
      IF(KNTI.LT.73) GO TO 525
      CALL BUFF(LCM,LCMV,BUFOUT,KNTO+1)
      KNTI=1
      KNTO=1
      GO TO 500
C
600   CONTINUE
C
C   HAVE FOUND AN EQUALS SIGN
C
      BUFOUT(KNTO)=BUFIN(KNTI)
      KNTO=KNTO+1
      KNTI=KNTI+1
C
C   SEARCH
C
      IF(BUFIN(KNTI).NE.IBLANK) GO TO 525
```

```
      IC1=BUFIN(KNTI-3)
      IC2=BUFIN(KNTI-2)
      DO 625 J=1,3
      IF(IC1.NE.ICHKI(J)) GO TO 625
      INDX=J
      GO TO 630
625   CONTINUE
C
630   CONTINUE
C
      DO 650 J=1,35
      IF(IC2.NE.ICHK(J)) GO TO 650
      JNDX=J
      GO TO 660
650   CONTINUE
C
660   CONTINUE
C
      NDIG=NCH(INDX,JNDX)
C
      IVL=IVALU(INDX,JNDX)
C
      DO 680 N=1,NDIG
      ICH=IVL / (10 ** (NDIG - N))
      IF(ICH.GT.0) BUFOUT(KNTO)=ICHK(ICH)
      IF(ICH.EQ.0) BUFOUT(KNTO)=1HO
      KNTO=KNTO + 1
      IF(ICH.EQ.0) BUFOUT(KNTO)=1HO
      KNTO=KNTO + 1
      IVL=IVL - (ICH * (10 ** (NDIG - N)))
680   CONTINUE
C
      GO TO 525
C
```

```
700   CONTINUE
C
C   HAVE FOUND LEFT PARENTHESIS
C
      BUFOUT(KNTO)=BUFIN(KNTI)
      KNTO=KNTO+1
      KNTI=KNTI+1
C
C   SEARCH
C
      IC1=BUFIN(KNTI)
      IC2=BUFIN(KNTI+1)
      IF(IC2.NE.ICOMMA) GO TO 702
C
      BUFOUT(KNTO)=IC1
      BUFOUT(KNTO+1)=IC2
      KNTO=KNTO+2
      KNTI=KNTI+2
      IC1=BUFIN(KNTI)
      IC2=BUFIN(KNTI+1)
C
702   CONTINUE

      IF(BUFIN(KNTI+2).EQ.ICOMMA) GO TO 705
      IF(BUFIN(KNTI+2).EQ.IPR) GO TO 705
      GO TO 525
C
705   CONTINUE
C
      DO 710 J=1,3
      IF(IC1.NE.ICHKI(J)) GO TO 710
      INDX=J
      IF(IC1.NE.ICHKI(J)) GO TO 710
      INDX=J
```

```
      GO TO 715
710   CONTINUE
C
      GO TO 800
C
715   CONTINUE
C
      DO 725 J=1,35
      IF(IC2.NE.ICHK(J)) GO TO 725
      JNDX=J
      GO TO 750
C
725   CONTINUE
C
750   CONTINUE
C
      NDIG=NCH(INDX,JNDX)
C
      IVL=IVALU(INDX,JNDX)
C
      DO 770 N=1,NDIG
      ICH=IVL / (10 ** (NDIG - N))
      IF(ICH.GT.0) BUFOUT(KNTO)=ICHK(ICH)
      IF(ICH.EQ.0) BUFOUT(KNTO)=1HO
      KNTO=KNTO + 1
      IVL=IVL - (ICH * (10 ** (NDIG - N)))
770   CONTINUE
C
      KNTI=KNTI+2
C
800   CONTINUE
C
C   NOW LOOK FOR @)@ OR @,@ TO SEE WHAT
C   TO DO NEXT
```

```
C
      IF(BUFIN(KNTI).EQ.IPR) GO TO 525
      IF(BUFIN(KNTI).EQ.ICOMMA) GO TO 700
C
      DO 900 N=1,9
      IF(BUFIN(KNTI).NE.ICHK(N)) GO TO 900
      BUFOUT(KNTO)=BUFIN(KNTI)
      KNTI=KNTI+1
      KNTO=KNTO +1
      GO TO 800
900   CONTINUE
C
      IF(BUFIN(KNTI).NE.1HO) GO TO 525
      BUFOUT(KNTO)=BUFIN(KNTI)
      KNTO=KNTO+1
      KNTI=KNTI+1
      GO TO 800
C
2000  CONTINUE
C
      STOP
      END
*DECK BUFF
      SUBROUTINE BUFF(LCM,LCMV,BUFOUT,KNTO)
C
      INTEGER BUFOUT(100)
      DATA IBLANK/1H /
C
      IF(LCM.NE.LCMV) GO TO 200
      IF(BUFOUT(1).NE.1HC) GO TO 200
      IF(BUFOUT(2).NE.1HL) GO TO 200
C
      DO 100 I=1,4
      BUFOUT(I)=IBLANK
```

```
100   CONTINUE
C
200   CONTINUE
C
      KNTN=KNTO
C
      DO 225 IKNT=1,KNTO
      INDX=KNTO - IKNT + 1
      IF(BUFOUT(INDX).NE.IBLANK) GO TO 230
      KNTN=KNTN - 1
225   CONTINUE
C
230   CONTINUE
C
      KNTO=KNTN
C
      IF(KNTO.LE.72) GO TO 250
C
      WRITE(4,10) (BUFOUT(I),I=1,KNTO)
   10 FORMAT(72A1/5X,"*",72A1)
      WRITE(6,10) (BUFOUT(I),I=1,KNTO)
      GO TO 275
C
250   CONTINUE
C
      WRITE(4,20) (BUFOUT(I),I=1,KNTO)
20    FORMAT(72A1)
      WRITE(6,20) (BUFOUT(I),I=1,KNTO)
C
275   CONTINUE
C
      DO 300 I=1,100
      BUFOUT(I)=IBLANK
300   CONTINUE
```

```
C
      RETURN
      END

$
```

D.2 "UNDIMENSIONED" VESSEL COMMON BLOCKS

```
*COMDECK ABSORD
      COMMON/ABSORD/     IBEG1,ABSCIS(MV,NF),ORDINT(MV,NF),
     1                   NKBND,KBND(2,NL),NIBND,IBOUND(2,MU),
     2                   ISPEC(MU),NHFN(MU),NPFN(MU),NWFN(NL),
     3                   HVALUE(MU),PVALUE(MU),WVALUE(NL),
     4                   NFUNCT,NPTS(MV),IEND1
CLCM  LEVEL 2, IBEG1
*COMDECK ABCDT
      COMMON/ABCDT/      A(M1),B(M1),C(M1),DD(M1),TT(M1)
*COMDECK DROP
      COMMON/DROP/       IBEG27,DN(MC,MX),DNO(MC,MX),SND(MC,MX),
     1                   ETAEN(MC,MX),ETAENP(MC,MX),GMULT(MG),
     $                   ETA(MC,MX),SENT(MC,MX),SDENT(MC,MX),
     $                   ETANR(MG),RBUBBL(MC),
     2                   RDROPP(MC),DXS(NQ),RBUBB(MC),IEND27
CLCM  LEVEL 2, IBEG27
*COMDECK ENTRDAT
      COMMON/ENTRDAT/    IBEG2,SAVE(MC),VSAVE,IEND2
*COMDECK GAPDAT1
      COMMON/GAPDAT1/    IBEG4,ALAT(MC,MY),FACTOR(MG),IGAP(MG,3),
     1                   IGAPA(MG),IGAPB(MG),INODE(MC,MY),JGAP(MG,3),
     2                   KGAPA(MC,MY),KGAPB(MC,MY),RUVAB(MG,3),
     3                   RUVAT(MG,3),NAMGAP(MC),IEND4
CLCM  LEVEL 2, IBEG4
*COMDECK GAPDAT2
      REAL LENGTH
      COMMON/GAPDAT2/    IBEG5,AINTF(MC,MX),GAP(MG,MX),IK(MG),JK(MG),
     1                   LENGTH(MG),NK,IEND5
CLCM  LEVEL 2, IBEG5
*COMDECK GRDDAT
      COMMON/GRDDAT/     IBEG6,CD(MC,MX),GRID,NCD,IEND6
CLCM  LEVEL 2, IBEG6
```

```
*COMDECK HTCS
      COMMON /HTCS/      DHTCDT(NX,NR),DHTSDT(MX,NV),
     1                   FGAMA(NX,NR),FGAMS(MX,NV),
     3                   TLIQ(NX,NR),TLIQS(MX,NV),TVAP(NX,NR),
     4                   TVAPS(MX,NV)
CLCM  LEVEL 2, FGAMA,FGAMS,TLIQ,TLIQS,TVAP,TVAPS,
     1        DHTCDT,DHTSDT
*COMDECK HTDAT
      COMMON /HTGEN/     IBEG7,AXCON(NN,NH),COND(N1,NX),QLIQI(NX),
     1                   QLIQO(NX),RADCON(N6,N7),RHS(NN,NX),
     2                   SPHTS(N1,NX),TOLDJ(NN),TOLDR(NG,NH),
     2                   IEND7
CLCM  LEVEL 2, IBEG7
*COMDECK INJDAT
      COMMON/INJECT/     IBEG8,AINJT(MS),HINJ(MS),HGINJ(MS),
     1                   HLINJ(MS),PINJ(MS),VEINJ(MS),VINJ(MS),
     2                   VLINJ(MS),VVINJ(MS),WINJ(MS),WEINJ(MS),
     3                   WGINJ(MS),WLINJ(MS),IEND8
*COMDECK INTFHT
      COMMON/INTFHT/     IBEG28,HASCL(MC,MX),HASHL(MC,MX),
     1                   HASCV(MC,MX),HASHV(MC,MX),IEND28
CLCM  LEVEL 2, IBEG28
*COMDECK IOVDAT
      COMMON/IOVDAT/ JUN(M9),ILCMPT,MAXLENT,JFLAGT,ITRACT
      DIMENSION AJUN(M9)
      EQUIVALENCE (JUN(1),AJUN(1))
C
*COMDECK IOVDA1
      COMMON/IOVDAT/ JUN(4,M7),LCMPTR,MAXLEN,JFLAG,ITRAC
      DIMENSION AJUN(4,M7)
      EQUIVALENCE (JUN(1),AJUN(1))
C
C
*COMDECK LOOPDAT
```

```
      COMMON/LOOP/         IBEG9,DEDVS(5),FAPS(NS),
     1                     ISRC(NS),ISRF(NS),ISRL(NS),JSN(NS),
     2                     JUNS(NS),MSS(NS),NCSR,PSNEW(NS),
     3                     PSOLD(NS),SCRS(NS),SLC(NS),SLCV(NS),
     4                     SLE(NS),SLEV(NS),SVC(NS),SVCV(NS),
     5                     SVE(NS),SVEV(NS),SVL(NS),VELSL(NS),
     6                     VL(NS),VLFS(NS),VMS(NS),VSIG(NS),
     7                     VV(NS),VVFS(NS),IEND9
*COMDECK MATPRO
      COMMON /MATPRO/      IBEG10,CPF1(NP,MT),NTDP(MT),RCOLD(MT),
     1                     THCF(NP,MT),TPROP(NP,MT),IEND10
*COMDECK MCMX
      COMMON/MCMX/         IBEG11,ACONT(MC,MX),DHYD(MC,MX),
     1                     GAMA(MC,MX),GAMEXP(MC,MX),GAMAO(MC,MX),
     2                     GAMEXPO(MC,MX),
     2                     H(MC,MX),IASPEC(MC,MX),ISIJ(MC,MX),
     3                     P(MC,MX),QCHFF(MC,MX),QLIQ(MC,MX),
     4                     QVAP(MC,MX),TCHFF(MC,MX),
     5                     IEND11
CLCM  LEVEL 2, IBEG11
*COMDECK MOMNTD1
      COMMON/MOMENT/       IBEG12,
     1                     UEBOT(MC),UFEM(MC),UFGM(MC),
     2                     UFLM(MC),UFOLD(3,MG),
     3                     ULBOT(MC),UVBOT(MC),
     4                     VFEM(MC),VFGM(MC),VFLM(MC),
     5                     VWEJ(MC),VWGJ(MC),VWLJ(MC),
     6                     IEND12
CLCM  LEVEL 2, IBEG12
*COMDECK MOMNTD2
      COMMON/MOMEN/        IBEG13,ABOT(MC),AMOM(MC,MX),
     1                     ATOP(MC),IEND13
CLCM  LEVEL 2, IBEG13
*COMDECK MVYDT
```

```
      COMMON /MVYDAT/ MOVIE, TMOVIE,NCGRAF,IGRFOP,
     1                MXGDMP,MAXNDS,NXTLOC,
     2                NGDUMP,NCVAR,NGVAR,NRVAR,N1DVAR
      COMMON /MVYDT2/ NLLR,NCHLL(MZ),JSLL(MB,MZ),
     1                JCELL(MZ),ICLL(MB,MZ)
      COMMON /MVYDT3/ GRFX(7500),TGDMP(MI),INDCMP(MH,MI),
     1                IGRFIT(MH),IGDUM(15),GRFN(M5),IGRF(M5,2)
C
      DIMENSION IGRFX(1)
      EQUIVALENCE (IGRFX(1),GRFX(1))
C
CLCM  LEVEL 2, GRFX,IGRFX
C
*COMDECK OVS3
      COMMON /OVS3/ LPVCT,NVCT,LBDT,LENBDT,LVSIT,AP(M6),
     1              VSI(M8),BD(38,M8)
C
*COMDECK PROPDAT
      COMMON/PROPS/      IBEG14,CPFF(90),CPG(90),HH1(90),
     1                   HH2(90),HH3(90),HH4(90),HH5(90),
     2                   HH6(90),HHF(90),HHG(90),KKF(90),
     3                   KKG(90),PRLL(90),RHOFF(90),
     4                   RHOGG(90),SSIGMA(90),
     5                   UUF(90),UUG(90),IEND14
      COMMON/PROP1/      IBG14,CPF,CPFILM,CPL,DHFDP,DHGDP,
     1                   DVIDH,FRACS,FRACT,H11,H22,H33,
     2                   H44,H55,H66,HF,HFG,HG,IPROP,
     3                   IPROPP1,IPROPS,KF,KFILM,KG,
     4                   KL,NPROP,PATM,PATM12,PATM13,
     5                   PATM2,PATM3,PATM4,PREF,PREFOL,
     6                   PRFILM,PRL,RFILM,RHOF,RHOG,
     7                   SIGMA,TF,THN,UF,UFILM,UG,UL,
     8                   PP(90),TT(90),VFG,
     9                   CPGAS1,CPGAS2,PLOG,TGASP1,TGASP2,
```

```
     $                   IPROPP,IND14
CLCM  LEVEL 2, IBEG14
      REAL KF,KFILM,KG,KKF,KKG,KL
*COMDECK QUEN
      COMMON /QUEN/      IBEG15,ALFA,ALFAJ,ALFA1,ALFA2,ALFL,ALFLJ,
     1                   ALFL1,ALFL2,CHEN1,CHEN2,DDROP,DE,DPDTC,
     2                   FWD,FWG,G,GG,GLIQ,HSPL,HSPV,PL,QCHF,QHN,
     3                   REG,REGU,SCBMOD,SUPF,TCHF,TCHF1,TCHF2,
     4                   TG,TL,TMIN,TMBOIL,TMERGE,TSBOIL,TSPLIT,
     5                   VEJM,VGJM,VLJM,XA,XTTI,COEFP,COEFD,GG1,GG2,
     6                   GLIQ1,GLIQ2,GVAPC,HSPVC,ALFAF,ALFLF,
     7                   FRACTW,SDETB,IEND15
*COMDECK REBAL
      COMMON/REBAL/      RJACL(5,7),AIRSL(5),DEDVSL(5,ME),SCRSL(ME),
     1                   SOURSL(NZ),AABL(NZ,3),DELPL(NZ),AJ(ME),
     2                   BJ(ME),ILVL
*COMDECK RESPAR
      NFDIM=(NF)
      MVDIM=(MV)
      NEDIM=(NE)
      M5DIM=(M5)
      NNDIM=(NN)
      M8DIM=(M8)
      MADIM=(MA)
      MDDIM=(MD)
      MGDIM=(MG)
      MCDIM=(MC)
      MXDIM=(MX)
      NQDIM=(NQ)
      NADIM=(NA)
      MLDIM=(ML)
      MRDIM=(MR)
      NTDIM=(NT)
      NVDIM=(NV)
```

```
      N4DIM=(N4)
      N5DIM=(N5)
      MFDIM=(MF)
      N1DIM=(N1)
      NCDIM=(NC)
      MTDIM=(MT)
      NPDIM=(NP)
      N3DIM=(N3)
      M2DIM=(M2)
      MUDIM=(MU)
      MSDIM=(MS)
      NLDIM=(NL)
      NZDIM=(NZ)
      MYDIM=(MY)
      NSDIM=(NS)
      MHDIM=(MH)
      MIDIM=(MI)
*COMDECK RODTAB
      COMMON /RODS/      IBEG16,AXIALP(N3,NZ),CAREA(N1,MF),
     1                   DAX(NX,MR),DAXMIN(MR),DFUEL(MF),
     2                   DRIA(NC,MF),FTDENS(MF),HEATP(MF),
     3                   HEATPI(MF),HPERIM(NT),HPERIMI(NT),
     3                   DXT(NX,MR),HTCL(NX,NR),HTCLS(MX,NV),
     3                   HTCV(NX,NR),HTCVS(MX,NV),
     4                   IAXP(MR),IFTYP(MR),IMATYP(N1,MF),IMATIX(MF),
     5                   IMATOX(MF),ISTYP(NT),JFLEND(MR),JFLST(MR),
     6                   JFLUID(NZ,MR),JHT(NZ,MR),JHTEND(MR),
     7                   JSTEND(NT),LCS(NV),LR(MC,6),LS(MC,8),
     8                   LT(MC),MODE(NX,NR),MODES(MX,NV),
     9                   NNODES(MF),NRENODE(MR),NROD,NRODIN(MR),
     1                   NSROD,NSTR(NT),POWER,POLYF(MF),QROD(NX,NR),
     2                   QFRACT(N1,MF),RAD(N1,MF),RADB(N6,MF),
     3                   RADIAL(MR),RHOVOL(N1,MF),RMULS(NV),RODQ,
     4                   RMULT(NR),TSTR(NY,MX,NT),TROD(NN,NX,MR),
```

```
     $                     TFAVG(NZ,N8),
     5                     XC(NX,MR),OXIDET,ROX(NZ,MR),IEND16
CLCM  LEVEL 2, IBEG16
*COMDECK S
      DIMENSION S(M1)
*COMDECK SETUPD
      COMMON/SETUPD/       AFACT(ML,MA),AN(MC),DHYDN(MC),FINLET(MC),
     1                     GAPN(MG),NGAPS,
     2                     GIN,HIN,JAXL(ML,MA),
     3                     NAFACT,NAXL(ML),
     5                     PW(MC),IACTAB(MC),IAMTAB(MC),
     6                     IPWTAB(MC),IGATAB(MG)
CLCM  LEVEL2 , AFACT,AN,DHYDN,FINLET,GAPN,NGAPS,GIN,
CLCM 1         HIN,JAXL,NAFACT,NAXL,PW,IACTAB,IAMTAB,
CLCM 2         IPWTAB,IGATAB
*COMDECK SIMSOL
      INTEGER CELLNO
      COMMON/SIMSOL/       IBEG17,IREBAL,AAB(NB,NA),AAC(NE,NI),
     1                     CELLNO(MC,MX),CONS(NA),DELP(NA),
     2                     ERRIN,ICENT,ITIN,KA,KD,MCON(NA,NJ),
     3                     MSIM(NA),NCELL,NCELLS,NCOLS,NGRPS,
     4                     STORJ(NA,ND),IEND17
CLCM  LEVEL 2, IBEG17
      DIMENSION ISTORJ(NA,ND)
      EQUIVALENCE (ISTORJ(1,1),STORJ(1,1))
      COMMON/SIMS1/        AIRS(5),RJAC(5,NK),SOURS(NA),SOURST(NI)
*COMDECK SNKDAT
      COMMON/SINK/         IBEG18,AESINK(MS),ALIQS(MS),ALSINK(MS),
     1                     ASINK(MS),DXSINK(MS),DWESINK(MS),
     2                     DWGSINK(MS),DWLSINK(MS),HSINK(MS),
     3                     HGSINK(MS),HLSINK(MS),PSINK(MS),
     4                     RLSINK(MS),RVSINK(MS),SINKK(MS),
     5                     VESINK(MS),VLSINK(MS),VVSINK(MS),
     6                     WESINK(MS),WESINKO(MS),WGSINK(MS),
```

```
     7                 WGSINKO(MS),WLSINK(MS),WLSINKO(MS),
     8                 WSINK(MS),IEND18
*COMDECK SPLTDAT
      COMMON/SPLIT/    IBEG19,ICHAN(2,6,6),ISECT,ISECTS(NQ,6),
     1                 KCHANA(MC,11),KCHANB(MC,11),LCHAN(MC),
     2                 NSECTS,IDCHAN(MC),IDGAP(MG),NCSEC(MB,NQ),
     3                 ILOCS(8,MC),FWALL(MG),WKR(MG),PRINHS(NT),
     4                 PRINTC(MC),PRINTG(MG),PRINTR(MR),IEND19
CLCM  LEVEL 2, IBEG19
*COMDECK TAX
      COMMON/TAXAVG/ ATCAVG(NZ),ATFS(NZ),ATCIS(NZ)
C
C
      DIMENSION TFAAVG(NN)
C
*COMDECK TURBDAT
      COMMON/TURBDAT/  IBEG31,IFTURB,IVIS(NQ),ITURB(NQ),ZLMXXL(M2),
     1                 ZLMXXV(M2),F11(M2),F22(M2),TPRNOL(M2),
     2                 TPRNOV(M2),IEND31
*COMDECK TURBVAR
      COMMON/TURBVAR/  IBEG32,PTURBL(M2,M4),PTURBV(M2,M4),SYXL(M2),
     1                 SYXV(M2),SXYL(M3),SXYV(M3),SXYPL(M3),SXYPV(M3),
     2                 SYYIIL(M3),SYYIIV(M3),SYYJJL(M3),SYYJJV(M3),
     3                 SXXL(M2),SXXV(M2),SXXPL(M2),SXXPV(M2),SZYL(M3),
     4                 SZYV(M3),D12L(M3,M4),D12V(M3,M4),D32L(M2,M4),
     5                 D32V(M2,M4),QCTL(M2),QCTV(M2),XIL(M2,M4),
     6                 XIV(M2,M4),UZL(M2),UZV(M2),VZL(M3),VZV(M3),
     7                 VZPL(M3),VZPV(M3),IEND32
CLCM  LEVEL2 , IBEG32,PTURBL,PTURBV,SYXL,SYXV,SXYL,SXYV,SXYPL,SXYPV,
CLCM 1         SYYIIL,SYYIIV,SYYJJL,SYYJJV,SXXL,SXXV,SXXPL,
CLCM 2         SXXPV,SZYL,SZYV,D12L,D12V,D32L,D32V,QCTL,
CLCM 3         QCTV,XIL,XIV,UZL,UZV,VZL,VZV,VZPL,VZPV,IEND32
*COMDECK TWOPHAS
      COMMON/TWOPHAS/  IBEG20,AE(MC,MX),AEOLD(MC,MX),AL(MC,MX),
```

```
     1                   ALOLD(MC,MX),ALIQ(MC,MX),FEM(MC,MX),
     2                   FEMN(MC,MX),FGM(MC,MX),FGMN(MC,MX),
     3                   FLM(MC,MX),FLMN(MC,MX),HL(MC,MX),
     4                   HLN(MC,MX),HV(MC,MX),HVN(MC,MX),
     5                   RL(MC,MX),RV(MC,MX),VLN(MC,MX),
     6                   VVN(MC,MX),WEM(MG,MX),WEMO(MG,MX),
     7                   WGM(MG,MX),WGMO(MG,MX),WLM(MG,MX),
     8                   WLMO(MG,MX),IEND20
CLCM  LEVEL 2, IBEG20
*COMDECK VARS
      COMMON/VARS/        IBEG21,AHF(9,3),AHG(9,3),CCX(5,3),CCXX(5,3),
     1                   E1X(6),IEND21
*COMDECK VELDAT1
      COMMON/VEL/        IBEG22,SURF,VECTUV,VISL,VISV,WPERIM,IEND22
*COMDECK VELDAT2
      COMMON/VELS/       IBEG23,UEJ(MC),UEJM(MC),UEJP(MC),ULJ(MC),
     1                   ULJM(MC),ULJP(MC),UVJ(MC),UVJM(MC),
     1                   UVJP(MC),VEJ(MG),VEJP(MG),VLJ(MG),
     3                   VLJP(MG),VVJ(MG),VVJP(MG),IEND23
CLCM  LEVEL 2, IBEG23
*COMDECK VW
      COMMON/VW/         NLMGAP,KGAP1(MD),KGAP2(MD),KGAP3(MD),
     1                   VWGM(MG),VWLM(MG),VWEM(MG)
CLCM  LEVEL 2, NLMGAP
*COMDECK XKVARS
      COMMON/XKVARS/     IBEG24,XK(MC,MX),XKES(MS),
     1                   XKGE(MC,MX),XKL(MG,MX),XKLE(MG,MX),
     2                   XKVLS(MS),XKWLS,XKWLW(MG),XKWLX(MC),
     3                   XKWVS,XKWVW(MG),XKWVX(MC),XQ(MC),
     4                   FIVLB(MC,MX),FIVVB(MC,MX),
     5                   FIHLB(MG,MX),FIHVB(MG,MX),
     6                   XKWEX(MC),XKWEW(MG),
     7                   IEND24
CLCM  LEVEL 2, IBEG24
```

```
*COMDECK XTEMP
      COMMON/XTEMP/      IBEG25,DFEMDP(MC,MX),DFGMDP(MC,MX),
     1                   DFLMDP(MC,MX),DWEMDP(MG,MX),DWGMDP(MG,MX),
     2                   DWLMDP(MG,MX),IEND25
CLCM  LEVEL 2, IBEG25
*COMDECK XTRADAT
      INTEGER PRINTC,PRINTG,PRINTR,PRINHS
      COMMON /EXTRA/     IBEG26,AFLUX,ALEX,ALMAX,ALMIN,AXIAL(N3,NF),
     1                   AXIALT(N4,N5),DATE(2),DT,DX,DX1,DX2,DXI,
     2                   ECOUR,EOLDT,ETIME,FCOUR,FGPFF(NF),
     3                   FDT,FDTM,FQ(NF),GC,GSFRAC(6,MF),
     4                   I2,I3,I8,I9,IERROR,NGPFF,
     5                   IFOUTH,IOPT,IRSTRT,
     6                   IRTAB(N4,MR),ISTAB(N4,NT),ITERAT,J7,JNOFLO,
     7                   K11,KASE,KNOFLO,MATR(N1),MTSAVE,NAAH,NAX,
     8                   NAXN(N3),NAXP,NC,NCHANL,NDT,NDX,NDXP1,NFUELT,
     9                   NMAT,NMSK,NODER(N1),NOUT1,NOUT2,NOUT3,NOUT4,
     1                   NQ,NRAX(N4),NRT(N4),NRTAB,NSKIPX,NST(N4),
     2                   NTINT,NTSTRT,NTRYX,PI,POUT,POWR(NC),RADP(NC),
     3                   QAX,QREG(N1),RI144,TEXT(17),TIME(2),
     4                   TINT,TTIME,TREG(N1),TRINIT(N4,N5),
     5                   X(NZ),XLIQ,XLIQ1,XVAP,XVAP1,XZERR,
     6                   YGPFF(NF),Y(N3,NF),YQ(NF),Z,IEND26
*COMDECK GAPCON
      COMMON /GAPHTC/    IBEG29,AGFACT(NF,MF),AXJ(NF,MF),
     1                   GAPH(NX,MR),GAPTH(NZ,N8),GPCON(NZ,MR),
     1                   CAEXP(N8),CDEXP(NZ,N8),COLDGP(NZ,N8),
     1                   FATHX(N8),FDEXP(NZ,N8),GASMOL(MF),
     2                   HGAP(MR),IGAPC(MR),IDGPC(MF),
     3                   LRC(NZ,MR),PINT(NZ,N8),TGINT,TGLST,
     3                   NGPAX(MF),PGAP(MF),PGAS(MR),
     4                   ROUFC(MF),ROUFF(MF),VPLEN(MF),
     6                   FRDREL(NZ,N8),ICONF,IRELF,
     5                   IEND29
```

```
CLCM  LEVEL 2, IBEG29
*COMDECK ALPHN
      COMMON /ALPHN/      IBEG30,FTYPE(MF),HROD,NUCL,TUBE,WALL,
     1                    IEND30
      INTEGER FTYPE,HROD,NUCL,TUBE,WALL
*COMDECK INTFRD
      COMMON /INTFRD/ FIVL(MC),FIVV(MC),FIHLJ(MG),
     1                FIHLJP(MG),FIHVJ(MG),FIHVJP(MG),
     2                VFACE(MG),KCHEK(MG)

$
```

DISTRIBUTION

No. of Copies

OFFSITE

335 U.S. Nuclear Regulatory Commission
Division of Technical Information and Document Control
7920 Norfolk Avenue
Bethesda, MD 20014

5 James Han
U.S. Nuclear Regulatory Commission
7915 Eastern Avenue
M/S 1130-SS
Silver Spring, MD 20910

No. of Copies

ONSITE

50 Pacific Northwest Laboratory

MJ Thurgood (43)
Publishing Coordination (2)
Technical Information ED (5)

☆ U.S. GOVERNMENT PRINTING OFFICE: 1983-381-297:3012

NRC FORM 335 (11-81) U.S. NUCLEAR REGULATORY COMMISSION **BIBLIOGRAPHIC DATA SHEET**	1. REPORT NUMBER *(Assigned by DDC)* NUREG/CR-3046, Vol. 5 PNL-4385
4. TITLE AND SUBTITLE *(Add Volume No., if appropriate)* COBRA/TRAC - A Thermal Hydraulics Code for Transient Analysis of Nuclear Reactor Vessels and Primary Coolant Systems - Volume 5: Programmers' Manual	2. *(Leave blank)*
	3. RECIPIENT'S ACCESSION NO.
7. AUTHOR(S) A. S. Koontz and J. M. Cuta	5. DATE REPORT COMPLETED MONTH November YEAR 1982
9. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS *(Include Zip Code)* Pacific Northwest Laboratory PO Box 999 Richland, Washington 99352	DATE REPORT ISSUED MONTH March YEAR 1983
	6. *(Leave blank)*
	8. *(Leave blank)*
12. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS *(Include Zip Code)* Division of Accident Evaluation Office of Nuclear Regulatory Research U.S. Nuclear Regulatory Commission Washington, DC 20555	10. PROJECT/TASK/WORK UNIT NO.
	11. FIN NO. FIN B2391
13. TYPE OF REPORT Computer Code Manual	PERIOD COVERED *(Inclusive dates)*
15. SUPPLEMENTARY NOTES	14. *(Leave blank)*

16. ABSTRACT *(200 words or less)*

The COBRA/TRAC computer program has been developed to predict the thermal-hydraulic response of nuclear reactor primary coolant systems to small and large break loss-of-coolant accidents and other anticipated transients. The code solves the compressible three-dimensional, two-fluid, three-field equations for two-phase flow in the reactor vessel. The three fields are the vapor field, the continuous liquid field, and the liquid drop field. A five-equation drift flux model is used to model fluid flow in the primary system piping, pressurizer, pumps, and accumulators. The heat generation rate of the core is specified by input and no reactor kinetics calculations are included in the solution. This volume explains the details of COBRA/TRAC's working parts from the programmer's viewpoint. The code's overlay structure is discussed. The memory management and COMMON block manipulation are explained, as are the restart/dump logic and the graphics logic. Suggestions for code conversion to "non-LANL" CDC computers and non-CDC computers are given.

17. KEY WORDS AND DOCUMENT ANALYSIS

17a. DESCRIPTORS

17b. IDENTIFIERS/OPEN-ENDED TERMS

18. AVAILABILITY STATEMENT	19. SECURITY CLASS *(This report)*	21. NO. OF PAGES
Unlimited	Unclassified	
	20. SECURITY CLASS *(This page)*	22. PRICE
	Unclassified	$

NRC FORM 335 (11-81)