

SANDIA REPORT

SAND92-0694 • UC-405

Unlimited Release

Printed September 1992

SNLA LIBRARY



SAND92-0694
0001
UNCLASSIFIED

09/92
132P

STAC

SAFSIM Input Manual—A Compute Program for the Engineering Simulation of Flow Systems

Dean Dobranich

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A06
Microfiche copy: A01

SAFSIM Input Manual – A Computer Program for the Engineering Simulation of Flow Systems

Dean Dobranich
Nuclear Technology Department
Sandia National Laboratories
Albuquerque, NM 87185

Abstract

SAFSIM (System Analysis Flow SIMulator) is a FORTRAN computer program to simulate the integrated performance of systems involving fluid mechanics, heat transfer, and reactor dynamics. SAFSIM provides sufficient versatility to allow the engineering simulation of almost any system, from a backyard sprinkler system to a clustered nuclear reactor propulsion system. In addition to versatility, speed and robustness are primary goals of SAFSIM. SAFSIM contains three basic physics modules: (1) a one-dimensional finite element fluid mechanics module with multiple flow network capability; (2) a one-dimensional finite element structure heat transfer module with multiple convection and radiation exchange surface capability; and (3) a point reactor dynamics module with reactivity feedback and decay heat capability. SAFSIM can be used for gas (compressible) or liquid (incompressible) single-phase flow systems with primary emphasis on gases (or supercritical fluids). This document contains a description of all the information required to create an input file for SAFSIM execution.

Preface

This document is the second of three documents that describe the SAFSIM computer program. The first document (SAND92-0693) is a theory manual that describes SAFSIM's governing equations and numerical methods, along with their computer implementation. This second document is an input manual that provides the analyst with the information necessary to build a SAFSIM input model. The third document (SAND92-0695) is an application manual that provides various example problems, including benchmark problems. All three documents are currently based on Version 1.0 of SAFSIM. Upon completion of all three documents, the SAFSIM computer program will be released to one of the national software centers.

Table of Contents

	<u>page</u>
Abstract	i
Preface	ii
1.0 Overview of the SAFSIM Computer Program	1
1.1 Introduction	1
1.2 Fluid Mechanics	1
1.3 Structure Heat Transfer	4
1.4 Reactor Dynamics	5
1.5 Function-Controlled Variables	5
1.6 Status and Future Enhancements	6
2.0 Creating a SAFSIM Input Model	7
2.1 Creating a Fluid Mechanics Network Input Model	10
2.2 Creating a Structure Heat Transfer Input Model	12
2.3 Creating a Reactor Dynamics Input Model	13
2.4 Creating Function-Controlled Variables	14
3.0 Input Data Blocks	17
Block 1, Execution Data	19
Block 2, Fluid Equation of State and Property Data	26
Block 3, Fluid Mechanics Finite Element Data	30
Block 4, Fluid Mechanics Node Initialization Data	36
Block 5, Fluid Mechanics Boundary Condition Identification Data	38
Block 6, Fluid Mechanics Boundary Condition Set Identification Data	39
Block 7, Function-Controlled Variable Data for Fluid Mechanics	41
Block 8, Function Data for Fluid Mechanics	43
Block 9, Structure Heat Transfer Material Property Data	45
Block 10, Structure Heat Transfer Finite Element and Node Data	49
Block 11, Function-Controlled Variable Data for Heat Transfer	60
Block 12, Function Data for Heat Transfer	62
Block 13, Reactor Dynamics Data	64
Block 14, Function-Controlled Variable Data for Reactor Dynamics	70
Block 15, Function Data for Reactor Dynamics	72
Block 16, Function-Controlled Variable Data for the System	74
Block 17, Function Data for the System	76
Block 18, User-Defined Data	78

4.0 SAFSIM Output Files	79
4.1 Fluid Mechanics Output	80
4.2 Heat Transfer Output	81
4.3 Reactor Dynamics Output	82
4.4 Function and Function-Controlled Variables Output	83
4.5 Iteration Summary Output	83
Appendices	85
Appendix A: Function-Controlled Variable Identification Numbers	86
Appendix B: Function Identification Numbers	90
Appendix C: Heat Transfer Coefficient Correlation Identification Numbers	98
Appendix D: Sample User-Defined Subroutines	105
Index of Input Variables	118

1.0 Overview of the SAFSIM Computer Program

The following subsections provide a brief overview of SAFSIM's modeling capabilities. The theory manual (SAND92-0693) should be consulted for the details of the governing equations, the numerical methods, and their computer implementation. Also, the application manual (SAND92-0695) provides examples of SAFSIM input files.

1.1 Introduction

SAFSIM (System Analysis Flow SIMulator) is a FORTRAN computer program that provides engineering simulations of user-specified flow networks at the system level. It includes fluid mechanics, heat transfer, and reactor dynamics capabilities. SAFSIM provides sufficient versatility to allow the simulation of almost any flow system, from a backyard sprinkler system to a clustered nuclear reactor propulsion system. In addition to versatility, speed and robustness are primary goals of SAFSIM.

The motivation for SAFSIM is the desire to have a tool to provide quick and inexpensive engineering performance simulations of complicated systems. The simulations are intended to provide a first-look understanding of a systems' transient behavior during operational and off-normal conditions. It is also desired that this tool have the ability to accommodate changes in the problem definition via changes in an input file rather than changes in the computer program. Thus, all geometric and operational information is provided by the analyst, allowing the analyst to select a level of modeling detail consistent with the problem detail available.

SAFSIM currently contains three basic physics modules: (1) fluid mechanics, (2) structure heat transfer, and (3) reactor dynamics. All three modules (along with function-controlled variables) are explicitly coupled to allow the prediction of system performance. The analyst can employ any or all of the physics modules as the problem dictates.

1.2 Fluid Mechanics

The fluid mechanics module is based on a 1-D finite element model that allows the specification of a user-defined, single-phase flow network or multiple networks. Provisions for modeling either open or closed networks are incorporated. The fluid can be either a gas or a liquid; also, mixing models are included that allow the specification of multiple gases. The fluid mechanics module includes the solution of compressible thermal and mechanical energy equations. The mechanical energy equation is a combined momentum and

mass continuity equation. The thermal energy equation is a total energy equation (combined with the mass continuity equation) in which the momentum equation has been subtracted. The dynamic (unsteady-state) temperature term in the thermal energy equation and the dynamic (unsteady-state) mass flow rate term in the momentum equation are currently not included; however, their inclusion is planned for a future version. Also, the mass accumulation (unsteady-state density) term in the mass continuity equation is omitted at this time. SAFSIM is structured for a fully implicit implementation of the dynamic terms.

Super element capability is provided for the mechanical energy equation solution, greatly improving computational efficiency. Two correlations are available for determination of pipe flow friction factors, including effects of wall roughness. Correlations are also available for porous media friction factor calculation. Separate loss coefficients for both forward and reverse flow can be included. Also, loss coefficients for contractions and expansions (both gradual and abrupt) are automatically determined by SAFSIM if desired. The body force due to gravitational acceleration can be included if desired. An option exists to bypass the mechanical energy equation during problem execution. This is useful if the flow and pressure fields are not changing significantly.

Advection, conduction, and convection within the fluid are modeled in the thermal energy equation. Upwind finite elements are employed for the thermal energy equation with automatic determination of the optimum upwind factor based on the Peclet number. Automatic upwind elements based on Mach number are also planned for the mechanical energy equation, allowing the simulation of supersonic flow. Convective heat transfer from any solid structure to the fluid (using Newton's law of cooling) is based on the log-mean temperature difference. This improves accuracy, especially for convective heat transfer at low flow rates. More than one convective surface can be coupled to any fluid mechanics finite element.

Although the mechanical and thermal energy equations are solved separately, they are iteratively coupled to provide the solution to a total energy equation for the fluid. Iteration on the fluid density is used for this coupling. An equation of state provides fluid density as a function of pressure and temperature. An ideal equation of state is used for gases. For liquids, the density is specified as a second-order polynomial function of temperature. Fluid conductivity and specific heat are specified as third-order polynomial functions of temperature. Either a power-law or Sutherland-law equation can be used to specify fluid viscosity. Multiple temperature ranges can be specified for the three fluid properties. (All polynomial and viscosity law constants are provided by the analyst.) Also, an interface with SAFSIM is provided that allows the analyst to easily include his own equation of state and property data as functions of pressure and temperature.

Different time steps can be specified for each fluid mechanics flow network. Pressure or mass flow rate boundary conditions can be applied to any node of a flow network for the mechanical energy equation. Temperature or zero heat flux boundary conditions can be applied to any node for the thermal energy equation. Mass fraction boundary conditions for each gas of a multiple gas network can also be specified for any node.

Four special finite elements are included to increase modeling versatility: (1) a porous media element, (2) a choked flow boundary element, (3) a compressor/pump element, and (4) a distributed flow manifold element. A porous media finite element allows the user to specify an element porosity to simulate porous media. The analyst may select the Ergun, Achenbach, or Beek correlation's for calculation of the porous media friction factor. If none of these correlation's is adequate, the analyst can define his own via the input file. A choked flow boundary element allows the simulation of choked flow based on solution of the compressible, isentropic flow equations. Discharge coefficients can be added to this special finite element to account for any non-isentropic effects. This "analytic" element allows a convenient way to model choked flow in converging nozzles without the need for multiple elements. The compressor/pump finite element requires the input of a pump characteristic curve along with rated and operational speeds. The pump characteristic curve is adjusted for varying operational speeds based on the pump similarity laws. A distributed flow manifold finite element accounts for the fact that flow can enter or exit this element along its entire length and not just at the nodes. It provides a pseudo two-dimensional capability that is useful in modeling plena with inflow/outflow distributed along its length. This special element can also be used, if desired, to more accurately model the momentum transfer in tees and wyes (or other flow splitting devices).

Three equation solvers are available for the mechanical energy equation: (1) Gauss-Seidel (iterative), (2) Cholesky decomposition (direct), (3) and Gauss elimination (direct). A fourth solver (LU decomposition with iterative improvement) is planned for inclusion to handle very large networks in which round-off error accumulation is a problem. The analyst may select any of these solvers based on the specific problem; if a specific solver is not selected, SAFSIM attempts to use Gauss-Seidel iteration first. If convergence is not met, or if the coefficient matrix is not diagonally dominant, Cholesky decomposition is employed. This is a direct solver that operates only on terms within the semibandwidth of the coefficient matrix and is therefore relatively fast. If the coefficient matrix is not positive definite, Cholesky will fail and Gauss elimination is used. This is the slowest of the three solvers but the most general. This solver uses partial pivoting and is written so that only the terms within the bandwidth of the coefficient matrix are included, greatly increasing speed. The Cholesky and Gauss elimination solvers both require numbering of

the network nodes to minimize the bandwidth. The use of multiple numerical solvers adds to the robustness of the program.

Two solvers are available for solution of the thermal energy equation: (1) Gauss-Seidel, and (2) Gauss elimination. For flow problems that are advectively dominated (such as many gas flow problems), Gauss-Seidel provides a rapid solution for the fluid temperature field. Gauss elimination is included for flow problems that are not advectively dominated because the coefficient matrix may lose its diagonal dominance. Cholesky decomposition is not included for the thermal energy solution because the coefficient matrix is not symmetric due to the advective term in the equation.

1.3 Structure Heat Transfer

The structure heat transfer physics module is based on a 1-D finite element model that allows the user to model conduction in any heat transfer structure such as pipe walls, fuel elements or particles, vessel walls, thermocouples, etc. The user can include as many structures as desired. Multiple exchange surfaces allow the user to convectively or radiatively couple any heat transfer structure finite element to any fluid mechanics finite element(s). The module can be run in static or dynamic mode. In dynamic mode, SAFSIM automatically determines the optimum implicitness factor for each node of each structure at each time step. Time step size can be automatically calculated for each structure by the program or user specified.

Material properties can be temperature dependent if desired. Also, several options are available for supplying the property data, including tables, polynomials, power laws, and constants. An extensive built-in heat transfer coefficient correlation library includes correlation's for laminar and turbulent flows, for internal and external flow geometries, and for gases and liquids (including liquid metals). Automatic finite element generation for spherical, cylindrical, conical, and rectangular geometries is available. Temperature, heat flux, and convective/radiative boundary conditions can be specified. Also, finite elements from different heat transfer structures can be conductively (including a contact resistance) or radiatively (including view factors) coupled. This coupling, however, is explicit. A tridiagonal numerical solver provides a rapid solution for the node temperatures of all of the structures.

SAFSIM contains an automatic steady-state option that allows a consistent solution between fluid mechanics and heat transfer physics modules. A solution process is used in which the steady-state fluid mechanics solution is determined for all flow networks based on initial guesses for all structure wall temperatures. Then, the steady-state temperature distributions in all structures are calculated and the wall temperatures updated. Iteration between the steady-state fluid mechanics and heat transfer solutions continues until the

wall temperatures converge within user-specified convergence criteria. Steady-state solutions can also be found by performing a transient solution until all the time-derivative terms approach zero.

1.4 Reactor Dynamics

The reactor dynamics physics module is based on a point (0-D) kinetics model with feedback. Multiple reactors can be specified, and multiple feedback coefficients are allowed for each reactor to account for all system interactions. The analyst has complete control over how the feedback coefficients are defined. Also, several "control laws" are available to simulate control rod/drum reactivity control for reactor startup and shutdown simulations. Special-purpose control laws can be added to the program by the analyst if desired. The analyst can specify any number of delayed neutron groups and any number of decay heat groups. Initial precursor concentrations can be input or calculated automatically by SAFSIM based on steady-state conditions at the specified power level. A source term also can be included.

Two solvers are currently available for integration of the reactor dynamics equations: (1) Euler, and (2) Runge-Kutta-Fehlberg (RKF). Adaptive time stepping is employed by both solvers based on the desired relative truncation error. The Euler integrator employs step doubling to provide the truncation error estimate for time step selection. The RKF integrator determines an error estimate as the difference between a fourth- and fifth-order prediction. The analyst can switch between solvers during a problem if desired.

1.5 Function-Controlled Variables

In addition to the three physics modules, SAFSIM contains a unique input feature: function-controlled variables. This feature allows the analyst to specify many of the input variables as functions of the output variables. Examples of input variables include: convergence criteria, flow areas, conduction lengths, feedback coefficients, compressor speed, and boundary conditions. Examples of output variables include: pressure, temperature, velocity, and heat transfer coefficient. For example, the diameter of a fluid mechanics finite element can be specified as a function of a heat transfer finite element temperature to simulate the effect of thermal expansion on the flow field. Functions can be specified as functions of other functions to allow complex logic paths to be included via input. A library of functions is included along with a provision for the analyst to easily add his own. Examples of mathematical functions in the library include: cosine, square root, averaging, table lookup, polynomials, and conditional statements.

1.6 Status and Future Enhancements

SAFSIM is a functional computer program that runs on a personal computer and provides the analyst with a tool for rapidly obtaining engineering solutions to complex system analysis problems. Benchmarking and documentation are in progress. Also, additional enhancements are envisioned to make SAFSIM even more versatile, robust, and fast.

To expand the class of problems for which SAFSIM is applicable, several enhancements are envisioned. These enhancements are provided in the following non-prioritized list:

- (1) LU decomposition solver with iterative refinement for very large flow networks
- (2) Built-in bandwidth minimizer for the mechanical and thermal energy equations
- (3) Upwind elements for the mechanical energy equation
- (4) Blowdown tank option
- (5) Turbine finite element
- (6) Dynamic temperature term in the thermal energy equation
- (7) Dynamic mass flow rate term in the momentum equation
- (8) Dynamic density term in the mass continuity equation (mass accumulation)
- (9) Kagonove solver for reactor dynamics equations
- (10) Structural mechanics physics module with plasticity and creep (1-D)
- (11) Restart capability
- (12) 2-D tables and other special functions
- (13) Enhanced user-supplied subroutine interfaces
- (14) Pre- and post-processing capability (graphical interface)
- (15) Electromagnetic pump finite element for liquid metals
- (16) Accumulator model for liquid systems

2.0 Creating a SAFSIM Input Model

The theory manual (SAND92-0693), a companion document to this input manual, provides a complete description of SAFSIM's governing equations and numerical methods. The theoretical manual serves as an engineering system-modeling primer and provides essential information regarding the applicability of SAFSIM. Also, to make appropriate use of SAFSIM, a basic understanding of the following subjects would be useful: fluid mechanics, heat transfer, reactor dynamics, thermodynamics, finite element methods, and numerical methods.

As mentioned in Section 1.0, SAFSIM consists of three basic physics modules: (1) Fluid Mechanics, (2) Heat Transfer, and (3) Reactor Dynamics. The analyst can employ any or all of the modules as the problem dictates. The number of fluid mechanics flow networks, heat transfer structures, and nuclear reactors used in the model is entirely the decision of the analyst, within array dimensioning constraints. Thus, the analyst can build a system input model with a level of complexity consistent with the level of available problem definition. As more definition becomes available, or as the understanding of the system performance grows, the input model can be extended and modified accordingly.

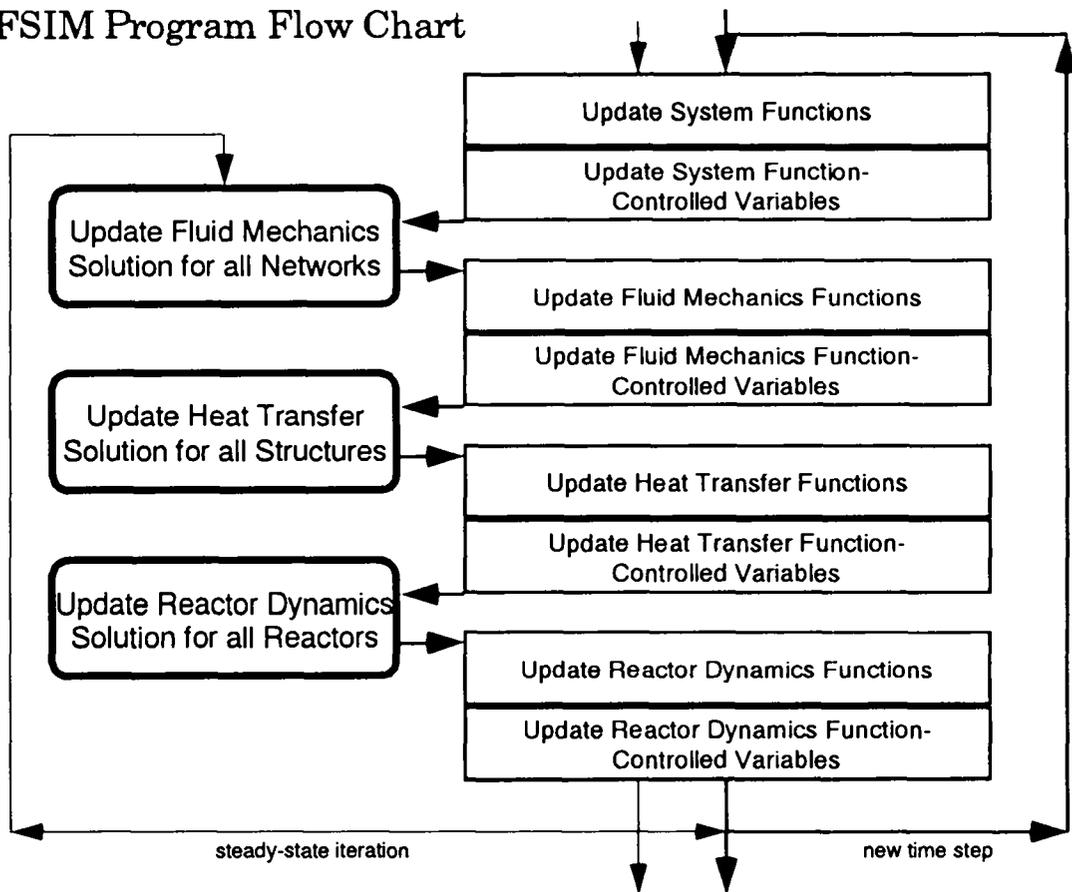
Building a system model requires the building of the fluid mechanics, heat transfer, and reactor dynamics models. The details of building these models are provided in the next three subsections. The analyst must ensure that the proper connections between the three models are provided. For example, a heat transfer structure with a convective boundary condition requires the input of the connecting fluid mechanics network and element identification numbers. If additional fluid mechanics elements are added, the connections must be updated to reflect the additions.

After the fluid mechanics, heat transfer, and nuclear reactor models are completed, function-controlled variables and functions may be specified. This allows the functional specification of many of SAFSIM's input variables. For example, a time-dependent pressure boundary condition could be implemented with a pressure versus time table using function-controlled variables. Another example is the ability to control a flow area at one location as a function of the temperature or pressure at some other location. The use of function-controlled variables is fully explained in Subsection 2.4. Because most of SAFSIM's input variables can be function controlled, this capability offers tremendous modeling flexibility.

A top level flow chart of the SAFSIM program is provided in the next figure. The flow chart shows the order that the physics modules, functions,

and function-controlled variables are updated for steady-state and transient calculations. This order should be kept in mind when creating the function and function-controlled variable data. System functions and function-controlled variables are updated only once for steady-state calculations. Typically, all time-dependent functions are included in the system blocks. However, any function or function-controlled variable can be included in any of the function or function-controlled variable blocks. In other words, including a function in the fluid mechanics function block specifies when that function is updated.

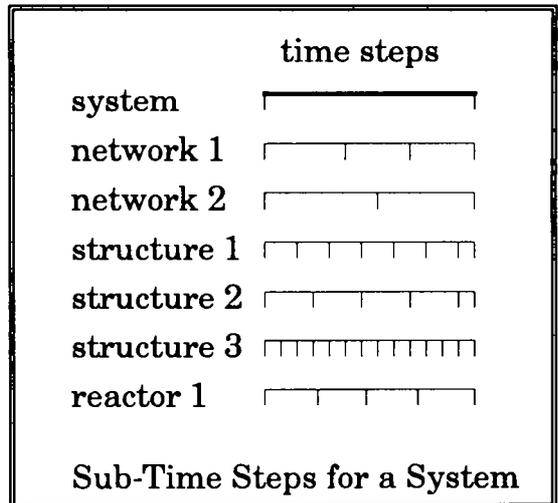
SAFSIM Program Flow Chart



The fluid mechanics, structure heat transfer, and reactor dynamics solutions, along with the function-controlled variables, are explicitly coupled to simulate the integrated performance of the entire system. Employing explicit coupling greatly increases program versatility and allows the analyst to specify different sub-time steps for each fluid mechanics network, heat transfer structure, and nuclear reactor in the system. Thus, sub-time steps can be chosen consistent with the characteristic time constant of the different physical phenomena. Automatic time step control for the heat transfer structures and nuclear reactors is provided by SAFSIM and can be implemented if desired. Also, the analyst can specify her own time step control

logic via function-controlled variables and functions. The following figure demonstrates the use of different sub-time steps for an arbitrary system.

The system time step is specified by the analyst and can be controlled via function-controlled variables if desired. Sub-time steps are truncated to coincide with the end of each system time step. For very rapid transients, the system time step can be reduced to more tightly couple the different parts of the system.



The size of an input model is constrained by the array dimensions within SAFSIM. All array dimensions are specified by parameter statements that can be modified to accommodate computer memory limitations and problem size requirements. The following table provides a list of the FORTRAN parameters that define the current array dimensioning within SAFSIM. The user can modify these parameters and then recompile and link SAFSIM to incorporate the changes. Reducing the parameter values reduces the computer memory requirements for SAFSIM.

Array Dimensioning Parameters

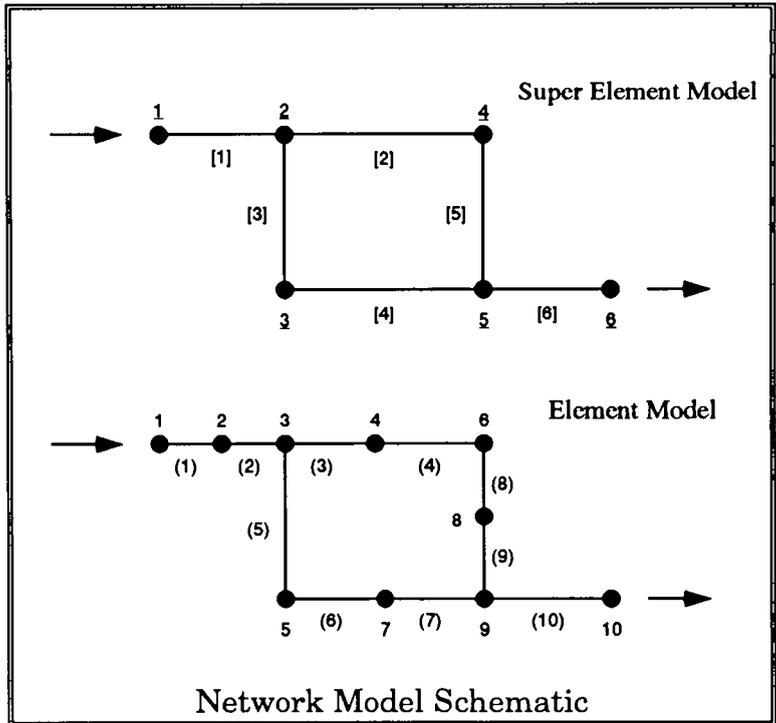
Parameter	Value	Description
MAXEL	500	maximum number of fluid mechanics elements
MAXELS	250	maximum number of super elements = MAXEL/2
MAXN	500	maximum number of fluid mechanics nodes
MAXNS	250	maximum number of super nodes = MAXN/2
MAXSETS	25	maximum number of boundary condition set IDs
MAXC	10	maximum number of fluid mechanics element connections allowed to a single node
MAXNOZ	25	maximum number of nozzle (choked flow boundary) elements
MAXCMPSR	25	maximum number of compressor elements
MAXDF	50	maximum number of distributed flow elements
MAXES	25	maximum number of elements per super element
MAXNTR	4	maximum number of temperature ranges for fluid property data
MAXG	4	maximum number of gases (or liquids)
MAXNW	5	maximum number of fluid mechanics flow networks
MAXFCV	200	maximum number of function-controlled variables

MAXF	200	maximum number of functions
MAXFCON	25	maximum number of functions that are constants
MAXFTAB	25	maximum number of functions that are tables
MAXFPOL	25	maximum number of functions that are polynomials
MAXPAIR	30	maximum number of function table x,y pairs
MAXORP1	5	maximum polynomial order + 1 for functions
MAXFTAV	10	maximum number of functions that are used for calculating average structure temperatures
MAXFTE	100	maximum number of table entries for average temperature functions
MAXFDAV	10	maximum number of functions that are used for calculating average fluid density
MAXFDE	100	maximum number of table entries for average density functions
MAXHS	200	maximum number of heat transfer structures
MAXELH	500	maximum number of heat transfer elements
MAXNH	50	maximum number of nodes per structure
MAXEX	3	maximum number of exchange surfaces per heat transfer element
MAXEXT	350	maximum number of exchange surfaces for all heat transfer elements
MAXFLUXA	3	maximum number of heat flux surfaces per element
MAXDNG	25	maximum number of delayed neutron groups
MAXDHG	15	maximum number of decay heat groups
MAXRX	3	maximum number of nuclear reactors
MAXREAC	10	maximum number of differential reactivity feedback terms per reactor

2.1 Creating a Fluid Mechanics Network Input Model

As mentioned in Section 1.0, super elements are employed for solution of the mechanical energy equation (which provides pressures and mass flow rates) and greatly enhance computational efficiency. However, super elements cannot be used for the thermal energy equation (which provides fluid temperatures) because of the convective heat transfer terms. Therefore, it is necessary to create two element networks -- a super element network for the mechanical energy equation and an element network for the thermal energy equation. The two networks communicate at the super nodes with respect to pressure. Super nodes are those nodes associated with a super element.

The accompanying figure depicts the two input models for a sample flow network. Super element identification (ID) numbers are enclosed in brackets and element ID numbers are enclosed in parentheses. Super node ID numbers are underlined. The super element model contains six super elements and six super nodes. The element model contains ten elements and ten nodes. Super element [1] contains two elements: (1) and (2); whereas super element [3] contains only one element: (5). The super node ID numbers were selected to minimize the bandwidth of the pressure coefficient matrix. This should always be done to maximize computational efficiency. Automatic bandwidth minimization is planned for a future version of SAFSIM. The node ID numbers were also selected to minimize the bandwidth, but this time for the temperature coefficient matrix.



Two different strategies for selecting node ID numbers should be considered for each problem: (1) bandwidth minimization and (2) flow direction dependent. Flow direction dependent refers to the numbering of nodes in ascending order based on the direction of fluid flow. This strategy should be selected for problems that are advectively dominated, as is typical with gases. This means a node temperature is a function primarily of the upstream nodes. In the sample network, if the flow is advectively dominated, the temperature at node 9 is dependent only on the temperatures of nodes 7 and 8. Such problems can employ the Gauss-Seidel iterative method for solution of the node temperatures, using a single iteration. Although this node numbering strategy is very efficient, it has two shortcomings. If the flow reverses direction, the nodes are numbered non-optimally for Gauss-Seidel. And, if the flow is not advectively dominated, Gauss-Seidel iteration will probably work but it may not be the optimal numerical method. Therefore, bandwidth minimization should be selected if flow reversals are expected or if the flow is not advectively dominated. Gauss elimination is employed for such flows. Advective dominance is diminished if the flow rate is very small or if the fluid conductivity is very large (as may be the case for liquid metals).

Automatic bandwidth minimization is planned for a future version of SAFSIM. In the future version, the nodes would be numbered based on the flow direction dependent strategy. A second set of node ID numbers would be created by SAFSIM based on the bandwidth minimization strategy. SAFSIM would then choose the node ID numbers and the appropriate numerical method based on the current fluid conditions.

The node ID numbers discussed in the previous paragraph refer to global nodes. In addition to global nodes, every element (and every super element) is associated with two local nodes. The specification of local nodes determines the sign (+/-) of the computed flow rate for that element. For example, element (6) of the sample flow network may have global node 5 assigned to local node 1 and global node 7 assigned to local node 2. In this case, if the flow is in the direction as indicated by the arrow (from left to right), the computed flow rate will be positive. If the local node ID number assignments are reversed, the computed flow rate will be negative. Thus, flow rates are positive if flow is from local node 1 to local node 2 of the element. The same rules apply to super elements and super nodes; however, consistency of local node assignments is required. Thus, if global node 5 is assigned to local node 1 of element (6), global super node 3 must also be assigned as local super node 1 of super element [4]. The analyst should choose a local node numbering scheme for each specific problem to facilitate easy interpretation of the mass flow rate output.

2.2 Creating a Structure Heat Transfer Input Model

Creating a structure heat transfer input model is somewhat easier than creating a fluid mechanics input model because heat transfer element networks are not allowed. Thus, all elements for a structure are connected in series and each structure has just two ends. The first node (end 1) is numbered 1 and the last node (end 2) is numbered NEH+1, where NEH is the number of finite elements for the structure.

The geometry of a structure is specified by three variables: conduction length, area, and volume. These three variables must be supplied for each heat transfer finite element. For non-uniform area structures, an appropriate element area must be selected. One choice is to use the algebraic average of the node areas. Another choice is to use the element volume divided by the element length. For cylindrical geometry, with conduction in the radial direction, the appropriate area to use is the log-mean area determined as

$$A' = \frac{2\pi H \Delta r}{\ln(r_o/r_i)}. \quad (1)$$

A' is the effective conduction area, H is the height of the cylinder, Δr is the element conduction length, and r_o and r_i are the outer and inner radii, respectively. This equation should be used for all elements except for the center element of a solid cylindrical structure ($r_i=0$), where the volume divided by the length should be used. The appropriate area for spherical geometry is given by

$$A' = 4\pi r_i r_o. \quad (2)$$

Again, the volume divided by the length should be used for the center element. SAFSIM's automatic geometry option uses these equations for cylindrical and spherical geometries. In addition, the automatic geometry option allows specification of partial cylindrical and spherical sections.

A useful feature of SAFSIM is the ability to specify multiple convection exchange surfaces for each finite element of a structure and to couple each exchange surface to different fluid mechanics elements. This capability allows a pseudo multidimensional conduction capability. Thus, a plate can be modeled with convection to one fluid stream on one side of the plate and to a different fluid stream on another side of the plate. An additional exchange surface could be specified for radiation from a side of the plate. It is also possible to couple one heat transfer structure to another heat transfer structure using functions and function-controlled variables.

2.3 Creating a Reactor Dynamics Input Model

Creating a reactor dynamics input model is relatively easy because the reactor is treated as a point (0-D); thus, no explicit geometry information is required. However, one input feature requires further explanation: the specification of differential reactivity feedback terms (variable DREAC in Block 13 data). The input for reactor dynamics only requires the specification of the desired number of terms and their initial values. To specify the equation that defines the feedback terms requires the use of function-controlled variables. Thus, all DREAC terms must be specified as function-controlled variables. Because function-controlled variables allow the analyst to specify input variables via user-selected functions, the analyst is not locked into any predefined equation for reactivity feedback. For example, a feedback term could be defined as a fourth-order polynomial function of an average structure temperature. Also, the requirement of using function-controlled variables to define feedback reactivity terms is consistent with the explicit coupling of the fluid mechanics, heat transfer, and reactor dynamics physics modules.

2.4 Creating Function-Controlled Variables

A powerful feature of SAFSIM is the ability to specify many of the input variables as functions of the output variables. This allows the user to define input variables with a functional dependence of the form: $y = x$ or $y = f(x)$. y is a function-controlled input variable, x is a signal-variable function, and $f()$ is a specified mathematical function. In the Block data input that follows in Section 3.0, function and function-controlled variable input can be specified in different Blocks depending on the order the analyst wants the function or variable to be updated. For example, including a fluid mechanics variable in the function-controlled variables for reactor dynamics data Block specifies that the variable is to be updated after the reactor dynamics solution is updated.

Perhaps the best way to explain the implementation of functions and function-controlled variables is by example.

Example 1: Specify the flow length (DX) of fluid mechanics element 7 of flow network 1 to be a function of time as specified by a table lookup.

For this example, the function-controlled variable, DX, has an ID of 36 (from Appendix A). The associated argument function number (IAFN) is arbitrarily assigned the number 15. This number is used in specification of the function. Thus, the function-controlled variable is defined by the following input variables to be included in Function-Controlled Variable Data.

```
IFCVID=36 (from Appendix A)
FCVNAME='element 7 length, NW1'
IAFN=15
FCVM=1.0
L1=1 (the network number)
L2=7 (the element number)
L3 is not used
```

Next, the information to define the argument function (IAFN=15) is provided. This information is included in the Function Data. The table lookup mathematical function has an ID of 114 (from Appendix B).

```
IFN=15 (referenced by the function-controlled variable IAFN number)
IFID=114 (from Appendix B)
FNAME='DX vs. time table'
FM=1.0
FVALUE=0.1
Supplemental variables:
    number of data pairs=5
```

argument IFN=20 (to be specified by a separate function)
data pairs (used to specify the table entries):
0.0, 0.1, 1.0, 0.11,
2.0, 0.115, 3.0, 0.12,
5.0, 0.125

Now it is necessary to define function 20, the argument function for the table lookup. This function is a signal-variable function as opposed to a mathematical function.

IFN=20 (referenced by the function argument IFN number)
IFID=1 (from Appendix B)
FNAME='Problem Time'
FM=1.0
FVALUE=0.0
No supplemental variables are required for this function.

Because functions are updated in the order they are listed in the input file, function IFN=20 should be listed before function IFN=15 so that the current time is used in the table lookup.

Example 2: Specify the second differential reactivity feedback (DREAC) term such that $DREAC=2.5E-6 \cdot \Delta T$. T is the average temperature of heat transfer structures 2 and 3, which each contains 3 elements.

For this example, the function-controlled variable, DREAC, has an ID of 98 (from Appendix A). The associated argument function number (IAFN) is arbitrarily assigned the number 25. This number is used in specification of the function. Thus, the function-controlled variable is defined by the following input variables to be included in Function-Controlled Variable Data.

IFCVID=98 (from Appendix A)
FCVNAME='moderator feedback'
IAFN=-25
FCVM=2.5E-6
L1=1 (the reactor number)
L2=2 (the feedback term number)
L3 is not used

Next, the information to define the argument function (IAFN=25) is provided. This information is included in the Function Data. The function ID of 122 (from Appendix B) defines the mathematical function for averaging heat transfer structure temperatures. Note that the IAFN number for the function-controlled variable is specified as a negative number. This indicates that the

change in the argument function (new-time value minus the old-time value) is to be used.

IFN=25 (referenced by the function-controlled variable IAFN number)

IFID=122 (from Appendix B)

FNAME='average moderator temperatur'

FM=1.0

FVALUE=300.0

Supplemental variables:

number of heat transfer structure elements=6

(structure number, element number, weighting factor) for each element:

2, 1, 1.0,

2, 2, 1.0,

2, 3, 1.0,

3, 1, 1.0,

3, 2, 1.0,

3, 3, 1.0

Example 3: Specify the second differential reactivity feedback (DREAC) term such that $DREAC=(4.2E-7/T)\cdot\Delta T$. T is the average temperature of heat transfer structures 2 and 3, which each contain 3 elements. This is similar to example 2 except the constant multiplying ΔT has been replaced by a constant divided by T. This example is explained using standard mathematical nomenclature.

The basic form of the desired functional relationship can be expressed as $y=f_1\{f_2[f_3(x)],\Delta f_3(x)\}$, where y is the function-controlled variable (DREAC) and $f_1\{ \}$ is the multiplication mathematical function (IFID=133 from Appendix B). This function has two argument functions: $f_2[]$ and $f_3()$. $f_2[]$ is the first argument function and is defined as the inverse mathematical function (IFID=137); this function is to include a multiplier of 4.2E-7. $\Delta f_3()$ is the second argument function and is defined as an averaged temperature mathematical function (IFID=122, as in example 2). A Δ sign preceding the function indicates that the change in the function is to be used; this is specified in the input by including a minus sign in front of the argument function ID number. This function requires supplemental argument variables specifying the structure and element numbers to include in the averaging process. These arguments are represented by the variable x. $f_2[]$ requires one argument function: $f_3()$. In summary, $f_3(x)$ is the average temperature, T; $\Delta f_3(x)$ is the change in the average temperature, ΔT ; $f_2[f_3(x)]$ is the inverse of $f_3(x)$, $1/T$, multiplied by 4.2E-7; and $f_1\{ \}$ is the function that multiplies 4.2E-7/T by ΔT . Although this example is somewhat complicated, it demonstrates the extensive flexibility available to the analyst via the input file.

3.0 Input Data Blocks

The SAFSIM input file is divided into eighteen data Blocks. The variables required for each of these Blocks are presented line by line in this section. Each line to be entered is numbered and marked with a  symbol. Special notes pertaining to data Blocks and lines are marked with a  symbol. Comment lines can be added anywhere within the input file as desired. Comment lines are indicated by an upper- or lower-case c or by an asterisk (*) in the first column. The liberal use of comment lines is encouraged, especially for complex systems.

All input variables are read in free format and therefore are independent of column location in the data line. Variables can be separated by either blank space or by a comma. Comma delimiters are recommended because they serve as a field place holder for each of the variables entered on the data line. Any variable that is not entered is read in as a null field. Thus, default values are not overwritten by the read statement. Commas can be used to mark the null fields. Thus, if a line calls for three variables, entering ,, indicates that the default values for all three variables are to be used. If a comma is omitted, the last variable is read from the next data line containing an entry. If desired, each of the three variables can be entered on a separate data line.

All non-character input variables are defined according to the FORTRAN implicit types. Thus, variables beginning with the letters I, J, K, L, M, and N are defined as integer variables and variables beginning with one of the other letters (A - H, and O - Z) are defined as real (floating point) variables. Some of the variables are read in as character data as indicated by their *description* on the following pages. These character variables must be enclosed by apostrophes; for example 'NAME'. Do not include apostrophes embedded within these character variables. One exception exists with regard to enclosing apostrophes: the END variable at the end of each data Block. This character variable must not include enclosing apostrophes.

SAFSIM employs SI/mKs units for all variables. The required unit for each variable is provided at the end of the variable *description*. The following table provides frequently used conversion factors for quick reference.

Table of Conversion Factors

Quantity	Given in	Multiplied by	Gives
length	ft	0.3048	m
	in	0.0254	m
area	ft ²	0.092903	m ²
	in ²	6.4516E-4	m ²
volume	ft ³	0.028317	m ³
	in ³	1.6387E-5	m ³
	U.S.gal	3.7854E-3	m ³
velocity	ft/min	0.00508	m/s
mass	lb _m	0.45359	kg
	ton (2000 lb _m)	907.18	kg
mass flow rate	lb _m /h	1.260E-4	kg/s
	lb _m /s	0.4536	kg/s
	lb _m /min	0.00756	kg/s
power	Btu/h	0.2931	W (J/s)
	kcal/h	1.163	W
	ft·lb _f /s	1.3558	W
	hp	735.5	W
heat flux	Btu/h·ft ²	3.1546	W/m ²
	kcal/s·cm ²	41.868	W/m ²
heat transfer coefficient	Btu/h·ft ² °F	5.6784	W/m ² ·K
	kcal/s·cm ² °C	41.868	W/m ² ·K
pressure	lb _f /ft ²	47.9	Pa
	lb _f /in ² (psi)	6894.8	Pa
	bar	1.0E5	Pa
	atm	1.01325E5	Pa
thermal conductivity	Btu/ft·h °F	1.7308	W/m·K
	kcal/m·h °C	1.163	W/m·K
density	lb _m /ft ³	16.0185	kg/m ³
	lb _m /U.S.gal	119.7	kg/m ³
specific heat	Btu/lb _m °F	4186.8	J/kg·K
	kcal/kg °C	4186.8	J/kg·K
dynamic viscosity	P (poise)	0.1	Pa·s
	lb _m /ft·h	4.134E-4	Pa·s
	lb _m /ft·s	1.4482	Pa·s
temperature	°F	T (K) = [T (°F) + 459.67]/1.8	

Block 1, Execution Data:

line1: TITLE1

<u>Variable</u>	<u>Description</u>
TITLE1	any descriptive title of up to 126 characters may be entered; enclosing apostrophes required; do not embed apostrophes within the variable; default: 'PROGRAM SAFSIM: Systems Analysis'; units: none

line2: TITLE2

<u>Variable</u>	<u>Description</u>
TITLE2	any descriptive title of up to 126 characters may be entered; enclosing apostrophes required; do not embed apostrophes within the variable; default: 'FLUID MECHANICS, HEAT TRANSFER, REACTOR DYNAMICS'; units: none

line3: RESTART, MAXSS,AERRTW,RERRTW,RELAXTW

<u>Variable</u>	<u>Description</u>
RESTART	restart flag, enter 'on' or 'off'; enclosing apostrophes required; default: 'off'; units: none; (not yet functional)
MAXSS	the maximum number of steady-state iterations for wall temperature allowed; if MAXSS=0, the steady-state option is bypassed; default: 250; units: none
AERRTW	the desired absolute error for steady-state wall temperature iterations; default: 0.25; units: K
RERRTW	the desired relative error for steady-state wall temperature iterations; this criteria is used only if the absolute convergence criteria (AERRTW) is not satisfied; default: 2.5E-4; units: none
RELAXTW	the relaxation parameter used for steady-state wall temperature iterations; default: 1.7; units: none

 **line4: TSTART, TEND, ISTART, TIMESTEP**

<u>Variable</u>	<u>Description</u>
TSTART	the problem start time; default: 0.0; units: s
TEND	the problem end time; default: 1.0; units: s
ISTART	the problem start time step number; default: 0; units: none;
TIMESTEP	the problem time step for the system; separate sub-time steps will also be set for each fluid network, heat transfer structure, and nuclear reactor; default: 0.1; units: s

 **line5: PRNTINT, NPLT, NDUMP**

<u>Variable</u>	<u>Description</u>
PRNTINT	the problem print interval for the system; printed output is generated every PRNTINT seconds for all flow networks, heat transfer structures, nuclear reactors, and function-controlled variables; default: 1.0; units: s
NPLT	the problem plot frequency for the system; unformatted output for all flow networks, heat transfer structures, nuclear reactors, and function-controlled variables is written to the respective plot files every NPLT system time steps; default: 25; units: none
NDUMP	the problem dump frequency for the system; unformatted output is written to the dump file every NDUMP system time steps; default: 100; units: none; (not yet functional)

 **line6: AGRAV, FLOED, TURLAM, SPEEDCO, THD**

<u>Variable</u>	<u>Description</u>
AGRAV	the acceleration due to gravity; default: 9.80665; units: m/s ²
FLOED	the default finite element flow rate; finite elements with the variable FLOES left blank default to the value of FLOED; default: 1.0E-10; units: s

- TURLAM the Reynolds number at which transition from laminar to turbulent flow occurs for friction factor calculations; default: 3000.0; units: none
- SPEEDCO the fractional cutoff speed (operational/rated) below which compressor finite elements are assumed not to be operating; default: 0.05; units: none
- THD the default value for the initial node temperatures for all heat transfer structures; default: 300.0; units: K

 **line7: PMCORR**

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

PMCORR	the correlation to be used for determining the friction factor for porous media finite elements; choices are 'ACHENBACH', 'ERGUN', and 'BEEK'; if any other name is entered, the user must enter five constants on line 8 to define the correlation; maximum of 9 characters allowed; enclosing apostrophes required; default: 'ACHENBACH', units: none
--------	---

 **line8: PMC1, PMC2, PMC3, PMC4, PMC5**

 enter this line only if a porous media correlation other than one of the PMCORR choices is selected; the form of the correlation: friction factor = $(1-\epsilon)/\epsilon^3 \cdot (PMC2 \cdot ((1-\epsilon)/Re)^{PMC4} + PMC3 \cdot ((1-\epsilon)/Re)^{PMC5} + PMC1)$; ϵ is the porosity and Re is the Reynolds number based on the superficial velocity and the particle diameter; the default values correspond to the 'ACHENBACH' correlation

Variable	Description
----------	-------------

PMC1	first constant for defining a porous media friction factor correlation; default: 1.75; units: none
PMC2	second constant for defining a porous media friction factor correlation; default: 320.0; units: none
PMC3	third constant for defining a porous media friction factor correlation; default: 20.0; units: none
PMC4	fourth constant for defining a porous media friction factor correlation; default: 1.0; units: none

PMC5 fifth constant for defining a porous media friction factor correlation; default: 0.4; units: none

 **line9: NUMNW**

<u>Variable</u>	<u>Description</u>
NUMNW	the total number of independent flow networks for this problem; default: 1; units: none

 repeat lines 10 through 16 for each independent flow network

 **line10: NW, MEE, ISKIPCD, NWNAM**

<u>Variable</u>	<u>Description</u>
NW	the identification number of the flow network for which the data in lines 10 through 16 applies; the networks must be numbered sequentially starting from 1 but can be listed non-sequentially in the input file if desired; NW must be less than or equal to NUMNW; default: none; units: none
MEE	an on/off switch for the mechanical energy equation solution; if MEE = 1, the mechanical energy equation is solved; if MEE = 0, the mechanical energy equation is bypassed; default: 1; units: none
ISKIPCD	an on/off switch for the use of Cholesky decomposition for the solution of the mechanical energy equation; if ISKIPCD = 1, Cholesky is skipped; if ISKIPCD = 0, Cholesky is not skipped; Cholesky should be skipped if the pressure coefficient matrix is not positive definite; default: 0; units: none
NWNAM	any descriptive name of up to 24 characters to identify the flow network in the output files; enclosing apostrophes required; default: 'FLUID MECHANICS NETWORK'; units: none

 **line11: DTFM, NPRNTF, IDYNFM**

<u>Variable</u>	<u>Description</u>
DTFM	the desired time step for the fluid mechanics solution; each network can have a different time step; DTFM is prevented

from exceeding the current system time step; default: 3.0E-3;
units: s

NPRNTF the print frequency for this network; fluid mechanics output is written to the output file every NPRNTF fluid mechanics time steps; this output is in addition to that generated every PRNTINT seconds as specified in Block 1 data; default: 10000; units: none

IDYNFM an on/off switch for the dynamic (unsteady-state) terms in the fluid mechanics equations (time derivatives of temperature, mass flow rate, and density); if IDYNFM = 1, the dynamic terms are included; if IDYNFM = 0, the dynamic terms are not included; default: 0; units: none; (only the static option is currently available)

line12: MAXIP, MAXIM, MAXID, MAXIT, MAXIC, MAXIX

<u>Variable</u>	<u>Description</u>
MAXIP	the maximum number of iterations allowed for solution of the nodal pressure equations using Gauss-Seidel iteration; if this number is exceeded, Cholesky decomposition or Gauss elimination is used; if MAXIP = 0, Gauss-Seidel iteration is skipped; default: 10, units: none
MAXIM	the maximum number of fixed-point iterations allowed for solution of the element mass flow rate field; if this number is exceeded, an error message is issued and execution ends; default: 50; units: none
MAXID	the maximum number of fixed-point iterations allowed for solution of the element density field via the equation of state; this iteration couples the mechanical and thermal energy equations; if this number is exceeded, an error message is issued and execution ends; default: 50; units: none
MAXIT	the maximum number of iterations allowed for solution of the nodal temperature equations using Gauss-Seidel iteration; if this number is exceeded, Gauss elimination is used; if MAXIT = 0, Gauss-Seidel iteration is skipped; default: 5, units: none
MAXIC	the maximum number of fixed-point iterations allowed for solution of the element temperature field; this iteration updates the element specific heat values based on the calculated

node temperatures; if this number is exceeded, an error message is issued and execution ends; default: 5; units: none

MAXIX the maximum number of Gauss-Seidel iterations allowed for solution of the nodal mass fraction equations for each gas of a multiple gas network; if this number is exceeded, Gauss elimination is used; if **MAXIX** = 0, Gauss-Seidel iteration is skipped; default: 10; units: none

line13: DRERRP, DRERRM, DRERRD

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

DRERRP	the desired relative error for solution of the nodal pressure equations using Gauss-Seidel iteration; default: 1.0E-9; units: none
---------------	--

DRERRM	the desired relative error for solution of the element mass flow rate field; default: 1.0E-4; units: none
---------------	---

DRERRD	the desired relative error for solution of the element fluid density field; default: 1.0E-3; units: none
---------------	--

line14: DRERRT, DRERRC, DRERRX

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

DRERRT	the desired relative error for solution of the nodal temperature equations using Gauss-Seidel iteration; default: 1.0E-4; units: none
---------------	---

DRERRC	the desired relative error for solution of the element temperature field; default: 1.0E-3; units: none
---------------	--

DRERRX	the desired relative error for solution of the nodal temperature equations using Gauss-Seidel iteration; default: 1.0E-3; units: none
---------------	---

line15: RELAXP, RELAXM, RELAXD

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

RELAXP	the relaxation parameter used in the Gauss-Seidel iterative solution for nodal pressure; RELAXP must be greater than 0.0 and less than or equal to 2.0; default: 1.1; units: none
---------------	--

RELAXM the relaxation parameter used in the fixed-point iterative solution for the element mass flow rate field; RELAXM must be greater than 0.0 and less than or equal to 2.0; default: 1.2; units: none

RELAXD the relaxation parameter used in the fixed-point iterative solution for the element fluid density field; RELAXD must be greater than 0.0 and less than or equal to 2.0; default: 1.0; units: none

 **line16: RELAXT, RELAXC, RELAXX**

<u>Variable</u>	<u>Description</u>
RELAXT	the relaxation parameter used in the Gauss-Seidel iterative solution for nodal temperature; RELAXT must be greater than 0.0 and less than or equal to 2.0; default: 1.0; units: none
RELAXC	the relaxation parameter used in the fixed-point iterative solution for the element temperature field; RELAXC must be greater than 0.0 and less than or equal to 2.0; default: 1.0; units: none
RELAXX	the relaxation parameter used in the Gauss-Seidel iterative solution for the nodal mass fraction equations; RELAXX must be greater than 0.0 and less than or equal to 2.0; default: 1.0; units: none

 the relaxation parameter is defined such that for any variable v, an updated value is calculated as: $v = \text{RELAX} \cdot v^i + (1.0 - \text{RELAX}) \cdot v^{i-1}$; i is the current iteration number; use overrelaxation (RELAX greater than 1.0) only for an already convergent equation set, and use underrelaxation (RELAX less than 1.0) for an oscillating equation set; use RELAX = 1.0 as a conservative value that may not be optimum with respect to minimizing iterations, but usually results in convergence

 **line17: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 1 data; do not include enclosing apostrophes

Block 2, Fluid Equation of State and Property Data:

☞ repeat line 18 for each independent flow network

☞ line18: NW, NGAS, GAS(igas=1,NGAS)

<u>Variable</u>	<u>Description</u>
NW	the identification number of the network for which the data in line 18 applies; NW must be less than or equal to NUMNW; the NW values can be listed in any order; default: none; units: none
NGAS	the number of fluids in independent flow network NW; NGAS must be set to 1 if the fluid is to be specified as a liquid (see variable RGAS); default: 1; units: none
GAS	the name of the gas (or liquid), for each of the NGAS fluids; any name of up to 15 characters may be specified; igas is the gas constituent number; the properties are specified on the following data lines; enclosing apostrophes required; default: none, units: none

☞ repeat lines 19 through 25 for each gas (or liquid) to be included in Block 2 data; the first gas listed is idgas=1,... ; idgas is used in the specification of functions and function-controlled variables; property data can be included but not referenced if desired

☞ line19: GAS, IEOSO, RGAS, DENC(k=1,3)

<u>Variable</u>	<u>Description</u>
GAS	the name of the gas (or liquid); any name of up to 15 characters may be specified; this name must match exactly with one of the names specified on the previous line(s); enclosing apostrophes required; default: 'HYDROGEN', units: none
IEOSO	the equation of state option for determining the density for this fluid; three options are available: 1 -- ideal gas law, 2 -- second order polynomial function of temperature (useful for incompressible liquids), and 3 -- user-specified (the user must

modify Functions UEOS, UCP, UCV, UCOND, and UVIS);
default: 1; units: none

RGAS the gas constant to be used in an ideal gas equation of state for density (IEOSO = 1); if IEOSO equals 2 (liquid equation of state, only one liquid per network allowed), RGAS is the bulk compressibility modulus and 3 values for variable DENC are required to define the equation of state; if IEOSO equals 3, RGAS should represent an approximate gas constant to be used for speed of sound and multiple gas mixing calculations; default: 4124.18 (corresponding to hydrogen); units: J/kg·K

DENC the 3 constants required to specify density as a function of temperature using a second order polynomial for simulating a liquid equation of state; required only if IEOSO equals 2; the form of the equation for liquid density: $\text{density} = \text{DENC1} + \text{DENC2} \cdot T + \text{DENC3} \cdot T^2$; default: none; density units: kg/m³, temperature units: K

☞ include the next six lines only if IEOSO=1 or 2; for IEOSO=3 (user-specified equation of state), the user must supply fluid properties via FORTRAN Functions UCP, UCV, UCOND, and UVIS; the default values for the constants correspond to hydrogen and are valid only for the temperature range from 273 K to 1800 K

📎 **line20: VISM, NTRV, TRV(k=1,NTRV+1)**

<u>Variable</u>	<u>Description</u>
VISM	the viscosity multiplier for all temperature ranges; default: 1.0; units: none
NTRV	the number of temperature ranges for which viscosity constants are to be specified; default: 1, units: none
TRV	the temperature range values for viscosity evaluation; enter the minimum temperature first [TRV1, TRV2, ..., TRV(NTRV+1)]; default: none; units: K

☞ repeat line 21 for each temperature range as specified by NTRV; viscosity can be specified using the Sutherland law or power law equations; Sutherland law: $\mu = \text{VISM} \cdot (\text{VOV} \cdot (T/\text{TOV})^{1.5} \cdot ((\text{TOV} + \text{SNV}) / (T + \text{SNV})))$
power law: $\mu = \text{VISM} \cdot (\text{VOV} \cdot (T/\text{TOV})^{\text{SNV}})$; temperature units: K

 **line21: VOV, TOV, SNV, LAW**

<u>Variable</u>	<u>Description</u>
VOV	viscosity function constant; default: 8.411E-6; units: Pa·s
TOV	viscosity function constant; default: 273.111; units: K
SNV	viscosity function constant; default: 0.68; units: none
LAW	the law to use to specify viscosity; either 'SUTHERLAND' or 'POWER'; enclosing apostrophes required; default: 'POWER', units: none

 **line22: CPM, NTRC, TRC(k=1,NTRC+1)**

<u>Variable</u>	<u>Description</u>
CPM	the constant pressure specific heat multiplier for all temperature ranges; default: 1.0; units: none
NTRC	the number of temperature ranges for which specific heat constants are to be specified; default: 1, units: none
TRC	the temperature range values for specific heat evaluation; enter the minimum temperature first [TRC1, TRC2, ..., TRC(NTRC+1)]; default: none; units: K

 repeat line 23 for each temperature range as specified by NTRC; specific heat is specified as a third order polynomial: $c_p = ACP + BCP \cdot T + CCP \cdot T^2 + DCP \cdot T^3$; specific heat units: J/kg·K; temperature units: K

 **line23: ACP, BCP, CCP, DCP**

<u>Variable</u>	<u>Description</u>
ACP	specific heat function constant; default: 14439.5
BCP	specific heat function constant; default: -0.9504
CCP	specific heat function constant; default: 1.9856E-3
DCP	specific heat function constant; default: -4.3175E-7

 **line24: CONDM, NTRK, TRK(k=1,NTRK+1)**

<u>Variable</u>	<u>Description</u>
CONDM	the conductivity multiplier for all temperature ranges; default: 1.0; units: none
NTRK	the number of temperature ranges for which conductivity constants are to be specified; default: 1, units: none
TRK	the temperature range values for conductivity evaluation; enter the minimum temperature first [TRK1, TRK2, ..., TRK(NTRK+1)]; default: none; units: K

 repeat line 25 for each temperature range as specified by NTRK; conductivity is specified as a third order polynomial: $k = ACOND + BCOND \cdot T + CCOND \cdot T^2 + DCOND \cdot T^3$; conductivity units: W/m·K; temperature units: K

 **line25: ACOND, BCOND, CCOND, DCOND**

<u>Variable</u>	<u>Description</u>
ACOND	conductivity function constant; default: 8.099E-3
BCOND	conductivity function constant; default: 6.689E-4
CCOND	conductivity function constant; default: -4.158E-7
DCOND	conductivity function constant; default: 1.562-10

 **line26: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 2 data; do not include enclosing apostrophes

Block 3, Fluid Mechanics Finite Element Data:

☞ repeat lines 27 through 36 for each independent flow network

☞ repeat lines 27 through 35 for each super element of the network

☞ line27: JES, N1S, N2S, NE

<u>Variable</u>	<u>Description</u>
JES	the super element identification (ID) number; super elements can be listed in any order but must be sequentially numbered for each network, starting with 1; default: none; units: none
N1S	the global super node ID number for the first local node of the super element; flow from local node 1 to local node 2 is defined to be the positive direction; default: none; units: none
N2S	the global super node ID number for the second local node of the super element; flow from local node 1 to local node 2 is defined to be the positive direction; default: none; units: none
NE	the number of elements in this super element; default: none; units: none

☞ repeat lines 28 through 34 for each NE elements in this super element

☞ line28: JE, NAME, N1, N2

<u>Variable</u>	<u>Description</u>
JE	the element ID number; the elements must be listed in the order that they are connected to form the super element; elements must be sequentially numbered for each network, starting with 1; default: none; units: none
NAME	any descriptive name of 10 characters or less to identify the finite element in the output files; enclosing apostrophes required; the first character of NAME is used to specify special purpose elements; three special purpose elements currently exist, their first-character symbol identifiers are (1) < or > for a nozzle element, (2) @ or & for a compressor element, and (3) (or) for a distributed flow manifold element; for example:

enter '>Nozzle 1' to specify a choked flow boundary (nozzle) element with the name Nozzle 1; a nozzle element must terminate a flow network with a boundary condition specified at N2, and nozzle elements cannot be used if the network fluid is a liquid; the last character of the variable NAME can be used to mark any element as a distributed flow manifold terminator element; thus, including the symbol [or] as the tenth character of NAME indicates that the element is a distributed flow manifold terminator element (all elements connected to a manifold element, except other manifold elements, are assumed to be manifold feed lines unless designated as a terminator element); default: 'PIPE'; units: none

- N1 the global node ID number for the first local node of the element; positive flow is defined as flow from local node 1 to local node 2; default: none; units: none
- N2 the global node ID number for the second local node of the element; positive flow is defined as flow from local node 1 to local node 2; default: none; units: none

 **line29: DX, ED, DZ, POR**

<u>Variable</u>	<u>Description</u>
DX	the flow length of the fluid mechanics finite element; DX is used in the element pressure drop calculation, and in the fluid conduction calculation; default: none; units: m
ED	the equivalent diameter for the finite element; ED is used in the element pressure drop calculation; ED is typically a hydraulic diameter with corrections applied for irregular cross sections; for porous media finite elements (see POR), ED is an effective particle diameter of the porous media; default: none; units: m
DZ	the change in elevation from local node 2 to local node 1 ($z_2 - z_1$); used in the element pressure drop calculation; default: 0.0; units: m
POR	the element porosity for porous media flow; entering a value of POR less than 1.0 flags this element as a porous media element for which pressure drop is calculated based on the friction factor correlation defined by PMCORR in Block 1 data; default: 1.0; units: none

 **line30: FA1, FA2, FAE, VOL**

<u>Variable</u>	<u>Description</u>
FA1	the flow area at local node 1 of the finite element; used in the calculation of the dynamic (velocity) pressure in the momentum equation; for porous media elements, enter the superficial area (the flow area if no porous media were present); default: $(\pi/4) \cdot ED^2$; units: m ²
FA2	the flow area at local node 2 of the finite element; used in the calculation of the dynamic (velocity) pressure in the momentum equation; for porous media elements, enter the superficial area (the flow area if no porous media were present); default: FA1; units: m ²
FAE	the flow area for the finite element; used in the element pressure drop calculation and the fluid conduction calculation; for porous media elements, enter the superficial area (the flow area if no porous media were present); default: $(FA1 + FA2)/2$ or $VOL/(DX \cdot POR)$ if a value for VOL is entered; units: m ²
VOL	the volume of the finite element available for fluid; used in the system pressure calculation for static, closed loop networks, and in the dynamic terms of the fluid thermal energy equation and mechanical energy equation if activated (see variable IDYNFM in Block 1 data); if a value for VOL is entered, FAE will be recalculated as $VOL/(DX \cdot POR)$; default: $FAE \cdot DX \cdot POR$; units: m ³

 **line31: RELRUF, AK12, AK21, ALOD, IALOD, COSA**

<u>Variable</u>	<u>Description</u>
RELRUF	the relative roughness of the finite element surface; relative roughness is defined as the absolute roughness divided by the equivalent diameter (ED); used in the friction factor calculation; if RELRUF = 0.0, a smooth surface is assumed; RELRUF is not used for porous media elements, leave field blank; default: 0.0; units: none
AK12	the added loss coefficient (sometimes referred to as a form loss coefficient or as a K factor) for flow from local node 1 to local node 2 (defined as the positive flow direction); if the element is

specified as a nozzle, AK12 is interpreted as a nozzle discharge coefficient; loss coefficients are typically used to account for the pressure losses associated with sudden expansions or contractions; loss coefficients are based on the node fluid velocity at the smallest flow area (FA1 or FA2); default: 0.0 for uniform area elements (FA1=FA2=FAE) and for porous media elements (POR<1.0); 0.98 for nozzle elements; for elements with gradual and abrupt area changes (expansions and contractions), AK12 is automatically determined based on the values of FA1, FA2, FAE, and DX; units: none

AK21 the added loss coefficient (sometimes referred to as a form loss coefficient or as a K factor) for flow from local node 2 to local node 1 (defined as the negative flow direction); if the element is specified as a nozzle, AK21 is interpreted as a nozzle discharge coefficient; loss coefficients are typically used to account for the pressure losses associated with sudden expansions or contractions; loss coefficients are based on the node fluid velocity at the smallest flow area (FA1 or FA2); default: 0.0 for uniform area elements (FA1=FA2=FAE) and for porous media elements (POR<1.0); 0.98 for nozzle elements; for elements with gradual and abrupt area changes (expansions and contractions), AK12 is automatically determined based on the values of FA1, FA2, FAE, and DX; units: none

ALOD an equivalent L/D to be added to the actual value of L/D (DX/ED) for the calculation of element pressure drop; ALOD is typically used to account for pressure losses associated with pipe bends or valve gates; ALOD is based on the element velocity; default: 0.0; units: none

IALOD this variable is required only for distributed flow manifold elements (see NAME); IALOD is the option number for automatic determination of ALOD for distributed flow manifold elements; SAFSIM will calculate an effective value of ALOD each iteration based on the current flow conditions to account for turning losses, etc.; options: 0 -- none; 1 -- blowing/sucking (transpiration flow); 2 -- Tee, requires two distributed flow manifold elements in series and specification of the variable COSA; 3 -- user-specified option 1 (currently not available); default: 1; units: none

COSA the cosine of the angle between the main header and the feed line (measured clockwise from horizontal; thus, for an angle of 45°, COSA = 0.7071) for use with distributed flow manifold elements with the Tee option (IALOD = 2); COSA not required for other elements; default: 0.0; units: none

line32: QDIR

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

QDIR	the power added directly to the fluid; used to account for heat deposition in the fluid due to such sources as chemical reactions or nuclear heating; default: none; units: W (J/s)
------	---

 the next two lines must be included if the element is specified as a compressor element (see NAME); otherwise, skip these two lines

line33: RSPEED, SPEED

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

RSPEED	the compressors' rated speed; default: none; units: revolutions/s
--------	---

SPEED	the fractional operating speed of the compressor defined as the actual speed divided by the rated speed; default: 1.0; units: none
-------	--

line34: NPNTS, (CFLO, CDP)(k=1,NPNTS)

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

NPNTS	the number of data points in the compressor characteristic curve; this curve specifies the compressor performance as a table of pressure rise (CDP) versus mass flow rate (CFLO) at the compressors' rated speed; adjustments are automatically made to this curve for operational speeds different from the rated speed using the standard pump similarity laws; linear interpolation is used in the table; default: none; units: none
-------	---

CFLO	compressor mass flow rate table entry; the independent variable for the compressor characteristic curve defined as a table of pressure rise versus mass flow rate values; data for the table is entered as NPNTS (CFLO, CDP) pairs; default: none; units: kg/s
------	--

CDP compressor pressure rise table entry; the dependent variable for the compressor characteristic curve defined as a table of pressure rise versus mass flow rate values; data for the table is entered as NPNTS (CFLO, CDP) pairs; default: none; units: Pa

 **line35: FLOES**

<u>Variable</u>	<u>Description</u>
FLOES	the initial value for the super element mass flow rate; default: FLOED (as defined in Block 1 data); units: kg/s

 **line36: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 3 data for each network as specified by NUMNW in Block 1 data; do not include enclosing apostrophes

Block 4, Fluid Mechanics Node Initialization Data:

☞ repeat lines 37 through 39 for each independent flow network

☞ line37: PRESND, TND, XMND(n=1,NGAS)

<u>Variable</u>	<u>Description</u>
PRESND	the default pressure value for all nodes in the network; if a node value is not overwritten on the next line, PRESND is used as the initial node pressure; default: none; units: Pa
TND	the default fluid temperature value for all nodes in the network; if a node value is not overwritten on the next line, TND is used as the initial node temperature; default: none; units: K
XMND	the default mass fraction value for all nodes in the network for each gas of the network (see NGAS in Block 2 data); if a node value is not overwritten on the next line, XMND is used as the initial node mass fraction; one value of XMND must be entered for each gas of the network and the total of all XMND values must equal 1.0; default: none; units: none

☞ line38: PTX, NID, VALUE(s)

☞ repeat line 38 for each pressure, temperature, or mass fraction value for each node of the network to be overwritten

<u>Variable</u>	<u>Description</u>
PTX	a character variable indicating if the node value to be overwritten is pressure ('P'), temperature ('T'), or mass fraction ('X'); maximum of 8 characters allowed; only the first character is used; enclosing apostrophes required; the default values from the previous line are used if not overwritten on this line; default: none; units: none
NID	the global node identification (ID) number for which the PTX value is to be overwritten; default: none; units: none
VALUE	the value of the overwrite variable; if PTX = 'X', then NGAS values must be entered; default: none; units: Pa if PTX = 'P', K if PTX = 'T', and none if PTX = 'X'

line39: END

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 4 data for each network as specified by NUMNW in Block 1 data; do not include enclosing apostrophes

Block 5, Fluid Mechanics Boundary Condition Identification Data:

☞ repeat lines 40 through 41 for each independent flow network

☞ repeat line 40 for each super node with a specified boundary condition

☞ line40: BCKIND, IDS, IDSET

<u>Variable</u>	<u>Description</u>
BCKIND	a character variable that specifies the kind of boundary condition to be applied to super node IDS; maximum of 15 characters allowed; only the first character is used; enclosing apostrophes required; choices are 'PRESSURE', 'MASS FLOW RATE', and 'CLOSED LOOP P'; all networks must have at least one pressure boundary condition; 'CLOSED LOOP P' is a special pressure boundary condition used for flow networks that represent a closed loop and are run in static mode (see <i>variable</i> IDYNFM, line 11); only one 'CLOSED LOOP P' boundary condition allowed per network; default: none; units: none
IDS	the global super node identification (ID) number to which the boundary condition is to apply; default: none; units: none
IDSET	the boundary condition set ID number for this boundary node; more than one boundary node can reference this set ID; any number between 1 and 99999 is a valid set ID; boundary condition values for all set IDs are defined in Block 6 data; default: none; units: none

☞ line41: END

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 5 data for each network as specified by NUMNW in Block 1 data; do not include enclosing apostrophes

Block 6, Fluid Mechanics Boundary Condition Set Identification Data:

☞ repeat lines 42 through 43 for each independent flow network

☞ repeat line 42 for each set identification (ID) number; as many unique set ID numbers can be entered as desired, even if not used; however, every set ID number referenced in Block 5 data must have a corresponding set ID number in this data Block

☞ **line42: IDSET, PMVALUE, TVALUE, XVALUE(n=1,NGAS),POT**

<i>Variable</i>	<i>Description</i>
IDSET	the boundary condition set ID number; any number between 1 and 99999 is a valid set ID; default: none; units: none
PMVALUE	the pressure or mass flow rate value associated with this set ID; variable BCKIND in Block 5 data determines whether this is a pressure or mass flow rate value; default: none; units: Pa for pressure and kg/s for mass flow rate
TVALUE	the temperature value associated with this set ID; all boundary nodes must have an assigned TVALUE; setting TVALUE less than or equal to zero implements a zero heat flux boundary condition; this is useful for closed loop networks which require a 'CLOSED LOOP P' boundary condition applied at one super node, but in which a temperature boundary is not required; zero heat flux conditions can also be used for outflow boundaries where calculation of the exiting fluid temperature is desired; default: none; units: K
XVALUE	the mass fraction value associated with this set ID; all boundary nodes must have an assigned XVALUE; this value is used only if flow is into the adjacent element(s); NGAS values of XVALUE must be entered, one for each network gas defined in Block data 2; the sum of the XVALUES must equal 1.0; default: none; units: none
POT	enter this variable only when a 'CLOSED LOOP P' boundary condition has been specified in Block data 5; POT is the initial value of pressure divided by temperature for closed loop

networks; this value is used to adjust the network pressure based on the ideal gas law and is needed only in static mode (see *variable* IDYNFM) in which the mass accumulation term in the mass continuity equation is omitted; the value of POT can also be determined as the gas constant times the initial network density (because the global density remains constant for a closed loop); default: PMVALUE/TVALUE; units: Pa/K

 **line43: END**

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

END	this line must contain the character variable END in the first three columns to indicate the end of Block 6 data for each network as specified by NUMNW in Block 1 data; do not include enclosing apostrophes
-----	---

Block 7, Function-Controlled Variable Data for Fluid Mechanics:

repeat lines 44 through 45 for each function-controlled variable; this Block, along with function data Blocks, allows the user to define input variables with functional dependence of the form: $y = x$ or $y = f(x)$; y is a function-controlled input variable, x is a signal-variable function, and $f()$ is a specified mathematical function; functions can be specified as functions of other functions if desired; function-controlled variables are updated at each system time step

line44: IFCVID, FCVNAME, IAFN, FCVM

<u>Variable</u>	<u>Description</u>
IFCVID	the desired function-controlled variable identification (ID) number; all function-controlled variable ID numbers are provided in Appendix A ; this ID specifies the input variable that is to be functionally controlled; any variable in Appendix A can be specified; including a variable in this data Block designates that it is updated immediately after the fluid mechanics solution is computed for all networks; function-controlled variables are updated in the order they are listed in the input file; default: none; units: none
FCVNAME	any descriptive phrase of 28 characters or less to identify the function-controlled variable in the output files; default: each function-controlled variable has a unique default assigned to it that is provided in Appendix A; units: none
IAFN	the associated argument function number for this function-controlled variable; any number between 1 and 9999 is a valid argument function number; the value of IAFN must correspond to an IFN value in the function data defined for fluid mechanics, heat transfer, reactor dynamics, or the system; a negative value of IAFN can be entered, indicating the use of the change in the associated IFN function value as the argument function; thus, $y = x^{new} - x^{old}$ or $y = f(x^{new} - x^{old})$, if IAFN is entered as a negative number; x^{old} is the IFN function value at the previous system time step; default: none; units: none

FCVM a multiplier applied to the computed function-controlled variable; default: 1.0; units: none

 **line45: L1, L2, L3**

<u>Variable</u>	<u>Description</u>
L1	the first locator number to define the desired system location of the function-controlled variable; the definition of L1 depends on the selected value of IFCVID and is provided in Appendix A; example -- for function-controlled variable DX, L1 is defined as the associated network number, NW; default: none; units: none
L2	the second locator number to define the desired system location of the function-controlled variable; the definition of L2 depends on the selected value of IFCVID and is provided in Appendix A; example -- for function-controlled variable DX, L2 is defined as the associated element number, JE; default: none; units: none
L3	the third locator number to define the desired system location of the function-controlled variable; the definition of L3 depends on the selected value of IFCVID and is provided in Appendix A; default: none; units: none

 **line46: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 7 data; do not include enclosing apostrophes

Block 8, Function Data for Fluid Mechanics:

☞ repeat line 47 and any associated supplemental line(s) for each function; this Block, along with function-controlled variable data Blocks, allows the user to define input variables with functional dependence of the form: $y = x$ or $y = f(x)$; y is a function-controlled input variable, x is a signal-variable function, and $f()$ is a specified mathematical function; functions can be specified as functions of other functions if desired; functions are updated at each system time step

☞ line47: IFN, IFID, FNAME, FM, FVALUE

<u>Variable</u>	<u>Description</u>
IFN	the user-specified function number; any number between 1 and 9999 is a valid function number; this number may be referenced in function-controlled variable data by the IAFN variable, or by function supplemental <i>variables</i> (see the next line); functions are updated in the order they are listed in the input file, independent of the value of IFN; default: none; units: none
IFID	the function identification (ID) number; all function ID numbers are provided in Appendix B ; any function in Appendix B can be specified; including a function in this data Block designates that it is updated immediately after the fluid mechanics solution is computed for all networks; available functions include dependent output variables (signal-variable functions) such as pressure and temperature, and mathematical functions such as cosine and square root; mathematical functions require specification of associated argument function(s) on the following line; default: none; units: none
FNAME	any descriptive phrase of 28 characters or less to identify the function in the output files; default: each function has a unique default assigned to it that is provided in Appendix B; units: none
FM	a multiplier applied to the computed function; default: 1.0; units: none
FVALUE	the initial value of the function; default: 0.0; units: none

 **line48:** Supplemental *variables* associated with the function

 each function may require supplemental lines to fully specify the function; Appendix B contains the supplemental *variables* required by each function; care should be taken to ensure the proper supplemental *variables* are entered in the correct order for each function

 **line49: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 8 data; do not include enclosing apostrophes

Block 9, Structure Heat Transfer Material Property Data:

☞ repeat lines 50 through 60 for each structure material; the materials are internally numbered sequentially, starting with 1 for the first material listed; material property data can be included but not referenced if desired

☞line50: MATNAME, DENHS

<u>Variable</u>	<u>Description</u>
MATNAME	any descriptive name of 12 characters or less to identify the material; this name is referenced by the conduction heat transfer input in Block 10; enclosing apostrophes required; default: none; units: none
DENHS	the density of the material; default: none; units: kg/m ³

☞line51: MSHTYPE, NP, SHM

<u>Variable</u>	<u>Description</u>
MSHTYPE	the material specific heat data entry type; five types are available for entering material property data: 1 -- ntable (non-uniform temperature data spacing), 2 -- utable (uniform temperature data spacing; this table is faster but requires the temperature data entries to be uniformly spaced), 3 -- polynomial of order NP, 4 -- power law, and 5 -- constant; linear interpolation is used for both tables; default: none; units: none
NP	if MSHTYPE = 1 or 2, NP is the number of data points in the specific heat versus temperature table; if MSHTYPE = 3, NP is the polynomial order; NP is not used for MSHTYPE = 4 or 5 (leave field blank); default: none; units: none
SHM	a multiplier applied to the specific heat; default: 1.0; units: none

☞ enter the next line only if MSHTYPE = 1 or 2 (non-uniform or uniform linear interpolation tables)

 **line52: (T, SH)(k=1, NP)**

<u>Variable</u>	<u>Description</u>
T	temperature table entry; the independent variable for the specific heat versus temperature table; data for the table is entered as NP (T, SH) pairs; default: none; units: K
SH	specific heat table entry; the dependent variable for the specific heat versus temperature table; data for the table is entered as NP (T, SH) pairs; default: none; units: J/kg·K

 enter the next line only if MSHTYPE = 3 (polynomial)

 **line53: SHC(k=1, NP+1)**

<u>Variable</u>	<u>Description</u>
SHC	the NP+1 constants required to specify specific heat as a function of temperature using an NP order polynomial; the form of the polynomial: specific heat = SHC1 + SHC2·T + SHC3·T ² + ... ; default: none; specific heat units: J/kg·K, temperature units: K

 enter the next line only if MSHTYPE = 4 (power law)

 **line54: SHC(k=1,3)**

<u>Variable</u>	<u>Description</u>
SHC	the 3 constants required to specify specific heat as a power law function of temperature; the form of the power law: specific heat = SHC1·T ^{SHC2} + SHC3; default: none; specific heat units: J/kg·K, temperature units: K

 enter the next line only if MSHTYPE = 5 (constant)

 **line55: SH**

<u>Variable</u>	<u>Description</u>
SH	the constant specific heat value; default: none; units: J/kg·K

line56: MCDTYPE, NP, CDM

<u>Variable</u>	<u>Description</u>
MCDTYPE	the material conductivity data entry type; five types are available for entering material property data: 1 -- ntable (non-uniform temperature data spacing), 2 -- utable (uniform temperature data spacing; this table is faster but requires the temperature data entries to be uniformly spaced), 3 -- polynomial of order NP, 4 -- power law, and 5 -- constant; linear interpolation is used for both tables; default: none; units: none
NP	if MCDTYPE = 1 or 2, NP is the number of data points in the conductivity versus temperature table; if MCDTYPE = 3, NP is the polynomial order; NP is not used for MCDTYPE = 4 or 5 (leave field blank); default: none; units: none
CDM	a multiplier applied to the conductivity; default: 1.0; units: none

☞ enter the next line only if MCDTYPE = 1 or 2 (non-uniform or uniform linear interpolation tables)

line57: (T, CD)(k=1,NP)

<u>Variable</u>	<u>Description</u>
T	temperature table entry; the independent variable for the conductivity versus temperature table; data for the table is entered as NP (T, CD) pairs; default: none; units: K
CD	conductivity table entry; the dependent variable for the conductivity versus temperature table; data for the table is entered as NP (T, CD) pairs; default: none; units: W/m-K

☞ enter the next line only if MCDTYPE = 3 (polynomial)

line58: CDC(k=1,NP+1)

<u>Variable</u>	<u>Description</u>
CDC	the NP+1 constants required to specify conductivity as a function of temperature using an NP order polynomial; the

form of the polynomial: conductivity = CDC1 + CDC2·T + CDC3·T² + ... ; default: none; conductivity units: W/m·K, temperature units: K

 enter the next line only if MCDTYPE = 4 (power law)

 **line59: CDC(k=1,3)**

<u>Variable</u>	<u>Description</u>
CDC	the 3 constants required to specify conductivity as a power law function of temperature; the form of the power law: conductivity = CDC1·T ^{CDC2} + CDC3; default: none; conductivity units: W/m·K, temperature units: K

 enter the next line only if MCDTYPE = 5 (constant)

 **line60: CD**

<u>Variable</u>	<u>Description</u>
CD	the constant conductivity value; default: none; units: W/m·K

 **line61: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 9 data; do not include enclosing apostrophes

Block 10, Structure Heat Transfer Finite Element and Node Data:

☞ repeat lines 62 through 84 for each heat transfer structure

☞ **line62: KH, NAMEHS, COPIES, QPPPM, KHDYN**

<u>Variable</u>	<u>Description</u>
KH	the heat transfer structure identification (ID) number; structures must be numbered sequentially starting at 1, but can be listed non-sequentially in the input file if desired; default: none; units: none
NAMEHS	any descriptive name of up to 24 characters to identify the structure in the output files; enclosing apostrophes required; default: 'HEAT TRANSFER STRUCTURE'; units: none
COPIES	the number of copies of the structure that contribute heat to a convectively coupled fluid mechanics finite element; this serves as a multiplier on the heat transferred convectively (q); thus, the total heat transferred to all coupled fluid mechanics elements is determined as q*COPIES; this is useful if several identical structures are coupled to the same fluid mechanics element and conduction in only one of the structures is modeled; default: 1.0; units: none
QPPPM	a multiplier applied to all of the element QPPPs (volumetric heat generation rates entered on line72) for this structure; default: 1.0; units: none, or the product of QPPPM and QPPP must have units of W/m ³
KHDYN	an on/off switch for the dynamic (unsteady-state) temperature term; if KHDYN = 1, the dynamic term is included; if KHDYN = 0, the dynamic term is not included; default: 1; units: none

☞ **line63: NEH, DTMINH, FETS, NPRNTH**

<u>Variable</u>	<u>Description</u>
NEH	the number of finite elements in this structure; SAFSIM internally numbers all elements consecutively, with element

number 1 adjacent to end 1 and element number NEH adjacent to end 2 of the structure; default: none; units: none

DTMINH the minimum-allowed desired time step for this structure; the current value of the system time step is used as the maximum-allowed structure time step; default: 0.001, units: s

FETS the fraction of the explicit time step limit for determining the actual structure time step; an explicit time step limit is determined at each time step iteration for each structure; this limit would be the maximum-allowed time step, based on stability requirements, if an explicit solution scheme were employed; SAFSIM uses an adaptive semi-implicit solution scheme that allows larger time steps, but the explicit time step limit provides a reasonable time step estimate; this limit is multiplied by FETS to obtain the actual time step; if desired, FETS can be greater than 1.0, but the structure time step is limited to the current value of the system time step; also, FETS can equal 0.0 as long as DTMINH is greater than 0.0; default: 0.5; units: none

NPRNTH the print frequency for this heat transfer structure; heat transfer output is written to the output file every NPRNTH heat transfer structure time steps; this output is in addition to that generated every PRNTINT seconds as specified in Block 1 data; default: 10000; units: none

 **line64: GEOM, DIM1, DIM2, DIM3, DIM4**

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

GEOM	the heat transfer structure geometry descriptor; this activates automatic calculation of the element geometry (conduction areas and volumes) based on the values of DIM1, DIM2, DIM3, and DIM4; the available choices are 'CYLINDRICAL', 'SPHERICAL', 'RECTANGULAR', 'LINEAR', and 'NONE'; only the first character is used; enclosing apostrophes required; 'CYLINDRICAL' provides calculation of element area based on a log-mean average of the node areas; 'SPHERICAL' provides calculation of element area based on a square root average of the node areas; 'RECTANGULAR' provides calculation of element area based on a uniform area; 'LINEAR' provides calculation of element area based on linear interpolation between the areas at the two ends of the structure; 'NONE' indicates that no geometry is selected and
-------------	---

that all values of AREA and VOLH must be specified; default: 'NONE'; units: none

DIM1 the first dimension required for use of the automatic element geometry option as selected by GEOM; the interpretation of DIM1 depends on GEOM and is provided in the following table; default: provided in the following table; units: m, m², or radians depending on GEOM selection

DIM2 the second dimension required for use of the automatic element geometry option as selected by GEOM; the interpretation of DIM2 depends on GEOM and is provided in the following table; default: provided in the following table; units: m, m², or radians depending on GEOM selection

DIM3 the third dimension required for use of the automatic element geometry option as selected by GEOM; the interpretation of DIM3 depends on GEOM and is provided in the following table; default: 0.0 or none; units: m

DIM4 the fourth dimension required for use of the automatic element geometry option as selected by GEOM; the interpretation of DIM4 depends on GEOM and is provided in the following table; default: none; units: m

Required Input for Automatic Geometry Options

GEOM	DIM1	DIM2	DIM3	DIM4
'CYLINDRICAL'	$\Delta\theta$ default: 2π units: radian	Height default: 1.0 units: m	Inner Radius default: 0.0 units: m	Outer Radius default: none units: m
'SPHERICAL'	$\Delta\theta$ default: 2π units: radian	$\Delta\phi$ default: π units: radian	Inner Radius default: 0.0 units: m	Outer Radius default: none units: m
'RECTANGULAR'	Width default: none units: m	Height default: none units: m	end1 position default: 0.0 units: m	end2 position default: none units: m
'LINEAR'	end1 Area default: none units: m ²	end2 Area default: none units: m ²	end1 position default: 0.0 units: m	end2 position default: none units: m
'NONE'	- not used leave blank	- not used leave blank	end1 position default: 0.0 units: m	end2 position default: none units: m

line65: BCH1

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

BCH1	a character variable that specifies the kind of boundary condition to be applied to end 1 of the structure; only the first character is used; enclosing apostrophes required; choices are 'TEMPERATURE', 'FLUX', and 'CONVECTION'; default: 'FLUX'; units: none
------	---

☞ enter the next line only if BCH1 = 'TEMPERATURE'

line66: TBC1

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

TBC1	the temperature boundary condition for end 1 of the structure; default: none; units: K
------	--

☞ enter the next line only if BCH1 = 'FLUX'

line67: FLUXBC1, AFLUX1

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

FLUXBC1	the heat flux boundary condition for end 1 of the structure; a positive value of FLUXBC1 indicates heat <u>out</u> of the structure; default: 0.0; units: W/m ²
---------	--

AFLUX1	the available heat flux surface area for end 1 of the structure; default: calculated end 1 area based on GEOM selection; units: m ²
--------	--

☞ enter the next 3 lines only if BCH1 = 'CONVECTION'

line68: NHBC1

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

NHBC1	the number of convective boundary conditions applied to end 1 of the structure; default: 1; units: none
-------	---

☞ repeat the next 2 lines for each 'CONVECTION' boundary condition for end 1 as specified by NHBC1

📎line69: NWH1, JEH1

<u>Variable</u>	<u>Description</u>
NWH1	the ID number of the fluid mechanics network to which the convective boundary at end 1 is coupled; default: none; units: none
JEH1	the ID number of the fluid mechanics element of network NWH1 to which the convective boundary at end 1 is coupled; default: none; units: none

📎line70: AEX, EDH, IHTCLAM, IHTCTUR, TURLAMH, AUXL, AUXT

<u>Variable</u>	<u>Description</u>
AEX	the available surface area for the convection boundary condition at end 1; default: calculated end 1 area based on GEOM selection; units: m ²
EDH	the equivalent diameter for the end 1 boundary convection heat transfer calculation; default: ED (from Block 3 data) for the coupled fluid mechanics element, as specified by JEH1; units: m
IHTCLAM	the ID number of the desired laminar flow heat transfer coefficient correlation; all heat transfer coefficient correlation ID numbers are provided in Appendix C ; default: 2; units: none
IHTCTUR	the ID number of the desired turbulent flow heat transfer coefficient correlation; all heat transfer coefficient correlation ID numbers are provided in Appendix C ; default: 6; units: none
TURLAMH	the Reynolds number at which transition from laminar to turbulent flow occurs for heat transfer coefficient calculations; default: TURLAM (from Block 1 data); units: none
AUXL	auxiliary data for the selected laminar flow heat transfer coefficient correlation (IHTCLAM); some correlations require auxiliary data, such as an entrance length; required auxiliary

data for each correlation is provided in Appendix C; default: 1.0; units: dependent on selected correlation

AUXT auxiliary data for the selected turbulent flow heat transfer coefficient correlation (IHTCTUR); some correlations require auxiliary data, such as an entrance length; required auxiliary data for each correlation is provided in Appendix C; default: 1.0; units: dependent on selected correlation

repeat lines 71 through 77 for each conduction finite element of this structure as specified by NEH; SAFSIM internally numbers elements consecutively, with element number 1 adjacent to end 1 and element number NEH adjacent to end 2 of the structure

line71: DXH, AREA, VOLH

<u>Variable</u>	<u>Description</u>
DXH	the conduction length of the finite element; default: uniform element length based on $(DIM4 - DIM3)/NEH$; units: m
AREA	the conduction cross-sectional area for the finite element; default: based on GEOM selection; units: m^2
VOLH	the volume of the finite element; default: based on GEOM selection; units: m^3

line72: MAT, QPPP

<u>Variable</u>	<u>Description</u>
MAT	a descriptive name of 12 characters or less to identify the material for this element; this name must exactly match one of the names provided in Block 9 material property data; enclosing apostrophes required; default: none; units: none
QPPP	the volumetric heat generation rate for this element; a positive value of QPPP indicates heat added to the structure; default: 0.0; units: W/m^3 , or the product of QPPPM and QPPP must have units of W/m^3

line73: NFLUXA

<u>Variable</u>	<u>Description</u>
NFLUXA	the desired number of heat flux surfaces for this element; default: 0; units: none

☞ repeat the next line for each heat flux surface for this element as specified by NFLUXA

line74: ASURF, QPP

<u>Variable</u>	<u>Description</u>
ASURF	the available area of the heat flux surface; default: none; units: m ²
QPP	the heat flux value for the surface; a positive value of QPP indicates heat <u>out</u> of the structure; default: none; units: W/m ²

line75: NEX

<u>Variable</u>	<u>Description</u>
NEX	the desired number of convective exchange surfaces for this element; default: 0; units: none

☞ repeat the next 2 lines for each convective exchange surface for this element as specified by NEX

line76: NWHE, JEHE

<u>Variable</u>	<u>Description</u>
NWHE	the ID number of the fluid mechanics network to which the convective exchange surface is coupled; default: none; units: none
JEHE	the ID number of the fluid mechanics element of network NWHE to which the convective exchange surface is coupled; default: none; units: none

📎line77: AEX, EDH, IHTCLAM, IHTCTUR, TURLAMH, AUXL, AUXT

<u>Variable</u>	<u>Description</u>
AEX	the available area for the convective exchange surface; default: none; units: m ²
EDH	the equivalent diameter for the convection heat transfer calculation; default: ED (from Block 3 data) for the coupled fluid mechanics element, as specified by JEHE; units: m
IHTCLAM	the ID number of the desired laminar flow heat transfer coefficient correlation; all heat transfer coefficient correlation ID numbers are provided in Appendix C ; default: 2; units: none
IHTCTUR	the ID number of the desired turbulent flow heat transfer coefficient correlation; all heat transfer coefficient correlation ID numbers are provided in Appendix C ; default: 6; units: none
TURLAMH	the Reynolds number at which transition from laminar to turbulent flow occurs for heat transfer coefficient calculations; default: TURLAM (from Block 1 data); units: none
AUXL	auxiliary data for the selected laminar flow heat transfer coefficient correlation (IHTCLAM); some correlations require auxiliary data, such as an entrance length; required auxiliary data for each correlation is provided in Appendix C; default: 1.0; units: dependent on selected correlation
AUXT	auxiliary data for the selected turbulent flow heat transfer coefficient correlation (IHTCTUR); some correlations require auxiliary data, such as an entrance length; required auxiliary data for each correlation is provided in Appendix C; default: 1.0; units: dependent on selected correlation

📎line78: BCH2

<u>Variable</u>	<u>Description</u>
BCH2	a character variable that specifies the kind of boundary condition to be applied to end 2 of the structure; only the first character is used; enclosing apostrophes required; choices are

'TEMPERATURE', 'FLUX', and 'CONVECTION'; default:
'FLUX'; units: none

☞ enter the next line only if BCH2 = 'TEMPERATURE'

☞ **line79: TBC2**

<u>Variable</u>	<u>Description</u>
TBC2	the temperature boundary condition for end 2 of the structure; default: none; units: K

☞ enter the next line only if BCH2 = 'FLUX'

☞ **line80: FLUXBC2, AFLUX2**

<u>Variable</u>	<u>Description</u>
FLUXBC2	the heat flux boundary condition for end 2 of the structure; a positive value of FLUXBC2 indicates heat <u>out</u> of the structure; default: 0.0; units: W/m ²
AFLUX2	the available heat flux surface area for end 2 of the structure; default: calculated end 1 area based on GEOM selection; units: m ²

☞ enter the next 3 lines only if BCH2 = 'CONVECTION'

☞ **line81: NHBC2**

<u>Variable</u>	<u>Description</u>
NHBC2	the number of convective boundary conditions applied to end 2 of the structure; default: 1; units: none

☞ repeat the next 2 lines for each 'CONVECTION' boundary condition for end
2 as specified by NHBC2

📌 **line82: NWH2, JEH2**

<u>Variable</u>	<u>Description</u>
NWH2	the ID number of the fluid mechanics network to which the convective boundary at end 2 is coupled; default: none; units: none
JEH2	the ID number of the fluid mechanics element of network NWH2 to which the convective boundary at end 2 is coupled; default: none; units: none

📌 **line83: AEX, EDH, IHTCLAM, IHTCTUR, TURLAMH, AUXL, AUXT**

<u>Variable</u>	<u>Description</u>
AEX	the available surface area for the convection boundary condition at end 2; default: calculated end 2 area based on GEOM selection; units: m ²
EDH	the equivalent diameter for the end 2 boundary convection heat transfer calculation; default: ED (from Block 3 data) for the coupled fluid mechanics element, as specified by JEH2; units: m
IHTCLAM	the ID number of the desired laminar flow heat transfer coefficient correlation; all heat transfer coefficient correlation ID numbers are provided in Appendix C ; default: 2; units: none
IHTCTUR	the ID number of the desired turbulent flow heat transfer coefficient correlation; all heat transfer coefficient correlation ID numbers are provided in Appendix C ; default: 6; units: none
TURLAMH	the Reynolds number at which transition from laminar to turbulent flow occurs for heat transfer coefficient calculations; default: TURLAM (from Block 1 data); units: none
AUXL	auxiliary data for the selected laminar flow heat transfer coefficient correlation (IHTCLAM); some correlations require auxiliary data, such as an entrance length; required auxiliary data for each correlation is provided in Appendix C; default: 1.0; units: dependent on selected correlation

AUXT auxiliary data for the selected turbulent flow heat transfer coefficient correlation (IHTCTUR); some correlations require auxiliary data, such as an entrance length; required auxiliary data for each correlation is provided in Appendix C; default: 1.0; units: dependent on selected correlation

 **line84: TNH(k=1,NEH+1)**

<u>Variable</u>	<u>Description</u>
TNH	the initial temperature for each of the NEH+1 nodes of the heat transfer structure; default: THD (from data Block 1); if a value is entered for the first node, it becomes the new default for all remaining nodes; units: K

 **line85: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 10 data; do not include enclosing apostrophes

Block 11, Function-Controlled Variable Data for Heat Transfer:

repeat lines 86 through 87 for each function-controlled variable; this Block, along with function data Blocks, allows the user to define input variables with functional dependence of the form: $y = x$ or $y = f(x)$; y is a function-controlled input variable, x is a signal-variable function, and $f()$ is a specified mathematical function; functions can be specified as functions of other functions if desired; function-controlled variables are updated at each system time step

line86: IFCVID, FCVNAME, IAFN, FCVM

<u>Variable</u>	<u>Description</u>
IFCVID	the desired function-controlled variable identification (ID) number; all function-controlled variable ID numbers are provided in Appendix A ; this ID specifies the input variable that is to be functionally controlled; any variable in Appendix A can be specified; including a variable in this data Block designates that it is updated immediately after the heat transfer solution is computed for all structures; function-controlled variables are updated in the order they are listed in the input file; default: none; units: none
FCVNAME	any descriptive phrase of 28 characters or less to identify the function-controlled variable in the output files; default: each function-controlled variable has a unique default assigned to it that is provided in Appendix A; units: none
IAFN	the associated argument function number for this function-controlled variable; any number between 1 and 9999 is a valid argument function number; the value of IAFN must correspond to an IFN value in the function data defined for fluid mechanics, heat transfer, reactor dynamics, or the system; a negative value of IAFN can be entered, indicating the use of the change in the associated IFN function value as the argument function; thus, $y = x^{new} - x^{old}$ or $y = f(x^{new} - x^{old})$, if IAFN is entered as a negative number; x^{old} is the IFN function value at the previous system time step; default: none; units: none

FCVM a multiplier applied to the computed function-controlled variable; default: 1.0; units: none

 **line87: L1, L2, L3**

<u>Variable</u>	<u>Description</u>
L1	the first locator number to define the desired system location of the function-controlled variable; the definition of L1 depends on the selected value of IFCVID and is provided in Appendix A; example -- for function-controlled variable DXH, L1 is defined as the associated structure number, KH; default: none; units: none
L2	the second locator number to define the desired system location of the function-controlled variable; the definition of L2 depends on the selected value of IFCVID and is provided in Appendix A; example -- for function-controlled variable DXH, L2 is defined as the associated element number; default: none; units: none
L3	the third locator number to define the desired system location of the function-controlled variable; the definition of L3 depends on the selected value of IFCVID and is provided in Appendix A; default: none; units: none

 **line88: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 11 data; do not include enclosing apostrophes

Block 12, Function Data for Heat Transfer:

repeat line 89 and any associated supplemental line(s) for each function; this Block, along with function-controlled variable data Blocks, allows the user to define input variables with functional dependence of the form: $y = x$ or $y = f(x)$; y is a function-controlled input variable, x is a signal-variable function, and $f()$ is a specified mathematical function; functions can be specified as functions of other functions if desired; functions are updated at each system time step

line89: IFN, IFID, FNAME, FM, FVALUE

<u>Variable</u>	<u>Description</u>
IFN	the user-specified function number; any number between 1 and 9999 is a valid function number; this number may be referenced in function-controlled variable data by the IAFN variable, or by function supplemental <i>variables</i> (see the next line); functions are updated in the order they are listed in the input file, independent of the value of IFN; default: none; units: none
IFID	the function identification (ID) number; all function ID numbers are provided in Appendix B ; any function in Appendix B can be specified; including a function in this data Block designates that it is updated immediately after the heat transfer solution is computed for all networks; available functions include dependent output variables (signal-variable functions) such as pressure and temperature, and mathematical functions such as cosine and square root; mathematical functions require specification of associated argument function(s) on the following line; default: none; units: none
FNAME	any descriptive phrase of 28 characters or less to identify the function in the output files; default: each function has a unique default assigned to it that is provided in Appendix B; units: none
FM	a multiplier applied to the computed function; default: 1.0; units: none
FVALUE	the initial value of the function; default: 0.0; units: none

 **line90: Supplemental *variables* associated with the function**

 each function may require supplemental lines to fully specify the function; Appendix B contains the supplemental *variables* required by each function; care should be taken to ensure the proper supplemental *variables* are entered in the correct order for each function

 **line91: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 12 data; do not include enclosing apostrophes

Block 13, Reactor Dynamics Data:

☞ repeat lines 92 through 111 for each nuclear reactor included in the system

☞ line92: KR, RXNAME, KSOLVE

<u>Variable</u>	<u>Description</u>
KR	the reactor identification number; reactors must be numbered sequentially starting at 1, but can be listed non-sequentially in the input file if desired; default: none; units: none
RXNAME	any descriptive name of up to 24 characters to identify the reactor in the output files; enclosing apostrophes required; default: 'Nuclear Reactor'; units: none
KSOLVE	specifies the numerical method for integration of the reactor dynamics equations; available options: 1 -- fifth-order Runge-Kutta-Felberg method using fourth- and fifth-order solutions to estimate the truncation error for adaptive time step control; 2 -- first-order Euler's method using step doubling to estimate the truncation error for adaptive time step control; default: 1; units: none

☞ line93: GENTIME, SOURCE, RDN

<u>Variable</u>	<u>Description</u>
GENTIME	the effective neutron generation time, defined as the prompt neutron lifetime divided by the effective neutron multiplication factor (k_{eff}); default: 1.0E-4; units: s
SOURCE	the added neutron source term for the reactor; default: 0.0; units: W/s or any user-selected unit consistent with the other reactor dynamics data
RDN	the initial value of the neutron power (or neutron density,...); default: 1.0E-5; units: W or any user-selected unit consistent with the other reactor dynamics data

line94: DELTR, DTRDMIN, DTRDMAX, DRERRR, NPRNTRD

<u>Variable</u>	<u>Description</u>
DELTR	the initial time step for the reactor dynamics solution; default: GENTIME; units: s
DTRDMIN	the minimum-allowed time step for the reactor dynamics solution; default: 0.1·GENTIME; units: s
DTRDMAX	the maximum-allowed time step for the reactor dynamics solution; default: 100·GENTIME; units: s
DRERRR	the desired relative truncation error for the reactor dynamics solution; the Runge-Kutta-Felhberg option (KSOLVE = 1) determines truncation error based on the difference between a fourth- and fifth-order solution; the Euler option (KSOLVE = 2) determines truncation error based on the difference between solutions using full and one-half time steps; default: 1.0E-6; units: none
NPRNTRD	the print frequency for this reactor; reactor dynamics output is written to the output file every NPRNTRD reactor dynamics time steps; this output is in addition to that generated every PRNTINT seconds as specified in Block 1 data; default: 10000; units: none

line95: RDST, AUXRD(kaux=1,3)

<u>Variable</u>	<u>Description</u>
RDST	the start time for initiation of the reactor dynamics solution; solution of the reactor dynamics equations does not begin until the problem time reaches RDST; default: 0.0; units: s
AUXRD	auxiliary data for reactor dynamics programmed reactivity algorithms; three values of AUXRD can be entered for use in user-defined programmed reactivity algorithms; these algorithms can be provided by the user as FORTRAN subroutines if desired and are useful for implementing complex reactor control laws not provided by SAFSIM; user-defined control laws are selected by variables KOPTPR and NPRD; AUXRD values can be function controlled; default: 0.0; units: use dependent

line96: NDNG

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

NDNG	the number of delayed neutron precursor groups; default: 6; units: none
------	---

☞ enter the next 2 lines only if a non-zero value for NDNG is entered in the previous line; if NDNG is left blank, the 6 default values of XLAM and BETAG are used

line97: XLAM(k=1,NDNG)

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

XLAM	the delayed neutron precursor decay constants; NDNG values of XLAM must be entered; default: 0.0124, 0.0305, 0.111, 0.301, 1.14, 3.01; units: s ⁻¹
------	---

line98: BETAG(k=1,NDNG)

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

BETAG	the effective delayed neutron fractions; NDNG values of BETAG must be entered; default: 0.00021, 0.00142, 0.00127, 0.00257, 0.00075, 0.00027; units: none
-------	---

line99: CDN(k=1,NDNG)

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

CDN	the initial delayed neutron precursor concentrations; NDNG values of CDN must be entered; this line must be entered unless NDNG = 0; default: steady-state concentration based on initial value of RDN; units: W or any user-selected unit consistent with the other reactor dynamics data
-----	--

line100: NDNG

<u>Variable</u>	<u>Description</u>
-----------------	--------------------

NDHG	the number of decay heat precursor groups; default: 11; units: none
------	---

☞ enter the next 2 lines only if a non-zero value for NDHG is entered in the previous line; if NDHG is left blank, the 11 default values of XLAMH and EFRACG are used

📎line101: XLAMH(k=1,NDHG)

<u>Variable</u>	<u>Description</u>
XLAMH	the decay heat precursor decay constants; NDHG values of XLAMH must be entered; default: 1.772, 0.5774, 0.06743, 0.006214, 4.739E-4, 4.81E-5, 5.344E-6, 5.726E-7, 1.036E-7, 2.959E-8, 7.585E-10; units: s ⁻¹

📎line102: EFRACG(k=1,NDHG)

<u>Variable</u>	<u>Description</u>
EFRACG	the decay heat precursor fractions; NDHG values of EFRACG must be entered; default: 0.00299, 0.00825, 0.0155, 0.01935, 0.01165, 0.00645, 0.00231, 0.00164, 0.00085, 0.00043, 0.00057; units: none

📎line103: CDH(k=1,NDHG)

<u>Variable</u>	<u>Description</u>
CDH	the initial decay heat precursor concentrations; NDHG values of CDH must be entered; this line must be entered unless NDHG = 0; default: steady-state concentration based on initial value of RDN; units: W-s or any user-selected unit consistent with the other reactor dynamics data

📎line104: NREAC

<u>Variable</u>	<u>Description</u>
NREAC	the number of differential reactivity feedback terms ($\Delta\rho = \rho^{\text{new}} - \rho^{\text{old}}$) to be used for this reactor; these terms couple the reactor dynamics solution to the fluid mechanics and heat transfer solutions; these terms can be function controlled; default: 1; units: none

☞ repeat the next line for each NREAC differential reactivity feedback term

line105: FBNAME, DREACI

<u>Variable</u>	<u>Description</u>
FBNAME	any descriptive name of 12 characters or less to identify the differential reactivity feedback term (DREAC) in the output files; enclosing apostrophes required; default: 'FEEDBACK'; units: none
DREACI	the initial value for the integration of the differential reactivity feedback term, DREAC (DREAC is function controllable); thus: $DREACI = DREACI + (DREAC^{new} - DREAC^{old})$; default: 0.0; units: none ($\Delta k/k$)

line106: KOPTPR, NPPR

<u>Variable</u>	<u>Description</u>
KOPTPR	the programmed reactivity ($d\rho/dt$) option number; 4 options are available: 1 -- table, 2 -- polynomial of order NPPR, 3 -- constant, and 4 -- user-defined via FORTRAN subroutines UPRIN, UPRINIT, and UPR; default: none; units: none
NPPR	if KOPTPR = 1, NPPR is the number of data points in the programmed reactivity versus time table; if KOPTPR = 2, NPPR is the polynomial order; NPPR is not used for KOPTPR = 3; if KOPTPR = 4, NPPR is the user-defined programmed reactivity selection number; default: none; units: none

☞ enter the next line only if KOPTPR = 1 (linear interpolation table)

line107: (t,PR)(k=1,NPPR)

<u>Variable</u>	<u>Description</u>
t	time table entry; the independent variable for the programmed reactivity versus time table; data for the table is entered as NPPR (t, PR) pairs; time t is relative to the reactor dynamics start time (RDST); default: none; units: s
PR	programmed reactivity table entry; the dependent variable for the programmed reactivity ($d\rho/dt$) versus time table; data for the table is entered as NPPR (t, PR) pairs; this option is useful for defining reactor scrams; default: none; units: s^{-1}

☞ enter the next line only if KOPTPR = 2 (polynomial)

☞ **line108: PRC(k=1,NPPR)**

<u>Variable</u>	<u>Description</u>
PRC	the NPPR+1 constants required to specify programmed reactivity (dp/dt) as a function of time using an NPPR order polynomial; the form of the polynomial: $PR = PRC1 + PRC2t + PRC3t^2 + \dots$; time t is relative to the reactor dynamics start time (RDST); default: none; units: s^{-1}

☞ enter the next line only if KOPTPR = 3 (constant)

☞ **line109: PR**

<u>Variable</u>	<u>Description</u>
PR	the constant programmed reactivity value (dp/dt); default: none; units: s^{-1}

☞ enter the next line(s) only if KOPTPR = 4 (user-defined)

☞ **line110: user-defined via subroutines UPRIN, UPRINT, and UPR**

☞ sample subroutines are provided in Appendix D

☞ **line111: PRI**

<u>Variable</u>	<u>Description</u>
PRI	the initial value for the programmed reactivity; this is used to initialize the time integration of the programmed reactivity; default: steady-state value based on initial values of RDN and SOURCE; units: none ($\Delta k/k$)

☞ **line112: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 13 data; do not include enclosing apostrophes

Block 14, Function-Controlled Variable Data for Reactor Dynamics:

repeat lines 113 through 114 for each function-controlled variable; this Block, along with function data Blocks, allows the user to define input variables with functional dependence of the form: $y = x$ or $y = f(x)$; y is a function-controlled input variable, x is a signal-variable function, and $f()$ is a specified mathematical function; functions can be specified as functions of other functions if desired; function-controlled variables are updated at each system time step

line113: IFCVID, FCVNAME, IAFN, FCVM

<u>Variable</u>	<u>Description</u>
IFCVID	the desired function-controlled variable identification (ID) number; all function-controlled variable ID numbers are provided in Appendix A ; this ID specifies the input variable that is to be functionally controlled; any variable in Appendix A can be specified; including a variable in this data Block designates that it is updated immediately after the reactor dynamics solution is computed for all reactors; function-controlled variables are updated in the order they are listed in the input file; default: none; units: none
FCVNAME	any descriptive phrase of 28 characters or less to identify the function-controlled variable in the output files; default: each function-controlled variable has a unique default assigned to it that is provided in Appendix A; units: none
IAFN	the associated argument function number for this function-controlled variable; any number between 1 and 9999 is a valid argument function number; the value of IAFN must correspond to an IFN value in the function data defined for fluid mechanics, heat transfer, reactor dynamics, or the system; a negative value of IAFN can be entered, indicating the use of the change in the associated IFN function value as the argument function; thus, $y = x^{\text{new}} - x^{\text{old}}$ or $y = f(x^{\text{new}} - x^{\text{old}})$, if IAFN is entered as a negative number; x^{old} is the IFN function value at the previous system time step; default: none; units: none

FCVM a multiplier applied to the computed function-controlled variable; default: 1.0; units: none

 **line14: L1, L2, L3**

<u>Variable</u>	<u>Description</u>
L1	the first locator number to define the desired system location of the function-controlled variable; the definition of L1 depends on the selected value of IFCVID and is provided in Appendix A; example -- for function-controlled variable DREAC, L1 is defined as the associated reactor number, KR; default: none; units: none
L2	the second locator number to define the desired system location of the function-controlled variable; the definition of L2 depends on the selected value of IFCVID and is provided in Appendix A; example -- for function-controlled variable DREAC, L2 is defined as the associated feedback term number; default: none; units: none
L3	the third locator number to define the desired system location of the function-controlled variable; the definition of L3 depends on the selected value of IFCVID and is provided in Appendix A; default: none; units: none

 **line15: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 14 data; do not include enclosing apostrophes

Block 15, Function Data for Reactor Dynamics:

repeat line 116 and any associated supplemental line(s) for each function; this Block, along with function-controlled variable data Blocks, allows the user to define input variables with functional dependence of the form: $y = x$ or $y = f(x)$; y is a function-controlled input variable, x is a signal-variable function, and $f()$ is a specified mathematical function; functions can be specified as functions of other functions if desired; functions are updated at each system time step

line116: IFN, IFID, FNAME, FM, FVALUE

<u>Variable</u>	<u>Description</u>
IFN	the user-specified function number; any number between 1 and 9999 is a valid function number; this number may be referenced in function-controlled variable data by the IAFN variable, or by function supplemental <i>variables</i> (see the next line); functions are updated in the order they are listed in the input file, independent of the value of IFN; default: none; units: none
IFID	the function identification (ID) number; all function ID numbers are provided in Appendix B ; any function in Appendix B can be specified; including a function in this data Block designates that it is updated immediately after the reactor dynamics solution is computed for all reactors; available functions include dependent output variables (signal-variable functions) such as pressure and temperature, and mathematical functions such as cosine and square root; mathematical functions require specification of associated argument function(s) on the following line; default: none; units: none
FNAME	any descriptive phrase of 28 characters or less to identify the function in the output files; default: each function has a unique default assigned to it that is provided in Appendix B; units: none
FM	a multiplier applied to the computed function; default: 1.0; units: none
FVALUE	the initial value of the function; default: 0.0; units: none

✎ **line117:** Supplemental *variables* associated with the function

✎ each function may require supplemental lines to fully specify the function; Appendix B contains the supplemental *variables* required by each function; care should be taken to ensure the proper supplemental *variables* are entered in the correct order for each function

✎ **line118: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 15 data; do not include enclosing apostrophes

Block 16, Function-Controlled Variable Data for the System:

repeat lines 119 through 120 for each function-controlled variable; this Block, along with function data Blocks, allows the user to define input variables with functional dependence of the form: $y = x$ or $y = f(x)$; y is a function-controlled input variable, x is a signal-variable function, and $f()$ is a specified mathematical function; functions can be specified as functions of other functions if desired; function-controlled variables are updated at each system time step

line119: IFCVID, FCVNAME, IAFN, FCVM

<u>Variable</u>	<u>Description</u>
IFCVID	the desired function-controlled variable identification (ID) number; all function-controlled variable ID numbers are provided in Appendix A ; this ID specifies the input variable that is to be functionally controlled; any variable in Appendix A can be specified; including a variable in this data Block designates that it is updated at the beginning of each system time step iteration; variables that are functions of time should be included in this data Block; function-controlled variables are updated in the order they are listed in the input file; default: none; units: none
FCVNAME	any descriptive phrase of 28 characters or less to identify the function-controlled variable in the output files; default: each function-controlled variable has a unique default assigned to it that is provided in Appendix A; units: none
IAFN	the associated argument function number for this function-controlled variable; any number between 1 and 9999 is a valid argument function number; the value of IAFN must correspond to an IFN value in the function data defined for fluid mechanics, heat transfer, reactor dynamics, or the system; a negative value of IAFN can be entered, indicating the use of the change in the associated IFN function value as the argument function; thus, $y = x^{\text{new}} - x^{\text{old}}$ or $y = f(x^{\text{new}} - x^{\text{old}})$, if IAFN is entered as a negative number; x^{old} is the IFN function value at the previous system time step; default: none; units: none

FCVM a multiplier applied to the computed function-controlled variable; default: 1.0; units: none

 **line120: L1, L2, L3**

<u>Variable</u>	<u>Description</u>
L1	the first locator number to define the desired system location of the function-controlled variable; the definition of L1 depends on the selected value of IFCVID and is provided in Appendix A; example -- for function-controlled variable DREAC, L1 is defined as the associated reactor number, KR; default: none; units: none
L2	the second locator number to define the desired system location of the function-controlled variable; the definition of L2 depends on the selected value of IFCVID and is provided in Appendix A; example -- for function-controlled variable DREAC, L2 is defined as the associated feedback term number; default: none; units: none
L3	the third locator number to define the desired system location of the function-controlled variable; the definition of L3 depends on the selected value of IFCVID and is provided in Appendix A; default: none; units: none

 **line121: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 16 data; do not include enclosing apostrophes

Block 17, Function Data for the System:

repeat line 122 and any associated supplemental line(s) for each function; this Block, along with function-controlled variable data Blocks, allows the user to define input variables with functional dependence of the form: $y = x$ or $y = f(x)$; y is a function-controlled input variable, x is a signal-variable function, and $f()$ is a specified mathematical function; functions can be specified as functions of other functions if desired; functions are updated at each system time step

line122: IFN, IFID, FNAME, FM, FVALUE

<u>Variable</u>	<u>Description</u>
IFN	the user-specified function number; any number between 1 and 9999 is a valid function number; this number may be referenced in function-controlled variable data by the IAFN variable, or by function supplemental <i>variables</i> (see the next line); functions are updated in the order they are listed in the input file, independent of the value of IFN; default: none; units: none
IFID	the function identification (ID) number; all function ID numbers are provided in Appendix B ; any function in Appendix B can be specified; including a function in this data Block designates that it is updated at the beginning of each system time step iteration; functions that are functions of time should be included in this data Block; available functions include dependent output variables (signal-variable functions) such as pressure and temperature, and mathematical functions such as cosine and square root; mathematical functions require specification of associated argument function(s) on the following line ; default: none; units: none
FNAME	any descriptive phrase of 28 characters or less to identify the function in the output files; default: each function has a unique default assigned to it that is provided in Appendix B; units: none
FM	a multiplier applied to the computed function; default: 1.0; units: none
FVALUE	the initial value of the function; default: 0.0; units: none

 **line123: Supplemental *variables* associated with the function**

 each function may require supplemental lines to fully specify the function; Appendix B contains the supplemental *variables* required by each function; care should be taken to ensure the proper supplemental *variables* are entered in the correct order for each function

 **line124: END**

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 17 data; do not include enclosing apostrophes

Block 18, User-Defined Data:

☞ this data block is reserved for any input variables the user chooses, consistent with any user-specified subroutines; it is the responsibility of the user to provide the proper interface between the user subroutines, the user variables, and SAFSIM; subroutine UOUT writes the user-defined output and is called every PRNTINT seconds as specified in Block 1 data

☞line125: user-defined variables

<u>Variable</u>	<u>Description</u>
user1	...
user2	...
...	

☞line126: END

<u>Variable</u>	<u>Description</u>
END	this line must contain the character variable END in the first three columns to indicate the end of Block 18 data; do not include enclosing apostrophes

4.0 SAFSIM Output Files

The output of SAFSIM is placed in separate files according to content. For example, heat transfer output is written to a single file and reactor dynamics output is written to a different file. This helps to reduce the size of any single output file. Also, unformatted files are created that contain data intended for use in plotting programs. These files contain all output variables and may be very large. Special purpose computer programs can then be created to extract the desired data and rewrite it in the desired format; perhaps in tabular format for inclusion in a spreadsheet. The following table provides a brief description of all files associated with SAFSIM.

SAFSIM Files

Unit	File Name	Description
1	SAFSIM.DMP	the unformatted dump file for restarts (not in use)
2	SAFSIM.OF	the Function and Function-Controlled Variable formatted output file
3	SAFSIM.OIN	this file contains the problem input data after processing and is formatted
4	SAFSIM.IN	the input data file created by the user
5	INPUT.SS	the input data file with all comment lines removed (deleted automatically after execution)
6	SAFSIM.OFM	the fluid mechanics formatted output file
7	SAFSIM.OHT	the heat transfer formatted output file; this file is created only if heat transfer structures are included in the input model
8	SAFSIM.OXM	the fluid mechanics mass fraction output file; this file is created only if multiple gases are specified
9	SAFSIM.ORD	the reactor dynamics formatted output file; this file is created only if nuclear reactors are included in the input model
10	SAFSIM.OIT	this file contains iteration summary information regarding fluid mechanics solution and is formatted
11	SAFSIM.PFM	the fluid mechanics unformatted output file
12	SAFSIM.PHT	the heat transfer unformatted output file; this file is created only if heat transfer structures are included in the input model
13	SAFSIM.PRD	the reactor dynamics unformatted output file; this file is created only if nuclear reactors are included in the input model
14	SAFSIM.PF	the Function unformatted output file
15	SAFSIM.OU	the user-defined formatted output file

The following four subsections provide more detail about the contents of the output files for fluid mechanics, heat transfer, reactor dynamics, function and function-controlled variables, and fluid mechanics iteration summary.

4.1 Fluid Mechanics Output

All fluid mechanics output is written for each flow network, starting with network number 1. The variables are grouped into three sections: nodal variables, element variables, and element property variables. If a compressor or nozzle element is included in the input model, additional output sections are provided, with information relevant to that special element. The heading E# indicates the element identification number, and the headings NODE1 and NODE2 indicate the global node identification numbers for local nodes 1 and 2 of the element, respectively. The following table provides definitions of the abbreviations used for the fluid mechanics and mass fraction output.

Fluid Mechanics Output Abbreviations

<u>Abbreviation</u>	<u>Definition</u>
P1	fluid pressure at local node 1 of the element
P2	fluid pressure at local node 2 of the element
T1	fluid temperature at local node 1 of the element
T2	fluid temperature at local node 2 of the element
D1	fluid density at local node 1 of the element
D2	fluid density at local node 2 of the element
FLOW	element fluid mass flow rate
VELOCITY	element fluid velocity (pore velocity for porous media elements)
P	element fluid pressure
T	element fluid temperature
FRIC FAC	element friction factor
Qflow	flow power (losses due to compressibility and viscous dissipation); positive Q indicates power <u>into</u> the fluid
Qconv	convection power added to the fluid from all heat transfer structure elements coupled to the fluid mechanics element; positive Q indicates power <u>into</u> the fluid
HA	the product of H (the heat transfer coefficient) and A (the heat transfer surface area) for all coupled heat transfer elements (HA is the sum of all coupled HA's)
DEN	element fluid density
MACH #	element fluid Mach number
DVISC	element fluid dynamic viscosity
Cp	element fluid constant pressure specific heat

GAMMA	element constant pressure specific heat divided by constant volume specific heat for the fluid
COND	element fluid thermal conductivity
REYNOLDS #	element fluid Reynolds number (based on pore velocity for porous media elements)
PRANDTL #	element fluid Prandtl number
PECLET #	element fluid Peclet number based on element length
Pt	fluid pressure at the throat (local node 2) of a nozzle (choked flow boundary) element
TEMPt	fluid temperature at the throat (local node 2) of a nozzle (choked flow boundary) element
DENt	fluid density at the throat (local node 2) of a nozzle (choked flow boundary) element
X	element gas constituent mass fraction (for multiple gas networks)
X1	gas constituent mass fraction at local node 1 of the element (for multiple gas networks)
X2	gas constituent mass fraction at local node 2 of the element (for multiple gas networks)
PARTIAL P	gas constituent partial pressure of the element (for multiple gas networks)

In the output, the fluid velocity for flow in a porous media is the pore (or interstitial void) velocity and not the superficial velocity. Also, the Reynolds number is based on the pore velocity and the particle diameter for porous media elements. Otherwise, the Reynolds number is based on the equivalent diameter of the element. The Peclet number (the product of a different Reynolds number and the Prandtl number) uses the element length as the characteristic distance in the Reynolds number. The flow and fluid properties provided in the SAFSIM.OFM file are for the fluid mixture if more than one gas constituent is specified in the input file. The flow and fluid properties for the individual gas constituents are provided in the SAFSIM.OXM file.

4.2 Heat Transfer Output

All heat transfer output is written for each heat transfer structure, starting with structure number 1. Boundary condition output data for end 1 of the structure is presented first, followed by element and node output data; this data is followed by boundary condition output data for end 2. The heading e/n# indicates either the element or node identification number. The following table provides definitions of the abbreviations used for the heat transfer output.

Heat Transfer Output Abbreviations

<u>Abbreviation</u>	<u>Definition</u>
I FAC	node implicitness factor (1.0 is fully implicit)
Tn	node material temperature
dT/dt	rate of temperature change (new-time temperature minus old-time temperature divided by time step) at the node
Te	element material temperature
dT/dX	material temperature gradient for the element (a positive gradient indicates increasing temperature in the direction of increasing node number)
COND	element material thermal conductivity
C	element material specific heat
DEN	element material density
Q"	surface heat flux (positive Q" indicates power out of the structure)
Q	heat flow rate (positive Q indicates power out of the structure)
NW#	the coupled fluid mechanics network ID number
E#	the coupled fluid mechanics element ID number
HTC	heat transfer coefficient for the surface
Twall	element wall (surface) temperature
Tfluid	the fluid temperature for the coupled fluid mechanics element
FILMDROP	Twall minus Tfluid

The structure internal power is the total power added to the structure by internal volumetric heating. Positive internal power indicates power into the structure. The exchange power is the power exchanged with any coupled fluid mechanics elements or coupled heat transfer elements. Positive exchange power indicates power out of the structure. The net power is the internal power minus the exchange power. If the net power is positive, more power is being added to the structure than is being taken away by convection or radiation.

4.3 Reactor Dynamics Output

All reactor dynamics output is written for each reactor, starting with reactor number 1. The following table provides definitions of the abbreviations used for the reactor dynamics output.

Reactor Dynamics Output Abbreviations

<u>Abbreviation</u>	<u>Definition</u>
dPOWER/dt	rate of neutron power change (new-time power minus old-time power divided by time step)
dPeff/dt	rate of effective thermal power change (new-time power minus old-time power divided by time step)
dC/dt	rate of delayed-neutron precursor concentration change (new-time concentration minus old-time concentration divided by time step)
dH/dt	rate of decay heat precursor concentration change (new-time concentration minus old-time concentration divided by time step)
dRHO/dt	rate of reactivity change (new-time reactivity minus old-time reactivity divided by time step)

The effective reactor power is determined as the prompt neutron power plus the decay power from all decay heat groups. The prompt neutron power is defined as the total neutron power times the fraction of neutron power deposited promptly ($1.0-\eta$). η is the total decay heat fraction. Reactivity output is presented as a dimensionless variable ($\Delta k/k$) and in units of $\$$ ($\Delta k/k\beta$), where k is the effective neutron multiplication factor and β is the total delayed neutron fraction.

4.4 Function and Function-Controlled Variables Output

The function and function-controlled variable output data is listed first for fluid mechanics, second for heat transfer, third for reactor dynamics, and last for the system. The functions and function-controlled variables are numbered consecutively under the COUNTER heading. The F # is the function number specified by input variable IFID. The ID #'s come from Appendices A and B. Two values are provided for the function output: the function value for the current time and the new-time function value minus the old-time function value.

4.5 Iteration Summary Output

This output provides a summary of the number of iterations required for the various fluid mechanics solutions for every time step iteration. This output is useful for optimizing convergence criteria and iteration maximums. It also provides valuable information if the program fails.

MEE indicates if the mechanical energy equation solution is on (1) or off (0). ID is the number of density iterations required for convergence of the density field and IM is the number of mass flow rate iterations required for

solution of the mass flow rate field. ICD indicates the number of times the Cholesky decomposition solver is employed for solution of the pressure equation. IGE indicates the number of times the Gauss elimination solver is employed for solution of the pressure equation. IP(MAX) is the maximum number of Gauss-Seidel iterations required for solution of the pressure equations for the current density iteration. IP(MIN) is the minimum number of iterations and IP(AVG) is the average number of Gauss-Seidel iterations. The average is determined as the total number of Gauss-Seidel iterations for all mass flow rate iterations divided by the number of mass flow rate iterations for the current density iteration. IC is the number of iterations required for solution of the element temperature solution. IGET is the number of times Gauss elimination is used to solve the fluid temperature equations. IT(MAX), IT(MIN), and IT(AVG) are the maximum, minimum, and average number of iterations, respectively, for solution of the nodal temperature field using Gauss-Seidel iteration for the current density iteration. IX(i) is the number of iterations required for solution of the mass fraction field for gas constituent i. Each gas constituent has a value of IX printed.

Appendices

Appendix A: Function-Controlled Variable Identification Numbers

Appendix B: Function Identification Numbers

**Appendix C: Heat Transfer Coefficient Correlation Identification
Numbers**

Appendix D: Sample User-Defined Subroutines

**Appendix A:
Function-Controlled Variable Identification Numbers**

ID	Variable	Default Name	L1	L2	L3
1	TIMESTEP	'SYSTEM TIMESTEP'	-	-	-
2	PRNTINT	'SYSTEM PRINT INTERVAL'	-	-	-
3	NPLT	'SYSTEM PLOT FREQUENCY'	-	-	-
4	NDUMP	'SYSTEM DUMP FREQUENCY'	-	-	-
5	MAXIP	'MAX # OF PRESSURE ITERATIONS'	NW	-	-
6	MAXIM	'MAX # FLOWRATE ITERATIONS'	NW	-	-
7	MAXID	'MAX # OF DENSITY ITERATIONS'	NW	-	-
8	MAXIT	'MAX # OF NODETEMP ITERATIONS'	NW	-	-
9	MAXIC	'MAX # OF ELEMTEMP ITERATIONS'	NW	-	-
10	MAXIX	'MAX # OF MASSFRAC ITERATIONS'	NW	-	-
11	DRRRP	'RELATIVE ERROR FOR PRESSURE'	NW	-	-
12	DRRRM	'RELATIVE ERROR FOR FLOWRATE'	NW	-	-
13	DRRRD	'RELATIVE ERROR FOR DENSITY'	NW	-	-
14	DRRRT	'RELATIVE ERROR FOR NODETEMP'	NW	-	-
15	DRRRC	'RELATIVE ERROR FOR ELEMTEMP'	NW	-	-
16	DRRRX	'RELATIVE ERROR FOR MASSFRAC'	NW	-	-
17	RELAXP	'RELAX PARAMETER FOR PRESSURE'	NW	-	-
18	RELAXM	'RELAX PARAMETER FOR FLOWRATE'	NW	-	-
19	RELAXD	'RELAX PARAMETER FOR DENSITY'	NW	-	-
20	RELAXT	'RELAX PARAMETER FOR NODETEMP'	NW	-	-
21	RELAXC	'RELAX PARAMETER FOR ELEMTEMP'	NW	-	-
22	RELAXX	'RELAX PARAMETER FOR MASSFRAC'	NW	-	-
23	AGRAV	'ACCELERATION DUE TO GRAVITY'	-	-	-
24	POT	'NETWORK DENSITY-RGAS'	NW	-	-

ID	Variable	Default Name	L1	L2	L3
25	VOLNW	'NETWORK VOLUME'	NW	-	-
26	MEE	'MECH ENERGY EQUATION FLAG'	NW	-	-
27	IDYNFM	'DYNAMIC FLAG FOR FM'	NW	-	-
28	blank	currently not in use	-	-	-
29	blank	currently not in use	-	-	-
30	RGAS	'GAS CONSTANT'	idgas	-	-
31	VISM	'FLUID VISCOSITY MULTIPLIER'	idgas	-	-
32	CPM	'FLUID SPECIFIC HEAT MULTIPLIER'	idgas	-	-
33	CONDM	'FLUID CONDUCTIVITY MULTIPLIER'	idgas	-	-
34	TEND	'PROBLEM END TIME'	-	-	-
35	BC (based on Block 5 data)	'FM BOUNDARY CONDITION'	IDSET	BC kind # 1 for mass flow rate; 2 for pressure; 3 for temperature; 4,5, ... for mass fraction for igas= 1,2, ..., NGAS	
36	DX	'ELEMENT LENGTH'	NW	JE	-
37	ED	'ELEMENT EQUIV DIAMETER'	NW	JE	-
38	DZ	'ELEMENT ELEVATION CHANGE'	NW	JE	-
39	POR	'ELEMENT POROSITY'	NW	JE	-
40	FA1	'ELEMENT FLOW AREA 1'	NW	JE	-
41	FA2	'ELEMENT FLOW AREA 2'	NW	JE	-
42	FAE	'ELEMENT FLOW AREA'	NW	JE	-
43	VOL	'ELEMENT VOLUME'	NW	JE	-
44	REL RUF	'ELEMENT RELATIVE ROUGHNESS'	NW	JE	-
45	AK12	'ELEMENT LOSS COEFFICIENT 12'	NW	JE	-
46	AK21	'ELEMENT LOSS COEFFICIENT 21'	NW	JE	-
47	ALOD	'ELEMENT ADDED L/D'	NW	JE	-
48	QDIR	'ELEMENT DIRECT POWER'	NW	JE	-
49	SPEED	'COMPRESSOR FRACTIONAL SPEED'	NW	JE	-
50	DTFM	'FLUID MECHANICS TIMESTEP'	NW	-	-
51	NPRNTF	'FLUID MECH PRINT FREQ'	NW	-	-
52	blank	currently not in use	-	-	-
53	blank	currently not in use	-	-	-
54	KHDYN	'STRUCTURE DYNAMIC FLAG'	KH	-	-

ID	Variable	Default Name	L1	L2	L3
55	DENHS	'STRUCTURE MATERIAL DENSITY'	material number	-	-
56	SHM	'SPECIFIC HEAT MULTIPLIER'	material number	-	-
57	CDM	'CONDUCTIVITY MULTIPLIER'	material number	-	-
58	COPIES	'NUMBER OF STRUCTURE COPIES'	KH	-	-
59	FETS	'FRACTION OF EXPLICIT DT'	KH	-	-
60	QPPPM	'POWER DENSITY MULTIPLIER'	KH	-	-
61	TBC1	'TEMPERATURE, END 1'	KH	-	-
62	TBC2	'TEMPERATURE, END 2'	KH	-	-
63	DTMINH	'MINIMUM HT TIMESTEP'	KH	-	-
64	FLUXBC1	'HEAT FLUX, END 1'	KH	-	-
65	FLUXBC2	'HEAT FLUX, END 2'	KH	-	-
66	AFLUX1	'HEAT FLUX AREA, END 1'	KH	-	-
67	AFLUX2	'HEAT FLUX AREA, END 2'	KH	-	-
68	NPRNTH	'HT PRINT FREQUENCY'	KH	-	-
69	AEX	'CONVECTIVE EXCHANGE AREA'	KH	jeh element #	jex surface #
70	AEX	'B CONVECTIVE EXCHANGE AREA' (for boundaries)	KH	end (1 or 2)	jex surface #
71	EDH	'CONVECTIVE EQUIVALENT D'	KH	jeh element #	jex surface #
72	EDH	'B CONVECTIVE EQUIVALENT D' (for boundaries)	KH	end (1 or 2)	jex surface #
73	IHTCLAM	'HTC CORRELATION #, LAMINAR'	KH	jeh element #	jex surface #
74	IHTCLAM	'B HTC CORRELATION #, LAM.' (for boundaries)	KH	end (1 or 2)	jex surface #
75	IHTCTUR	'HTC CORRELATION #, TURBULANT'	KH	jeh element #	jex surface #
76	IHTCTUR	'B HTC CORRELATION #, TUR.' (for boundaries)	KH	end (1 or 2)	jex surface #
77	TURLAMH	'CONVECTIVE TRANSITION RE #'	KH	jeh element #	jex surface #
78	TURLAMH	'B CONVECTIVE TRANS. RE #' (for boundaries)	KH	end (1 or 2)	jex surface #
79	AUXL	'AUXILIARY DATA, LAMINAR'	KH	jeh element #	jex surface #
80	AUXL	'B AUXILIARY DATA, LAMINAR' (for boundaries)	KH	end (1 or 2)	jex surface #
81	AUXT	'AUXILIARY DATA, TURBULANT'	KH	jeh element #	jex surface #
82	AUXT	'B AUXILIARY DATA, TURBULEN' (for boundaries)	KH	end (1 or 2)	jex surface #
83	DXH	'STRUCTURE ELEMENT LENGTH'	KH	jeh element #	-

ID	Variable	Default Name	L1	L2	L3
84	AREA	'STRUCTURE ELEMENT AREA'	KH	jeh element #	-
85	VOLH	'STRUCTURE ELEMENT VOLUME'	KH	jeh element #	-
86	QPPP	'STRUCTURE ELEMENT QPPP'	KH	jeh element #	-
87	QPP	'STRUCTURE ELEMENT FLUX'	KH	jeh element #	jfluxa surface #
88	ASURF	'STRUCTURE ELEMENT S.AREA'	KH	jeh element #	jfluxa surface #
89	GENTIME	'NEUTRON GENERATION TIME'	KR	-	-
90	SOURCE	'NEUTRON SOURCE TERM'	KR	-	-
91	DTRDMIN	'MINIMUM TIMESTEP FOR RD'	KR	-	-
92	DTRDMAX	'MAXIMUM TIMESTEP FOR RD'	KR	-	-
93	DRERR	'DESIRED REL.ERR FOR RD'	KR	-	-
94	NPRNTRD	'PRINT FREQUENCY FOR RD'	KR	-	-
95	RDST	'RX DYNAMICS START TIME'	KR	-	-
96	AUXRD	'RD AUXILIARY DATA'	KR	kaux	-
97	KSOLVE	'RD INTEGRATOR OPTION #'	KR	-	-
98	DREAC	'FEEDBACK DIFFERENTIAL RHO'	KR	kreac term #	-
99	blank	currently not in use	-	-	-
100	blank	currently not in use	-	-	-

NOTE: some of the L (locator) variables are not explicitly defined in the Block data input; these variables (listed in lower case) are defined here:

- idgas: the gas ID number based on the order the gases are listed on line 19 of Block 2 data
- igas: the gas constituent number based on the order the gases are listed on line 18 of Block 2 data
- BC kind #: the corresponding number for the kind of boundary condition assigned for the referenced IDSET from Block 5 data
- jeh: the structure finite element number based on the order the elements are listed in Block 10 data
- jex: the exchange surface number for convection based on the order the exchange surfaces are listed in Block 10 data
- end: the boundary end number (1 or 2) for variables that are defined at the boundaries of heat transfer structures in Block 10 data
- jfluxa: the surface area number for heat flux based on the order the surfaces are listed in Block 10 data
- kaux: the auxiliary data number based on the order they are listed in Block 13 data
- kreac: the feedback term number based on the order the feedback terms are listed in Block 13 data

Appendix B: Function Identification Numbers

Signal-Variable Functions			supplemental <i>variables</i>		
ID	Variable	Default Name	L1	L2	L3
1	TIME	'PROBLEM TIME'	-	-	-
2	DELT	'SYSTEM delta-TIME'	-	-	-
3	ISTEP	'SYSTEM TIMESTEP #'	-	-	-
4	TIME-DT/2	'TIME - delta-TIME/2'	-	-	-
5	TIME-DT	'TIME - delta-TIME'	-	-	-
6	DELTF	'FLUID MECHANICS TIMESTEP'	NW	-	-
7	ISTF	'FLUID MECHANICS TIMESTEP #'	NW	-	-
8	PRESN	'NODE FLUID PRESSURE'	NW	i (node #)	-
9	TN	'NODE FLUID TEMPERATURE'	NW	i (node #)	-
10	FLONL	'LOCAL NODE FLUID FLOW RATE'	NW	JE	ilocal (1 or 2)
11	XMN	'NODE GAS MASS FRACTION'	NW	i (node #)	igas
12	PRESE	'ELEMENT FLUID PRESSURE'	NW	JE	-
13	TE	'ELEMENT FLUID TEMPERATURE'	NW	JE	-
14	FLOE	'ELEMENT FLUID FLOW RATE'	NW	JE	-
15	DENE	'ELEMENT FLUID DENSITY'	NW	JE	-
16	FF	'ELEMENT FRICTION FACTOR'	NW	JE	-
17	QFLOE	'ELEMENT FLOW POWER'	NW	JE	-
18	QCV	'ELEMENT CONVECTIVE POWER'	NW	JE	-
19	HAFM	'ELEMENT HTC-S.AREA'	NW	JE	-
20	T2-T1	'ELEMENT FLUID delta-T'	NW	JE	-
21	P2-P1	'ELEMENT delta-P'	NW	JE	-
22	REY	'ELEMENT REYNOLDS NUMBER'	NW	JE	-
23	PRNL	'ELEMENT PRANDTL NUMBER'	NW	JE	-
24	VELE	'ELEMENT FLUID VELOCITY'	NW	JE	-
25	AK	'ELEMENT LOSS COEFFICIENT'	NW	JE	-
26	GRAVE	'ELEMENT GRAVITY HEAD'	NW	JE	-
27	ER	'ELEMENT RESISTNC COEFFICIENT'	NW	JE	-
28	RGASM	'ELEMENT MIXTURE GAS CONSTANT'	NW	JE	-

Signal-Variable Functions			supplemental variables		
ID	Variable	Default Name	L1	L2	L3
29	MACH#	'ELEMENT MACH #'	NW	JE	-
30	GAMMA	'ELEMENT SPECIFIC HEAT RATIO'	NW	JE	-
31	PP	'ELEMENT PARTIAL PRESSURE'	NW	JE	igas
32	XME	'ELEMENT GAS MASS FRACTION'	NW	JE	igas
33	WISE	'ELEMENT FLUID VISCOSITY'	NW	JE	igas (0 for mixture)
34	CPE	'ELEMENT FLUID Cp'	NW	JE	igas (0 for mixture)
35	CONDE	'ELEMENT FLUID CONDUCTIVITY'	NW	JE	igas (0 for mixture)
36	CVE	'ELEMENT FLUID Cv'	NW	JE	igas (0 for mixture)
37	blank	currently not in use	-	-	-
38	blank	currently not in use	-	-	-
39	blank	currently not in use	-	-	-
40	PANW	'NETWORK-AVERAGED PRESSURE' (open networks)	NW	-	-
41	TANW	'NETWORK-AVERAGED TEMPERATURE' (open networks)	NW	-	-
42	DANW	'NETWORK-AVERAGED DENSITY'	NW	-	-
43	PZ	'NOZZLE THROAT PRESSURE'	NW	JE	-
44	ZMT	'NOZZLE THROAT MACH #'	NW	JE	-
45	PANW	'NETWORK-AVERAGED PRESSURE' (closed networks)	NW	-	-
46	TANW	'NETWORK-AVERAGED TEMPERATURE' (closed networks)	NW	-	-
47	blank	currently not in use	-	-	-
48	DENH	'STRUCTURE MATERIAL DENSITY'	KH	jeh element #	-
49	SHH	'STRUCTURE MATERIAL SP.HEAT'	KH	jeh element #	-
50	CONDH	'STRUCTURE MATERIAL CONDUCTVY'	KH	jeh element #	-
51	TEH	'STRUCTURE ELEMENT TEMPERATUR'	KH	jeh element #	-
52	DTDX	'STRUCTURE ELEMENT dT/dX'	KH	jeh element #	-
53	DELTH	'ACTUAL HEAT TRANSFER TIMESTEP'	KH	-	-
54	ETSTEP	'EXPLICIT TIMESTEP LIMIT'	KH	-	-

Signal-Variable Functions			supplemental variables		
ID	Variable	Default Name	L1	L2	L3
55	ISTH	'STRUCTURE TIMESTEP #'	KH	-	-
56	QPOW	'STRUCTURE ELEMENT POWER'	KH	jeh element #	-
57	TAH	'STRUCTURE AVERAGE T'	KH	-	-
58	QTOH	'STRUCTURE TOTAL Q'	KH	-	-
59	TMAXH	'STRUCTURE MAXIMUM T'	KH	-	-
60	TMINH	'STRUCTURE MINIMUM T'	KH	-	-
61	TNH	'STRUCTURE NODE TEMPERATURE'	KH	ih node #	-
62	DTDT	'STRUCTURE NODE dT/dt'	KH	ih node #	-
63	FACTI	'NODE IMPLICITNESS FACTOR'	KH	ih node #	-
64	TWALL	'CONVECTIVE WALL T'	KH	jeh element #	jex surface #
65	TWALL	'B CONVECTIVE WALL T' (for boundaries)	KH	end (1 or 2)	jex surface #
66	TEX	'CONVECTIVE EXCHANGE T'	KH	jeh element #	jex surface #
67	TEX	'B CONVECTIVE EXCHANGE T' (for boundaries)	KH	end (1 or 2)	jex surface #
68	HTC	'HEAT TRANSFER COEFFICIENT'	KH	jeh element #	jex surface #
69	HTC	'B HEAT TRANSFER COEFFICIENT'	KH	end (1 or 2)	jex surface #
70	REYH	'CONVECTIVE REYNOLDS #'	KH	jeh element #	jex surface #
71	REYH	'B CONVECTIVE REYNOLDS #' (for boundaries)	KH	end (1 or 2)	jex surface #
72	FILMDRP	'CONVECTIVE FILM DROP'	KH	jeh element #	jex surface #
73	FILMDRP	'B CONVECTIVE FILM DROP' (for boundaries)	KH	end (1 or 2)	jex surface #
74	QCVH	'CONVECTIVE Q'	KH	jeh element #	jex surface #
75	QCVH	'B CONVECTIVE Q' (for boundaries)	KH	end (1 or 2)	jex surface #
76	QPP	'CONVECTIVE QPP'	KH	jeh element #	jex surface #
77	QPP	'B CONVECTIVE QPP' (for boundaries)	KH	end (1 or 2)	jex surface #
78	blank	currently not in use	-	-	-
79	ISRDR	'REACTOR DYNAMICS TIMESTEP #'	KR	-	-
80	DELTR	'REACTOR DYNAMICS REL ERROR'	KR	-	-

Signal-Variable Functions			supplemental variables		
ID	Variable	Default Name	L1	L2	L3
81	RDN	'REACTOR NEUTRON POWER'	KR	.	.
82	DNDT	'REACTOR DYNAMICS dPower/dt'	KR	.	.
83	RDNE	'EFFECTIVE REACTOR POWER'	KR	.	.
84	DNEDT	'REACTOR DYNAMICS dPeff/dt'	KR	.	.
85	DECPOW	'REACTOR DECAY POWER'	KR	.	.
86	KEFF	'NEUTRON MULTIPLICATION FACTR'	KR	.	.
87	PERINV	'REACTOR INVERSE PERIOD'	KR	.	.
88	PERINVE	'E. REACTOR INVERSE PERIOD'	KR	.	.
89	CDN	'PRECURSOR CONCENTRATION'	KR	igroup	.
90	DCDT	'DELAYED NEUTRON dC/dt'	KR	igroup	.
91	CDH	'DECAY HEAT CONCENTRATION'	KR	igroupd	.
92	DHDT	'DECAY HEAT dH/dt'	KR	igroupd	.
93	DREACT	'DIFFERENTIAL FB REACTIVITY' (total)	KR	.	.
94	DRHODT	'd(PROGRAMMED REACTIVITY)/dt'	KR	.	.
95	REACT	'TOTAL REACTIVITY'	KR	.	.
96	RERRR	'RD RELATIVE TRUNCATION ERROR'	KR	.	.
97	DREACI	'INTEGRATED DIFF.REACTIVITY'	KR	kreact term #	.
98	PRI	'INTEGRATED PROG.REACTIVITY'	KR	.	.
99	blank	currently not in use	.	.	.
100	blank	currently not in use	.	.	.
101	blank	currently not in use	.	.	.
102	blank	currently not in use	.	.	.
103	blank	currently not in use	.	.	.
104	blank	currently not in use	.	.	.
105	blank	currently not in use	.	.	.
106	blank	currently not in use	.	.	.
107	blank	currently not in use	.	.	.
108	blank	currently not in use	.	.	.
109	blank	currently not in use	.	.	.

Mathematical Functions		supplemental variables
ID	Default Name/Description	description for each line
110	'INPUT CONSTANT'	1: constant value
111	'POLYNOMIAL'	1: polynomial order, argument IFN # 2: $a_0, a_1, \dots, a_{order+1}$
112	'LINEAR INTERPOLATION TABLE' non-uniform x data spacing allowed	1: number of data pairs, argument IFN # 2: x,y data pairs
113	'LINEAR TABLE W/ EQUAL X INT' uniform x data spacing required	1: number of data pairs, argument IFN # 2: x,y data pairs
114	'LINEAR ADVANCING-ONLY TABLE' x data must be in ascending order, useful when time is x variable	1: number of data pairs, argument IFN # 2: x,y data pairs
115	'QUADRATIC INT TABLE' currently not available	1: number of data pairs, argument IFN # 2: x,y data pairs
116	'2-D LINEAR INT TABLE' currently not available	1: number of data triplets, x argument IFN #, y argument IFN # 2: x,y,z data triplets
117	'STEP TABLE' currently not available	1: number of data pairs, argument IFN # 2: x,y data pairs
118	'ADVANCING-ONLY STEP TABLE' currently not available	1: number of data pairs, argument IFN # 2: x,y data pairs
119	currently not in use	-
120	currently not in use	-
121	currently not in use	-
122	'AVERAGED TEMPERATURE' mass averaging of heat transfer structure element temperatures	1: # of heat transfer structure elements 2: (KH, jeh, weighting factor) for each structure element
123	'AVERAGED FLUID DENSITY' volume averaging of fluid mechanics element densities	1: # of fluid mechanics elements 2: (NW, JE, weighting factor) for each fluid mechanics element
124	blank, currently not in use	-
125	'USER-DEFINED FUNCTION #1' user must modify subroutine UFUN	1: # of IFN argument functions (maximum of 5), IFN #'s
126	'USER-DEFINED FUNCTION #2' user must modify subroutine UFUN	1: # of IFN argument functions (maximum of 5), IFN #'s
127	'USER-DEFINED FUNCTION #3' user must modify subroutine UFUN	1: # of IFN argument functions (maximum of 5), IFN #'s
128	'USER-DEFINED FUNCTION #4' user must modify subroutine UFUN	1: # of IFN argument functions (maximum of 5), IFN #'s
129	'USER-DEFINED FUNCTION #5' user must modify subroutine UFUN	1: # of IFN argument functions (maximum of 5), IFN #'s
130	currently not in use	-
131	'SUMMATION'	1: # of IFN argument functions (maximum of 5), IFN #'s
132	'SUBTRACTION'	1: argument 1 IFN #, argument 2 IFN #
133	'MULTIPLICATION'	1: # of IFN argument functions (maximum of 5), IFN #'s

Mathematical Functions		supplemental variables
ID	Default Name/Description	description for each line
134	'DIVISION'	1: argument 1 IFN #, argument 2 IFN #
135	'POWER' / x^y	1: x argument IFN #, y argument IFN #
136	'EXPONENTIAL' / e^x	1: x argument IFN #
137	'INVERSE'	1: argument IFN #
138	'NATURAL LOG'	1: argument IFN #
139	'LOG (BASE10)'	1: argument IFN #
140	'ABSOLUTE VALUE'	1: argument IFN #
141	'SQUARE ROOT'	1: argument IFN #
142	'MAXIMUM'	1: # of IFN argument functions (maximum of 5), IFN #'s
143	'MINIMUM'	1: # of IFN argument functions (maximum of 5), IFN #'s
144	'SINE' / argument in radians	1: argument IFN #
145	'COSINE' / argument in radians	1: argument IFN #
146	'TANGENT' / argument in radians	1: argument IFN #
147	'ARCSINE' / result in radians	1: argument IFN #
148	'ARCCOSINE' / result in radians	1: argument IFN #
149	'ARCTANGENT' / result in radians	1: argument IFN #
150	'SIGN (+/-)' result=sign of argument 2 times value of argument 1	1: argument 1 IFN #, argument 2 IFN #
151	'MOD' result=MOD(argument1,argument2)	1: argument 1 IFN #, argument 2 IFN #
152	'UNIT SQUARE WAVE 1' result between 0 and 1	1: argument IFN # (in radians, for example: $2\pi \cdot \text{frequency} \cdot \text{TIME}$)
153	'UNIT SQUARE WAVE 2' result between -1 and 1	1: argument IFN # (in radians, for example: $2\pi \cdot \text{frequency} \cdot \text{TIME}$)
154	'IF(TIME>F1)F=F2'	1: argument 1 IFN #, argument 2 IFN #
155	'IF(F1 > F2)F=F3 ELSE F4' if value of argument 1 > than value of argument 2, then result=argument 3, else result=argument 4	1: argument 1 IFN #, argument 2 IFN #, argument 3 IFN #, argument 4 IFN #
156	'IF(F1>F2.AND.F3>F4)F=F5 ELSE F6'	1: argument 1 IFN #, argument 2 IFN #, argument 3 IFN #, argument 4 IFN #, argument 5 IFN #, argument 6 IFN #
157	'IF(F1>F2.OR.F3>F4)F=F5 ELSE F6'	1: argument 1 IFN #, argument 2 IFN #, argument 3 IFN #, argument 4 IFN #, argument 5 IFN #, argument 6 IFN #
158	'GO TO (F1,F2,F3), F4' if F4<0,result=F1, if F4=0,result=F2, if F4>0,result=F3	1: argument 1 IFN #, argument 2 IFN #, argument 3 IFN #, argument 4 IFN #
159	currently not in use	-
160	currently not in use	-

Mathematical Functions		supplemental variables
ID	Default Name/Description	description for each line
161	'RADIATIVE QPP, SURFACE 1' radiation from flux surface 1 of element 1 to flux surface 1 of element 2; $QPP=C \cdot \sigma \cdot (T_1^4 - T_2^4)$, $C=1/(\text{argument 1 value} + \text{argument 3 value} + (A_1/A_2) \cdot \text{argument 2 value})$; T is temperature, A is area of the element flux surface	1: (KH, jeh1) for flux surface 1 of element 1, (KH, jeh2) for flux surface 1 of element 2 (see NFLUXA in Block 10 data) 2: argument 1 IFN # = $(1-\epsilon_1)/\epsilon_1$, argument 2 IFN # = $(1-\epsilon_2)/\epsilon_2$, argument 3 IFN # = $1/f_{12}$; ϵ is emissivity, and f_{12} is the view factor from element 1 to element 2 flux surfaces
162	'RADIATIVE QPP, SURFACE 2' radiation from flux surface 2 of element 1 to flux surface 2 of element 2; $QPP=C \cdot \sigma \cdot (T_1^4 - T_2^4)$, $C=1/(\text{argument 1 value} + \text{argument 3 value} + (A_1/A_2) \cdot \text{argument 2 value})$; T is temperature, A is area of the element flux surface	1: (KH, jeh1) for flux surface 2 of element 1, (KH, jeh2) for flux surface 2 of element 2 (see NFLUXA in Block 10 data) 2: argument 1 IFN # = $(1-\epsilon_1)/\epsilon_1$, argument 2 IFN # = $(1-\epsilon_2)/\epsilon_2$, argument 3 IFN # = $1/f_{12}$; ϵ is emissivity, and f_{12} is the view factor from element 1 to element 2 flux surfaces
163	'RADIATIVE QPP, SURFACE 3' radiation from flux surface 3 of element 1 to flux surface 3 of element 2; $QPP=C \cdot \sigma \cdot (T_1^4 - T_2^4)$, $C=1/(\text{argument 1 value} + \text{argument 3 value} + (A_1/A_2) \cdot \text{argument 2 value})$; T is temperature, A is area of the element flux surface	1: (KH, jeh1) for flux surface 3 of element 1, (KH, jeh2) for flux surface 3 of element 2 (see NFLUXA in Block 10 data) 2: argument 1 IFN # = $(1-\epsilon_1)/\epsilon_1$, argument 2 IFN # = $(1-\epsilon_2)/\epsilon_2$, argument 3 IFN # = $1/f_{12}$; ϵ is emissivity, and f_{12} is the view factor from element 1 to element 2 flux surfaces
164	'CONDUCTIVE QPP, SURFACE 1' conduction from flux surface 1 of element 1 to flux surface 1 of element 2; $QPP=C \cdot (T_1 - T_2)$, $C=1/(\text{argument 1 value}/k_1 + (A_1/A_2) \cdot \text{argument 2 value}/k_2 + A_1 \cdot \text{argument 3 value})$; T is temperature, A is area of the element flux surface, k is element conductivity	1: (KH, jeh1) for flux surface 1 of element 1, (KH, jeh2) for flux surface 1 of element 2 (see NFLUXA in Block 10 data) 2: argument 1 IFN # = Δx_1 , argument 2 IFN # = Δx_2 , argument 3 IFN # = $1/(h \cdot A)_c$; Δx is conduction length, and $1/(h \cdot A)_c$ is the contact resistance between element 1 and element 2 flux surfaces
165	'CONDUCTIVE QPP, SURFACE 2' conduction from flux surface 2 of element 1 to flux surface 2 of element 2; $QPP=C \cdot (T_1 - T_2)$, $C=1/(\text{argument 1 value}/k_1 + (A_1/A_2) \cdot \text{argument 2 value}/k_2 + A_1 \cdot \text{argument 3 value})$; T is temperature, A is area of the element flux surface, k is element conductivity	1: (KH, jeh1) for flux surface 2 of element 1, (KH, jeh2) for flux surface 2 of element 2 (see NFLUXA in Block 10 data) 2: argument 1 IFN # = Δx_1 , argument 2 IFN # = Δx_2 , argument 3 IFN # = $1/(h \cdot A)_c$; Δx is conduction length, and $1/(h \cdot A)_c$ is the contact resistance between element 1 and element 2 flux surfaces
166	'CONDUCTIVE QPP, SURFACE 3' conduction from flux surface 3 of element 1 to flux surface 3 of element 2; $QPP=C \cdot (T_1 - T_2)$, $C=1/(\text{argument 1 value}/k_1 + (A_1/A_2) \cdot \text{argument 2 value}/k_2 + A_1 \cdot \text{argument 3 value})$; T is temperature, A is area of the element flux surface, k is element conductivity	1: (KH, jeh1) for flux surface 3 of element 1, (KH, jeh2) for flux surface 3 of element 2 (see NFLUXA in Block 10 data) 2: argument 1 IFN # = Δx_1 , argument 2 IFN # = Δx_2 , argument 3 IFN # = $1/(h \cdot A)_c$; Δx is conduction length, and $1/(h \cdot A)_c$ is the contact resistance between element 1 and element 2 flux surfaces

Mathematical Functions		supplemental variables
ID	Default Name/Description	description for each line
167	'RADIATIVE QPP, BC' radiation from a flux boundary of one structure to a flux boundary of another structure; $QPP=C \cdot \sigma \cdot (T_1^4 - T_2^4)$, $C=1/(\text{argument 1 value}+\text{argument 2 value}+\text{argument 3 value})$; T is temperature	1: (KH, end#) for the first flux boundary, (KH, end#) for the second flux boundary (see FLUXBCn in Block 10 data) 2: argument 1 IFN # = $(1-\epsilon_1)/\epsilon_1$, argument 2 IFN # = $(A_1/A_2) \cdot (1-\epsilon_2)/\epsilon_2$, argument 3 IFN # = $1/f_{12}$; ϵ is emissivity, and f_{12} is the view factor from first flux boundary to second flux boundary, A is area of the boundary flux surface
168	'CONDUCTIVE QPP, BC' conduction from a flux boundary of one structure to a flux boundary of another structure; $QPP=C \cdot (T_1 - T_2)$, $C=1/(\Delta x_1/k_1+(A_1/A_2) \cdot \Delta x_2/k_2+ A_1 \cdot \text{argument 1 value})$; T is temperature, Δx is conduction length, A is area of the flux boundary, k is element conductivity	1: (KH, end#) for the first flux boundary, (KH, end#) for the second flux boundary (see FLUXBCn in Block 10 data) 2: argument 1 IFN # = $1/(h \cdot A)_c$; $1/(h \cdot A)_c$ is the contact resistance between the first and second flux boundaries
169	currently not in use	-
170	currently not in use	-

NOTE: some of the supplemental variables are not explicitly defined in the Block data input; these variables (listed in lower case) are defined here:

- idgas: the gas ID number based on the order the gases are listed on line 19 of Block 2 data
- igas: the gas constituent number based on the order the gases are listed on line 18 of Block 2 data
- i: the fluid mechanics node ID number based on Block 3 data
- ilocal: the fluid mechanics local node number (either 1 or 2) based on Block 3 data
- jeh: the structure finite element number based on the order the elements are listed in Block 10 data
- jex: the exchange surface number for convection based on the order the exchange surfaces are listed in Block 10 data
- end: the boundary end number (1 or 2) for variables that are defined at the boundaries of heat transfer structures in Block 10 data
- ih: the structure node ID number based on Block 10 data
- igrp: the delayed neutron precursor group ID number based on the order the groups are listed in Block 13 data
- igrpd: the decay heat group ID number based on the order the groups are listed in Block 13 data
- krec: the feedback term number based on the order the feedback terms are listed in Block 13 data

Appendix C: Heat Transfer Coefficient Correlation Identification Numbers

Forced Convection		Internal Flow		
ID	Correlation ^{ref} ;Description[AUX]	Range	flow	fluid
1	Constant Surface Flux ¹ ; fully developed	$Pr > 0.6$	lam	gas/liq
2	Constant Surface Temperature ¹ ; fully developed	$Pr > 0.6$	lam	gas/liq
3	Hausen ² ; thermally fully developed with entrance effects [AUX=distance from entrance]	$Pr \gg 1$ or unheated entrance	lam	gas/liq
4	Modified Sieder & Tate ² ; with entrance effects [AUX=distance from entrance]	$Re \cdot Pr \cdot D/L > 10$ $0.48 < Pr < 16700$	lam	gas
5	Modified Sieder & Tate ² ; with entrance effects [AUX=distance from entrance]	$Re \cdot Pr \cdot D/L > 10$ $0.48 < Pr < 16700$	lam	liquid
6	Petukhov ² ; consistent with friction factor correlation for smooth surfaces	$1E4 < Re < 5E6$ $0.5 < Pr < 2000$	tur	gas
7	Petukhov ² ; consistent with friction factor correlation for smooth surfaces	$1E4 < Re < 5E6$ $0.5 < Pr < 2000$	tur	liquid
8	Colburn ²	$E4 < Re < E6$ $0.7 < Pr < 160$	tur	gas/liq
9	Dittus-Boelter ²	$E4 < Re < E6$ $0.7 < Pr < 160$ $Tw - T < 6$ liquids $Tw - T < 60$ gases	tur	gas/liq
10	Sieder & Tate ²	$1E4 < Re < 1E6$ $0.7 < Pr < 160$	tur	gas
11	Sieder & Tate ²	$1E4 < Re < 1E6$ $0.7 < Pr < 160$	tur	liquid
12	Taylor ³ ; with entrance effects [AUX=distance from entrance]	-	tur	gas
13	Nusselt ² ; with entrance effects	$10 < L/D < 400$	tur	gas/liq
14	Notter & Sleicher ⁴	$E4 < Re < E6$ $0.1 < Pr < E4$	tur	gas/liq
15	Organic Fluid Flow ⁵	-	tur	organ
16	Skupinski ¹ ; constant surface flux; liquid metals in smooth circular tubes	$3.6E3 < Re < 9E5$ $100 < Pe < 1E4$ $L/D > 10$	tur	liqmet
17	Seban & Shimazaki ¹ ; constant surface temperature; liquid metals in smooth circular tubes	$Pe > 100$ $L/D > 60$	tur	liqmet
18	Azer & Chao ² ; constant surface temperature; liquid metals in tubes	$Pe > 15000$ $Pr < 0.1$	tur	liqmet
19	Sliecher ² ; constant surface temperature; liquid metals in tubes	$Re < 5E5$ $0.004 < Pr < 0.1$	-	liqmet
20	currently not in use	-	-	-
21	currently not in use	-	-	-
22	currently not in use	-	-	-

Miscellaneous Convection		Internal/External Flow		
ID	Correlation ^{ref} ;Description[AUX]	Range	flow	fluid
23	Constant [AUX=heat transfer coefficient]	-	-	-
24	Generic 1; $h_{tc}=AUX \cdot T_{wall}-T_{fluid} $ [AUX=constant]	-	-	-
25	Generic 2; $h_{tc}=AUX \cdot T_{wall}-T_{fluid} ^{0.2}$ [AUX=constant]	-	-	-
26	Generic 3; $h_{tc}=AUX \cdot T_{wall}-T_{fluid} ^{0.25}$ [AUX=constant]	-	-	-
27	Generic 4; $h_{tc}=AUX \cdot (T_{wall}+T_{fluid})/2.0$ [AUX=constant]	-	-	-
28	Generic 5; $h_{tc}=AUX \cdot Re^{0.8}$ [AUX=constant]	-	-	-
29	Radiative; $h_{tc}=\sigma \cdot AUX \cdot (T_{wall}^2+T_{fluid}^2) \cdot (T_{wall}+T_{fluid})$ [AUX= ϵ for convex surface enclosed by large concave surface, or $AUX=1/((1-\epsilon_1)/\epsilon_1 + 1/f_{12} + (A_1/A_2) \cdot (1-\epsilon_2)/\epsilon_2)$; ϵ is the surface emissivity, f_{12} is the view factor from surface 1 to surface 2, σ is the Stephan-Boltzmann constant ($5.67E-8$ W/m ² ·K ⁴) and A is the surface area]	-	-	-
30	currently not in use	-	-	-
31	user-supplied htc #1; user must modify subroutine UHTC	-	-	-
32	user-supplied htc #2; user must modify subroutine UHTC	-	-	-
33	user-supplied htc #3; user must modify subroutine UHTC	-	-	-

Forced Convection		External Flow		
ID	Correlation ^{ref} ;Description/[AUX]	Range	flow	fluid
34	Denton ⁶ ; flow over packed bed with porosity=0.37	500<Re<5E4	-	gas/liq
35	Jeschar ⁷ ; flow over packed bed	250<Re<5.5E4	-	gas/liq
36	Achenbach ⁸ ; flow over packed bed	Re<3E5	-	gas/liq
37	Beek ⁹ ; heat transfer from wall of packed bed of cylinders to a gas	40<Re<2000	-	gas
38	Beek ⁹ ; heat transfer from wall of packed bed of spheres to a gas	40<Re<2000	-	gas
39	Packed Bed Flow ¹ ; flow over packed bed [AUX=1.0 for spheres, AUX=0.79 for cylinders with L/D=1, AUX= 0.71 for cubes]	90<Re<4000 Pr=0.7	-	gas
40	currently not in use	-	-	-
41	currently not in use	-	-	-
42	Flow Over A Sphere ¹⁰ ; modified for Pr variations	17<Re<7E5	-	gas
43	Whitaker ¹ ; flow over a sphere	3.5<Re<7.6E4 0.71<Pr<380	-	gas
44	Whitaker ¹ ; flow over a sphere	3.5<Re<7.6E4 0.71<Pr<380	-	liquid
45	Witte ⁹ ; liquid metal flow over a sphere	3.6E4<Re<2E5	-	liqmet
46	Ranz & Marshall ¹ ; convection from a free falling liquid drop	-	-	-
47	Modified Ranz & Marshall ¹ ; convection from a free falling liquid drop with oscillations [AUX=falling distance]	-	-	-
48	currently not in use	-	-	-
49	Hilpert ¹ ; cylinder in cross flow	0.4<Re<4E5	-	gas
50	Hilpert ¹ ; cylinder in cross flow	0.4<Re<4E5	-	liquid
51	Hilpert ¹ ; corner-on square in cross flow	5E3<Re<1E5	-	gas
52	Hilpert ¹ ; flat-on square in cross flow	5E3<Re<1E5	-	gas
53	Hilpert ¹ ; corner-on hexagon in cross flow	5E3<Re<1E5	-	gas
54	Hilpert ¹ ; flat-on hexagon in cross flow	5E3<Re<1E5	-	gas
55	Hilpert ¹ ; vertical flat plate in cross flow	4E3<Re<1.5E5	-	gas
56	Zhulkauskas ¹ ; cylinder in cross flow	1<Re<1E6 0.7<Pr<500	-	gas
57	Zhulkauskas ¹ ; cylinder in cross flow	1<Re<1E6 0.7<Pr<500	-	liquid
58	Churchill & Bernstein ¹ ; cylinder in cross flow	Re·Pr>0.2	-	gas
59	Churchill & Bernstein ¹ ; cylinder in cross flow	Re·Pr>0.2	-	liquid
60	Zhulkauskas ¹ ; aligned cylindrical tube bank in cross flow	1E3<Re<2E6 0.7<Pr<500	-	gas
61	Zhulkauskas ¹ ; aligned cylindrical tube bank in cross flow	1E3<Re<2E6 0.7<Pr<500	-	liquid
62	Zhulkauskas ¹ ; staggered cylindrical tube bank in cross flow [AUX=0.4 if St/SI>2 and AUX=0.35·(St/SI) ^{0.2} if St/SI<2; St is transverse pitch (⊥ to flow), SI is longitudinal pitch (to flow)]	1E3<Re<2E6 0.7<Pr<500	-	gas

Forced Convection		External Flow		
ID	Correlation ^{ref} ;Description[AUX]	Range	flow	fluid
63	Zhulkauskas ¹ ; staggered cylindrical tube bank in cross flow [AUX=0.4 if $St/SI > 2$ and $AUX=0.35 \cdot (St/SI)^{0.2}$ if $St/SI < 2$; St is transverse pitch (\perp to flow), SI is longitudinal pitch (\parallel to flow)]	$1E3 < Re < 2E6$ $0.7 < Pr < 500$	-	liquid
64	currently not in use	-	-	-
65	currently not in use	-	-	-
66	Flow Over a Flat Plate ¹ ; local, set EDH to the distance from plate leading edge to the center of the fluid mechanics element	$Pr > 0.6$	lam	gas
67	Flow Over a Flat Plate ¹ ; local, set EDH to the distance from plate leading edge to the center of the fluid mechanics element	$Pr > 0.6$	lam	liquid
68	Flow Over a Flat Plate ¹ ; average, set EDH to the plate length	$Pr > 0.6$	lam	gas
69	Flow Over a Flat Plate ¹ ; average, set EDH to the plate length	$Pr > 0.6$	lam	liquid
70	Flow Over a Flat Plate ¹ ; local, set EDH to the distance from plate leading edge to the center of the fluid mechanics element	$0.6 < Pr < 60$	tur	gas
71	Flow Over a Flat Plate ¹ ; local, set EDH to the distance from plate leading edge to the center of the fluid mechanics element	$0.6 < Pr < 60$	tur	liquid
72	Flow Over a Flat Plate ¹ ; average, set EDH to the plate length [AUX= $0.037 \cdot Re_c^{0.8} - 0.664 \cdot Re_c^{0.5}$; Re_c is the critical Reynolds number, usually = 5E5]	$5E5 < Re < 1E8$ $0.6 < Pr < 60$	lam/ tur	gas
73	Flow Over a Flat Plate ¹ ; average, set EDH to the plate length [AUX= $0.037 \cdot Re_c^{0.8} - 0.664 \cdot Re_c^{0.5}$; Re_c is the critical Reynolds number, usually = 5E5]	$5E5 < Re < 1E8$ $0.6 < Pr < 60$	lam/ tur	liquid
74	Liquid Metal Flow Over a Flat Plate ¹ ; local, set EDH to the distance from plate leading edge to the center of the fluid mechanics element	$Pr < 0.05$ $Pe > 100$	lam	liqmet
75	currently not in use	-	-	-

Free and Natural Convection		External Flow		
ID	Correlation ^{ref} ;Description/[AUX]	Range	flow	fluid
76	Churchill & Chu ¹ ; free convection from a vertical surface, or non-vertical surface if angle is less than 60° (measured from vertical) [AUX=L ³ ·cos(θ); L is surface length, and θ is the angle from vertical]	0<Ra<1E9 θ<60°	lam	gas
77	Churchill & Chu ¹ ; free convection from a vertical surface, or non-vertical surface if angle is less than 60° (measured from vertical) [AUX=β·L ³ ·cos(θ); β is volumetric coefficient of thermal expansion, L is surface length, and θ is the angle from vertical]	0<Ra<1E9 θ<60°	lam	liquid
78	Churchill & Chu ¹ ; free convection from a vertical surface [AUX=L ³ ; L is surface length]	all Ra	lam/ tur	gas
79	Churchill & Chu ¹ ; free convection from a vertical surface [AUX=β·L ³ ; β is volumetric coefficient of thermal expansion, and L is surface length]	all Ra	lam/ tur	liquid
80	McAdams ¹ ; free convection from a horizontal plate; either the upper surface of a heated plate or the lower surface of a cooled plate [AUX=L ³ ; L is surface length; use L=As/P, As is surface area and P is surface perimeter]	1E4<Ra<1E11	-	gas
81	McAdams ¹ ; free convection from a horizontal plate; either the upper surface of a heated plate or the lower surface of a cooled plate [AUX=β·L ³ ; β is volumetric coefficient of thermal expansion, L is surface length; use L=As/P, As is surface area and P is surface perimeter]	1E4<Ra<1E11	-	liquid
82	McAdams ¹ ; free convection from a horizontal plate; either the upper surface of a cooled plate or the lower surface of a heated plate [AUX=L ³ ; L is surface length; use L=As/P, As is surface area and P is surface perimeter]	1E5<Ra<1E10	-	gas
83	McAdams ¹ ; free convection from a horizontal plate; either the upper surface of a cooled plate or the lower surface of a heated plate [AUX=β·L ³ ; β is volumetric coefficient of thermal expansion, L is surface length; use L=As/P, As is surface area and P is surface perimeter]	1E5<Ra<1E10	-	liquid
84	Morgan ¹ ; free convection from a horizontal cylinder [AUX=D ³ ; D is diameter of cylinder]	1E-10<Ra<1E12	-	gas
85	Morgan ¹ ; free convection from a horizontal cylinder [AUX=β·D ³ ; β is volumetric coefficient of thermal expansion, and D is diameter of cylinder]	1E-10<Ra<1E12	-	liquid

Free and Natural Convection		External Flow		
ID	Correlation ^{ref} ;Description/[AUX]	Range	flow	fluid
86	Churchill & Chu ¹ ; free convection from a horizontal cylinder [AUX=D ³ ; D is diameter of cylinder]	1E-5<Ra<1E12	-	gas
87	Churchill & Chu ¹ ; free convection from a horizontal cylinder [AUX=β·D ³ ; β is volumetric coefficient of thermal expansion, and D is diameter of cylinder]	1E-5<Ra<1E12	-	liquid
88	Churchill & Chu ¹ ; free convection from a sphere [AUX=D ³ ; D is diameter of sphere]	Ra>1E11 Pr>0.7	-	gas
89	Churchill & Chu ¹ ; free convection from a sphere [AUX=β·D ³ ; β is volumetric coefficient of thermal expansion, and D is diameter of sphere]	Ra>1E11 Pr>0.7	-	liquid
90	currently not in use	-	-	-

Appendix C References

1. Incropera, F. P. and D. P. DeWitt, Introduction to Heat Transfer, (John Wiley & Sons, Inc., 1985).
2. Chapman, A. J., Fundamentals of Heat Transfer, (Macmillan Publishing Company, 1987).
3. Taylor, M. F., Correlation of Local Heat-Transfer Coefficients for Single-Phase Turbulent Flow of Hydrogen in Tubes with Temperature Ratios to 23, NASA TN D-4332, Jan. 1968.
4. Ozisik, M. N., Basic Heat Transfer, (McGraw-Hill, 1977).
5. El-Wakil, M. M., Nuclear Heat Transport, (The American Nuclear Society, 1978).
6. Eckert, E. R. G. and R. M. Drake, Analysis of Heat and Mass Transfer, (Hemisphere Publishing Corporation, 1987).
7. J. Radestock and R. Jeschar, "Study of the Flow Through the Column of Charges in the Blast Furnace," Stahl u. Eisen, 90, (1970, p.1249).
8. Achenbach, E., "Heat Transfer and Pressure Drop of Pebble Beds Up to High Reynolds Number," Int. Heat Transfer Conf, (Munich, 1982).
9. Kreith, F. and M. S. Bohn, Principles of Heat Transfer, (Harper & Row, Fourth Edition, 1986).
10. Karlekar, B. V. and R. M. Desmond, Engineering Heat Transfer, (West Publishing Company, 1977).

Appendix D: Sample User-Defined Subroutines

This appendix provides examples of user-defined subroutines and should be used as a guide for creating specific-purpose programming for use within SAFSIM.

```

1 C *****
2 C
3 C THIS FILE CONTAINS SUBROUTINES AND FUNCTIONS THAT CAN BE MODIFIED
4 C BY THE USER TO MEET SPECIFIC NEEDS.
5 C MANY VARIABLES IN THE CALLING STATEMENTS HAVE BEEN PROVIDED TO
6 C PROVIDE FLEXIBILITY TO THE USER. MANY OF THESE MAY NOT BE USED AND
7 C WILL GENERATE A COMPILER WARNING THAT SHOULD BE IGNORED.
8 C
9 C LIST OF FUNCTIONS AND SUBROUTINES THAT CAN BE MODIFIED BY THE USER:
10 C FUNCTION DESCRIPTION
11 C UFUN FUNCTIONS FOR USE WITH FUNCTION CONTROLLED VARIABLES
12 C UEOS EQUATION OF STATE FOR CALCULATION OF FLUID DENSITY
13 C UCP CONSTANT PRESSURE SPECIFIC HEAT FOR FLUID
14 C UCV CONSTANT VOLUME SPECIFIC HEAT FOR FLUID
15 C UCOND CONDUCTIVITY FOR FLUID
16 C UVIS VISCOSITY FOR FLUID
17 C SUBROUTINE DESCRIPTION
18 C UHTC HEAT TRANSFER COEFFICIENTS
19 C UPRIN PROGRAMMED REACTIVITY INPUT FOR REACTOR DYNAMICS
20 C UPRINIT PROGRAMMED REACTIVITY INITIALIZATION FOR REACTOR DYNAMICS
21 C UPR PROGRAMMED REACTIVITY CALCULATIONS
22 C UIN SPECIAL-PURPOSE INPUT
23 C UOUT SPECIAL-PURPOSE OUTPUT
24 C
25 C
26 C *****
27 C
28 C
29 C FUNCTION UFUN(IUSN,IU,NAF)
30 C
31 C EVALUATE USER-SUPPLIED FUNCTIONS
32 C
33 C CALLED BY SUBROUTINE(S): FUP
34 C
35 C CALLS SUBROUTINE(S): NONE
36 C
37 C CALLS FUNCTIONS: NONE
38 C
39 C IMPLICIT REAL*8 (A-H,O-Z)
40 C
41 C PARAMETER (MAXFCV=200,MAXF=200,MAXFCON=25,MAXFTAB=25,MAXFPOL=25)
42 C PARAMETER (MAXPAIR=30,MAXORP1=5)
43 C PARAMETER (MAXFTAV=10,MAXFTE=100)
44 C PARAMETER (MAXFDAV=10,MAXFDE=100)
45 C
46 C CHARACTER FNAME*28
47 C
48 C COMMON/FDATA/IFN(MAXF),IFID(MAXF),IPNTRF(MAXF,6),FNAME(MAXF),
49 C + FM(MAXF),FVALUE(0-MAXF:MAXF),FCON(MAXFCON),LF(MAXF,4),
50 C + FTABX(MAXFTAB,MAXPAIR),FTABY(MAXFTAB,MAXPAIR),
51 C + FPOL(MAXFPOL,MAXORP1),IAFNF(MAXF,6),
52 C + IHSN(MAXFTAV,MAXFTE),IEN(MAXFTAV,MAXFTE),
53 C + TWF(MAXFTAV,MAXFTE),INWN(MAXFDAV,MAXFDE),
54 C + IFEN(MAXFDAV,MAXFDE),DWF(MAXFDAV,MAXFDE),
55 C + NFSYS,NFFM,NFHT,NFRD
56 C
57 C
58 C IUSN IS USER FUNCTION NUMBER 1,2,3,4, OR 5
59 C IU IS THE FUNCTION ID NUMBER AND
60 C NAF IS THE NUMBER OF ARGUMENT FUNCTIONS
61 C ARGUMENT FUNCTION ID NUMBERS ARE STORED IN IA=IPNTRF(IU,L+1),
62 C WHERE L=1 THRU NAF (5 MAX).
63 C
64 C GO TO (101,102,103,104,105), IUSN
65 C 101 CONTINUE
66 C dRHO/dT FOR THE Be MODERATOR BASED ON A CURVE FIT OF DATA;
67 C FVALUE(IA) IS THE AVERAGE MODERATOR TEMPERATURE.
68 C IA=IPNTRF(IU,NAF+1)
69 C UFUN=-9.76D-5+4.4398E-8*FVALUE(IA)+(0.040958199D0+
70 C + 1.175074577D0/FVALUE(IA))/FVALUE(IA)
71 C RETURN
72 C 102 CONTINUE
73 C RETURN
74 C 103 CONTINUE
75 C RETURN
76 C 104 CONTINUE
77 C RETURN
78 C 105 CONTINUE
79 C
80 C RETURN
81 C END

```

```
82 C
83 C *****
84 C
85 C     SUBROUTINE UHTC(IUS,DF,DH,AUX,PF,FF,RE,PR,CD,V,D,TW,T,AGRAV,USH)
86 C
87 C USER-SUPPLIED SUBROUTINE TO IMPLEMENT SPECIAL HEAT TRANSFER COEFFICIENT
88 C CORRELATIONS; THE USER CAN MODIFY THIS SUBROUTINE AS DESIRED
89 C
90 C DEFINITIONS:
91 C IUS IS THE USER-SUPPLIED CORRELATION NUMBER: 1,2, OR 3
92 C DF IS THE CHARACTERISTIC LENGTH FOR FLOW
93 C DH IS THE CHARACTERISTIC LENGTH FOR HEAT TRANSFER
94 C AUX IS ANY AUXILIARY DATA REQUIRED BY THE CORRELATION
95 C PF IS THE PACKING FRACTION FOR POROUS MEDIA FLOW ELEMENTS
96 C FF IS THE FRICTION FACTOR
97 C RE IS THE REYNOLDS NUMBER BASED ON DH
98 C PR IS THE PRANDTL NUMBER
99 C CD IS FLUID CONDUCTIVITY
100 C V IS FLUID VISCOSITY
101 C D IS FLUID DENSITY
102 C TW IS THE WALL TEMPERATURE
103 C T IS THE FLUID TEMPERATURE
104 C AGRAV IS THE ACCELERATION DUE TO GRAVITY
105 C USH IS THE CALCULATED USER-SUPPLIED HEAT TRANSFER COEFFICIENT
106 C
107 C CALLED BY SUBROUTINE(S): NONE
108 C CALLED BY FUNCTION(S): HTCCOR
109 C
110 C CALLS SUBROUTINE(S):
111 C
112 C
113 C     IMPLICIT REAL*8 (A-H,O-Z)
114 C
115 C     GO TO (1,2,3), IUS
116 C     USH=1.0D-12
117 C     1 CONTINUE
118 C EXAMPLE:
119 C MIXED CONVECTION IN A HORIZONTAL PIPE (TURBULENT GAS FLOW);
120 C AUX MUST BE ENTERED AS THE LENGTH
121 C CHAPMAN, P.395
122 C     VF=V*(TF/T)**0.7
123 C     TF=(T+TW)*0.5D0
124 C     D=D*T/TF
125 C     GR=AGRAV*(AUX**3)*(D**2)*DABS(TW-T)/(TF*VF**2)
126 C     USH=CD/DH*4.69*(RE**0.27)*(PR**0.21)*(GR**0.07)*(DH/AUX)**0.36
127 C     RETURN
128 C
129 C     2 CONTINUE
130 C PLACE USER-SUPPLIED FORTRAN HERE
131 C     USH=1.0D-12
132 C     RETURN
133 C
134 C     3 CONTINUE
135 C PLACE USER-SUPPLIED FORTRAN HERE
136 C     USH=1.0D-12
137 C
138 C
139 C     RETURN
140 C     END
141 C
142 C *****
143 C
144 C     SUBROUTINE UPRIN(KR)
145 C
146 C USER-SUPPLIED SUBROUTINE TO READ IN DATA TO BE USED IN THE CALCULATION
147 C OF PROGRAMMED REACTIVITY. KR IS THE REACTOR #.
148 C THIS SUBROUTINE IS MEANT TO BE MODIFIED BY THE USER TO IMPLEMENT
149 C CAPABILITIES SPECIFIC TO HIS NEEDS. EXAMPLES ARE PROVIDED TO GUIDE
150 C THE USER IN ADDING HIS OWN PROGRAM. USE ARRAYS IN COMMON BLOCK PRDATA.
151 C SEE ALSO: UPRINIT, UPR
152 C
153 C CALLED BY SUBROUTINE(S): INPUT
154 C CALLED BY FUNCTION(S): NONE
155 C
156 C CALLS SUBROUTINE(S):
157 C
158 C
159 C     IMPLICIT REAL*8 (A-H,O-Z)
160 C
161 C     PARAMETER (MAXDNG=25,MAXDHG=15,MAXRX=3,MAXREAC=10)
162 C     PARAMETER (MAXPAIR=30)
```

```
163 C
164 CHARACTER RXNAME*24,FBNAME*12
165 CHARACTER RESTART*4
166 C
167 COMMON/EXEC1/TSTART,TEND,TIME,TIMESTEP,DELT,PRNTINT,ISTEP,
168 + ISTART,IPRINT,TIMPRNT,NPLT,NDUMP,RESTART
169 COMMON/RDDATA/NRX,NDNG(MAXRX),NDHG(MAXRX),RXNAME(MAXRX),
170 + BETA(MAXRX),OMEFAC(MAXRX),GENTIME(MAXRX),
171 + RDN(MAXRX),SOURCE(MAXRX),XLAM(MAXDNG,MAXRX),
172 + BETAG(MAXDNG,MAXRX),CDN(MAXDNG,MAXRX),ISRD(MAXRX),
173 + XLAMH(MAXDHG,MAXRX),EFACG(MAXDHG,MAXRX),
174 + CDH(MAXDHG,MAXRX),NREAC(MAXRX),DREAC(MAXREAC,MAXRX),
175 + REACT(MAXRX),DTRDMIN(MAXRX),DRERRR(MAXRX),
176 + DELTR(MAXRX),NPRNTRD(MAXRX),RDNE(MAXRX),RDNO(MAXRX),
177 + DELTRL(MAXRX),CDNO(MAXDNG,MAXRX),CDHO(MAXDHG,MAXRX),
178 + DTRDMAX(MAXRX),RDNEO(MAXRX),RDST1(MAXRX),PRI(MAXRX),
179 + AUXRD(3,MAXRX),KSOLVE(MAXRX),RERRR(MAXRX),
180 + DREACI(MAXREAC,MAXRX),PERINV(MAXRX),PERINVE(MAXRX),
181 + DREACIT(MAXRX),FBNAME(MAXREAC,MAXRX)
182 COMMON/PRDATA/KOFTPR(MAXRX),NPAIRSPR(MAXRX),DRHODT(MAXRX),
183 + XTPR(MAXRX,MAXPAIR),YPR(MAXRX,MAXPAIR),IPR(MAXRX)
184 COMMON/RDCONS/CONN(MAXRX),CONC(MAXDNG,MAXRX),FC(15),FCA(9)
185 C
186 C
187 GO TO (101,102), NPAIRSPR(KR)
188 101 CONTINUE
189 C MIT-SNL MINIMUM TIME CONTROL LAW #1:
190 C INPUT: TARGET POWER, TARGET POWER/INITIAL POWER,
191 C TARGET TIME INTERVAL, A CONSTANT, MAX dRHO/dt
192 C TARGET POWER IS THE DESIRED POWER TO BE REACHED WITHIN THE TARGET
193 C TIME INTERVAL. THE CONSTANT EQUALS N*DT WHERE N IS THE NUMBER OF
194 C TIMES TO UPDATE INVERSE PERIOD RATE PER CONTROL LAW CYCLE TIME, DT.
195 C THIS IS BASED ON DIGITAL IMPLIMENTATION OF THE CONTROL LAW.
196 C MAX dRHO/dt IS MAX ALLOWED INSERTION RATE IN DIMENSIONLESS UNITS
197 READ(5,*)XTPR(KR,1),XTPR(KR,2),XTPR(KR,3),XTPR(KR,4),XTPR(KR,5)
198 RETURN
199 102 CONTINUE
200 C MIT-SNL MINIMUM TIME CONTROL LAW #2:
201 C INPUT(LINE1): TARGET POWER, TARGET POWER/INITIAL POWER,
202 C TARGET TIME INTERVAL, A CONSTANT, MAX ROTATION RATE
203 C INPUT(LINE2): INITIAL DRUM POSTION
204 C TARGET POWER IS THE DESIRED POWER TO BE REACHED WITHIN THE TARGET
205 C TIME INTERVAL. THE CONSTANT EQUALS N*DT WHERE N IS THE NUMBER OF
206 C TIMES TO UPDATE INVERSE PERIOD RATE PER CONTROL LAW CYCLE TIME, DT.
207 C THIS IS BASED ON DIGITAL IMPLIMENTATION OF THE CONTROL LAW.
208 C MAX ROTATION RATE IS MAX ALLOWED DRUM ROTATION RATE IN radians/s
209 C INITIAL DRUM POSITION (radians)
210 C NOTE: A POSITIVE ROTATION IS CLOCKWISE AND GIVES A NEGATIVE
211 C REACTIVITY INSERTION (ROTATES POISON TOWARD CORE)
212 C ALSO, AUXRD(1,KR) IS THE MAX DRUM REACTIVITY WORTH WHICH CAN BE
213 C FUNCTION CONTROLLED
214 READ(5,*)XTPR(KR,1),XTPR(KR,2),XTPR(KR,3),XTPR(KR,4),XTPR(KR,5)
215 C INITIAL DRUM POSITION (radians)
216 READ(5,*)YPR(KR,2)
217 C
218 C
219 RETURN
220 END
221 C
222 C *****
223 C
224 SUBROUTINE UPRINIT(KR)
225 C
226 C USER-SUPPLIED SUBROUTINE TO INITIALIZE DATA TO BE USED IN THE
227 C CALCULATION OF PROGRAMMED REACTIVITY. KR IS THE REACTOR #.
228 C THIS SUBROUTINE IS MEANT TO BE MODIFIED BY THE USER TO IMPLEMENT
229 C CAPABILITIES SPECIFIC TO HIS NEEDS. EXAMPLES ARE PROVIDED TO GUIDE
230 C THE USER IN ADDING HIS OWN PROGRAM. USE ARRAYS IN COMMON BLOCK PRDATA.
231 C SEE ALSO: UPRIN, UPR
232 C
233 C CALLED BY SUBROUTINE(S): INIT
234 C CALLED BY FUNCTION(S): NONE
235 C
236 C CALLS SUBROUTINE(S):
237 C
238 C
239 IMPLICIT REAL*8 (A-H,O-Z)
240 C
241 PARAMETER (MAXDNG=25,MAXDHG=15,MAXRX=3,MAXREAC=10)
242 PARAMETER (MAXPAIR=30)
243 C
```

```
244 CHARACTER RXNAME*24,FBNAME*12
245 CHARACTER RESTART*4
246 C
247 COMMON/EXEC1/TSTART,TEND,TIME,TIMESTEP,DELT,PRNTINT,ISTEP,
248 + ISTART,IPRINT,TIMPRNT,NPLT,NDUMP,RESTART
249 COMMON/RDDATA/NRX,NDNG(MAXRX),NDHG(MAXRX),RXNAME(MAXRX),
250 + BETA(MAXRX),OMEFRAC(MAXRX),GENTIME(MAXRX),
251 + RDN(MAXRX),SOURCE(MAXRX),XLAM(MAXDNG,MAXRX),
252 + BETAG(MAXDNG,MAXRX),CDN(MAXDNG,MAXRX),ISR(D,MAXRX),
253 + XLAMH(MAXDHG,MAXRX),EFRACG(MAXDHG,MAXRX),
254 + CDH(MAXDHG,MAXRX),NREAC(MAXRX),DREAC(MAXREAC,MAXRX),
255 + REACT(MAXRX),DTRDMIN(MAXRX),DRERRR(MAXRX),
256 + DELTR(MAXRX),NPRNTRD(MAXRX),RDNE(MAXRX),RDNO(MAXRX),
257 + DELTRL(MAXRX),CDNO(MAXDNG,MAXRX),CDHO(MAXDHG,MAXRX),
258 + DTRDMAX(MAXRX),RDNEO(MAXRX),RDST1(MAXRX),PRI(MAXRX),
259 + AUXRD(3,MAXRX),KSOLVE(MAXRX),RERRR(MAXRX),
260 + DREACI(MAXREAC,MAXRX),PERINV(MAXRX),PERINVE(MAXRX),
261 + DREACIT(MAXRX),FBNAME(MAXREAC,MAXRX)
262 COMMON/PRDATA/KOPTPR(MAXRX),NPAIRSPR(MAXRX),DRHODT(MAXRX),
263 + XTPR(MAXRX,MAXPAIR),YPR(MAXRX,MAXPAIR),IPR(MAXRX)
264 COMMON/RDCONS/CONN(MAXRX),CONC(MAXDNG,MAXRX),FC(15),FCA(9)
265 C
266 C
267 GO TO (101,102),NPAIRSPR(KR)
268 CONTINUE
269 C 101 MIT-SNL MINIMUM TIME CONTROL LAW #1:
270 C DESIRED INVERSE PERIOD
271 YPR(KR,1)=DLOG(XTPR(KR,2))/XTPR(KR,3)
272 RETURN
273 C 102 CONTINUE
274 C MIT-SNL MINIMUM TIME CONTROL LAW #2:
275 C DESIRED INVERSE PERIOD
276 YPR(KR,1)=DLOG(XTPR(KR,2))/XTPR(KR,3)
277 C YPR(KR,3) IS THE INITIAL DRUM WORTH BASED ON INITIAL ANGLE, YPR(KR,2)
278 YPR(KR,3)=AUXRD(1,KR)*DCOS(YPR(KR,2))
279 C
280 C
281 RETURN
282 END
283 C
284 C *****
285 C
286 SUBROUTINE UPR(KR,PR,DT)
287 C
288 C USER-SUPPLIED SUBROUTINE TO BE USED FOR THE CALCULATION
289 C OF PROGRAMMED REACTIVITY. KR IS THE REACTOR #, DELTR IS THE
290 C PREVIOUS REACTOR DYNAMICS TIMESTEP AND PR IS THE PROGRAMMED
291 C TIME-DERIVATIVE REACTIVITY TO BE DETERMINED (dRHO/dt).
292 C THIS SUBROUTINE IS MEANT TO BE MODIFIED BY THE USER TO IMPLEMENT
293 C CAPABILITIES SPECIFIC TO HIS NEEDS. EXAMPLES ARE PROVIDED TO GUIDE
294 C THE USER IN ADDING HIS OWN PROGRAM.
295 C SEE ALSO: UPRIN, UPRINIT
296 C
297 C CALLED BY SUBROUTINE(S): NONE
298 C CALLED BY FUNCTION(S): PROGDR
299 C
300 C CALLS SUBROUTINE(S):
301 C
302 C
303 IMPLICIT REAL*8 (A-H,O-Z)
304 C
305 PARAMETER (MAXDNG=25,MAXDHG=15,MAXRX=3,MAXREAC=10)
306 PARAMETER (MAXPAIR=30)
307 PARAMETER (SMALLER=1.0D-16)
308 C
309 CHARACTER RXNAME*24,FBNAME*12
310 CHARACTER RESTART*4
311 C
312 COMMON/EXEC1/TSTART,TEND,TIME,TIMESTEP,DELT,PRNTINT,ISTEP,
313 + ISTART,IPRINT,TIMPRNT,NPLT,NDUMP,RESTART
314 COMMON/RDDATA/NRX,NDNG(MAXRX),NDHG(MAXRX),RXNAME(MAXRX),
315 + BETA(MAXRX),OMEFRAC(MAXRX),GENTIME(MAXRX),
316 + RDN(MAXRX),SOURCE(MAXRX),XLAM(MAXDNG,MAXRX),
317 + BETAG(MAXDNG,MAXRX),CDN(MAXDNG,MAXRX),ISR(D,MAXRX),
318 + XLAMH(MAXDHG,MAXRX),EFRACG(MAXDHG,MAXRX),
319 + CDH(MAXDHG,MAXRX),NREAC(MAXRX),DREAC(MAXREAC,MAXRX),
320 + REACT(MAXRX),DTRDMIN(MAXRX),DRERRR(MAXRX),
321 + DELTR(MAXRX),NPRNTRD(MAXRX),RDNE(MAXRX),RDNO(MAXRX),
322 + DELTRL(MAXRX),CDNO(MAXDNG,MAXRX),CDHO(MAXDHG,MAXRX),
323 + DTRDMAX(MAXRX),RDNEO(MAXRX),RDST1(MAXRX),PRI(MAXRX),
324 + AUXRD(3,MAXRX),KSOLVE(MAXRX),RERRR(MAXRX),
```

```

325 + DREACI(MAXREAC,MAXRX),PERINV(MAXRX),PERINVE(MAXRX),
326 + DREACIT(MAXRX),FBNAME(MAXREAC,MAXRX)
327 COMMON/PRDATA/KOPTPR(MAXRX),NPAIRSPR(MAXRX),DRHODT(MAXRX),
328 + XTPR(MAXRX,MAXPAIR),YPR(MAXRX,MAXPAIR),IPR(MAXRX)
329 COMMON/RDCONS/CONN(MAXRX),CONC(MAXDNG,MAXRX),FC(15),FCA(9)
330 C
331 C
332 GO TO (101,102,103), NPAIRSPR(KR)
333 PR=0.000
334 C
335 C
336 101 CONTINUE
337 C
338 C MIT-SNL MINIMUM TIME CONTROL LAW #1:
339 C
340 C XTPR(KR,1) IS THE TARGET POWER
341 C XTPR(KR,2) IS THE TARGET POWER/INITIAL POWER
342 C XTPR(KR,3) IS THE TARGET TIME INTERVAL
343 C XTPR(KR,4) IS A CONSTANT THAT EQUALS N*DT WHERE N IS THE NUMBER OF
344 C TIMES TO UPDATE INVERSE PERIOD RATE PER CONTROL LAW CYCLE TIME, DT.
345 C THIS IS BASED ON DIGITAL IMPLIMENTATION OF THE CONTROL LAW.
346 C XTPR(KR,5) IS THE MAXIMUM-ALLOWED REACTIVITY INSERTION RATE (/s)
347 C YPR(KR,1) IS THE INITIAL INVERSE PERIOD; USE THIS VALUE UNTIL
348 C TARGET POWER IS ACHIEVED AND THEN USE INVERSE PERIOD OF ZERO.
349 C
350 IF(RDN(KR).LT.0.99D0*XTPR(KR,1))THEN
351 WH=YPR(KR,1)
352 ELSE
353 WH=1.0D2*((1.0D0-RDN(KR)/XTPR(KR,1))*YPR(KR,1)
354 ENDIF
355 C DETERMINE PRECURSOR-WEIGHTED EFFECTIVE MULTIGROUP DECAY PARAMETER
356 XLAMP=0.000
357 D=0.000
358 DO 5 KN=1,NDNG(KR)
359 C=CND(KN,KR)*XLAM(KN,KR)
360 D=D+C
361 XLAMP=XLAMP+C*XLAM(KN,KR)
362 5 CONTINUE
363 XLAMP=XLAMP/DMAX1(D,SMALLER)
364 C DETERMINE CORRECTIVE TIME RATE OF CHANGE OF INVERSE PERIOD
365 WDOT=(WH-PERINV(KR))/XTPR(KR,4)
366 C NOW ESTIMATE NEW dRHO/dt
367 SUMBL=0.000
368 DO 10 KN=1,NDNG(KR)
369 SUMBL=SUMBL+BETAG(KN,KR)*(XLAM(KN,KR)-XLAMP)
370 10 CONTINUE
371 DPR=(BETA(KR)-REACT(KR))*WH-XLAMP*REACT(KR)-SUMBL+GENTIME(KR)*
372 + (WDOT+WH*(WH+XLAMP))
373 C LIMIT dRHO/dt TO MAX ALLOWED
374 PR=DMIN1(DABS(DPR),XTPR(KR,5))
375 PR=DSIGN(PR,DPR)
376 RETURN
377 C
378 C
379 C
380 102 CONTINUE
381 C
382 C MIT-SNL MINIMUM TIME CONTROL LAW #2 (INCLUDES DRUM ROTATIONS):
383 C
384 C XTPR(KR,1) IS THE TARGET POWER
385 C XTPR(KR,2) IS THE TARGET POWER/INITIAL POWER
386 C XTPR(KR,3) IS THE TARGET TIME INTERVAL
387 C XTPR(KR,4) IS A CONSTANT THAT EQUALS N*DT WHERE N IS THE NUMBER OF
388 C TIMES TO UPDATE INVERSE PERIOD RATE PER CONTROL LAW CYCLE TIME, DT.
389 C THIS IS BASED ON DIGITAL IMPLIMENTATION OF THE CONTROL LAW.
390 C XTPR(KR,5) IS THE MAXIMUM-ALLOWED DRUM ROTATION RATE (radians/s)
391 C YPR(KR,1) IS THE INITIAL INVERSE PERIOD; USE THIS VALUE UNTIL
392 C TARGET POWER IS ACHIEVED AND THEN USE INVERSE PERIOD OF ZERO.
393 C YPR(KR,2) IS THE DRUM ROTATION ANGLE IN radians
394 C NOTE: A POSITIVE ROTATION IS CLOCKWISE AND GIVES A NEGATIVE
395 C REACTIVITY INSERTION (ROTATES POISON TOWARD CORE)
396 C YPR(KR,3) IS THE CURRENT DRUM WORTH
397 C AUXRD(1,KR) IS THE MAX DRUM WORTH WHICH CAN BE FUNCTION CONTROLLED
398 C
399 IF(RDN(KR).LT.0.99D0*XTPR(KR,1))THEN
400 WH=YPR(KR,1)
401 ELSE
402 WH=1.0D2*((1.0D0-RDN(KR)/XTPR(KR,1))*YPR(KR,1)
403 ENDIF
404 C DETERMINE PRECURSOR-WEIGHTED EFFECTIVE MULTIGROUP DECAY PARAMETER
405 XLAMP=0.000
    
```

```

406      D=0.000
407      DO 15 KN=1,NDNG(KR)
408          C=CDN(KN,KR)*XLAM(KN,KR)
409          D=D+C
410          XLAMP=XLAMP+C*XLAM(KN,KR)
411      15 CONTINUE
412          XLAMP=XLAMP/DMAX1(D,SMALLER)
413 C DETERMINE CORRECTIVE TIME RATE OF CHANGE OF INVERSE PERIOD
414      WDOT=(WH-PERINV(KR))/XTPR(KR,4)
415 C NOW ESTIMATE NEW DRHO/dt
416      SUMBL=0.000
417      DO 20 KN=1,NDNG(KR)
418          SUMBL=SUMBL+BETAG(KN,KR)*(XLAM(KN,KR)-XLAMP)
419      20 CONTINUE
420      DPR=(BETA(KR)-REACT(KR))*WH-XLAMP*REACT(KR)-SUMBL+GENTIME(KR)*
421      + (WDOT+WH*(WH+XLAMP))
422 C DETERMINE DRHO/dTHETA
423      DRDTH=-AUXRD(1,KR)*DSIN(YPR(KR,2))
424 C DETERMINE DESIRED DRUM ROTATION RATE
425      THDOT=DPR/DRDTH
426 C LIMIT THDOT TO MAX ALLOWED
427      THDOTL=DMIN1(DABS(THDOT),XTPR(KR,5))
428      THDOT=DSIGN(THDOTL,THDOT)
429 C DETERMINE NEW DRUM POSITION
430      YPR(KR,2)=YPR(KR,2)+THDOT*DT
431 C LIMIT THETA TO GT 0.5 DEGREES AND LT 179.5 DEGREES
432      YPR(KR,2)=DMIN1(3.13287D0,YPR(KR,2))
433      YPR(KR,2)=DMAX1(0.00873D0,YPR(KR,2))
434      ROLD=YPR(KR,3)
435      YPR(KR,3)=AUXRD(1,KR)*DCOS(YPR(KR,2))
436      PR=(YPR(KR,3)-ROLD)/DT
437      RETURN
438 C
439 C
440 C
441 103 CONTINUE
442 C
443 C
444 C
445      RETURN
446      END
447 C
448 C *****
449 C
450      SUBROUTINE UIN(IUERR)
451 C
452 C THIS SUBROUTINE IS USEFUL FOR SPECIFYING SPECIAL INPUT DATA TO BE
453 C USED IN SUBROUTINE UOUT WHICH ALLOWS SPECIAL OUTPUT DATA TO BE
454 C WRITTEN TO FILE SAFSIM.OU (UNIT 15) OR ANY OTHER UNIT THE USER DESIRES.
455 C UOUT IS CALLED EVERY PRINTING TIME STEP ITERATIONS. IF IUERR IS SET TO
456 C 74, EXECUTION WILL BE TERMINATED. BE SURE TO INCLUDE AND END LINE AT
457 C END OF BLOCK DATA 18.
458 C
459 C CALLED BY SUBROUTINE(S): INPUT
460 C CALLS SUBROUTINE(S): ?
461 C
462      IMPLICIT REAL*8 (A-H,O-Z)
463 C
464      CHARACTER DUMMY*126
465 C
466 C PLACE ANY USER-DEFINED COMMON BLOCKS HERE <-----
467      COMMON/UDATA1/ISPOUT
468 C
469 C
470      IUERR=0
471      ISPOUT=0
472 C
473      5 CONTINUE
474      READ(5,1000,END=6)DUMMY
475 C CHECK FOR END OF INPUT DATA BLOCK
476      IF(DUMMY(1:3).EQ.'END'.OR.DUMMY(1:3).EQ.'end')GO TO 6
477 C IF NOT THE END, REREAD LINE FOR DATA
478      BACKSPACE (5)
479 C
480 C PLACE DESIRED FORTRAN READ STATEMENTS HERE <-----
481      READ(5,*)ISPOUT
482 C
483      GO TO 5
484 C
485      6 CONTINUE
486 C

```

```
487 1000  FORMAT(A)
488 C
489         RETURN
490         END
491 C
492 C *****
493 C
494         SUBROUTINE UOUT
495 C
496 C THIS SUBROUTINE IS USEFUL FOR SPECIFYING SPECIAL OUTPUT DATA TO BE
497 C WRITTEN TO FILE SAFSIM.OU (UNIT 15) OR ANY OTHER UNIT THE USER DESIRES.
498 C UOUT IS CALLED EVERY PRNTINT TIME STEP ITERATIONS.
499 C
500 C CALLED BY SUBROUTINE(S): INPUT
501 C CALLS SUBROUTINE(S): ?
502 C
503 C
504         IMPLICIT REAL*8 (A-H,O-Z)
505 C
506         PARAMETER (MAXEL=500,MAXN=500,MAXSETS=25,MAXC=10)
507         PARAMETER (MAXNOZ=25,MAXCMPSR=25,MAXDF=50)
508         PARAMETER (MAXELS=MAXEL/2+1,MAXNS=MAXN/2+1,MAXES=25)
509         PARAMETER (BIG=1.0D12,SMALL=1.0D-12)
510         PARAMETER (MAXNTR=4,MAXG=4,NGEL=MAXG*MAXEL)
511         PARAMETER (MAXNW=5,MAXSBW=MAXN/2+1)
512         PARAMETER (JFLAG1=5,JFLAG2=25)
513         PARAMETER (UGAS=8314.41D0)
514         PARAMETER (PI=3.141592654D0)
515         PARAMETER (MAXFCV=200,MAXF=200,MAXFCON=25,MAXFTAB=25,MAXFPOL=25)
516         PARAMETER (MAXPAIR=30,MAXORP1=5)
517         PARAMETER (MAXF6=MAXF*6,MAXF4=MAXF*4,MAXF2P1=MAXF*2+1)
518         PARAMETER (MAXFCV3=MAXFCV*3)
519         PARAMETER (MAXHS=200,MAXELH=500,MAXEX=3,MAXEXT=350)
520         PARAMETER (MAXFLUXA=3,MAXNH=50,MAXNTH=MAXELH*MAXHS)
521         PARAMETER (MEXTNW=MAXEXT*MAXNW,MAXHS4=MAXHS*4)
522         PARAMETER (MAXDNG=25,MAXDHG=15,MAXRX=3,MAXREAC=10)
523         PARAMETER (MAXNR=MAXDNG*MAXRX,MAXHR=MAXDHG*MAXRX,
524 *             MAXRR=MAXREAC*MAXRX,MAX3R=3*MAXRX)
525         PARAMETER (MAXFTAV=10,MAXFTE=100)
526         PARAMETER (MAXFDAV=10,MAXFDE=100)
527 C
528         CHARACTER TITLE1*126,TITLE2*126,NAME*10
529         CHARACTER GAS*12,PMCORR*10
530         CHARACTER RESTART*4,FNAME*28,FCVNAME*28,NWNAME*24
531         CHARACTER NAMEHS*24,MATNAME*12,GEOM*1
532         CHARACTER RXNAME*24,FBNAME*12
533 C
534 C
535 C COMMON BLOCKS
536 C -----
537         COMMON/ERROR/IERR
538         COMMON/GASCON/NGAS(MAXG),IEOSO(MAXG),RGAS(MAXG),GAS(MAXG),
539 +         DENC(MAXG,3),IDGAS(MAXNW,MAXG),IEOS(MAXNW),NGFLAG,NGIN
540         COMMON/VISCON/VOV(MAXG,MAXNTR),TOV(MAXG,MAXNTR),
541 +         SNV(MAXG,MAXNTR),LAWV(MAXG,MAXNTR),NTRV(MAXG),
542 +         TRV(MAXG,MAXNTR+1),VISM(MAXG)
543         COMMON/CPCON/ACP(MAXG,MAXNTR),BCP(MAXG,MAXNTR),
544 +         CCP(MAXG,MAXNTR),DCP(MAXG,MAXNTR),NTRC(MAXG),
545 +         TRC(MAXG,MAXNTR+1),CPM(MAXG)
546         COMMON/CONDCON/ACOND(MAXG,MAXNTR),BCOND(MAXG,MAXNTR),
547 +         CCOND(MAXG,MAXNTR),DCOND(MAXG,MAXNTR),
548 +         NTRK(MAXG),TRK(MAXG,MAXNTR+1),CONDM(MAXG)
549         COMMON/BC1/IBCN(MAXNS),IBC(MAXNS,2),ISN(MAXSETS),IBCNN(MAXNS)
550         COMMON/BC2/BC(MAXSETS,MAXG+2)
551         COMMON/BC3/NMBC,NPBC,NBBC,NBC(MAXNW),NBCS(MAXNW)
552         COMMON/TITLES/TITLE1,TITLE2
553         COMMON/EXEC1/TSTART,TEND,TIME,TIMESTEP,DELT,PRNTINT,ISTEP,
554 +         ISTART,IPRINT,TIMPRNT,NPLT,NDUMP,RESTART
555         COMMON/EXEC2/MAXIP(MAXNW),MAXIM(MAXNW),MAXID(MAXNW),
556 +         MAXIT(MAXNW),MAXIC(MAXNW),MAXIX(MAXNW),
557 +         DRERRP(MAXNW),DRERRM(MAXNW),DRERRD(MAXNW),
558 +         DRERRT(MAXNW),DRERRC(MAXNW),DRERRX(MAXNW),
559 +         RELAXP(MAXNW),RELAXM(MAXNW),RELAXD(MAXNW),
560 +         RELAXT(MAXNW),RELAXC(MAXNW),RELAXX(MAXNW),
561 +         OMRELP(MAXNW),OMRELM(MAXNW),OMRELD(MAXNW),
562 +         OMRELT(MAXNW),OMRELC(MAXNW),OMRELX(MAXNW)
563         COMMON/EXEC3/IPMAX(MAXNW),IPMIN(MAXNW),RIPAVG(MAXNW),
564 +         ITERP,ITERM(MAXNW),ITERD(MAXNW),ITERT,ITERC(MAXNW),
565 +         ITERX(MAXNW,MAXG),ITMAX(MAXNW),ITMIN(MAXNW),
566 +         RITAVG(MAXNW),ICD(MAXNW),IGE(MAXNW),MEE(MAXNW),
567 +         ISKIPCD(MAXNW),IGET(MAXNW)
```

568 COMMON/EXEC4/FLOED,AGRAV,TURLAM,SPEEDCO,MAXSS,THD,
569 + AERRTW,RERRTW,RELAXTW
570 COMMON/ELDATA1/NAME(MAXEL),N1(MAXEL),N2(MAXEL),
571 + DX(MAXEL),ED(MAXEL),RELRF(MAXEL),DZ(MAXEL),
572 + FA1(MAXEL),FA2(MAXEL),FAE(MAXEL),AK12(MAXEL),
573 + AK21(MAXEL),ALOD(MAXEL),POR(MAXEL),VOL(MAXEL),
574 + QDIR(MAXEL),FLOE(MAXEL),XME(MAXEL),MAXG)
575 COMMON/ELDATA2/NUMEL,JDF(MAXDF),DFF(MAXEL),NDF(MAXNW),
576 + RFLOW1(MAXEL),RFLOW2(MAXEL)
577 COMMON/ELDATA3/AK(MAXEL),FF(MAXEL),CON1(MAXEL),CON2(MAXEL),
578 + CON3(MAXEL),CON4(MAXEL),CON5(MAXEL),CON6(MAXEL),
579 + CFKE(MAXEL),REY(MAXEL),DIR(MAXEL),UWF(MAXEL),
580 + ER(MAXEL),DENE(MAXEL),PRESE(MAXEL),TE(MAXEL),
581 + QFLOE(MAXEL),WFE(MAXEL),QEFF(MAXEL),PRNL(MAXEL),
582 + HAFM(MAXEL),PEC(MAXEL),GRAVE(MAXEL)
583 COMMON/ELDATA4/CONDE(MAXEL,0:MAXG),VISE(MAXEL,0:MAXG),
584 + CPE(MAXEL,0:MAXG),CVE(MAXEL,0:MAXG),RGASH(MAXEL)
585 COMMON/ELDATA1S/N1S(MAXELS),N2S(MAXELS),FLOES(MAXELS),
586 + NE(MAXELS),JSE(MAXES,MAXELS)
587 COMMON/ELDATA2S/NUMELS
588 COMMON/ELDATA3S/ERS(MAXELS),GRAVES(MAXELS)
589 COMMON/NDATA1/PRESN(MAXN),TN(MAXN),XMN(MAXN,MAXG),DENN(MAXN)
590 COMMON/NDATA2/GF(MAXNS,0:MAXC),GRAV(MAXNS),CVECTOR(MAXNS),
591 + IEQNP(MAXNS)
592 COMMON/NDATA3/NUMN,NUMEQP,NUMEQT
593 COMMON/NDATA4/GM(MAXN,0:MAXC),QVECTOR(MAXN),IEQNT(MAXN),
594 + GT(MAXN,0:MAXC),XVECTOR(MAXN)
595 COMMON/NDATA1S/FLONS(MAXNS),FLONAS(MAXNS),OFLONAS(MAXNS),
596 + PRESNS(MAXNS)
597 COMMON/NDATA2S/NUMNS,NNS(MAXNS),NN(MAXN)
598 COMMON/CONNECT1/ICQ(MAXN,0:MAXC),ICQT(MAXN,0:MAXC),
599 + ICCQ(MAXN,0:MAXC),ICCQT(MAXN,0:MAXC),
600 + FOS(MAXN,0:MAXC),FOST(MAXN,0:MAXC)
601 COMMON/CONNECT2/INNT(MAXN),NEQT(MAXNW)
602 COMMON/CONNCP1/ICM(MAXNS,0:MAXC),ICMP(MAXNS,0:MAXC),
603 + ICCM(MAXNS,0:MAXC),ICCMP(MAXNS,0:MAXC),
604 + SGN(MAXNS,0:MAXC),SGNP(MAXNS,0:MAXC)
605 COMMON/CONNCP2/INNP(MAXNS),NEQP(MAXNW),ISBW(MAXNW)
606 COMMON/NWDATA/NUMNW,LASTN(0:MAXNW),LASTE(0:MAXNW),VOLNW(MAXNW),
607 + NWNAME(MAXNW),POT(MAXNW),CNW(MAXNW),TANW(MAXNW),PANW(MAXNW),
608 + LASTES(MAXNW),LASTNS(MAXNW),DELTF(MAXNW),DTFM(MAXNW),
609 + ISTF(MAXNW),NPRNTF(MAXNW),IDYNFM(MAXNW)
610 COMMON/CHDATA/ICHNG(MAXNW),NCHNG(MAXNW)
611 COMMON/PMDATA/PMC(5),PMCORR
612 COMMON/PMFFV/PF(MAXEL),FFCF(MAXEL),PFS(MAXELS),FFCFS(MAXELS)
613 COMMON/FCVDATA/IFCVID(MAXFCV),IPNTRV(MAXFCV,2),FCVM(MAXFCV),
614 + IAFN(MAXFCV),IAFNR(MAXFCV),FCVNAME(MAXFCV),
615 + LFCV(MAXFCV,3),NFCVSY, NFCVFM, NFCVHT, NFCVRD, NFCBC
616 COMMON/FDATA/IFN(MAXF),IFID(MAXF),IPNTRF(MAXF,6),FNAME(MAXF),
617 + FM(MAXF),FVALUE(0-MAXF:MAXF),FCON(MAXFCON),LF(MAXF,4),
618 + FTABX(MAXFTAB,MAXPAIR),FTABY(MAXFTAB,MAXPAIR),
619 + FPOL(MAXFPOL,MAXORP1),IAFNF(MAXF,6),
620 + IHSN(MAXFTAV,MAXFTE),IEN(MAXFTAV,MAXFTE),
621 + TWF(MAXFTAV,MAXFTE),INWN(MAXFNAV,MAXFDE),
622 + IFEN(MAXFNAV,MAXFDE),DWF(MAXFNAV,MAXFDE),
623 + NFSYS,NFFM,NFHT,NFRD
624 COMMON/NOZDAT/NOZ(MAXNW),JNOZ(MAXNOZ),PZ(MAXNOZ),DTZ(MAXNOZ)
625 COMMON/CMPSRDAT/NCMP SR(MAXNW),JCMP SR(MAXCMP SR),RSPEED(MAXCMP SR),
626 + SPEED(MAXCMP SR),NCPNTS(MAXCMP SR),JCMP SR S(MAXCMP SR),
627 + CTABX(MAXCMP SR,MAXPAIR),CTABY(MAXCMP SR,MAXPAIR)
628 COMMON/MATDAT1/NMATS,MATNAME(MAXFTAB),DENHS(MAXFTAB),
629 + SHH(MAXFTAB),CDM(MAXFTAB),MSHTYPE(MAXFTAB),
630 + MCTYPE(MAXFTAB),NPAIRSSH(MAXFTAB),
631 + NPAIRSC(MAXFTAB),XTSH(MAXFTAB,MAXPAIR),
632 + XTCD(MAXFTAB,MAXPAIR),YSH(MAXFTAB,MAXPAIR),
633 + YCD(MAXFTAB,MAXPAIR)
634 COMMON/MATDAT2/DENH(MAXELH),SHH(MAXELH),CONDH(MAXELH),
635 + TEH(MAXELH)
636 COMMON/HSDATA1/NAMEHS(MAXHS),NEH(MAXHS),COPIES(MAXHS),GEOM(MAXHS),
637 + IBCH1(MAXHS),IBCH2(MAXHS),TBC1(MAXHS),TBC2(MAXHS),
638 + NHBC1(MAXHS),NWH1(MAXHS,MAXEX),JEH1(MAXHS,MAXEX),
639 + JEH1S(MAXHS,MAXEX),NHBC2(MAXHS),NWH2(MAXHS,MAXEX),
640 + JEH2(MAXHS,MAXEX),JEH2S(MAXHS,MAXEX),JHF(MAXHS),
641 + FLUXBC1(MAXHS),AFLUX1(MAXHS),FLUXBC2(MAXHS),
642 + AFLUX2(MAXHS),ETSTEP(MAXHS),ISTH(MAXHS),
643 + DELTH(MAXHS),DTMINH(MAXHS),NPRNT(MAXHS),NHS,
644 + QPPPM(MAXHS),FETS(MAXHS),KH0YN(MAXHS)
645 COMMON/HSDATA2/IEXT(MAXNW),IEX(MAXNW),JEH(MAXEXT,MAXNW),
646 + AEX(MAXEXT,MAXNW),AEXC(MAXEXT,MAXNW),
647 + EDH(MAXEXT,MAXNW),IHTCLAM(MAXEXT,MAXNW),
648 + IHTCTUR(MAXEXT,MAXNW),TURLAMH(MAXEXT,MAXNW),

```

649 + TWALL(MAXEXT,MAXNW),TEX(MAXEXT,MAXNW),
650 + HTC(MAXEXT,MAXNW),AUXL(MAXEXT,MAXNW),
651 + AUXT(MAXEXT,MAXNW),REYH(MAXEXT,MAXNW),
652 + 1EXRS(MAXNW)
653 COMMON/HSDATA3/DXH(MAXELH),AREA(MAXELH),VOLH(MAXELH),QPPP(MAXELH),
654 + MATID(MAXELH),NFLUXA(MAXELH),QPP(MAXELH,MAXFLUXA),
655 + ASURF(MAXELH,MAXFLUXA),NEX(MAXELH),QPOW(MAXELH),
656 + NWHE(MAXELH,MAXEX),JEHE(MAXELH,MAXEX),DIM(MAXHS,4)
657 COMMON/HSDATA4/GK(MAXNH,3),RVECT(MAXNH),TNH(MAXNTH),TNHO(MAXNTH),
658 + FACTI(MAXNTH)
659 COMMON/RDDATA/NRX,NDNG(MAXRX),NDHG(MAXRX),RXNAME(MAXRX),
660 + BETA(MAXRX),OMEFAC(MAXRX),GENTIME(MAXRX),
661 + RDN(MAXRX),SOURCE(MAXRX),XLAM(MAXDNG,MAXRX),
662 + BETAG(MAXDNG,MAXRX),CDN(MAXDNG,MAXRX),ISRD(MAXRX),
663 + XLAMH(MAXDHG,MAXRX),EFACG(MAXDHG,MAXRX),
664 + CDH(MAXDHG,MAXRX),NREAC(MAXRX),DREAC(MAXREAC,MAXRX),
665 + REACT(MAXRX),DTRDMIN(MAXRX),DRERRR(MAXRX),
666 + DELTR(MAXRX),NPRNTRD(MAXRX),RDNE(MAXRX),RDNO(MAXRX),
667 + DELTRL(MAXRX),CDNO(MAXDNG,MAXRX),CDHO(MAXDHG,MAXRX),
668 + DTRDMAX(MAXRX),RDNEO(MAXRX),RDST1(MAXRX),PRI(MAXRX),
669 + AUXRD(3,MAXRX),KSOLVE(MAXRX),RERRR(MAXRX),
670 + DREACI(MAXREAC,MAXRX),PERINV(MAXRX),PERINVE(MAXRX),
671 + DREACIT(MAXRX),FBNAME(MAXREAC,MAXRX)
672 COMMON/RDCONS/CONN(MAXRX),CONC(MAXDNG,MAXRX),FC(15),FCA(9)
673 COMMON/PRDATA/KOFTPR(MAXRX),NPAIRSPR(MAXRX),DRHODT(MAXRX),
674 + XTPR(MAXRX,MAXPAIR),YPR(MAXRX,MAXPAIR),IPR(MAXRX)

```

675 C
 676 C PLACE USER-DEFINED COMMON BLOCKS HERE <-----

677 C
 678 C
 679 C PLACE USER-DEFINED OUTPUT HERE <-----
 680 COMMON/UDATA1/ISPOUT

681 C
 682 C WRITE NODE STAGNATION PRESSURES FOR EACH NETWORK, IF REQUESTED
 683 C (ISPOUT IS SPECIFIED IN USER-DEFINED INPUT DATA)

```

684 C  

685 IF(ISPOUT.GT.0)THEN
686 IP=0
687 JP=0
688 LN1=1
689 WRITE(15,998)
690 DO 2 NW=1,NUMNW
691 LN2=LASTE(NW)
692 NN2=LASTN(NW)
693 C
694 WRITE(15,1000)TIME,NW
695 DO 10 J=LN1,LN2
696 JMJP=J-JP
697 I1=N1(J)
698 I2=N2(J)
699 GAMMA=CPE(J,0)/CVE(J,0)
700 V1=FLOE(J)/(DENN(I1)*POR(J)*FA1(J))
701 XM1=V1/DSQRT(GAMMA*RGASM(J)*TN(I1))
702 GM1O2=(GAMMA-1.0D0)*0.5D0
703 GOGM1=GAMMA/(GAMMA-1.0D0)
704 PSN1=PRESN(I1)*(1.0D0+GM1O2*XM1*XM1)**GOGM1
705 V2=FLOE(J)/(DENN(I2)*POR(J)*FA2(J))
706 XM2=V2/DSQRT(GAMMA*RGASM(J)*TN(I2))
707 GM1O2=(GAMMA-1.0D0)*0.5D0
708 GOGM1=GAMMA/(GAMMA-1.0D0)
709 PSN2=PRESN(I2)*(1.0D0+GM1O2*XM2*XM2)**GOGM1
710 WRITE(15,1005)JMJP,I1-IP,I2-IP,PSN1,PSN2,PSN1/PRESN(I1),
711 PSN2/PRESN(I2),V1,V2,XM1,XM2,NAME(J)
712 10 CONTINUE
713 C
714 LN1=LN2+1
715 IP=NN2
716 JP=LN2
717 2 CONTINUE
718 ENDIF

```

719 C
 720 C
 721 C AND,
 722 C
 723 KR=1
 724 C THIS OUTPUT IS PROVIDED IF USER-DEFINED CONTROL LAW #2 IS SPECIFIED
 725 C
 726 C YPR(KR,2) IS THE DRUM ROTATION ANGLE IN radians
 727 C NOTE: A POSITIVE ROTATION IS CLOCKWISE AND GIVES A NEGATIVE
 728 C REACTIVITY INSERTION (ROTATES POISON TOWARD CORE)
 729 C YPR(KR,3) IS THE CURRENT DRUM WORTH

```
730 C AUXRD(1,KR) IS THE MAX DRUM WORTH WHICH CAN BE FUNCTION CONTROLLED
731 IF(KOPTPR(KR).GT.3.AND.NPAIRSPR(KR).EQ.2)THEN
732 WRITE(9,2000)TIME,YPR(KR,2)*57.29578,YPR(KR,3)/BETA(KR),
733 + AUXRD(1,KR)/BETA(KR)
734 ENDIF
735 C
736 C
737 C
738 998 FORMAT(' ','/' ,64('><'),/)
739 1000 FORMAT(' ','/' ,2X,'*** TIME (s) = ',1PE12.4,4X,
740 + 'FLOW NETWORK # ',13,3X,'***',/' ' ,3X,
741 + 'E#',1X,'NODE1',1X,'NODE2',4X,'Po1 (Pa)',5X,'Po2 (Pa)',3X,
742 + 'Po1/P1',3X,'Po2/P2',5X,'V1 (M/S)',5X,'V2 (M/S)',4X,
743 + 'MACH1',5X,'MACH2',6X,'NAME')
744 1005 FORMAT(' ',15,1X,14,2X,14,2X,2(2(1PE12.4,1X),2(OPF8.5,2X)),2X,A)
745 2000 FORMAT(' ',2X,'*TIME (s) = ',1PE12.4,4X,'DRUM ANGLE (degrees) = ',
746 + E12.4,/' ' ,3X,'DRUM WORTH ($) = ',E12.4,2X,
747 + 'MAXIMUM DRUM WORTH ($) = ',E12.4)
748 C
749 C
750 RETURN
751 END
752 C
753 C *****
754 C
755 FUNCTION UEOS(P,T,IOP)
756 C
757 C EVALUATE USER-SUPPLIED EQUATION OF STATE (FLUID DENSITY)
758 C BE SURE TO LINK ANY REQUIRED DATA BASES TO SAFSIM
759 C ALSO, NOTE THAT ALL VARIABLES AND FUNCTIONS IN SAFSIM
760 C ARE REAL*8; THUS, ANY USER-SUPPLIED FUNCTIONS SHOULD BE
761 C APPROPRIATELY DECLARED OR CONVERTED
762 C
763 IMPLICIT REAL*8 (A-H,O-Z)
764 C
765 PARAMETER (MAXEL=500,MAXG=4)
766 C
767 COMMON/ELDATA4/CONDE(MAXEL,0:MAXG),VISE(MAXEL,0:MAXG),
768 + CPE(MAXEL,0:MAXG),CVE(MAXEL,0:MAXG),RGASM(MAXEL)
769 C
770 C P IS FLUID PRESSURE
771 C T IS FLUID TEMPERATURE
772 C IOP-2 IS USER EOS OPTION NUMBER; THUS, IF MORE THAN ONE USER-
773 C SUPPLIED EQUATION OF STATE IS IMPLEMENTED, THE FIRST WILL HAVE
774 C IOP=3, THUS IOP-2=1 ...
775 C USE IOP TO INDICATE WHICH USER-SUPPLIED EOS TO USE, FOR EXAMPLE:
776 C GO TO (1,2,3), IOP-2
777 C
778 C ueos=ptdens(p,t)
779 CALL UPH2EOS2(P,T,RHO)
780 UEOS=RHO
781 C
782 RETURN
783 END
784 C
785 C *****
786 C
787 FUNCTION UCP(P,T,IOP,JE,IG)
788 C
789 C EVALUATE USER-SUPPLIED PROPERTY (CONSTANT PRESSURE SPECIFIC HEAT)
790 C BE SURE TO LINK ANY REQUIRED DATA BASES TO SAFSIM
791 C ALSO, NOTE THAT ALL VARIABLES AND FUNCTIONS IN SAFSIM
792 C ARE REAL*8; THUS, ANY USER-SUPPLIED FUNCTIONS SHOULD BE
793 C APPROPRIATELY DECLARED OR CONVERTED
794 C
795 IMPLICIT REAL*8 (A-H,O-Z)
796 C
797 PARAMETER (MAXEL=500,MAXG=4)
798 C
799 COMMON/ELDATA4/CONDE(MAXEL,0:MAXG),VISE(MAXEL,0:MAXG),
800 + CPE(MAXEL,0:MAXG),CVE(MAXEL,0:MAXG),RGASM(MAXEL)
801 C
802 C COMMON BLOCK ELDATA4 IS PROVIDED IN CASE THE USER NEEDS THE PROPERTY
803 C VALUES IN THE USER-SUPPLIED FUNCTIONS. JE AND IG ARE PROVIDED IN
804 C THE CALLING LIST TO ACCESS THE APPROPRIATE ELEMENT AND GAS;
805 C THUS, CP=CPE(JE,IG)
806 C
807 C JE IS THE ELEMENT NUMBER
808 C IG IS THE GAS ID NUMBER (0 FOR MIXTURE)
809 C P IS FLUID PRESSURE
810 C T IS FLUID TEMPERATURE
```

```
811 C IOP-2 IS USER EOS OPTION NUMBER; THUS; IF MORE THAN ONE USER-
812 C SUPPLIED EQUATION OF STATE IS IMPLEMENTED, THE FIRST WILL HAVE
813 C IOP=3, THUS IOP-2=1 ...
814 C USE IOP TO INDICATE WHICH USER-SUPPLIED EOS TO USE, FOR EXAMPLE:
815 C GO TO (1,2,3), IOP-2
816 C
817 C ucp=ptcp(p,t)
818 C CALL UPH2EOS1(P,T,CP,GAM,TK,VIS,A)
819 C UCP=CP
820 C CVE(JE,IG)=CP/GAM
821 C CONDE(JE,IG)=TK
822 C VISE(JE,IG)=VIS
823 C THE NEXT LINE IS OPTIONAL WHEN USING THE PH2 EOS (SEE NOTE BELOW)
824 C RGASM(JE)=A*A/(GAM*T)
825 C
826 C NOTE:
827 C for gases, an effective mixture gas constant (rgasm) is determined
828 C in subroutine eos based on the ideal gas law; thus,
829 C  $rgasm = p/(density*t)$ . however,
830 C if the user-supplied eos is for a liquid, the value of
831 C rgasm should be set to a value in this function that provides
832 C a reasonable estimate of sound speed based on  $\sqrt{gamma*rgasm*t}$ 
833 C for example,
834 C  $rgasm(j)=emod/(den*t)$ 
835 C where emod is the bulk compressibility modulus, den is liquid density,
836 C and emod and den are evaluated at temperature t.
837 C remember that liquids and gases cannot be mixed, so only redefine
838 C rgasm if only one fluid in the network is present or make sure
839 C that rgasm is a representative value for the mixture.
840 C
841 C RETURN
842 C END
843 C
844 C *****
845 C
846 C FUNCTION UCV(P,T,IOP,JE,IG)
847 C
848 C EVALUATE USER-SUPPLIED PROPERTY (CONSTANT VOLUME SPECIFIC HEAT)
849 C BE SURE TO LINK ANY REQUIRED DATA BASES TO SAFSIM
850 C ALSO, NOTE THAT ALL VARIABLES AND FUNCTIONS IN SAFSIM
851 C ARE REAL*8; THUS, ANY USER-SUPPLIED FUNCTIONS SHOULD BE
852 C APPROPRIATELY DECLARED OR CONVERTED
853 C
854 C IMPLICIT REAL*8 (A-H,O-Z)
855 C
856 C PARAMETER (MAXEL=500,MAXG=4)
857 C
858 C COMMON/ELDATA4/CONDE(MAXEL,0:MAXG),VISE(MAXEL,0:MAXG),
859 C + CPE(MAXEL,0:MAXG),CVE(MAXEL,0:MAXG),RGASM(MAXEL)
860 C
861 C COMMON BLOCK ELDATA4 IS PROVIDED IN CASE THE USER NEEDS THE PROPERTY
862 C VALUES IN THE USER-SUPPLIED FUNCTIONS. JE AND IG ARE PROVIDED IN
863 C THE CALLING LIST TO ACCESS THE APPROPRIATE ELEMENT AND GAS;
864 C THUS, CP=CPE(JE,IG)
865 C
866 C JE IS THE ELEMENT NUMBER
867 C IG IS THE GAS ID NUMBER (0 FOR MIXTURE)
868 C P IS FLUID PRESSURE
869 C T IS FLUID TEMPERATURE
870 C IOP-2 IS USER EOS OPTION NUMBER; THUS; IF MORE THAN ONE USER-
871 C SUPPLIED EQUATION OF STATE IS IMPLEMENTED, THE FIRST WILL HAVE
872 C IOP=3, THUS IOP-2=1 ...
873 C USE IOP TO INDICATE WHICH USER-SUPPLIED EOS TO USE, FOR EXAMPLE:
874 C GO TO (1,2,3), IOP-2
875 C
876 C ucv=ptcv(p,t)
877 C UCV=CVE(JE,IG)
878 C
879 C RETURN
880 C END
881 C
882 C *****
883 C
884 C FUNCTION UCOND(P,T,IOP,JE,IG)
885 C
886 C EVALUATE USER-SUPPLIED PROPERTY (FLUID CONDUCTIVITY)
887 C BE SURE TO LINK ANY REQUIRED DATA BASES TO SAFSIM
888 C ALSO, NOTE THAT ALL VARIABLES AND FUNCTIONS IN SAFSIM
889 C ARE REAL*8; THUS, ANY USER-SUPPLIED FUNCTIONS SHOULD BE
890 C APPROPRIATELY DECLARED OR CONVERTED
891 C
```

```
892      IMPLICIT REAL*8 (A-H,O-Z)
893 C
894      PARAMETER (MAXEL=500,MAXG=4)
895 C
896      COMMON/ELDATA4/CONDE(MAXEL,0:MAXG),VISE(MAXEL,0:MAXG),
897 +      CPE(MAXEL,0:MAXG),CVE(MAXEL,0:MAXG),RGASM(MAXEL)
898 C
899 C COMMON BLOCK ELDATA4 IS PROVIDED IN CASE THE USER NEEDS THE PROPERTY
900 C VALUES IN THE USER-SUPPLIED FUNCTIONS. JE AND IG ARE PROVIDED IN
901 C THE CALLING LIST TO ACCESS THE APPROPRIATE ELEMENT AND GAS;
902 C THUS, CP=CPE(JE,IG)
903 C
904 C JE IS THE ELEMENT NUMBER
905 C IG IS THE GAS ID NUMBER (0 FOR MIXTURE)
906 C P IS FLUID PRESSURE
907 C T IS FLUID TEMPERATURE
908 C IOP-2 IS USER EOS OPTION NUMBER; THUS; IF MORE THAN ONE USER-
909 C SUPPLIED EQUATION OF STATE IS IMPLEMENTED, THE FIRST WILL HAVE
910 C IOP=3, THUS IOP-2=1 ...
911 C USE IOP TO INDICATE WHICH USER-SUPPLIED EOS TO USE, FOR EXAMPLE:
912 C GO TO (1,2,3), IOP-2
913 C
914 C      ucond=ptcond(p,t)
915 C      UCOND=CONDE(JE,IG)
916 C
917 C      RETURN
918 C      END
919 C
920 C *****
921 C
922 C      FUNCTION UVIS(P,T,IOP,JE,IG)
923 C
924 C EVALUATE USER-SUPPLIED PROPERTY (FLUID VISCOSITY)
925 C BE SURE TO LINK ANY REQUIRED DATA BASES TO SAFSIM
926 C ALSO, NOTE THAT ALL VARIABLES AND FUNCTIONS IN SAFSIM
927 C ARE REAL*8; THUS, ANY USER-SUPPLIED FUNCTIONS SHOULD BE
928 C APPROPRIATELY DECLARED OR CONVERTED
929 C
930 C      IMPLICIT REAL*8 (A-H,O-Z)
931 C
932 C      PARAMETER (MAXEL=500,MAXG=4)
933 C
934 C      COMMON/ELDATA4/CONDE(MAXEL,0:MAXG),VISE(MAXEL,0:MAXG),
935 +      CPE(MAXEL,0:MAXG),CVE(MAXEL,0:MAXG),RGASM(MAXEL)
936 C
937 C COMMON BLOCK ELDATA4 IS PROVIDED IN CASE THE USER NEEDS THE PROPERTY
938 C VALUES IN THE USER-SUPPLIED FUNCTIONS. JE AND IG ARE PROVIDED IN
939 C THE CALLING LIST TO ACCESS THE APPROPRIATE ELEMENT AND GAS;
940 C THUS, CP=CPE(JE,IG)
941 C
942 C JE IS THE ELEMENT NUMBER
943 C IG IS THE GAS ID NUMBER (0 FOR MIXTURE)
944 C P IS FLUID PRESSURE
945 C T IS FLUID TEMPERATURE
946 C IOP-2 IS USER EOS OPTION NUMBER; THUS; IF MORE THAN ONE USER-
947 C SUPPLIED EQUATION OF STATE IS IMPLEMENTED, THE FIRST WILL HAVE
948 C IOP=3, THUS IOP-2=1 ...
949 C USE IOP TO INDICATE WHICH USER-SUPPLIED EOS TO USE, FOR EXAMPLE:
950 C GO TO (1,2,3), IOP-2
951 C
952 C      uvis=ptvisc(p,t)
953 C      UVIS=VISE(JE,IG)
954 C
955 C      RETURN
956 C      END
957 C
958 C *****
959 C
```

Index of Input Variables

ACOND, 29
ACP, 28
AERRTW, 19
AEX, 53, 55, 58
AFLUX1, 52
AFLUX2, 57
AGRAV, 20
AK12, 32
AK21, 33
ALOD, 33
AREA, 54
ASURF, 55
AUXL, 53, 56, 58
AUXRD, 65
AUXT, 54, 56, 58
BCH1, 52
BCH2, 56
BCKIND, 38
BCOND, 29
BCP, 28
BETAG, 66
CCOND, 29
CCP, 28
CD, 47, 48
CDC, 47, 48
CDH, 67
CDM, 47
CDN, 66
CDP, 35
CFLO, 34
CONDM, 29
COPIES, 49
COSA, 34
CPM, 28
DCOND, 29
DCP, 28
DELTR, 65
DENC, 27
DENHS, 45
DIM1, 51
DIM2, 51
DIM3, 51
DIM4, 51
DREAC, 68
DREACI, 68
DRERRC, 24
DRERRD, 24
DRERRM, 24
DRERRP, 24
DRERRR, 65
DRERRT, 24
DRERRX, 24
DTFM, 22
DTMINH, 50
DTRDMAX, 65
DTRDMIN, 65
DX, 31
DXH, 54
DZ, 31
ED, 31
EDH, 53, 56, 58
EFRACG, 67
END, 25, 29, 35, 37, 48, 59, 61, 63,
69, 71, 73, 75, 77, 78
FA1, 32
FA2, 32
FAE, 32
FBNAME, 68
FCVM, 42, 61, 71, 75
FCVNAME, 41, 60, 70, 74
FETS, 50
FLOED, 20
FLOES, 35
FLUXBC1, 52
FLUXBC2, 57
FM, 43, 62, 72, 76
FNAME, 43, 62, 72, 76
FVALUE, 43, 62, 72, 76
GAS, 26
GENTIME, 64
GEOM, 50
IAFN, 41, 60, 70, 74

IALOD, 33
idgas, 26
IDS, 38
IDSET, 38, 39
IDYNFM, 23, 32
IEOSO, 26, 27
IFCVID, 41, 60, 70, 74
IFID, 43, 62, 72, 76
IFN, 43, 62, 72, 76, 78
igas, 26
IHTCLAM, 53, 56, 58
IHTCTUR, 53, 56, 58
ISKIPCD, 22
ISTART, 20
JE, 30
JEH1, 53
JEH2, 57
JEHE, 55
JES, 30
KH, 49
KHDYN, 49
KOPTPR, 68
KR, 64
KSOLVE, 64
L1, 42, 61, 71, 75
L2, 42, 61, 71, 75
L3, 42, 61, 71, 75
LAW, 28
MAT, 54
MATNAME, 45
MAXIC, 23
MAXID, 23
MAXIM, 23
MAXIP, 23
MAXIT, 23
MAXIX, 24
MAXSS, 19
MCDTYPE, 47
MEE, 22
MSHTYPE, 45
N1, 31
N1S, 30
N2, 31
N2S, 30
NAME, 30
NAMEHS, 49
NDHG, 66
NDNG, 66
NDUMP, 20
NE, 30
NEH, 49
NEX, 55
NFLUXA, 54
NGAS, 26
NHBC1, 52
NHBC2, 57
NID, 36
NP, 45, 47
NPLT, 20
NPNTS, 34
NPPR, 68
NPRNTF, 23
NPRNTH, 50
NPRNTRD, 65
NREAC, 67
NSSLS, 19
NTRC, 28
NTRK, 29
NTRV, 27
NUMNW, 22
NW, 22, 26
NWH1, 53
NWH2, 57
NWHE, 55
NWNNAME, 22
PMC1, 21
PMC2, 21
PMC3, 21
PMC4, 21
PMC5, 22
PMCORR, 21
PMVALUE, 39
POR, 31
POT, 39
PR, 68, 69
PRC, 69
PRESND, 36
PRI, 69
PRNTINT, 20
PTX, 36

QDIR, 34
QPP, 55
QPPP, 49, 54
QPPPM, 49, 54
RDN, 64
RDST, 65
RELAXC, 25
RELAXD, 25
RELAXM, 25
RELAXP, 24
RELAXT, 25
RELAXTW, 19
RELAXX, 25
RELRUF, 32
RESTART, 19
RGAS, 27
RSPEED, 34
RXNAME, 64
SH, 46
SHC, 46
SHM, 45
SNV, 28
SOURCE, 64
SPEED, 34
SPEEDCO, 21
T, 46, 47
t, 68
TBC1, 52
TBC2, 57
TEND, 20
THD, 21
TIMESTEP, 20
TITLE1, 19
TITLE2, 19
TND, 36
TNH, 59
TOV, 28
TRC, 28
TRK, 29
TRV, 27
TSTART, 20
TURLAM, 21
TURLAMH, 53, 56, 58
TVALUE, 39
VALUE, 36
VISM, 27
VOL, 32
VOLH, 54
VOV, 28
XLAM, 66
XLAMH, 67
XMND, 36
XVALUE, 39

USER NOTES

USER NOTES

DISTRIBUTION:

- 1 Barattino, Maj. William
Philips Laboratory/TAS
Kirtland AFB, NM 87117-6008
- 1 Bennett, Gary L.
NASA Headquarters
Code RP
Washington, DC 20546
- 2 Attn: Frederick R. Best
Stacy K. Lee
Texas A&M University
Nuclear Engineering Dept.
129 Zachry Engineering Ctr.
College Station, TX
7843-3133
- 1 Bhattacharyya, Dr. Samit K.
Associate Director,
Engineering
Argonne National Laboratory
9700 South Cass-Bldg. 207
Argonne, IL 60439-4841
- 1 Bleeker, Lt. Col. Gary
Phillips Laboratory/VT-X
Kirtland AFB, NM 87117-6008
- 1 Bohl, Dick J.
Los Alamos National Laboratory
P. O. Box 1663 - MS-E561
Los Alamos, NM 87545
- 1 Borowski, Dr. Stanley K.
Nuclear/Aerospace Engineer
NASA Lewis Research Center
Adv Space Analysis Office
21000 Brookpark Road
Mail Stop AAC-2
Cleveland, OH 44135
- 1 Boyle, Maj. David R.
Philips Laboratory/AWYS
Kirtland AFB, NM 87117-6008
- 1 Breen, J. A.
Pratt & Whitney GESP
P. O. Box 109600
West Palm Beach, FL
33410-9600
- 1 Brengle, Bob
Rockwell International
Corporation
Rocketdyne Division
6633 Canoga Avenue
Canoga Park, CA 91303
- 1 Buden, David
Center for Nuclear Eng.
and Technology
INEL
P. O. Box 1625
Idaho Falls, ID 83415-2516
- 3 Attn: John Buska
Bill Ryder
R. T. Perry
Mail Stop K551
P. O. Box 1663
Los Alamos National Laboratory
Los Alamos, NM 87545
- 1 Carroll, Wade P.
U. S. Department of Energy
NE-52
19901 Germantown Road
Germantown, MD 20874
- 2 NASA Lewis Research Center
Attn: John Clark
Tom Miller
21000 Brookpark Rd.
Cleveland, OH 44135
- 1 Dahl, Wayne B.
Program Manager
Aerojet Propulsion Division
P. O. Box 13222
Department 9810
Sacramento, CA 95813
- 1 Difilippo, Felix C.
Oak Ridge National Lab
Oak Ridge, TN 37831-6363
- 1 Donahue, Benjamin B.
Senior Engr., Propulsion
Analysis
Boeing Aerospace
P. O. Box 240002
Mail Stop JX-23
Huntsville, AL 35824-6402

2 El-Genk, Mohammed
 Institute for Space Nuclear
 Power Studies
 Chemical and Nuclear
 Engineering Department
 University of New Mexico
 Albuquerque, NM 87131

1 Felty, Jim
 U. S. Dept. of Energy
 DP-241
 19901 Germantown Rd.
 Germantown, MD 20585

2 Emrich, Bill
 Jones, Lee
 MS EP53 Flash 4666
 NASA Marshall Space
 Flight Center, AL 35812

1 Gerrish, Harold
 NASA Marshall Space
 Flight Center
 Huntsville, AL 35812

1 Gunn, Dr. Stanley V.
 Program Manager
 Rockwell International Corp.
 Rocketdyne Division
 6633 Canoga Avenue
 Mail Stop 43B
 Canoga Park, CA 91303

1 Haloulakos, V. E.
 Engineering Specialist
 McDonnell Douglas Space Sys Co
 Internal Mail
 Code A3-Y833-13/3
 5301 Bolsa Avenue
 Huntington Beach, CA 92647

1 Haman, Nelson A.
 Argonne National Laboratory
 9700 So. Cass Ave.
 Building 207
 Argonne, IL 60439-4841

1 Hampston, Ken
 Phillips Laboratory/VT-X
 Kirtland AFB, NM 87117-6008

1 Haskin, Eric F.
 Institute for Space Nuclear
 Power Studies
 Chemical & Nuclear Engr. Dept.
 University of New Mexico
 Albuquerque, NM 87131

1 Haslett, Robert
 Project Director
 Grumman Aerospace Corp.
 MS B09-25
 Bethpage, NY 11714

1 Howe, Dr. Steve
 Program Coordinator
 Space Propulsion Tech
 Los Alamos National Lab
 P. O. Box 1663
 Mail Stop F643
 Los Alamos, NM 87545

1 Husser, Dwayne
 Babcock & Wilcox
 P. O. Box 11165
 Lynchburg, VA 24506-1165

1 Latham, T. S.
 United Technologies Research
 Center
 Silver Lane
 East Hartford, CT 06108

1 Lenard, Lt. Col. Roger
 Phillips Laboratory
 VT-X
 Kirtland AFB, NM 87117-6008

2 Los Alamos National
 Laboratory
 Attn: Mike Cappiello
 Dennis Liles
 MS K551
 P. O. Box 1663
 Los Alamos, NM 87545

1 Ludewig, Hans
 Brookhaven National Laboratory
 Building 701
 Upton, NY 11719

1 Luth, W. C.
 DOE/ER
 U. S. Dept. of Energy
 ER-15, J317/GTN
 19901 Germantown Road
 Germantown, MD 20585

1 Maise, George
 Brookhaven National Laboratory
 Bldg. 701
 Upton, NY 11793

<p>1 Mangus, Jim Program Manager Westinghouse Electric Corporation Box 158 Madison, PA 15663-0158</p>	<p>1 Powell, Dr. James R. Building 701 Brookhaven National Laboratory Upton, NY 11973</p>
<p>1 Martinell, John A. Idaho National Engineering Laboratory P. O. Box 1625 Idaho Falls, ID 83415</p>	<p>1 Reid, Hank Babcock & Wilcox Space Systems Engineering Mt. Athos Road - Rt. 726 P. O. Box 11165 Lynchburg, VA 24506-1165</p>
<p>1 Matson, Warren Idaho National Engineering Laboratory P. O. Box 1625 Idaho Falls, ID 83415</p>	<p>1 Roberds, Dr. Richard M. Professor, Engr Science & Math Univ of Tennessee Space Institute B. H. Goethert Parkway Tullahoma, TN 37388</p>
<p>1 McElroy, Steve H. Babcock & Wilcox P. O. Box 11165 Lynchburg, VA 24506-1165</p>	<p>1 Sapyta, Joseph J. Babcock & Wilcox Space Systems Engineering Mt. Athos Road - Rt. 726 P. O. Box 11165 - MC#08 Lynchburg, VA 24506-1165</p>
<p>1 Metzger, John D. Grumman Electronics Systems Division MS B09-25 Bethpage, NY 11714</p>	<p>1 Schuller, Michael J. Philips Laboratory/AWYS Kirtland AFB, NM 87117-6008</p>
<p>1 Moody, Capt. Jay Phillips Laboratory/VT-X Kirtland AFB, NM 87117-6008</p>	<p>1 Shepard, Kyle Staff Engineer Gen Dynamics Space Sys Div P. O. Box 85900 San Diego, CA 92138</p>
<p>1 Nichols, Don F. Philips Laboratory/AWYS Kirtland AFB, NM 87117</p>	<p>1 Sheppard, Susan J. Babcock & Wilcox Space Systems Engineering Mt. Athos Road - Rt. 726 P. O. Box 11165 - MC#08 Lynchburg, VA 24506-1165</p>
<p>1 Parsley, Randy C. Mgr, Rocket Conceptual Design Pratt & Whitney P. O. Box 109600 Mail Stop 714-65 West Palm Beach, FL 33410-9600</p>	<p>1 Smith, John M. NASA Lewis Research Center MS: 301-5 21000 Brookpark Rd. Cleveland, OH 44135</p>
<p>1 Pelaccio, Dennis G. Senior Staff Scientist Science App International Corporation 21151 Western Avenue Torrence, CA 90501</p>	<p>1 Solon, Marty Grumman Space Electronics Division South Oyster Bay Road MS B09-25 Bethpage, NY 11714</p>
<p>1 Perkins, Ken Brookhaven National Laboratory Bldg. 701 Upton, NY 11973</p>	

1	Tepes, Frank Grumman Space Electronics Division MS BO-25 Bethpage, NY 11714	1	8523-2 Central Technical Files
		5	7141 Technical Library
		1	7151 Technical Publications
		10	7613-2 Document Processing for DOE/OSTI
		1513	R. E. Hogan
1	Walter, Carl E. Lawrence Livermore National Laboratory 7000 East Avenue L-144 MS: L-197 P. O. Box 808 Livermore, CA 94550	5166	K. R. Boldt
		6400	N. R. Ortiz
		6404	D. A. Powers
		6412	G. W. Wyss
		6413	F. T. Harper
		6414	J. E. Kelly
		6418	S. L. Thompson
		6418	R. S. Longenbaugh
		6418	R. J. Gross
2	Walton, James NASA Lewis Research Center M.S. AAC-2 Cleveland, OH 44135	6429	K. E. Washington
		6474	L. O. Cropp
		6474	D. R. Gallup
		6474	J. H. Lee
1	Walton, Lewis A. Babcock & Wilcox 3315 Old Forest Rd. Lynchburg, VA 24506	6474	A. C. Marshall
		6474	R. R. Peters
		6474	F. J. Wyant
		6500	J. K. Rice
		6501	J. V. Walker
1	Warren, John Division of Defense Energy Projects Office of Nuclear Energy U. S. Dept. of Energy, NE-52 Washington, DC 20545	6512	D. F. Beck
		6513	G. C. Allen
		6513	C. D. Croessmann
		20	6513 D. Dobranich
		6513	C. D. Harmon
		6513	P. J. McDaniel
		6513	D. L. Miller
1	Watson, Ron Babcock & Wilcox Space Systems Engineering Mt. Athos Road - Rt. 726 P. O. Box 11165 Lynchburg, VA 24506-1165	6513	C. A. Ottinger
		6513	D. S. Oscar
		6513	G. F. Polansky
		6513	L. C. Sanchez
		6513	L. R. Shippers
		6514	R. L. Coats
		6514	E. J. Parma
1	Witter, Johnathan K. Dept. of Nuclear Engr. Massachussetts Institute of Technology NW12-306A Cambridge, MA 02139	6514	J. P. Petraglia
		6514	P. S. Pickard
		6514	A. J. Suo-Anttila
		6514	M. E. Vernon
		6514	S. A. Wright
		6515	M. Berman
		6515	V. J. Dandini
1	Zubrin, Robert Senior Engineer Martin Marietta Astronautics P. O. Box 179 Denver, CO 80201	6515	J. R. Hipp
		6515	W. H. McCulloch
		6515	G. S. Rightley
		6515	M. P. Sherman
		6515	M. F. Young
		6516	K. W. Hanks
1	Zuppero, A. MS-3413 EG&G Idaho, Inc. 1955 Freemont Ave. Idaho Falls, ID 83415-3413	6521	J. W. Bryson
		6523	J. S. Philbin