The Pennsylvania State University

The Graduate School

Department of Mechanical and Nuclear Engineering

DEVELOPMENT AND IMPLEMENTATION OF POINT KINETICS AND ASSOCIATED MODELS IN COBRA-TF

A Thesis in

Nuclear Engineering

by

Faisal Z. Raja

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2011

The thesis of Faisal Z. Raja was reviewed and approved* by the following:

Maria Avramova
Assistant Professor of Nuclear Engineering
Thesis Advisor

Kostadin Ivanov
Distinguished Professor of Nuclear Engineering

Arthur Thompson Motta
Professor of Nuclear Engineering
Chair of the Nuclear Engineering Program

*Signatures are on file in the Graduate School

ABSTRACT

The research consists of the implementation of a power calculation model in the reactor thermal-hydraulic analysis code, COBRA-TF, using the point kinetics approach. COBRA-TF in its current form has a fixed power model (power vs. time table). With the addition of the new point kinetics model, one should expect more realistic results since it accounts for the thermal-hydraulic feedback. Along with the point kinetics model, two other associated models are developed and implemented into the code: the decay heat model, and the feedback model. In order to verify the validity and accuracy of the new model in COBRA-TF, the reactor analysis code TRACE is used for code-to-code verification. A sample test problem, which consists of a ¼ core of a Pressurized Water Reactor being modeled as nine separate channels, is used to perform code-to-code verification using a 50 % loss of coolant flow transient. Based on this sample test problem, an input deck for COBRA-TF and an identical input model for TRACE are created. Steady state comparison results show consistent agreement between the two codes with respect to pressure, coolant density, coolant temperature, and liquid velocity. Transient results, which compare similar state parameter as well as reactor power and reactor multiplication factor, however, have shown some discrepancies especially for coolant density and channel temperature. TRACE predicts a flat temperature response whereas COBRA-TF predicts an increase in temperature. Reactor power and multiplication factor do show agreement between the two codes. Both COBRA-TF and TRACE predict a decline in reactor power. Similarly, both codes predict the reactor multiplication factor to have a downward slope from an initial value of one and it stays under one at the end of the transient.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

COBRA-TF	Coolant Boiling in Rod Arrays-Two Fluid
FBM	FeedBack Model
DHM	Decay Heat Model
CTF	Cobra-TF
LWR	Light Water Reactor
BWR	Boiling Water Reactor
PWR	Pressurized Water Reactor
IPM	Input Power Model
PKM	Point Kinetics Model
PK	Point Kinetics
ANS	American Nuclear Society
PJA	Prompt Jump Approximation
PARCS	Purdue Advanced Reactor Core Simulator
TRACE	TRAC/RELAP Advanced Computational Engine
SNAP	Symbolic Nuclear Analysis Package
T-H	Thermal-Hydraulic
PKE	Point Kinetics Equation
AFLUX	Axial Flux
DNBR	Departure from Nucleate Boiling Ratio
ASCII	American Standard Code for Information Interchange

ACKNOWLEDGEMENTS

I would like to express my profound appreciation to a list of people who gave me support and help for the completion of this work.

I thank my research advisor Dr. Maria Avramova for the guidance in my research and accurate decisions concerning my academic requirements.

I thank Dr. Kostadin Ivanov and Dr. Arthur Motta for spending time to read my thesis.

I thank Dr. Jack Brenizer and Dr. Martin Trethewey for their support and believing in my promise to finish what I started.

I would also like to express my gratitude to the faculty and staff of the Mechanical and Nuclear Engineering department.

Chapter 1

Introduction

The subject of this thesis is the development and implementation of the Point Kinetics Model in COBRA-TF, which from hereon after can be referred to as PKM. There are two associated models which are to be used with the PSU version of COBRA-TF, namely, FeedBack Model or FBM, and Decay Heat Model or DHM. There has been continued work and related development to further refine these models to increase their reliability and applicability to the problem on hand. As mentioned earlier, COBRA-TF (CTF), PSU version of COBRA, is being employed for development.

The methodology consists of the initial development of the Point Kinetics Model along with associated work on Feedback Model and Decay Heat Model. Initial work is focused on a basic working PKM model and to make sure it is stable and can be integrated into COBRA-TF with ease. Secondary, but equally important, focus is the modeling of Feedback Model and Decay Heat Model. The Feedback Model includes Doppler and Moderator feedback effects whereas the Decay Heat Model accounts for the fission product and actinide decay heat. The main goal here is to reach a point where all three of these models can be implemented into COBRA-TF for use in practical applications.

Just to give a brief over view of COBRA-TF, it is a thermal-hydraulics reactor analysis code based on the subchannel approach. Sub-channel approach based codes were initially developed for PWR applications to calculate the crossflow in the open lattice cores and solved for axial flow, and used empirical lateral pressure drop relationships to couple axial flows in adjacent channels. Most of these early developed codes utilized homogeneous equilibrium or drift flux

modeling of the two-phase flow. Further development resulted in these subchannel codes employing multi-fluid, multi-field representation of the two-phase flow. Today's subchannel codes have improved transverse momentum equations that allow the use of either Cartesian or subchannel coordinates. This allows one to use these codes for a fully 3-D treatment of reactor structures. Continued improvements have been made to subchannel codes. CTF is one of the codes that has benefited from these improvements. These improvements have mainly been focused on prediction of the Light Water Reactors' thermal margins, flow regime selection for safety analysis, turbulent mixing and void drift models, and spacer grid effects.

In the subchannel analysis, the control volume is of a size equivalent to a single fuel rod and its associated fluid. The fluid properties are averaged values within the subchannel. Constitutive relations are in turn needed in the governing equations for axial and lateral friction, form losses, energy exchanges between subchannels, and mass transfer between subchannels.

1.1. **COBRA-TF Overview**

CTF is the Reactor Dynamics and Fuel Management Group's version of thermal-hydraulic subchannel code COBRA-TF [1-3]. It stands for COolant Boiling in Rod Arrays – Two Fluid. It is used for best estimate analysis for LWR safety margins. It uses a 3-fields representation of two-phase flow. This configuration results in a set of nine time-averaged conservation equations written in a semi-implicit form using a donor cell differencing for the two convective quantities. It can be used either with rectangular or subchannel coordinates. The momentum equations can be represented either in Cartesian coordinates or subchannel coordinate system. One of the biggest benefits of subchannel approach is that any vertical, one-, two-, or three dimensional reactor vessel components can be modeled with good accuracy. The equations

for flow field in the reactor core are solved using a staggered difference scheme on the Eulerian mesh. The characteristics are as follows:

- velocities are obtained at the mesh faces
- state variables are obtained at the cell center
- the scalar mesh cell is characterized by cross-sectional area , height , and the width of the connections to the adjacent mesh cell,
- the momentum mesh cell is centered on the scalar cell face

The mesh cells are defined by input in terms of subchannels where a subchannel is a vertical stack of single mesh cells. Several subchannels can be connected together by gaps to model a region of the reactor vessel. Sections that occupy the same level then form a section. These sections are connected axially to complete the vessel mesh by specifying subchannel connections between sections. Heat transfer surfaces and solid structures that interact significantly with the fluid can be modeled with rods and unheated conductors. One thing that needs to be mentioned is the fact that subchannel formulation is often given preference over the Cartesian formulation for modeling complex and irregular analysis.

Input for the code is a simple text file where the data is arranged in the form of cards. Each card specifies a particular parameter. The code makes a few assumptions when solving he conservation equations

- Gravity is considered the only body force;
- No volumetric heat generation is assumed in the fluid;
- Radiation heat transfer is limited to rod-to-drop and rod-to-steam;
- Same pressure in all phases is assumed;
- Viscous dissipation is neglected in the enthalpy formulation of the energy equation.

1.2. Point Kinetics Model Background

"Exact" Point Kinetics Model (PKM) means that a given time-, space-, and energy-dependent neutronics model is condensed or lumped into the form of the Point Kinetics (PK) equations without applying simplifying approximations. The exact point kinetics equations are formally about the same as the traditional approximate PK equations. However, the derivation of the exact PK equations provides improved definition for the integral kinetics parameters, ρ (reactivity), β (delayed neutron fraction), and Λ (neutron generation time or neutron prompt lifetime) as well as all quantities involving p (amplitude function), ζ_k (reduced delayed neutron family/group fractions), and s (external source). The exact point kinetics equations were first derived by Henry [4]. The derivation of the final form of the point kinetics model can be found in other reactor analysis textbooks as well [5]. In addition to the exact point kinetics equations, there are approximate point kinetics equations that are based on certain simplifying assumptions. Most of the practical applications are based on the use of approximate "point kinetics equations".

The conceptual starting point of the derivation of the exact point kinetics equations from time-, space-, and energy-dependent neutronics equations is the *factorization* of the neutron flux into a purely time-dependent amplitude function, $p(t)$, and a space-, energy-, and time-dependent shape function, $\psi(r, E, t)$:

$$\phi(\mathbf{r}, E, t) = p(t) \cdot \psi(\mathbf{r}, E, t) \tag{1.1}$$

This factorization of the flux is not an approximation such as the separation of the flux as it is done in the approximate PK equations. In those equations the time dependence of the flux $\Phi(\mathrm{r}, E, t)$ is assumed to be separable in its space and energy dependence:

$$\Phi(r,E,t) = p(t)\psi(r,E) \tag{1.2}$$

The factorization merely splits one function into two. But then a second equation is required to make the factorization unique. Although flux factorization is general, employment is particularly advantageous if the flux variation with time consists primarily of flux amplitude changes with comparatively small time variations of the shape function. Such conditions are realized for most nuclear reactor transients. Some exceptions are special experiments that use an external neutron source in the form of a pulsed beam.

The second equation required to make the factorization unique can be used to shift the major time dependence into the amplitude function by constraining the time variation of the "magnitude" of the shape function. A convenient way to do this is to hold some integral (over space and energy) of the shape function constant in time. The derivation leads to a specific integral of ψ, which is held constant in time in order to obtain the desired form of the point kinetics equations.

Factorizing the neutron flux, applying an arbitrary weighting function, and constraining the variation of the flux shape do not introduce an approximation. In a sense, the factorization introduces a new degree of freedom (two instead of one function), which needs to be eliminated again by a constraint condition. The weight function only influences the precise split of the flux into amplitude and shape function as well as the definitions of all other quantities appearing in the point kinetics equations. The end result for the flux,

$$\phi(\mathbf{r},E,t) = p^{w}(t)\cdot\psi^{w}(\mathbf{r},E,t)\,, \tag{1.3}$$

is unique despite the dependence of the magnitude of both factors on the weighting function; the dependence of p and ψ on the choice of the weight function is indicated by the superscript w in Eq. (1.3).

Since the shape function $\psi(\mathbf{r}, E, t)$ is not known precisely, the resultant flux and particularly the flux amplitude $p(t)$ will only be approximate solutions. In this practical case then, both will depend on the choice of the weight function. It is thus advantageous to choose the weight function in such a way that it reduces the error resulting from inaccuracies in the shape function. Since the solution of the point kinetics equation is particularly sensitive to an error in the reactivity, a weight function should be chosen that reduces the effect of the shape function inaccuracies on the reactivity. The initial adjoint flux, $\phi_0^*(\mathbf{r}, E)$, fulfills this objectives. With the initial adjoint flux as a weighting function,

$$w(\mathbf{r}, E) = \phi_0^*(\mathbf{r}, E), \tag{1.4}$$

the constraint condition for the shape function is given by

$$\int_V \int_0^\infty \frac{\phi_0^*(\mathbf{r}, E)\psi(\mathbf{r}, E, t)}{\upsilon(E)} dE\, dV = K_0\,, \tag{1.5}$$

with K_0 being a constant.

The exact kinetics equations can be written in the form:

$$\dot{p}(t) = \frac{\rho(t) - \beta(t)}{\Lambda(t)} p(t) + \frac{1}{\Lambda_0} \sum_k \lambda_k \zeta_k(t) \tag{1.6}$$

$$\dot{\zeta}_k(t) = -\lambda_k \zeta_k(t) + \frac{F(t)}{F_0} \beta_k(t) p(t) \tag{1.7}$$

The integral kinetics parameters, ρ, β_k, and Λ are defined as integrals over space and energy of integrands that depend on the time-dependent space function, $\psi(\mathbf{r}, E, t)$. If $\psi(\mathbf{r}, E, t)$ could be calculated and used to find $\rho(t)$, $\beta_k(t)$, and $\Lambda(t)$ from their "exact" definitions, one could use them in the exact point kinetics equations to determine the amplitude function $p(t)$. However, the generation of such shape functions would require solutions of full $(\mathbf{r}, E, t)$-dependent problems. Approximate methods for such calculations comprise what is called "space-energy dependent dynamics". There is, however, definite need for simpler approximate methods. The most commonly used approach is called the "point reactor model".

The kinetics equations for the point reactor model are normally referred to as "point kinetics equations", without an adjective. They can be obtained from the exact point kinetics equations by introducing one major and two minor simplifications. The major simplification consists of neglecting the time dependence of the flux shape; that is the initial flux shape,

$$\psi(\mathbf{r}, E, t) \cong \phi_0(\mathbf{r}, E), \tag{1.6}$$

is used in forming the integral kinetics parameters. If $p_0 = 1$, the initial flux shape is equal to the initial flux $\phi_0(\mathbf{r}, E)$. The time dependence of the neutron flux is thus assumed separable from the spatial and energy $(\mathbf{r}, E)$ dependences.

The implementation of these approximations and the replacements into the exact point kinetics equations yields what is usually referred to as the point kinetics equations:

$$\dot{p}(t) = \frac{\rho(t) - \beta}{\Lambda} p(t) + \frac{1}{\Lambda} \sum_k \lambda_k \zeta_k(t) \tag{1.9}$$

$$\dot{\zeta}_k(t) = -\lambda_k \zeta_k(t) + \beta_k p(t) \tag{1.10}$$

1.3. **Model Basics**

The primary energy source for a nuclear-reactor power plant is the reactor core. In the following sections, the TRACE [6] model (which is based on the TRAC/PF1/MOD2 [7] model**)** is used as an example to show how the system thermal-hydraulic codes determine the core power. The total reactor power has two components (i.e. fission power and decay heat power). The TRACE decay heat models are based on the 1979 [8] and 1994 [9] ANS decay heat standards. Two reactor power methods are used by the code (in addition TRACE is coupled to PARCS for determination of reactor power by a full 3D transient neutronics calculation). In the first method, the user specifies power to be a constant or defined by a power table supplied through input. The table is a function of a system signal-variable parameter or a control-block output-signal parameter. Values between entries in the table are determined by linear interpolation. In the

second method, the user determines power from the solution of the point reactor kinetics equations with reactivity feedback. These equations specify the time behavior of the core power level with neutronic reactivity, ρ, which is a sum of programmed reactivity ρ_{prog} and feedback reactivity ρ_{fdbk}, as the driving parameter. The user inputs programmed reactivity (defining reactivity effects not accounted for by feedback reactivity, such as fuel reactivity and control-rod movement) with the same forms that define power in the first method (as a constant or by a table). The code evaluates feedback reactivity based on changes in the core-averaged fuel temperature, coolant temperature, coolant vapor (gas volume) fraction, and boron concentration. Note that the TRACE is a code for LWR analysis. In these reactors the light water is used as a coolant *and* a moderator.

Chapter 2

Development of Point Kinetics Model without Feedback

2.1. Point Kinetics Model (PKM)

To start things off, PSU version of COBRA, CTF, has been modified in order to implement the new models. Currently, there are two methods used by the code to calculate power. The default method, which currently exists in CTF comprises of the user specifying the power to be constant or defined by a power vs. time table supplied through the CTF input. This method is known as Input Power Model (IPM). PKM is employed in the second model to determine power from the solution of point reactor kinetics equations. These neutronic equations predict the core power level with neutronic reactivity, where the neutronic reactivity is a sum of both the programmed reactivity as well as the feedback reactivity. The former is input by the user and includes things such as fuel reactivity and control rod movement. With these input, the code then goes ahead and evaluates the feedback reactivity based on changes in the core-averaged fuel temperature, moderator temperature, moderator density, and boron concentration. To help speed up the implementation of PKM into CTF, similar methodology as the one used in TRACE and TRAC-PF1 was used. Even though TRACE is primarily PWR oriented model, the methodology still helps in devising PKM approach for CTF. Another addition is the implementation of algorithms based on boron tracking model to evaluate the core-averaged boron concentration, which is being developed in CTF within the framework of another Ph.D. thesis.

The utilized point-reactor-kinetics equations are a coupled set of (***I+1***) first-order differential equations defining the fission power P and the delayed-neutron precursor concentrations C_i as a function of time:

$$\frac{dP}{dt} = \frac{\rho - \beta}{\Lambda} P + \sum_{i=1}^{I} \lambda_i C_i + \frac{S}{\Lambda(1-\rho)} \tag{2.1}$$

$$\frac{dC_i}{dt} = -\lambda_i C_i + \frac{\beta_i P}{\Lambda} \qquad for \quad i = 1, 2.....I \tag{2.2}$$

where

P is the thermal power (W) that results from fission occurring at time t (s);

$k = 1 + \Delta k_{fdbk} + \Delta k_{prog}$ is the reactor multiplication factor;

$\rho = \frac{(k-1)}{k} = \rho_{prog} + \rho_{fdbk}$ is the neutronic reactivity, (including both programmed reactivity and feedback reactivity);

$\rho_{prog} = \Delta k_{prog} / k$ is programmed reactivity, an input value, based on the fuel reactivity and control rod movement;

$\rho_{fdbk} = \Delta k_{fdbk} / k$ is the feedback reactivity, evaluated by the code and based on changes in the core-average fuel temperature, coolant temperature, coolant vapor fraction, and boron concentration;

$\beta = \sum_{i=1}^{I} \beta_i$ is the total fraction of delayed neutrons where β_i is the fraction of delayed neutrons in group *i* and I is the number of delayed-neutron groups;

Λ is the effective prompt-neutron lifetime (s);

λ_i is the decay constant for the delayed-neutron precursors in group *i* (s^{-1});

C_i is the concentration of the delayed-neutron precursors in group *i* (W);

S is the thermal power (W) from the external source of neutrons in the reactor cores that are producing fission.

Kaganove method is used to solve these equations.

The thermal-power solution is applied to the decay-heat equations

$$\frac{dH_j}{dt} = -\lambda_j^H H_j + E_j P \qquad for \quad j = 1, 2.....J \tag{2.3}$$

where

P is the solution of Eqs. (2.1) and (2.2)

H_j is the energy of the decay-heat group *j* (W.sec);

λ_j^H is the decay constant for decay-heat group *j* (s^{-1});

E_j is the effective energy fraction of decay-heat group *j*;

and J is the number of decay-heat groups.

After solving each j^{th} equation represented by Eq. (2.3) for the decay-heat group concentration H_j, the code calculates the total thermal power generated in the reactor fuel at time t from prompt fission, precursor decay, delayed fission, and the external source:

$$P_{eff} = P + \sum_{j=1}^{J} \lambda_j^H H_j \tag{2.4}$$

As input, one requires the number of delayed-neutron groups, I; the delayed-neutron parameters, λ_i and β_i; the number of decay-heat groups, J; the decay-heat parameters, λ_j^H and E_j; and either the total thermal-power history $P(-t)$ for $t \le 0$ for the initial delayed-neutron precursor concentrations $C_i(0)$ and decay-heat concentrations, $H_j(0)$.

Usually by default one can internally set I to 6 and defines λ_i and β_i with the values in Table 2.1. If $I \leq 0$ and no total thermal-power history is an input, the code assumes that an initial steady-state condition exists to initialize $C_i(0)$ internally. One can set the time derivative in Eq. (2.5) to zero and calculate the initial values of C_i from the following:

$$C_i(0) = \frac{\beta_i}{\lambda_i \Lambda} P(0) \qquad for\ i = 1, 2....I \tag{2.5}$$

where $P(0)$ is the initial power specified through input.

Table 2.1: Delayed Neutron Constants

Group *i*	**Decay Constant**	**Neutron Fraction**
1	3.87	0.000169
2	1.40	0.000832
3	0.311	0.00264
4	0.115	0.00122
5	0.0317	0.00138
6	0.0127	0.000247

2.1.1. **Default Data for the Decay-Heat Groups**

If $J \leq 0$ is input, one can internally sets J to 69 and defines λ_j^H and ED_j with the values in Table 3.3. These decay heat parameters are obtained from the 1979 ANS decay-heat

standard [8]. If the default decay-heat groups are used or if the user inputs 69 or 71 decay-heat groups, the fraction of fission power due to each of the three isotopes, ^{235}U, ^{239}Pu, and ^{238}U, must be input. In addition, in order to convert the MeVs of decay energy per fission to a fraction of the total fission energy, the MeVs per fission for each of the three isotopes, ^{235}U, ^{239}Pu, and ^{238}U, must be input. ED_j in Table 3.3 is converted to E_j with the following equation:

$$E_j = ED_j / (\lambda_j^H Q_k) \tag{2.6}$$

where Q_k is the MeV/fission for isotope k. If $J \leq 0$ and no total thermal-power history is input, the code assumes that an initial steady-state condition exists to initialize $H_j(0)$ internally. The code sets the time derivative in Eq. (2.3) to zero and calculates the initial values of the H_j from the following:

$$H_j(0) = \frac{E_j}{\lambda_j^H} P(0) \qquad for\ j = 1, 2....J \tag{2.7}$$

where $P(0)$ is the initial steady-state power specified through input. This is equivalent to assuming an infinite reactor operating power at the initial power.

2.1.2. **Prompt-Fission Power History**

If the total thermal-power history $P(-t)$ is input, TRACE evaluates $C_i(0)$ and $H_j(0)$ from Eqs. (2.2) and (2.3). The $P(-t)$ consists of tabular data (t_l, P_l) pairs for $l = 1, 2, 3 \ldots L$, where $-t_{l+1} \le -t_l \le 0$. The code assumes that the total thermal power varies linearly between data pairs in the power history table, that is,

$$P(t) = a_l + b_l t \quad \text{, over the time interval } -t_{l+1} \le t \le -t_l \tag{2.8}$$

Substituting Eq. (2.8) into Eq. (2.2) and integrating the resulting equation analytically from $t = -t_L$ [where $C_i(-t_L) = 0$ is assumed] to $t = 0$ gives

The decay heat parameters are obtained from the 1979 ANS decay heat standard. If the default decay heat groups are used, the fraction of fission power due to each of the three isotopes, ^{235}U, ^{239}Pu, and ^{238}U must be input. Both the 1971 and 1979 decay heat standards have been implemented in the code. A detailed description of the two heat standards is given in chapter 3.

2.2. **Solution of the Point-Reactor Kinetics**

The code solves the point-reactor kinetics equations [Eqs.(2.1) through (2.3)] by the Kaganove method. The derivation of the Kaganove method approximates the time dependence of P and $k_{ex} = k - 1 = \dfrac{\rho}{(1-\rho)}$ over each integration time step by second-order polynomials and assumes

that $\Lambda(t) = \ell/[1+k_{ex}(t)]$, where ℓ is a constant. It is more appropriate to approximate the time dependence of P by a second-order polynomial, ρ by a first-order polynomial, and Λ by a constant if one linearly extrapolates its estimate of $\rho(t)$ over the fluid-dynamics time step to be evaluated and because the weak time dependence of Λ generally is unknown. These modified assumptions, when applied to the point-reactor-kinetics equations, simplify the form that these equations take when analytically integrated over the integration time step Δt. The derivation of those analytically integrated equations follows. One can then evaluate them for $\Delta t^n / \Delta t$ integration time steps during the Δt^n fluid-dynamics time step.

We assume that

$$P(t) = P(0) + P_1 t + P_2 t^2 , \tag{2.9}$$

$$\rho(t) = \rho(0) + \rho_1 t ,$$

and

$\Lambda, \lambda_i, \beta_i, \lambda_j^H,$ and E_j are constant for $0 \le t \le \Delta t$,

where

$$P(0), \rho(0), \rho_1 = \left[\left(\rho^{est}\right)^n - \rho(0)/\Delta t^n\right] \text{ are known values.} \tag{2.10}$$

Solving Eq. (2.2) for $C_i(t)$ in terms of a functional of the prompt-fission power $P(t)$ gives

$$C_i(t) = C_i(0)\exp[-\lambda_i t] + \frac{\beta_i}{\Lambda}\int_0^t \exp[-\lambda_i(t-\tau)]P(\tau)\,d\tau \qquad (2.11)$$

Substituting Eq. (2.9) into Eq. (2.1) and integrating the resulting equation term-by- term gives

$$P(t) = P(0) + \int_0^t \frac{\rho(\tau)P(\tau)}{\Lambda}\,d\tau - \sum_{i=1}^{I}\frac{\beta_i}{\Lambda}\int_0^t \exp[-\lambda_i(t-\tau)]P(\tau)\,d\tau \\ \sum_{i=1}^{I} C_i(0)[1-\exp[-\lambda_i(t)] + \frac{S}{\Lambda}\left[1-\rho(0)t-\rho_1\frac{t^2}{2}\right] \qquad (2.12)$$

Substituting Eq. (3.8) for $P(t)$ and Eq. (3.9) for $\rho(t)$ in Eq. (2.10), evaluating the integrals, and rearranging the resulting equation in terms of the P_1 and P_2 unknowns gives

$$a_1(t)P_1 + a_2(t)P_2 = q(t),$$

where

$$a_1(t) = t\Lambda - t^2\left[\frac{1}{2}\rho(0)+\frac{1}{3}\rho_1(t)\right] + \sum_{i=1}^{I}\beta_i t^2 I_{i_1}(t), \qquad (2.13)$$

$$a_2(t) = t^2\Lambda - t^3\left[\frac{1}{3}\rho(0)+\frac{1}{4}\rho_1(t)\right] + \sum_{i=1}^{I}\beta_i t^3 I_{i_2}(t),$$

$$q(t) = S + t\left[\rho(0)+\frac{1}{2}\rho_1(t)\right][P(0)-S] + \sum_{i=1}^{I}[\Lambda\lambda_i C_i(0) - \beta_i P(0)]tI_{i_0}(t),$$

and

$$t^{m+1} I_{im}(t) = \int_0^t \exp[-\lambda_i (t-\tau)] \tau^m d\tau \quad \text{, for } m = 0, 1, 2$$

The P_1 and P_2 polynomial coefficients are evaluated by requiring Eq. (2.11) to be satisfied for $t = \Delta t$ (at the end of the integration time step) and $t = \Delta t / 2$ (at the midpoint of the integration time step). Solving the two equations

$$a_1(\Delta t) P_1 + a_2(\Delta t) P_2 = q(\Delta t) \tag{2.14}$$

and

$$a_1\left(\frac{\Delta t}{2}\right) P_1 + a_2\left(\frac{\Delta t}{2}\right) P_2 = q\left(\frac{\Delta t}{2}\right) \tag{2.15}$$

for P_1 and P_2 gives

$$P_1 = \frac{\left[q(\Delta t) a_2\left(\frac{\Delta t}{2}\right) - q\left(\frac{\Delta t}{2}\right) a_2(\Delta t) \right]}{D} \tag{2.16}$$

and

$$P_2 = \frac{\left[q\left(\frac{\Delta t}{2}\right) a_1(\Delta t) - q(\Delta t) a_1\left(\frac{\Delta t}{2}\right) \right]}{D} \text{,}$$

where

$$D = a_1(\Delta t)a_2\left(\frac{\Delta t}{2}\right) - a_1\left(\frac{\Delta t}{2}\right)a_2(\Delta t)$$

(2.17)

The total thermal power at the end of the integration time step $t = \Delta t$ from Eq. (3.8) is

$$P(\Delta t) = P(0) + P_1\Delta t + P_2\Delta t^2 \tag{2.18}$$

If we know $P(t)$ for $0 \le t \le \Delta t$, we can evaluate $C_i(\Delta t)$ by substituting Eq. (3.8) into Eq. (2.9) and using

$$\Delta t I_{i_0}(\Delta t) = \frac{1 - \exp(-\lambda_i \Delta t)}{\lambda_i} \tag{2.19}$$

by analytically integrating its definition in Eq. (2.11):

$$C_i(\Delta t) = C_i(0)\exp[1 - \lambda_i \Delta t I_{i_0}(\Delta t)] + \frac{\beta_i}{\Lambda}[P(0)\Delta t I_{i_0}(\Delta t) + P_1\Delta t^2 I_{i_1}(\Delta t) + P_2\Delta t^3 I_{i_2}(\Delta t)], \quad \textit{for } i = 1, 2, \ldots, I \tag{2.20}$$

The code evaluates the decay-heat equations in the same way to give

$$H_j(\Delta t) = H_j(0)[1-\lambda_j^H \Delta t I_{j_0}(\Delta t)] + E_j[P(0)\Delta t I_{j_0}(\Delta t) + P_1 \Delta t^2 I_{j_1}(\Delta t) + P_2 \Delta t^3 I_{j_2}(\Delta t)], \quad for\ j = 1, 2, \ldots, J \tag{2.21}$$

where

$$\Delta t^{m+1} I_{j_m}(\Delta t) = \int_0^{\Delta t} \exp[-\lambda_j^H (\Delta t - \tau)] \tau^m d\tau$$

The code evaluates $\Delta t^{m+1} I_{i_m}(\Delta t)$ with the following recursion relation:

$$\Delta t^{m+1} I_{i_m}(\Delta t) = \frac{\Delta t^m - m \Delta t^m I_{i_{m-1}}(\Delta t)}{\lambda_i}, \text{ for } m = 1, 2 \tag{2.22}$$

once

$$\Delta t I_{i_0}(\Delta t) \frac{1-\exp(\lambda_i \Delta t)}{\lambda_i}$$

is first evaluated. For $\lambda_i \Delta t << 1$, Eq. (2.19) results in the loss of several least-significant digits of accuracy with each application of the recursion formula. Thus, for $\lambda_i \Delta t < 1$, code first evaluates $\Delta t^3 I_{i_2}(\Delta t)$ using the Maclaurin expansion,

$$\Delta t^{m+1} I_{i_m}(\Delta t) = \Delta t^{m+1} m! \left[\frac{1}{(m+1)!} - \frac{\lambda_i \Delta t}{(m+2)!} + \frac{(\lambda_i \Delta t)^2}{(m+3)!} - \cdots \right] \tag{2.23}$$

for *m* = 2, and then evaluates the reciprocal of the Eq. (2.22)

$$\Delta t^m I_{i_{m-1}}(\Delta t) = \frac{\Delta t^m - \lambda_i \Delta t^{m+1} I_{i_m}(\Delta t)}{m}, \text{ for } m = 2,1 \tag{2.24}$$

The accuracy of the second-order polynomial approximation for $P(t)$ can be increased by decreasing the integration time-step size Δt. We would like Δt to be as large as possible, however, while we maintain a desired level of accuracy in approximating $P(t)$. To achieve this, the following procedure for automatically adjusting the reactor-kinetics time step in the code is used. In the same manner that Eqs. (2.12) and (2.13) were defined; Eq. (2.11) also could be required to be satisfied at $t = \Delta t / 4$:

$$a_1\left(\frac{\Delta t}{4}\right)P_1 + a_2\left(\frac{\Delta t}{4}\right)P_2 = q\left(\frac{\Delta t}{4}\right) \tag{2.25}$$

The above equation together with Eq. (2.13), which is evaluated at $t = \Delta t / 2$, are solved over the time range:

$$P_1^* = \frac{q\left(\frac{\Delta t}{2}\right)a_2\left(\frac{\Delta t}{4}\right) - q\left(\frac{\Delta t}{4}\right)a_2\left(\frac{\Delta t}{2}\right)}{D^*} \tag{2.26}$$

and

$$P_2^* = \frac{q\left(\frac{\Delta t}{4}\right)a_1\left(\frac{\Delta t}{2}\right) - q\left(\frac{\Delta t}{2}\right)a_1\left(\frac{\Delta t}{4}\right)}{D^*}, \tag{2.27}$$

where

$$P_2^* = a_1\left(\frac{\Delta t}{2}\right)a_2\left(\frac{\Delta t}{4}\right) - a_1\left(\frac{\Delta t}{4}\right)a_2\left(\frac{\Delta t}{2}\right)$$

This solution over half the integration time step provides a more accurate value for $P(t)$ at $t = \Delta t/2$,

$$P^*\left(\frac{\Delta t}{2}\right) = P(0) + P_1^*\left(\frac{\Delta t}{2}\right) + P_2^*\left(\frac{\Delta t}{2}\right)^2 \tag{2.28}$$

then the solution over the full integration time step $t = \Delta t/2$

$$P\left(\frac{\Delta t}{2}\right) = P(0) + P_1\left(\frac{\Delta t}{2}\right) + P_2\left(\frac{\Delta t}{2}\right)^2 \tag{1.29}$$

The code uses the difference between these two values as a measure of the error in the solution over the full integration time step Δt

$$\varepsilon(\Delta t) = \left|\frac{P\left(\frac{\Delta t}{2}\right) - P^*\left(\frac{\Delta t}{2}\right)}{P^*\left(\frac{\Delta t}{2}\right)}\right| \tag{2.30}$$

It is desirable to increase Δt if the above e is too small and to decrease ε if Δt is too large. Toward this end, the code defines two error bounds, ε_1 and ε_2 , to provide the desired convergence such that,

1. if $\varepsilon_1 < \varepsilon(\Delta t) < \varepsilon_2$, the code maintains the present value of reactor-kinetics time step Δt ;
2. if $\varepsilon_1 < \varepsilon_2 \le \varepsilon(\Delta t)$), the code halves the value of Δt because the error in $P(\Delta t)$ is too large; and
3. if $\varepsilon(\Delta t) \le \varepsilon_1 < \varepsilon_2$, the code doubles the value of Δt because the error in $P(\Delta t)$ is too small.

When Δt is halved, the integration time step is reevaluated. When the third criterion is satisfied, the number of remaining integration time steps to be evaluated over the fluid-dynamics time step must be an even number for the integration time step to be doubled (and the number of remaining integration time steps to be halved). The code is programmed to use $\varepsilon_1 = 10^{-6}$ and $\varepsilon_2 = 10^{-4}$. A numerical study showed these values to yield a maximum fractional error less than 10^{-4} and an average fractional error less than 10^{-5} in the prompt-fission power solution. The point-reactor-kinetics solution usually requires $<1\%$ of the execution time of a thermal-hydraulic code.

2.3. Point Kinetics Test Problem

A test problem has been devised to verify the validity of the PKM model for steady state and transient conditions. It consists of ramp reactivity insertion without the thermal-hydraulic feedback. Listed below is a brief description of the problem:

In this problem, we use the PK code to solve ramp reactivity insertion problems in which no thermal-hydraulic (T/H) feedback is considered. The purpose of this problem is first to verify the point kinetics code for a transient for which analytic solution can be obtained, and then to understand the power response to insertions.

2.3.1. One Delayed Neutron Group

Consider a ramp in which the reactivity insertion is given in Figure 2.1. When only one delayed neutron group is used, an analytic solution exists. We use one group delayed neutron parameters to solve the ramp problem up to 10.0 sec. using the developed point kinetics algorithm. The one group of delayed neutron parameters is computed from the six group data given in Table 2.2.

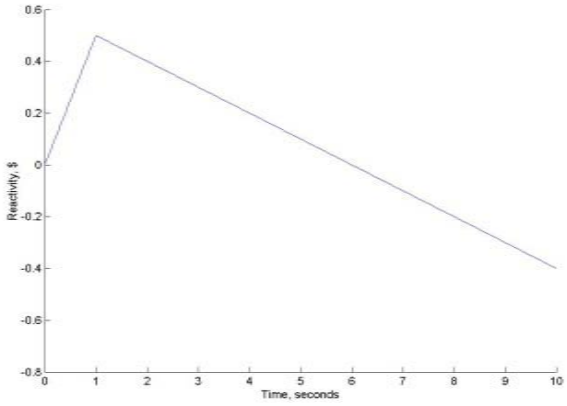

Figure 2.1: Reactivity Insertion

a) First, to simulate the PJA, let Λ=2.6E-15 seconds. We use δt=1ms and compare the power at t_m with the analytic solution. Then we increase the time step size and analyze the dependence of error on the time step size. Based on this analysis, we determine a proper time step size that gives an error of ~0.01%.

b) Then we consider the more realistic case with Λ=2.6E-5 sec. We run the case with the time step size determined in a). Then we can plot the reactivity, power, and precursor concentration and discuss the results.

c) By comparing the analytic solution with the PK solution up to 1.0 sec, we can draw a conclusion on the validity of the PJA.

2.3.2. **Six Delayed Neutron Groups**

Using six delayed neutron groups (attached sheet), we solve the ramp problem given in Part A.1

a) Then we can plot and compare the power and precursor changes with those of the one group case. We can try to explain the difference in power change behaviors between the six and one group.

b) We can locate the maximum power point and evaluate one group decay constant and compare this value with the values obtained with

$$\bar{\lambda} = \sum_k \beta_k \lambda_k / \beta \quad \text{, and} \quad \bar{\lambda}^{in} = \beta / \sum_k \beta_k / \lambda_k \qquad (2.31)$$

We then can conclude which out of three values is most meaningful to be used in one precursor group approximation.

Table 2.2: Six group data

Group	**1**	**2**	**3**	**4**	**5**	**6**
λ_i	0.0128	0.0318	0.119	0.3181	1.4027	3.9286
β_i **(%)**	0.02584	0.152	0.13908	0.30704	0.1102	0.02584

The solution and discussions are summarized below:

Part A.

A.1a Λ=2.6E-15, One delayed neutron group results from time step size Δt=1, 10, 20, 50, 100 ms are shown in Figure 2.2 (power) and Figure 2.3 (precursor).

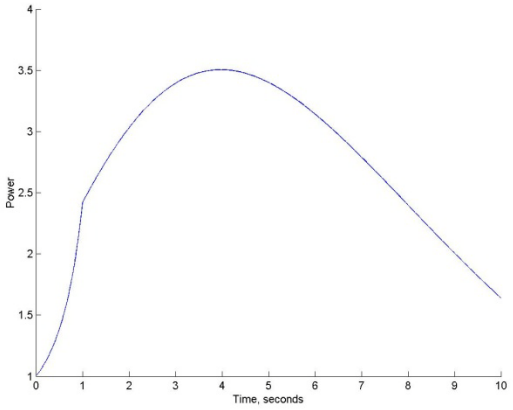

Figure 2.2: Power in problem A.1.a

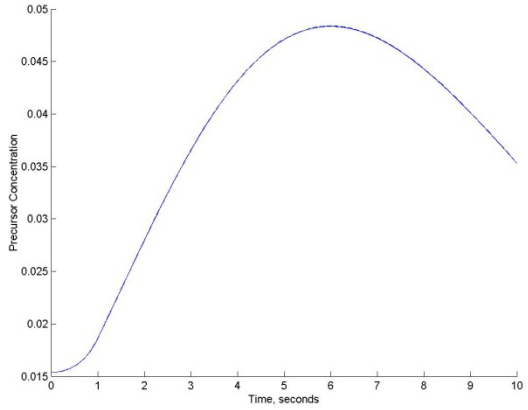

Figure 2.3: Precursor of problem A.1.a

From the figures, we can see the difference between the results from different time step sizes. The power at the point t=4s (peak power) is given in Table 2.3.

Table 2.3: Power at different time step sizes

Step Size (ms)	**Power at 4 second**	**Absolute error**	**relative error(%)**	$10^5 \times err/(\Delta t)^2$
1	3.503954	-	-	-
10	3.50399	3.6E-05	0.001027	1.027410748
20	3.504097	0.000143	0.004081	1.020275951
50	3.504845	0.000891	0.025428	1.01713664
100	3.507518	0.003564	0.101714	1.01713664

From the table, it is evident that the errors are proportional to the square of step size. The step size 20ms will be used for the following problems.

A.1b Λ=2.6E-5 One delayed neutron group results from time step size Δt=10ms are shown in Figure 2.4.

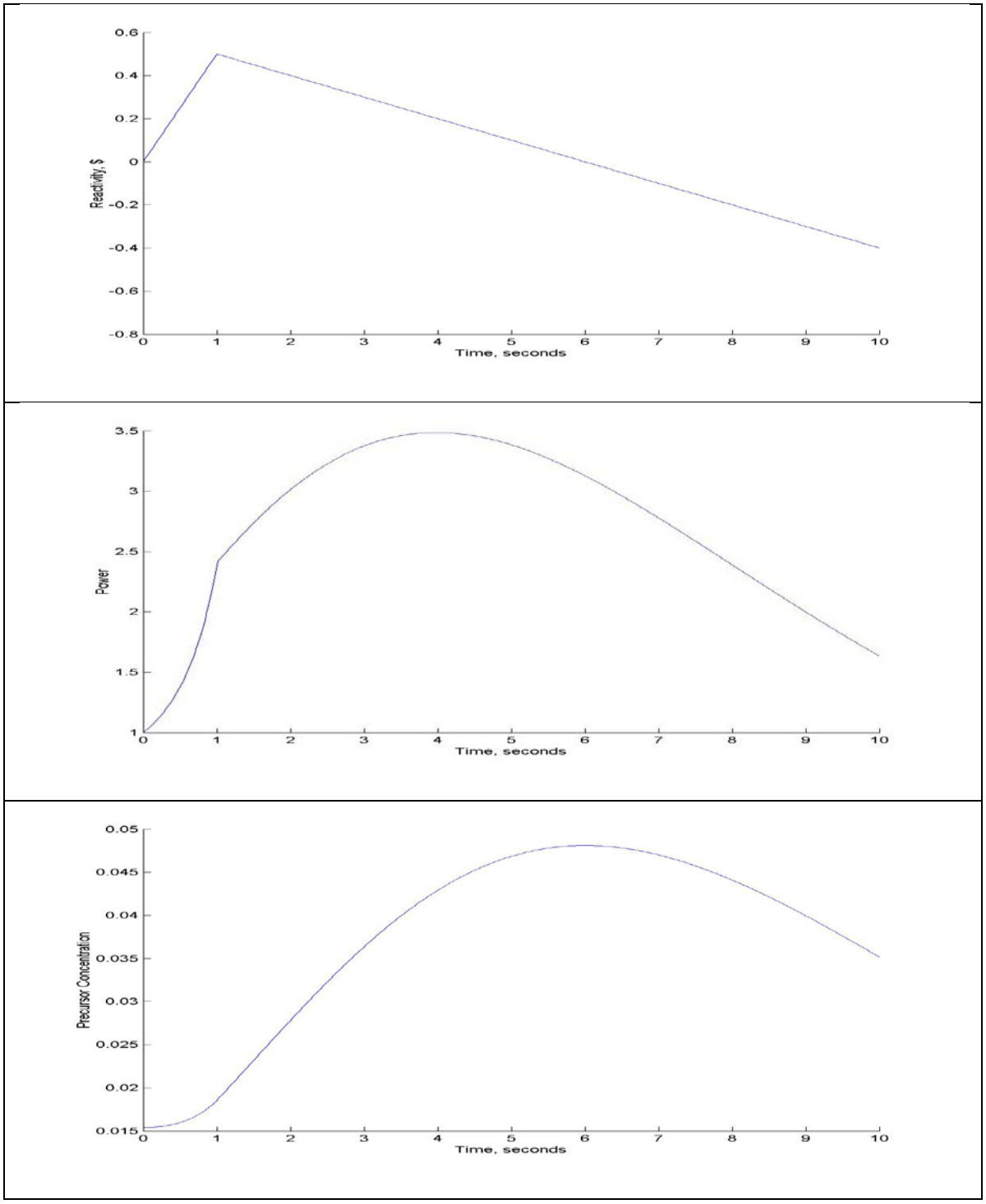

Figure 2.4: Reactivity, Power, and Precursor concentration for problem A.1b

From Figure 2.4, it is evident that before 1 second the power grows exponentially, but the precursors do not increase as fast as the power. From 1 second to 4 second, the power still increases but only asymptotically, whereas the precursors accumulate relatively fast. After 4 seconds, the power begins to decrease even though the reactivity is still positive. The precursor concentration continues to increase until 6 seconds, and then begins to decrease.

The reason for power decrease when reactivity is still positive can be explained by the Prompt Jump Approximation (PJA) formula: $\dot{P}(t) = \frac{\lambda\rho(t) + \dot{\rho}(t)}{\beta - \rho(t)} P(t)$. When $0 < \rho < \beta$, if $\dot{\rho} < 0$ then it is possible that

$\lambda\rho(t) + \dot{\rho}(t) < 0, \textit{ then } \dot{P}(t) < 0$, the power decreases.

The physical reason can be obtain from the original equation more clearly.

$$\Lambda\dot{P}(t) = \left[\rho(t) - \beta(t)\right]P(t) + \sum_k \lambda_k \zeta_k(t) \tag{2.32}$$

$$\Lambda\dot{P}(t)/P(t) = \rho(t) - \left[\beta(t) - \sum_k \lambda_k \zeta_k(t)/P(t)\right] \tag{2.33}$$

A positive reactivity cannot insure an increasing power without delay neutrons if $\rho < \beta$.

For the stationary state, $\beta(t) = \sum_k \lambda_k \zeta_k(t)/P(t)$, so inserting positive reactivity always make power increase. But during the transient, the precursors can accumulate more slowly than the increase in power, so $\beta(t) - \sum_k \lambda_k \zeta_k(t)/P(t) > 0$. If $\rho(t) < \beta(t) - \sum_k \lambda_k \zeta_k(t)/P(t)$, which happens when $\dot{\rho} < 0$, then the power will decrease, even though the reactivity is positive. In the PJA formula, $\dot{\rho}$ represents the effect of the precursors.

A.1c Analytic solution with Prompt Jump Approximation and the numerical solutions for Λ=2.6E-15, Λ=2.6E-5 are shown in figure 2.5

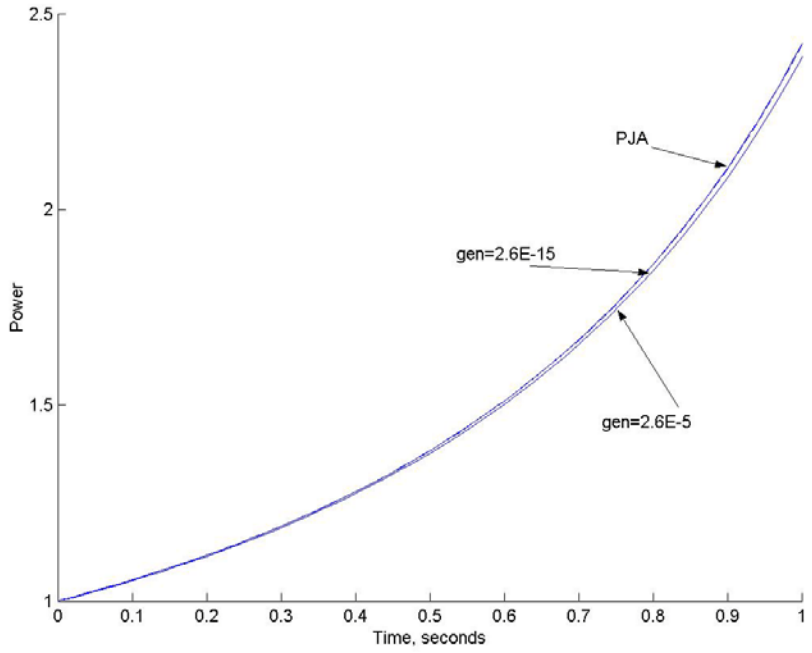

Figure 2.5: Analytic Solution with PJA and a Numerical Method

Also from Figure 2.5, we see that the analytic solution with Prompt Jump Approximation is very close to the solution from Λ=2.6E-15, it is also agrees with the solution from Λ=2.6E-5 before t<0.5s, after that there is small difference between solutions.

This suggests that the PJA is very good approximation. For the reactor with smaller neutron generation time, such as fast reactor, the result from PJA is very accurate. For the reactor with larger neutron generation time, the PJA is still a good approximation, but it will have some error.

The derivation of analytic solution with PJA is given here.

One delay neutron group point kinetics equation (PKE) is given below:

$$\Lambda \dot{P}(t) = (\rho(t) - \beta)P(t) + \lambda\zeta(t) \tag{2.34}$$

$$\dot{\zeta}(t) = -\lambda\zeta(t) + \beta P(t)$$

(2.35)

The Prompt Jump Approximation (PJA) is introduced below.

Assume: $\Lambda = 0$,

$$0 = (\rho(t) - \beta)P(t) + \lambda\zeta(t)$$

(2.36)

The derivative of (2.34) with respect to t is:

$$0 = \dot{\rho}(t)P(t) + (\rho(t) - \beta)\dot{P}(t) + \lambda\dot{\zeta}(t)$$

(2.37)

(235)+(2.36)+(2.37)/ λ :

$$\dot{\zeta}(t) = -\lambda\zeta(t) + \beta P(t) + (\rho(t) - \beta)P(t) + \lambda\zeta(t) + \frac{\dot{\rho}(t)P(t) + (\rho(t) - \beta)\dot{P}(t) + \lambda\dot{\zeta}(t)}{\lambda}$$

$$0 = \lambda\rho(t)P(t) + \dot{\rho}(t)P(t) + (\rho(t) - \beta)\dot{P}(t)$$

$$\dot{P}(t) = \frac{\lambda\rho(t) + \dot{\rho}(t)}{\beta - \rho(t)}P(t)$$

(2.38)

For a linear ramp

$$t < t_m \qquad \rho(t) = \frac{\rho_m}{t_m} t$$

$$\dot{\rho}(t) = \frac{\rho_m}{t_m}$$

$$\frac{dP(t)}{P(t)} = \frac{\lambda t + 1}{\frac{\beta}{\rho_m} t_m - t} dt = \frac{\lambda t + 1}{\tau - t} dt$$

$$\tau = \frac{\beta}{\rho_m} t_m$$

(2.39)

$$\ln \frac{P(t')}{P(0)} = \int_0^{t'} \frac{\lambda t + 1}{\tau - t} dt = \int_0^{t'} (\frac{\lambda \tau + 1}{\tau - t} - \lambda) dt = -\lambda t' - (\lambda \tau + 1) \ln \frac{\tau - t'}{\tau}$$

$$P(t) = P(0) \exp\left[-\lambda t - (\lambda \tau + 1) \ln \frac{\tau - t}{\tau} \right] = P(0) e^{-\lambda t} \exp\left[(\lambda \tau + 1) \ln \frac{\tau}{\tau - t} \right]$$

$$P(t) = P(0) e^{-\lambda t} \left[\frac{\tau}{\tau - t} \right]^{(\lambda \tau + 1)}$$

(2.40)

A.2 The PK results with six group of delay neutrons and 4 different definitions of one group delay neutron are shown in Figures 2.6, 2.7, 2.8..

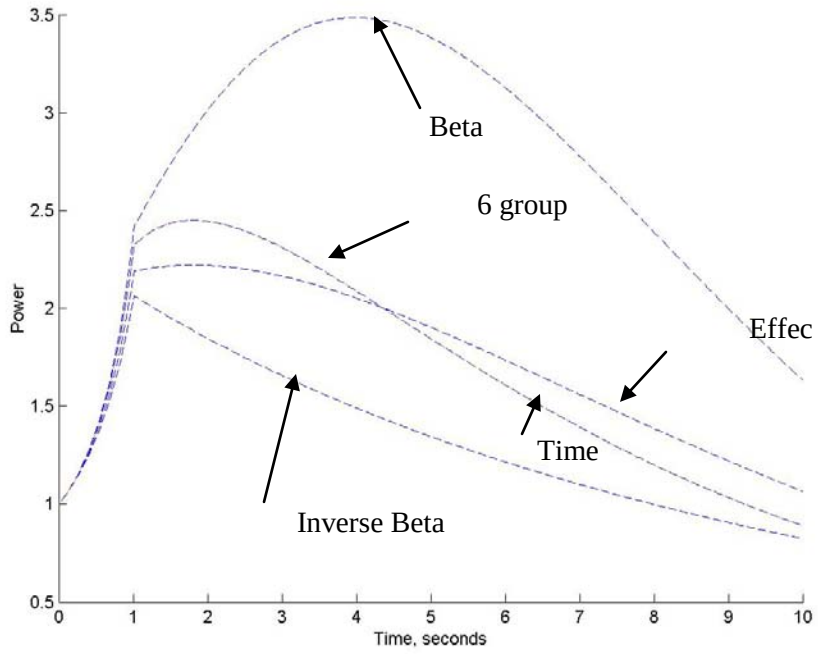

Figure 2.6: Power from six/one group delay neutron calculation

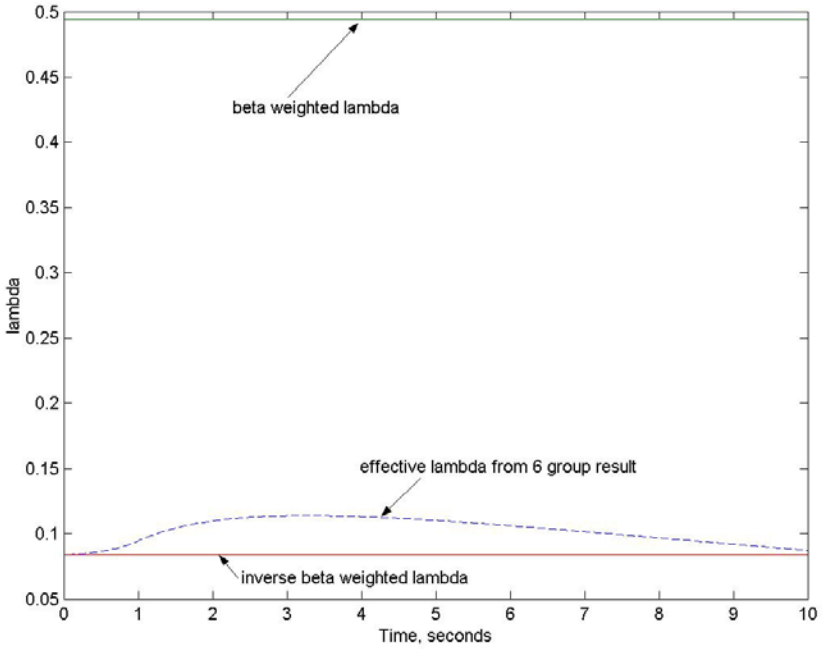

Figure 2.7: Effective one group lambda from 6 group delay neuron calculation

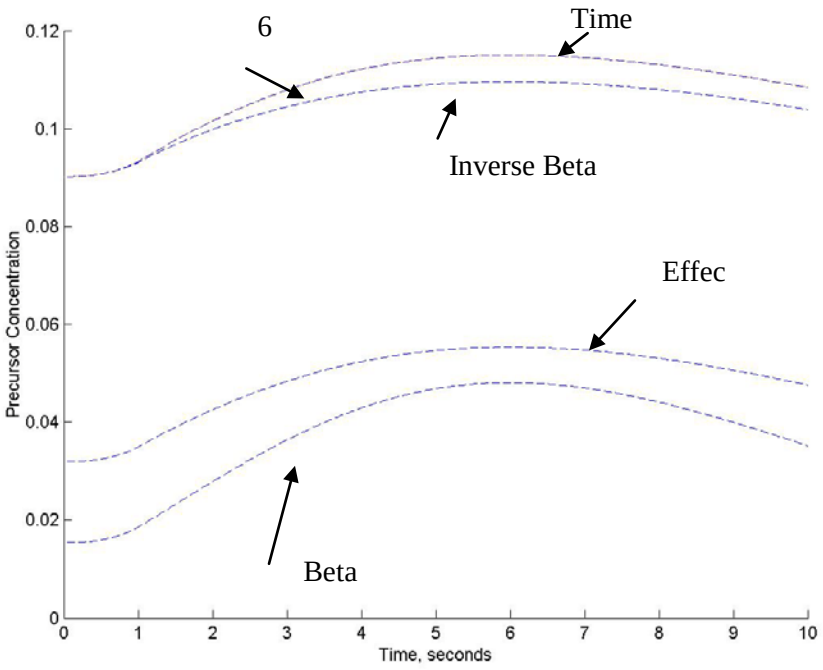

Figure 2.8: Precursors from six/one group delay neutron calculation

There are four types of lambdas for one group delay neutron:

1) Beta weighted

$$\overline{\lambda} = \sum_k \beta_k \lambda_k / \beta \qquad (=0.49405)$$

2) Inverse beta weighted

$$\overline{\lambda}^{in} = \beta / \sum_k \beta_k / \lambda_k \qquad (=0.084278)$$

3) Effective: find out the point of maximum power

t=1.8 ρ=0.42 p_m=2.449566 $\dot{P} = 0$

$$\dot{\rho} = \frac{-0.4 - 0.5}{10 - 1} = -0.1$$

from equation(30): $$\dot{P}(t) = \frac{\lambda\rho(t) + \dot{\rho}(t)}{\beta - \rho(t)} P(t)$$

$$\overline{\lambda}^e = -\dot{\rho} / \rho = 0.1 / 0.42 = 0.2381$$

4) Time dependent:

$$\overline{\lambda}(t) = \sum_k \zeta_k(t)\lambda_k / \sum_k \zeta_k(t) \qquad (2.41)$$

The time dependent lambda gives the best result, as expected. An effective lambda gives better results than the other two. However, these two lambdas depend on the results of six group delay neutron, which cannot be known “a priori” and therefore are not practical. Between beta weighted and inverse beta weighted, the inverse beta weighted is better for this problem, since it is not a rapid transient. As seen in Figure 2.5, the analytical solution employing Prompt Jump Approximation compares favorably with the numerical solution. Further verification of the standalone PKM alone verification is given in section 2.4. It consists of comparing the analytical solution of a simple PK equation to the one obtained through numerical means.

2.4. **Stand-alone PKM Verification**

The point kinetics routine is solved analytically and is then compared with the PKM numerical results.

$$n(t) = -\frac{\rho}{\beta - \rho}\exp\left(\frac{\beta - \rho}{\Lambda}t\right) + \frac{\beta}{\beta - \rho}\exp\left(\frac{\lambda\rho}{\beta - \rho}t\right) \tag{2.42}$$

Equation 2.42 is basically a simple problem to test the validity of the PKM numerical scheme implemented in COBRA-TF. As stated earlier, it is solved by analytical means and then solved using COBRA-TF. The results obtained in terms of fractional core power are then compared. The comparison results are shown in Figure 2.9.

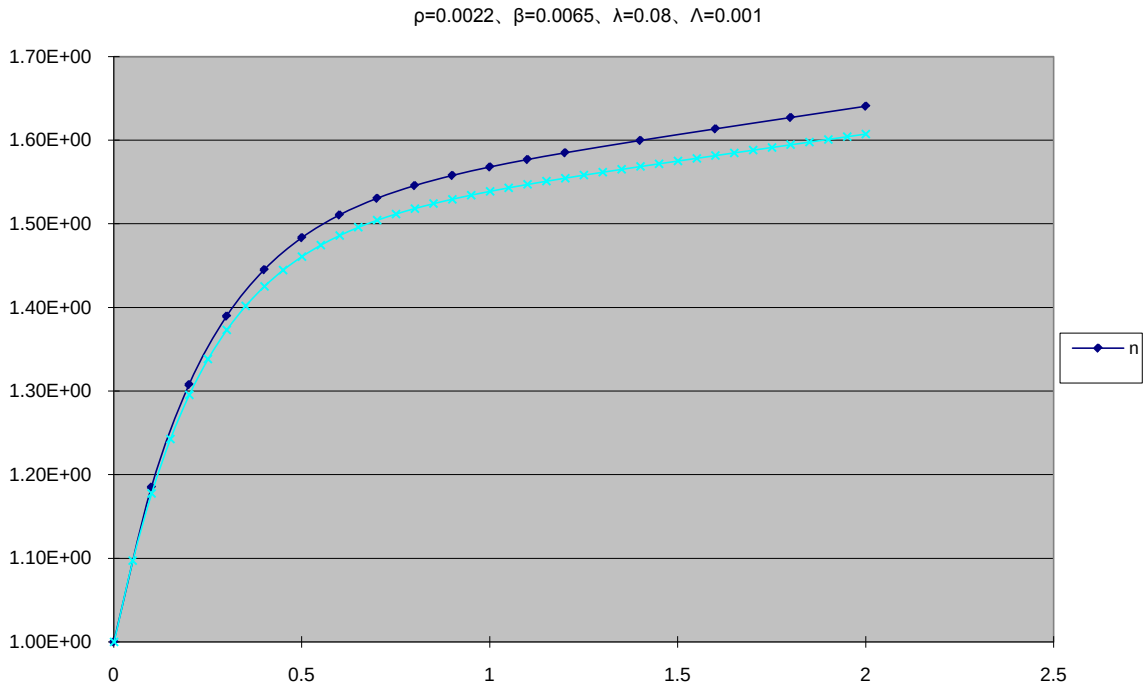

Figure 2.9: PKM and Analytical Solution Comparison (Fraction Core Power Change)

It is evident that there is a small discrepancy between the PKM and the analytical solutions. It is believed that a small error occurring at every time step, and its accumulation

through the transient calculation leads to the discrepancy shown in Figure 2.9. The discrepancy is independent of the degree of reactivity insertion, be it small or large.

Despite the minor discrepancy, the error accumulation is not enough to discount the favorable agreement between the PKM and the numerical solution. But, work continues to be done through COBRA-TF source code de-bugging to find the root cause of this minor error accumulation.

In order to give a brief idea of how the implementation is being done inside COBRA-TF, the input deck of CTF is noted below:

2.5. **Implementation of PKM in CTF**

```
*****************************************************************************
*
* GROUP 1 - Calculation Variables and Initial Conditions
*
*****************************************************************************
*
* CARD GROUP 1
*NGRP
    1
* Card 1.1
* NGAS IRFC EDMD IMIX ISOL NDM5 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
     1    2    1    2    0    0    0    0    0    0    0    0    0    0
* Card 1.2
*     GTOT      AFLUX     DHFRAC
     8.086     24.95        0.0
* Card 1.3
*     PREF        HIN       HGIN    VFRAC1    VFRAC2
    151.18   1340.40      288.4       1.0    0.9999
* Card 1.4
* GTP(1)    VFRAC(3)  GTP(2)    VFRAC(4)  GTP(3)    VFRAC(5)  GTP(4)
VFRAC(6)
air            .0001
*
```

Figure 2.10: COBRA-TF input deck

As can be seen, the linear power in KW/m is given as a fixed or constant value in group 1 of the AFLUX input card or it can be given in the form of a power versus time table. The original CTF

model used the Input Power Model (IPM). The PKM model is being implemented into CTF as a second option for reactor core power modeling so the user has the freedom to choose between a constant or tabulated value and the one calculated by PKM. The linear power given under AFLUX is used in PKM as initial power. Also, the code also contains the power shape change capability, which specifies core averaged relative axial and radial power distributions as a function of time. This capability is used in both IPM and PKM especially for the treatment of transient behavior. The reactor power calculated by PKM consists of two components—fission power and decay heat power—and these are calculated with point kinetics with reactivity feedback. All these options are selected in the CTF input deck either having user specified values or have the code pick the default value.

Listed below is a subroutine to read the required data:

```
c ... Point kinetics model      read(iin,*) i_power, error_r_mod, error_t_mod,  error_t_fuel,
error_b_conc
      if ((i_power.eq. 1)) then

c ...  Read number of delay groups, decay groups and history
      read(iin,*) ndgx, ndhx, nhist, i_pkm
           if (i_pkm. eq. 1) then
c ...  Read neutron generation time
      read(iin,*) tneut, react, rrpwmx
c ... Read precursor beta's and dcdn's (decay constant) for each delay group.
      read(iin,*) beta(i) (i=1, ndgx)
           read(iin,*) dcdn(i) (i=1, ndgx)
          endif
          if (nhis.eq.0) then
```

```
c ... Read precursor concentration
            read(iin,*) cdgn(i) (i=1, ndgx)
c ... Read decay group (dcdh) constants, energy fractions (edh), and concentrations (cdhn)      '
      read(iin,*) dcdh(j) (j=1, ndhx)
            read(iin,*) edh(j) (j=1, ndhx)
            read(iin,*) cdhn(j) (j=1, ndhx)
            endif


            read(iin,*) q235, q239, q238, qavg, r239pf


            read(iin,*) fisphi, rans, fp235, fp238
            endif
```

Development of a special subroutine called PKIN.F was done and in this subroutine an improved version of TRAC-PF1 point kinetics model is implemented including delayed precursor concentrations' equation solution based on both 1971 and 1979 ANS standard as options. Another subroutine called FEEDBACK_PKM.F supplemented by FEEDBACK.F is developed to calculate core average fuel temperature, moderator temperature, moderator density and boron concentration. The implementation of the above-described subroutines is as follows:

```
c-----------------------------------------------------------------------
c---- Select power option - IPM vs. PKM
c-----------------------------------------------------------------------
     if (i_power .eq. 1) then
       Th_timestep=delt
       timectf= timet
```

```
      if (calls_to_pkm.eq.0) then
        if (nstep.eq.1) write (*,*) "COBRA-TF Stand-alone Initialization"
      if (timet.gt.5.0) then
       write(*,*)"Start PKM Calculation"
        calls_to_pkm = calls_to_pkm + 1
        call feedback_PKM
        call pkin
       endif
      else
        calls_to_pkm = calls_to_pkm + 1
        call feedback_PKM
        call pkin
      end if
    elseif (i_power .eq. 0) then
     goto 50
    endif
 50 continue
```

The code structure inside COBRA-TF while implementing the subroutines is given as follows:

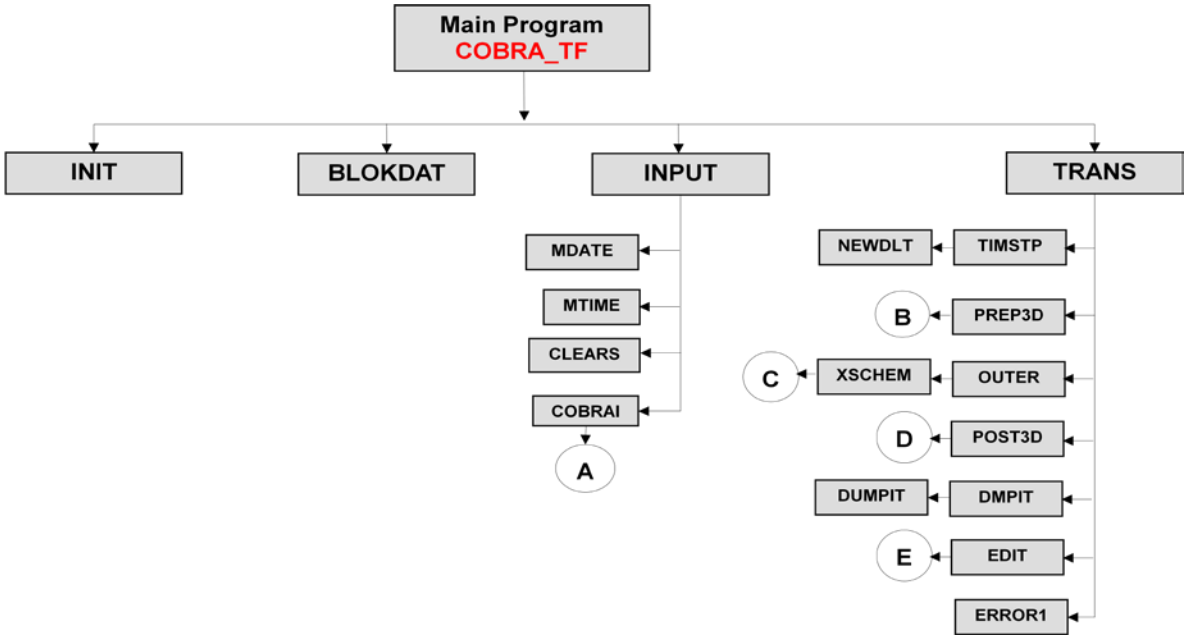

Figure 2.11: COBRA-TF Code Structure

The core feedback parameters, to be used in the PKM, are volume averaged based on the specifications in the CTF hydraulic and heat-structure nodalization while the calculated total power in the PKM is distributed according to the specified relative radial and axial power distributions specified for the given time step.

The selected time step by the time selection algorithm of CTF is used as initial guess for the time step for the PKM. This step is then automatically adjusted (sub-divided) by the PKM reactor-kinetic step selection procedure according to the specified error bounds of $\varepsilon_1 = 10^{-6}$ and = 10^{-4}. If $\varepsilon_1 < \varepsilon(\Delta t) < \varepsilon_2$ or $\varepsilon(\Delta t) \leq \varepsilon_1 < \varepsilon_2$ the model maintains the CTF step. If $\varepsilon_1 < \varepsilon_2 \leq \varepsilon(\Delta t)$ the model halves the value of Δt because the error in total power prediction is too large.

An overview of how the process works is given in Figure 2.12 as a calculation flow-chart.

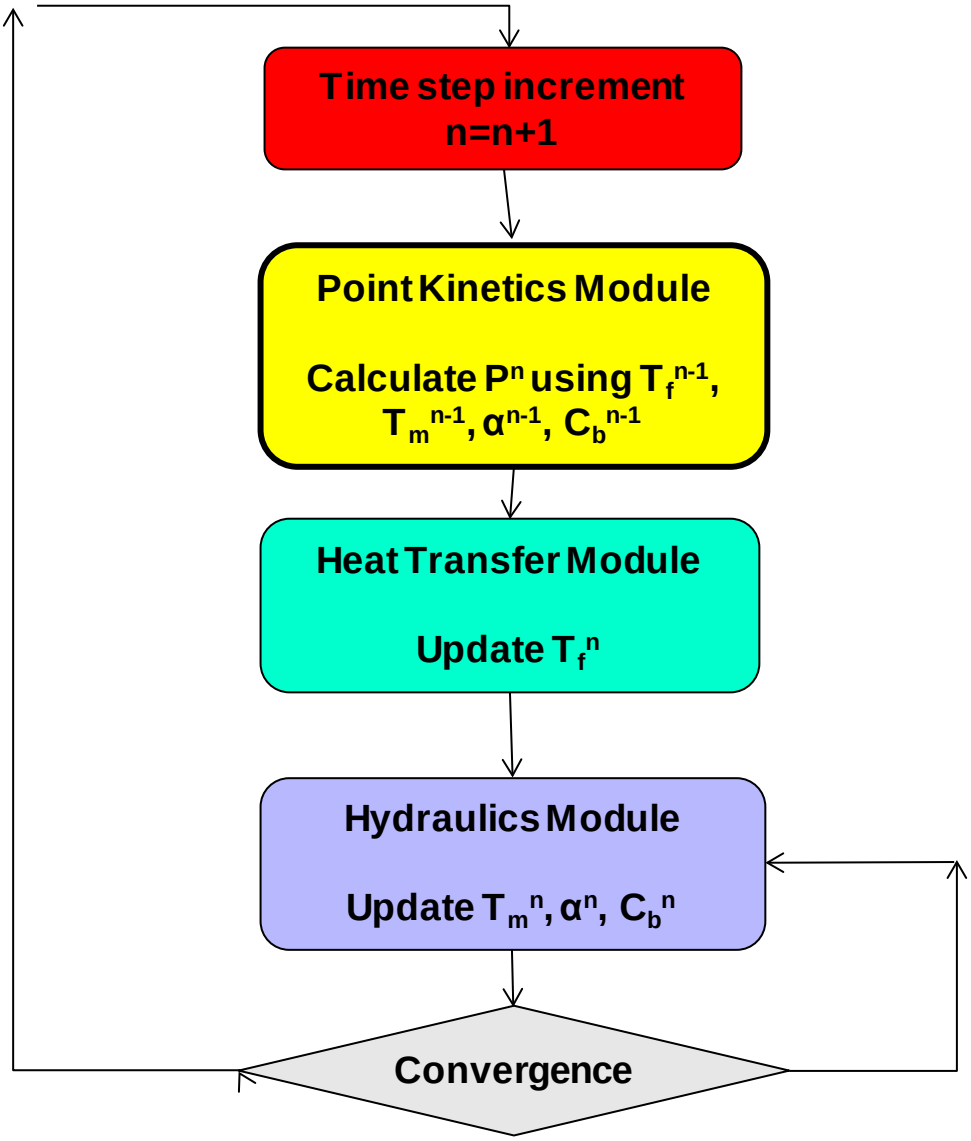

Figure 2.12: COBRA-TF Calculation Flow Chart

2.6. **Optional Coolant Density Reactivity Feedback**

The Point Kinetics Model in its current form takes into account coolant temperature feedback, coolant void fraction feedback, and boron concentration feedback. In order to render

the PKM more comprehensive, an optional coolant density feedback model is being implemented into the PK algorithm. This inclusion will make the model consistent with the current safety analysis practice. The Point Kinetics Model will be updated with this new option upon completion of the required work to achieve consistent results.

Chapter 3

Associated Models for Point Kinetics

3.1. Reactivity Feedback Model

The reactivity-feedback model is based on the assumption that only changes in the core averaged fuel temperature T_f, coolant temperature T_c, coolant vapor fraction α, and boron concentration (B_m boron concentration; or B_r ratio of boron mass to liquid-coolant mass) affect the neutron-multiplication reactivity of the reactor core. The code determines core-averaged values by applying fuel or coolant mass times power times volume-weighting factors to the temperatures and a power times volume-weighting factor to the coolant vapor fraction and boron concentration. These factors approximate the product of the local adjoint flux, neutron flux, and volume. The adjoint flux is commonly called the importance function as applied in perturbation theory. This weighting-factor product is from perturbation theory where it is used to weight spatially the change in reaction-rate cross sections to estimate their reactivity change. CTF approximates this weighting-factor product by the product of the affecting material mass, the fission-power density distribution raised to a user-specified power, and the mesh-cell volume. The averaging process over the core for the general reactivity-feedback parameter x is the following:

$$\bar{x} = \frac{\sum_i x_i \rho_i P_i^{\text{user-specified power}} V_i}{\sum_i \rho_i P_i^{\text{user-specified power}} V_i} \tag{3.1}$$

where P_i is the power, V_i is the volume of mesh cell i, and the assumptions are over all cells in the core. If x is T_f or T_c, ρ_i is the density of the fuel or coolant, respectively, in cell i; if x is α or the boron concentration (B_m or B_r), then ρ_i is set to 1.0 for all i.

Two options are available: 1) the fission reaction-rate cross-section distribution approximates the adjoint-flux distribution; and 2) the fission reaction-rate cross section is spatially constant, the neutron flux approximates the adjoint flux (Galerkin approximation). The choice of an appropriate option will depend on the fuel loading distribution in the reactor core. Keep in mind, however, that two levels of approximation are being made: first, that the affecting-material mass times the fission power density distribution raised to the user-specified power times the volume is a good approximation for the perturbation-theory product of the adjoint flux, neutron flux, and volume and, second, that the theoretical basis for the use of that weighting factor for reaction-rate cross sections can be applied to the reactivity-feedback parameters that we wish to average over the reactor core. The user has the option of replacing any of the three spatial shapes (fuel-rod radial, core horizontal plane, and core axial) in the fission-power density distribution with a different shape to weight the spatial averaging of reactivity-feedback parameters over the core. These spatial-shape replacements are specified through input as shown in Table 3.1.

Table 3.1: Reactivity Coefficient Forms

Form Number	Reactivity-Coefficient Form
0	$\frac{\partial k}{\partial x}$
1	$\frac{1}{k}\frac{\partial k}{\partial x} \cong \frac{\partial R}{\partial x}$
2	$x\frac{\partial k}{\partial x}$
3	$\frac{x}{k}\frac{\partial k}{\partial x} \cong x\frac{\partial R}{\partial x}$

The user defines a reactivity coefficient for each of the reactivity-feedback independent-variable parameters $x = T_f, T_c, \alpha, B_m(B_r)$ by choosing one of the reactivity-coefficient forms in Table 3.1. Each reactivity coefficient is defined through input by a table of reactivity-coefficient values that are dependent on all four reactivity feedback parameters. Each parameter has one or more values specified in defining the table. Defining one value for a parameter corresponds to the reactivity coefficient having no dependence on that reactivity-feedback parameter. Defining two or more values for a parameter gives the reactivity-coefficient dependence on one or more of the reactivity-feedback parameters. Linear interpolation is used to evaluate the reactivity coefficient for values of each parameter between its table values. Multidimensional linear-surface interpolation is performed when the reactivity coefficient is dependent on two or more of the reactivity-feedback parameters.

Boron in the derivative and boron dependence of a reactivity coefficient are defined in terms of the boron concentration in the coolant channel, B_m, in units of $kg \cdot m^{-3}$, a macroscopic density, or the boron mass to liquid-coolant mass ratio, B_r, in units of parts boron per million parts liquid coolant (ppm). The boron reactivity coefficient is based on the change in the amount of boron

without changing T_c and α when the coefficient is defined as a derivative of B_r. The boron dependence of a reactivity coefficient is important when the reactivity coefficient is sensitive to a shift in the neutron-energy spectrum. Through the proportionality to the amount of neutron capture in boron (B_m or B_r) dependence characterizes this shift.

All forms of boron in the reactor core are included in the evaluation of B_m or B_r for boron dependence of the reactivity coefficients: dissolved boric acid in the liquid coolant, boric acid plated on core structure, borosilicate glass in burnable-poison pins, and boron oxide in control-rod-cluster pins. The first two forms are based in boron convection into the reactor core throughout the modeled system and on plate-out of boric acid in the core as a result of coolant dry-out. The code user defines the burnable poison pin and control-rod-cluster pin forms through input with what would be their equivalent boron concentrations in the coolant channel. A first-order polynomial function of the core-averaged coolant temperature T_c, the coefficients of which are input parameters, describes the equivalent burnable-poison pin boron concentration:

$$BPP = BPP0 + BPP1 \cdot T_c \tag{3.2}$$

where BPP is an equivalent burnable poison pin boron concentration; $BPP0$ and $BPP1$ are polynomial coefficients for BPP.

Whereas the physical amount of boron in the burnable-poison pins is constant during the transient, its equivalent concentration for neutron capture increases with coolant temperature because of reduced spatial self-shielding of boron to neutrons in those pins. The amount of control-rod-cluster pin boron in the reactor core depends on the amount of control-rod insertion

into the core. Its equivalent concentration is input as a first order polynomial function of programmed reactivity ρ_{prog}:

$$BCR = BCR0 + BCR1 \cdot \rho_{prog} \tag{3.3}$$

where BCR is a control-rod cluster pin boron concentration; $BCR0$ and $BCR1$ are polynomial coefficients for BCR.

Both BPP and BCR in the previous two equations have units of macroscopic boron density ($kg \cdot m^{-3}$) in the coolant-channel volume. Programmed reactivity is assumed to be proportional to the amount of effective boron-mass change caused by control-rod movement in the core. Initially, when the reactor core is at steady-state conditions, $BCR = BCR0$.

Only the dissolved boric acid and that plated on solid structures in the reactor core are included in the evaluation of B_m or B_r for the boron reactivity-coefficient derivative parameter because the amount of boron in burnable-poison pins is constant during the transient that is being evaluated. Control-rod movement changes the amount of boron, but the reactivity change associated with this change is accounted for in the user-defined programmed reactivity ρ_{prog}.

After all four reactivity coefficients are evaluated by linear interpolation in their four dimensionally dependent tables; feedback reactivity is evaluated in terms of its change in the neutron multiplication constant over the last time step, Δt^{n-1},

$$k^{n-1}k^{n}\rho_{fdbk}^{n} = (\Delta k_{fdbk})^{n} = \sum_{i=1}^{4}\frac{1}{2}\left[\left(\frac{\partial k}{\partial x_i}\right)^{n} + \left(\frac{\partial k}{\partial x_i}\right)^{n-1}\right]\times\left[x_i^{n} - x_i^{n-1}\right] \qquad (3.4)$$

Where x_1 is T_f; x_2 is T_c; x_3 is α; and x_4 is B_m or B_r. The subscripts n and $n-1$ indicate that the parameter is evaluated at the beginning of the current time step or the beginning of the previous time step, respectively. The code obtains the value of $(\partial k/\partial x_i)^n$ in Eq. (3.2) by multiplying the reactivity coefficients defined by reactivity coefficient forms 1, 2, and 3 (Table 3.1) by k^{n-1}, $1/x_i^n$, and k^{n-1}/x_i^n, respectively. Using k^{n-1} rather than k^n (which is not yet known) to convert reactivity-coefficient forms 1 and 3 to $\partial k/\partial x_i$ is an approximation. The values of $\partial k/\partial x_i$ and x_i at time t^n (the start of the present time step) are evaluated as described above; their values at time t^{n-1} are those saved from their evaluation at the start of the previous time step.

The change in x_i over the last time step (with the other x_i parameters held constant in defining the reactivity coefficient) is defined in the last factor of Eq. (3.2). The term B_m or B_r includes only dissolved boric acid in the liquid coolant and boric acid plated on the core structure. There is no change in the B_r equivalent boron concentration in burnable-poison pins when T_f, T_c, and α are not considered to vary.

To determine reactivity feedback during the current time step $\Delta t^n = t^{n+1} - t^n$, the code would need to know the end-of-time-step values of T_f, T_c, α, B_m or B_r. To evaluate that reactor-core state requires knowing the current core state and the total energy-generation rate, $P_{eff}(t)$ over

the time step Δt^n. The neutronic reactivity $\rho(t)$ defined by the reactivity-feedback contribution, $\rho_{fdbk}(t)$, must be known to determine the value of $P_{eff}(t)$ from the solution of the point-reactor kinetics equations. Thus, the evaluation of reactivity feedback during the current time step requires first knowing what it is. The code handles this difficulty of needing to know the reactivity feedback in order to evaluate it by assuming its Δk feedback-reactivity rate is zero. The Δk programmed-reactivity rate is assumed to be the same as in the previous time step Δt^{n-1},

$$(k_{prog}^{est})^n = \Delta t^n \left\{ \frac{\left[k_{prog}^{n} - k_{prog}^{n-1} \right]}{\Delta t^{n-1}} \right\} \tag{3.5}$$

We estimate $(k_{prog})^n$ because the independent variable needed to evaluate its tabular definition generally is a function of the reactor-core state. We have tested a similar approximation for $(\Delta k_{fdbk}^{est})^n$ in the code, but near steady state it caused the sign of $(\Delta k_{fdbk}^{est})^n$ to change each time step. It is better to assume that $(\Delta k_{fdbk}^{est})^n$ is zero than to estimate its value based on its value in the previous time step. Thus, estimates for the reactor multiplication constant and neutronic reactivity for the reactor state at the end of the present time step are

$$(k^{est})^n = k^n + (\Delta k_{prog}^{est})^n \tag{3.6}$$

and

$$(\rho^{est})^n = \frac{\left[(k^{est})^n - 1\right]}{(k^{est})^n} \tag{3.7}$$

After each time-step solution, the code compares the actual programmed reactivity with its estimated value. The code corrects any discrepancy by applying

$$(\Delta k^{cor})^{n+1} = \left[(\Delta k_{prog})^n - (\Delta k_{prog}^{est})^n\right] \cdot \min\left(\frac{\Delta t^n}{\Delta t^{n+1}},\ 2\right) \tag{3.8}$$

during the next time step, Δt^{n+1}. To prevent a time-step reactivity correction from becoming very large by applying it over a time step $\Delta t^{n+1} << \Delta t^n$, the code constrains $(\Delta k^{cor})^{n+1}$ in Eq. (3.5) to be no more than twice the Δt^n time-step value. Including this correction and the actual reactivity-feedback change in Δk during the previous time step in the reactor multiplication constant end-of-time-step estimate gives Eq. (3.4) as a modified form:

$$(k^{est}) = k^n + (k_{prog}^{est})^n + (\Delta k^{cor})^n + (\Delta k_{fdbk})^{n-1} \tag{3.9}$$

In this procedure, we assume that $(\Delta k^{cor})^n$ and $(\Delta k_{fdbk})^{n-1}$ are small enough not to require an iterative evaluation of the present time-step solution and that $(\Delta k^{cor})^n$ is large enough to be included in the next time-step solution rather than be neglected. The code applies a similar estimate and correction procedure to the end-of-time-step power for the case in which the total core power is specified directly by a table.

3.2. **Decay Heat Model**

Decay Heat Model works in conjunction with the Feedback Model to provide all the necessary parameters to compute reactor power while employing PKM [10]. The code applies the thermal-power solution to the decay-heat equations

$$\frac{dH_j}{dt} = -\lambda_j^H H_j + E_j P \qquad for \quad j = 1, 2.....J \tag{3.10}$$

where

P is the solution of Eqs. (1) and (2)

H_j is the energy of the decay-heat group *j* (W.sec);

λ_j^H is the decay constant for decay-heat group *j* (s^{-1});

E_j is the effective energy fraction of decay-heat group *j*;

and J is the number of decay-heat groups.

After solving each j^{th} equation represented by Eq. (3.10) for the decay-heat group concentration H_j, the code calculates the total thermal power generated in the reactor fuel at time t from prompt fission, precursor decay, delayed fission, and the external source:

$$P_{eff} = P + \sum_{j=1}^{J} \lambda_j^H H_j \tag{3.11}$$

The code requires the number of delayed-neutron groups, I; the delayed-neutron parameters, λ_i and β_i; the number of decay-heat groups, J; the decay-heat parameters, λ_j^H and E_j; and either the total thermal-power history $P(-t)$ for $t \le 0$ for the initial delayed-neutron precursor concentrations $C_i(0)$ and decay-heat concentrations, $H_j(0)$.

By default the code internally sets I to 6 and defines λ_i and β_i with the values in Table 3.2. If $I \leq 0$ and no total thermal-power history is an input, the code assumes that an initial steady-state condition exists to initialize $C_i(0)$ internally. The code sets the time derivative in Eq. (3.12) to zero and calculates the initial values of the C_i from the following:

$$C_i(0) = \frac{\beta_i}{\lambda_i \Lambda} P(0) \qquad for\ i = 1, 2....I \tag{3.12}$$

where $P(0)$ is the initial power specified through input.

Table 3.2: Delayed-Neutron Constants

Group i	Decay Constant $\lambda_i(s^{-1})$	Neutron Fraction β_i
1	3.87	0.000169
2	1.40	0.000832
3	0.311	0.00264
4	0.115	0.00122
5	0.0317	0.00138
6	0.0127	0.000247

3.2.1. **Default Data for the Decay-Heat Groups**

If $J \leq 0$ is input, TRACE internally sets J to 69 and defines λ_j^H and ED_j with the values in Table 1.2. These decay heat parameters are obtained from the 1979 ANS decay-heat standard. If the default decay-heat groups are used or if the user inputs 69 or 71 decay-heat groups, the fraction of fission power due to each of the three isotopes, ^{235}U, ^{239}Pu, and ^{238}U, must be input. In addition, in order to convert the MeVs of decay energy per fission to a fraction of the total fission energy, the MeVs per fission for each of the three isotopes, ^{235}U, ^{239}Pu, and ^{238}U, must be input. ED_j in Table 3.3 is converted to E_j with the following equation:

$$E_j = ED_j / (\lambda_j^H Q_k) \tag{3.13}$$

where Q_k is the MeV/fission for isotope k. If $J \leq 0$ and no total thermal-power history is input, the code assumes that an initial steady-state condition exists to initialize $H_j(0)$ internally. The code sets the time derivative in Eq. (3) to zero and calculates the initial values of H_j from the following:

$$H_j(0) = \frac{E_j}{\lambda_j^H} P(0) \qquad for\ j = 1, 2....J \tag{3.14}$$

where $P(0)$ is the initial steady-state power specified through input. This is equivalent to assuming an infinite reactor operating power at the initial power.

Table 3.3: 1979 DH Const for U-235,Pu-239 Thermal Fission, and U-238 Fast Fission

Group No	Decay Constant λ_j^H (s^{-1})	MeV of decay energy per fission per second ED_j (MeV/fission-s)
for isotope U^{235}		
1	22.138	6.5057×10^{-01}
2	0.51587	5.1264×10^{-01}
3	0.19594	2.4384×10^{-01}
4	0.10314	1.3850×10^{-01}
5	0.033656	5.5440×10^{-02}
6	0.011681	2.2225×10^{-02}
7	0.0035870	3.3088×10^{-03}
8	0.0013930	9.3015×10^{-04}
9	0.00062630	8.0943×10^{-04}
10	0.00018906	1.9567×10^{-04}
11	5.4988×10^{-05}	3.2535×10^{-05}
12	2.0958×10^{-05}	7.5595×10^{-06}
13	1.0010×10^{-05}	2.5232×10^{-06}
14	2.5438×10^{-06}	4.9948×10^{-07}
15	6.6361×10^{-07}	1.8531×10^{-07}
16	1.2290×10^{-07}	2.6608×10^{-08}
17	2.7213×10^{-08}	2.2398×10^{-09}
18	4.3714×10^{-09}	8.1641×10^{-12}
19	7.5780×10^{-10}	8.7797×10^{-11}
20	2.4786×10^{-10}	2.5131×10^{-14}
21	2.2384×10^{-13}	3.2176×10^{-16}
22	2.4600×10^{-14}	4.5038×10^{-17}
23	1.5699×10^{-14}	7.4791×10^{-17}
for isotope Pu^{239}		
24	10.02	2.083×10^{-01}
25	0.6433	3.853×10^{-01}
26	0.2186	2.213×10^{-01}
27	0.1004	9.460×10^{-02}
28	0.03728	3.531×10^{-02}
29	0.01435	2.292×10^{-02}

Table 3.3 (contd.): 1979 DH Const for U-235,Pu-239 Thermal Fission, and U-238 Fast Fission

Group No	Decay Constant $\lambda_j^H (s^{-1})$	MeV of decay energy per fission per second ED_j (MeV/fission-s)
30	0.004549	3.946×10^{-03}
31	0.001328	1.317×10^{-03}
32	0.0005356	7.052×10^{-04}
33	0.0001730	1.432×10^{-04}
34	4.881×10^{-05}	1.765×10^{-05}
35	2.006×10^{-05}	7.347×10^{-06}
36	8.319×10^{-06}	1.747×10^{-06}
37	2.358×10^{-06}	5.481×10^{-07}
38	6.450×10^{-07}	1.671×10^{-07}
39	1.278×10^{-07}	2.112×10^{-08}
40	2.466×10^{-08}	2.996×10^{-09}
41	9.378×10^{-09}	5.107×10^{-11}
42	7.450×10^{-10}	5.703×10^{-11}
43	2.426×10^{-10}	4.138×10^{-14}
44	2.210×10^{-13}	1.088×10^{-15}
45	2.640×10^{-14}	2.454×10^{-17}
46	1.380×10^{-14}	7.557×10^{-17}
for isotope U^{238}		
47	3.2881	1.2311
48	0.93805	1.1486
49	0.37073	7.0701×10^{-01}
50	0.11118	2.5209×10^{-01}
51	0.036143	7.1870×10^{-02}
52	0.013272	2.8291×10^{-02}
53	0.0050133	6.8382×10^{-03}
54	0.0013655	1.2322×10^{-03}
55	0.00055158	6.8409×10^{-04}
56	0.00017873	1.6975×10^{-04}
57	4.9032×10^{-05}	2.4182×10^{-05}
58	1.7058×10^{-05}	6.6356×10^{-06}
59	7.0465×10^{-06}	1.0075×10^{-06}

Table 3.3 (contd.): 1979 DH Const for U-235,Pu-239 Thermal Fission, and U-238 Fast Fission

Group No	Decay Constant λ_j^H (s^{-1})	MeV of decay energy per fission per second ED_j (MeV / fission-s)
60	2.3190×10^{-06}	4.9894×10^{-07}
61	6.4480×10^{-07}	1.6352×10^{-07}
62	1.2649×10^{-07}	2.3355×10^{-08}
63	2.5548×10^{-08}	2.8094×10^{-09}
64	8.4782×10^{-09}	3.6236×10^{-11}
65	7.5130×10^{-10}	6.4577×10^{-11}
66	2.4188×10^{-10}	4.4963×10^{-14}
67	2.2739×10^{-13}	3.6654×10^{-16}
68	9.0536×10^{-14}	5.6293×10^{-17}
69	5.6293×10^{-15}	7.1602×10^{-17}

In TRACE there is an option to use the 1994 ANS decay heat standard [5], which uses higher number of decay-heat groups – 94 groups. Recently the ANS 2005 [11] decay heat standard was released and might also be utilized if desired. The Decay Heat Model in CTF, however, has been programmed with the two older ANS heat standards. The ANS/ANSI 5.1-1971 decay heat standard is listed in Table 3.4. This standard has been implemented for legacy purposes.

Table 3.4: Decay-Heat Constants ANS/ANSI 5.1-1971 for U-235 Thermal Fission

Group No.		
1	6.587E+00	2.658E+00
2	1.490E-01	4.619E-01
3	2.730E-01	6.069E-02
4	2.173E-02	5.593E-03
5	1.961E-03	6.872E-04
6	1.025E-04	6.734E-05
7	4.923E-06	6.413E-06
8	2.679E-07	6.155E-07
9	1.452E-08	8.288E-08
10	1.893E-09	1.923E-08
11	1.633E-10	1.214E-09

3.2.2. **Prompt-Fission Power History**

If the total thermal-power history $P(-t)$ is input, TRACE evaluates $C_i(0)$ and $H_j(0)$ from Eqs. (2.2) and (3.10). The $P(-t)$ consists of tabular data (t_l, P_l) pairs for $l = 1,2,3\ldots L$, where $-t_{l+1} \le -t_l \le 0$. The code assumes that the total thermal power varies linearly between data pairs in the power history table, that is,

$$P(t) = a_l + b_l t \quad \text{, over the time interval } -t_{l+1} \le t \le -t_l \tag{3.15}$$

Substituting Eq. (3.15) into Eq. (2.2) and integrating the resulting equation analytically from $t = -t_L$ [where $C_i(-t_L) = 0$ is assumed] to $t = 0$ gives

$$C_i(0) = \frac{\beta_i}{\lambda_i \Lambda}\left\{\left(a_l - \frac{b_l}{\lambda_i}\right)(1 - exp[-\lambda_i(t_{l+1} - t_l)]) + b_l(t_{l+1} exp[-\lambda_i(t_{l+1} t_l)] - t_l) - exp[-\lambda_i t_l]\right\} \tag{3.16}$$

$$H_j(0) = \frac{E_j}{\lambda_j^H \Lambda}\left\{\left(a_l - \frac{b_l}{\lambda_j^H}\right)\left(1 - exp\left[-\lambda_j^H(t_2 - t_l)\right]\right) + b_l\left(t_2\, exp\left[-\lambda_j^H(t_2 - t_l)\right] - t_l\right) - exp\left[-\lambda_j^H t_l\right]\right\} \tag{3.17}$$

The decay heat parameters are obtained from the 1979 ANS decay heat standard. If the default decay heat groups are used, the fraction of fission power due to each of the three isotopes, ^{235}U, ^{239}Pu, and ^{238}U must be input. Both the 1979 ANS heat standard and the 1971 heat standard have been implemented in the code. The major difference is that there are 23 pair variables in the newer heat standard whereas the 1971 heat standard only has 11 such pair variables.

3.3. **Optional Constant Actinide Decay Heat**

In its default form, PKM computes the actinide decay heat when the user selects IANS=2 (71-group decay heat), which is an appropriate option for best-estimate calculation. But, when PKM is used as a conservative model, the capability to keep the actinide decay heat constant throughout is needed. This option has been programmed into the decay heat algorithm. The user, based on an input flag, is given an option to have PKM compute the actinide decay heat or keep it constant throughout the transient.

Chapter 4

Point Kinetics Code-to-Code Verification Using TRACE

4.1. Objective

This chapter describes and analyzes the code to code verification needed for the newly implemented Point Kinetics Model in CTF. For comparison purposes, TRACE is used and the test problem is run for both the steady state and the transient cases

4.2. Introduction

In order to verify that the code is performing as intended, a simplified code-to-code verification using TRACE has been conducted. TRACE has a built-in PKM model with various options to tailor to the user. This capability gives an ideal opportunity for the user to compare an identical problem by running it on the two codes. For our case, an identical problem will be run on CTF and TRACE, first with constant power and then with PKM activated in both codes.

The sample problem used is PWR Sample Case 1 [12]. The sample case consists of a ¼ core of pressurized water reactor, and in CTF it has been modeled using 9 channels where channel 3 is the hottest channel and channel 9 is the largest channel. The data from these two channels will be analyzed for comparison purposes. The data is given in two sets; a base case without PKM and a secondary case with PKM activated. For comparison, pressure drop within the channel, the liquid temperature, liquid density, liquid velocity, and surface temperature are plotted for comparison in both channels. Additional case is run with the 50 % loss of flow transient.

In order to facilitate the generation of TRACE input, SNAP was used. The Symbolic Nuclear Analysis Package (SNAP) consists of a suite of integrated applications designed to simplify the process of performing thermal-hydraulic analysis using codes such as TRACE. Along with SNAP, JEdit was used which is a type of text editor integrated into SNAP. It makes it a lot easier to transfer the SNAP models into ASCII text files for analysis.

4.3. **Test Case Description**

The test case used for comparison is explained below along with the pertinent parameters used in the analysis in both TRACE and CTF. The parameters used in TRACE are identical to what is being used in CTF. In the original document with the CTF input file, cross-flow was enabled within the channels. For this steady-state test case, cross-flow has been turned off in CTF, and TRACE deck is created without any connections among the nine channels to mimic the CTF configuration.

For the verification of the different models implemented in CTF, the following case, given in Table 4.1 has been selected:

Table 4.1: Validation and Verification Test Case

Name of the case	**Case1**
Reference	COBRA-TF V&V matrix
Case Description	PWR core containing 121 14×14 fuel assemblies, each with 179 fuel rods, 16 guide tubes, and 1 instrumentation tube. Fuel assembly characteristics are given in **Table 4.2**
COBRA-TF Model	Quarter core model The hot assembly is located at the core center (the asymmetric location of the instrumentation tube can be neglected)
Time-dependence	• Stationary • 50 % Loss of Flow Transient
Pressure	156.3 bars
Radial power distribution	Non-uniform
Axial power distribution	Chopped cosine with a peak value of 1.55
Linear heat flux	22.21 kW/m
Total inlet mass flow rate	1645.35 kg/s
Inlet enthalpy	1251.255 kJ/kg
Models in COBRA-TF to be verified	• Code performance at steady state and flow transient conditions

Table 4.2: Full Fuel Assembly Specifications

Parameter	
Fuel rod outside diameter (cm)	1.077
Guide tube outside diameter (cm)	1.369
Instrumentation tube outside diameter (cm)	1.077
Fuel rod pitch (cm)	1.412
Fuel assembly pitch (cm)	19.82
Fuel assembly dimensions	14×14
Gap between fuel assemblies (cm)	0.051
Number of fuel rods	179
Number of guide tubes	16
Number of instrumentation tubes	1
Fuel active length (cm)	241.3

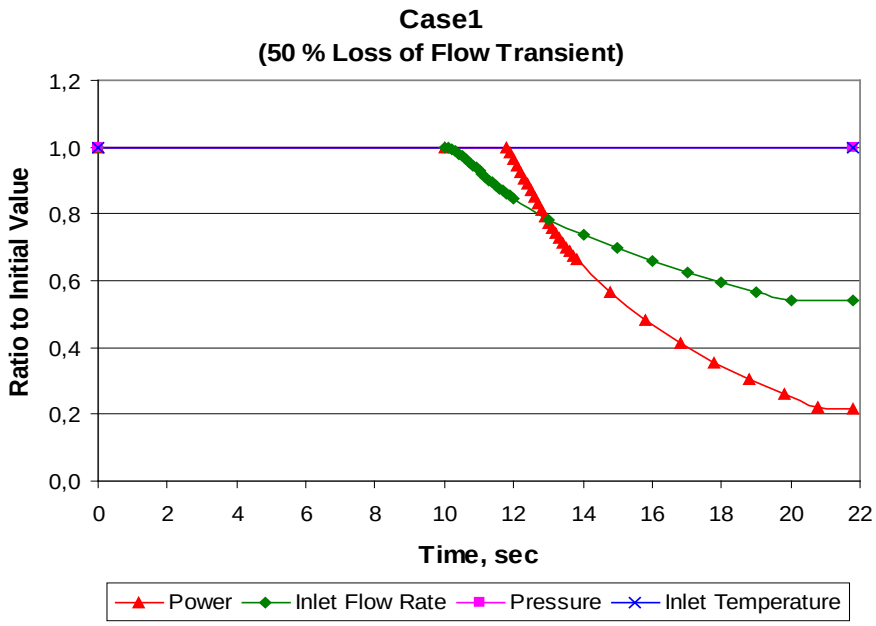

Figure 4.1: 50 percent loss of flow transient

A sample case has been developed to test the implementation of the PKM model and then compare the results obtained with or without PKM in CTF with those from TRACE. The test case is calculated with CTF as well as with TRACE. This helps validate the results among the different models being compared.

4.3.1. **Test Case COBRA-TF Model**

The simulated Pressurized Water Reactor (PWR) core contains 121 14×14 fuel assemblies, each with 179 fuel rods, 16 guide tubes, and 1 instrumentation tube. A quarter of the core was modelled with CTF.

The model layout is shown in Figure 4.2. The so-called macrocell is comprised of subchannels 1 through 7 is shown in Figure 4.3. The subchannels surrounding the hottest fuel rod have been modelled exactly as subchannels 1 through 4 as given in the problem specification. Surrounding this area are lumped channels 5, 6, and 7. The remaining parts of the four fuel assemblies are modelled as channel 8 and the rest of the quarter core is modelled as channel 9. The fuel assembly characteristics are given in Table 4.2.

The axial power distribution is modelled as a chopped cosine with a peak value of 1.55. The radial power distribution is modelled on a subchannel basis, where for each subchannel a pseudo fuel rod is modelled, which represents the sum of the fractional powers of the different rods in the subchannel. For the subchannels 1 through 4, the modelled rods have the power of the hottest fuel rod in the assembly. A summary of the power factors for each subchannel is given in Table 4.3.

Table 4.3: Subchannel Power Factors and Rod Multipliers

Subchannel	**Rod Multiplier**	**Power Factor**	**Rod Fraction**	**Total Fuel Rods**
1	1	1.55769	0.7440	0.744
2	1	1.55769	0.7450	0.745
3	1	1.55769	0.9890	0.989
4	1	1.55769	0.7470	0.747
5	12	1.52627	0.8333	10.000
6	33	1.50229	0.9621	31.750
7	120	1.51086	0.8979	107.750
8	272	1.51087	0.9191	250.000
9	5488	0.95895	0.9133	5012.000

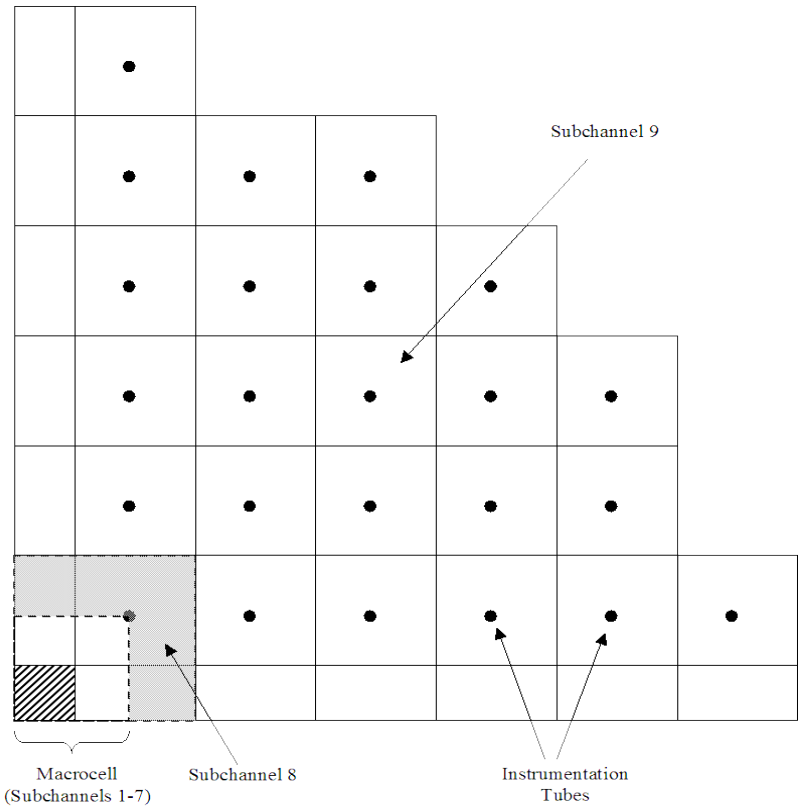

Figure 4.2: Layout of the ¼ core model

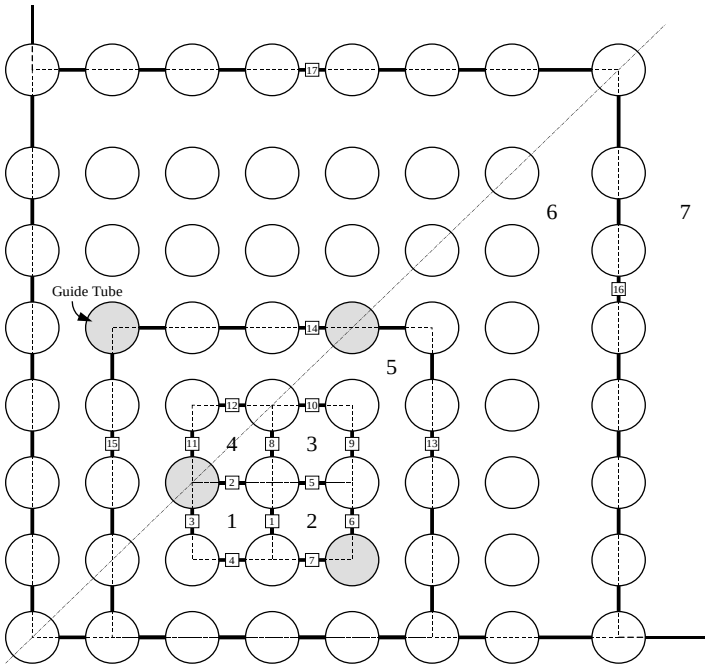

Figure 4.3: Subchannel layout of the macrocell

The core model includes five spacer grids in the heated length. Of these five spacer grids, four are mixing spacers and one is a structural spacer. The axial locations of each spacer grid and its type are listed in Table 4.4. The grid loss coefficients for each channel and for each type of spacer are listed in Table 4.5. Table 4.6 summarizes the CTF modelling options used in the calculation.

Table 4.4: Spacer grid locations and types

Axial location (cm)	**Type**
2.1	structural
21.8	mixing
41.5	mixing
61.1	mixing
80.8	mixing

Table 4.5: Local spacer loss coefficients at Re = 500,000

Loss Coefficients		
Subchannel	Mixing Spacer	Structural Spacer
1	0.91	1.13
2	0.91	1.13
3	0.68	1.04
4	0.91	1.13
5	0.81	1.10
6	0.85	0.94
7	0.79	1.04
8	0.79	1.02
9	0.79	1.02

Table 4.6: Parameters of the COBRA-TF model for PWR Sample Case 1

Units in input file	SI units
Direct heating of the coolant (by neutron moderation)	none
Initial pressure of the fluid	equal to the outlet pressure
Initial enthalpy of the fluid	equal to the inlet enthalpy
Heat loss to ambient	none
Initial rod temperature	280 °C
Critical Heat Flux Correlation	W3 with a constant Tong "F" factor equal to 1.1
Radiation	none
Fuel pellet model (inside the rods)	none
Forcing functions for rod power	Constant in the first 10 s, followed by a reduction of 21.6 % of the nominal power in 11.8 s
Forcing functions for the inlet flow rate	Constant in the first 10 s, followed by a reduction of 54.1 % of the nominal flow rate in 11.8 s
Mixing models	• Standard turbulent mixing model according to user specified mixing coefficient β: $\beta = 0.051$
Boundary condition type	Inlet: mass flow rate and enthalpy Outlet: pressure and enthalpy
Simulation time	21.8 s in total: 10 s to reach stationary conditions followed by 11.8 s transient
No. of test points in the calculation	1

4.3.2. **COBRA-TF steady state calculations**

CTF must be run as a “null” transient to reach “quasi” steady-state condition. The results shown in this analysis represent 10 seconds CTF run at constant power and flow rate. The axial distributions of liquid enthalpy and mass flow rate for the hottest subchannel (subchannel # 3 in Figure 4.3) are shown in Figure 4.4 and Figure 4.5. Figure 4.6 and Figure 4.7 depict, respectively, the pressure drop along the subchannel length and the heat flux axial distribution. The W3 critical heat flux correlation with a constant Tong “F” factor equal to 1.1 is used to calculate the departure from nucleate boiling ration (DNBR). DNBR predictions are given Figure 4.8.

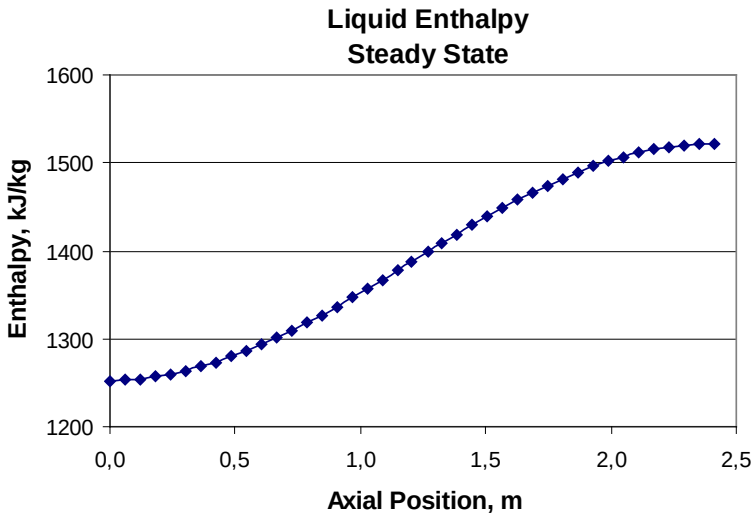

Figure 4.4: Subchannel # 3 Liquid Enthalpy at Steady-State

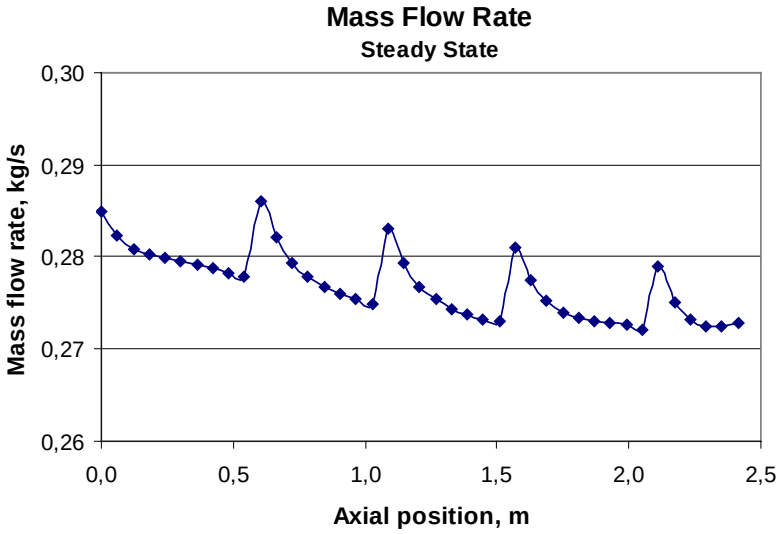

Figure 4.5: Subchannel # 3 Liquid Mass Flow Rate at Steady-State

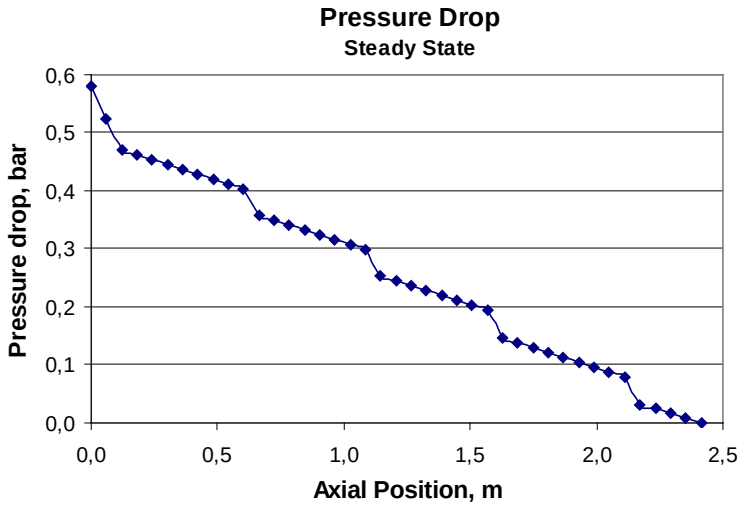

Figure 4.6: Subchannel # 3 Pressure Drop at Steady-State

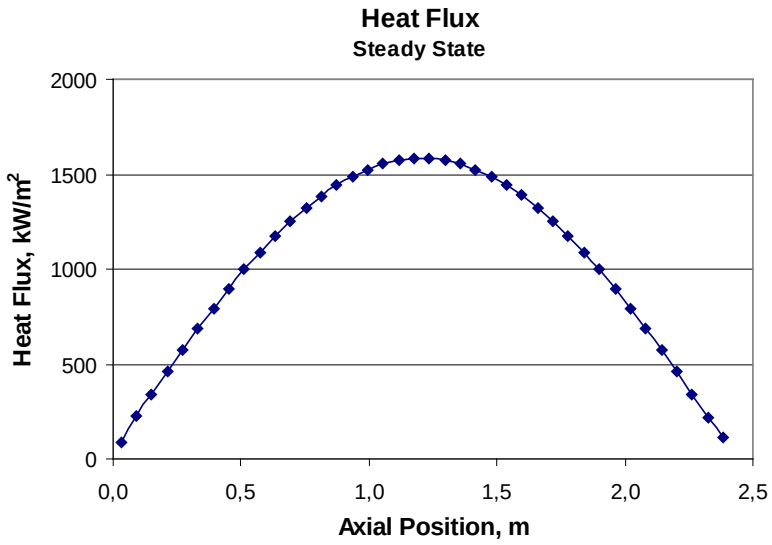

Figure 4.7: Rod # 3 Heat Flux at Steady-State

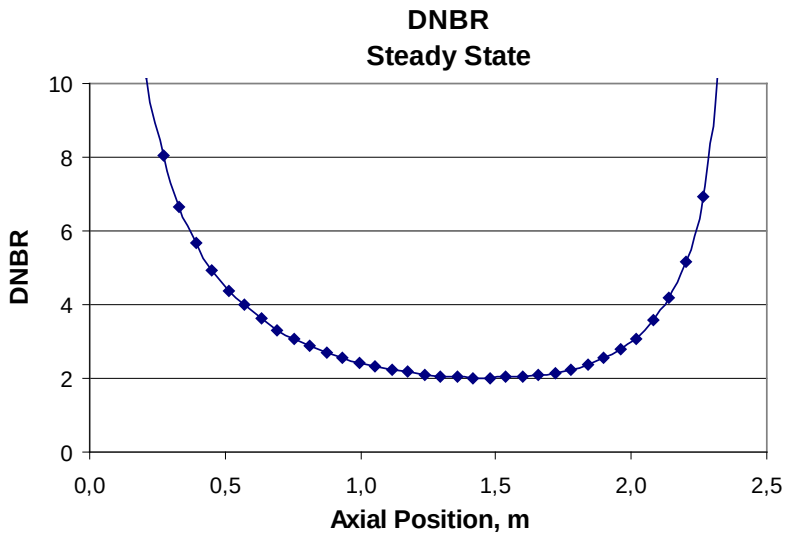

Figure 4.8: Subchannel # 3 DNBR at Steady-State

4.3.3. **COBRA-TF transient calculations**

A 50% loss of flow transient is modeled with CTF. The core inlet flow rate is reduced with 50 % for 10 seconds. At approximately 1.8 seconds after the transient starts, the power drop is also initialized and the core power level is reduced with 80% for 10 seconds. The time evaluation of the minimum DNBR ratio can be seen in Figure 4.9. The maximum heat flux to flow ratio is predicted at 1.8 seconds into the transient and as a result the minimum DNBR ratio is reached at that time. Figure 4.10 and Figure 4.11 show the hottest rod heat flux profile and DNBR at that time.

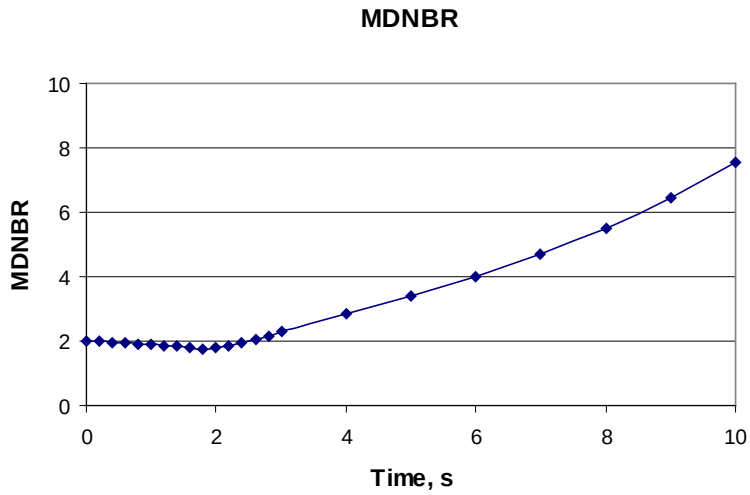

Figure 4.9: Subchannel # 3 MDNBR, Time Evaluation

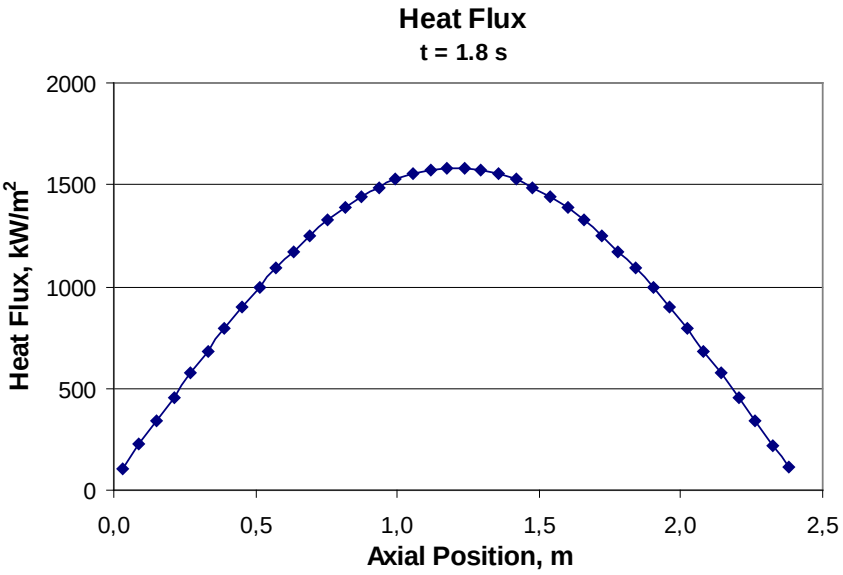

Figure 4.10: Rod # 3 Heat flux axial profile at the most limited transient time, t_{trans}=1.8 s

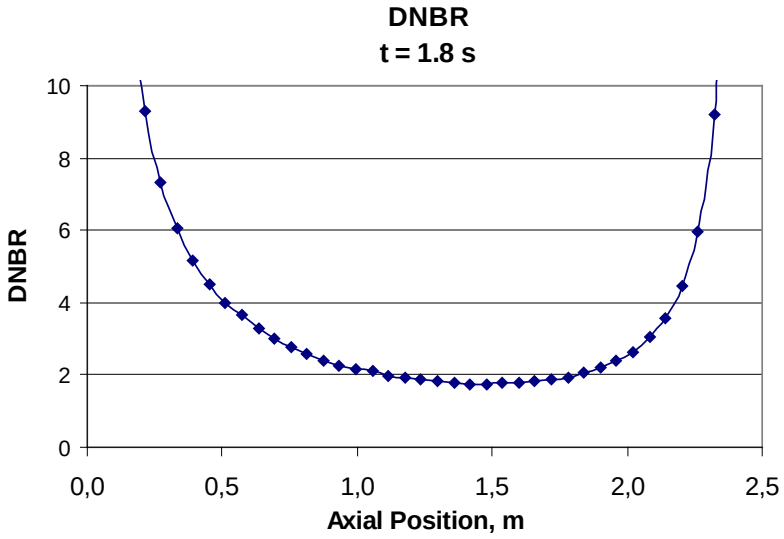

Figure 4.11: Subchannel #3 DNBR axial evaluation at the most limited transient time, t_{trans}=1.8 s

Sample input decks have been created in both CTF and TRACE to compare the implementation of PKM in the code.

4.4. **Results and Analysis**

Before we proceed, it would be prudent to explain the approach used in TRACE in order to mimic the test case parameters. For our analysis, TRACE version 5.0 with patch 1.0 is used. The temporal method used is semi-implicit with the transient calculation option set to "Steady State" and "Transient". SNAP with a TRACE plugin is employed to generate the test case input model. Additionally, the text editor, JEdit, is used to open up ASCII file generated by SNAP.

After internal collaboration, it was determined that the best approach to model the 9 channels was by using the pipe components in SNAP and then adding heat structures to those pipe components. These heat structures would then be provided power using the power component. One has the option to provide the type of power option in the power component, be it constant power, or point kinetics with constant reactivity. For the sake of avoiding complexity, cross-flow option is turned off in CTF, and, similarly, the pipes used to model the channels are not connected.

A set of 9 pipes are modeled, one for each of the 9 channels. The pipe numbers correspond to the channel numbers used in CTF. The heat structures attached to the pipes provide heat to the coolant flowing inside the pipes. The point kinetics parameters used in TRACE are identical to the ones being used in CTF. Figure 4.12 through Figure 4.44 show the comparison between the two codes.

4.4.1. **Steady State Case**

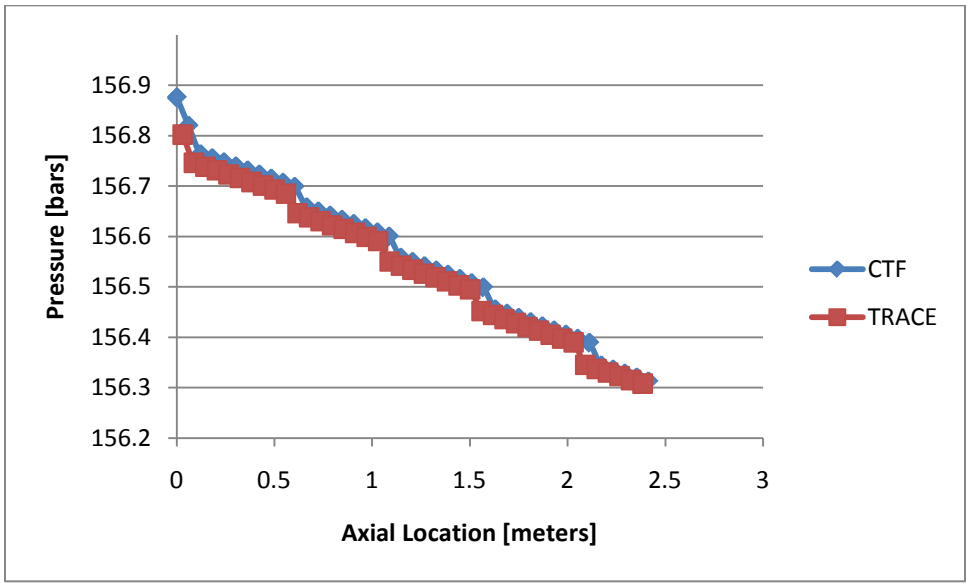

Figure 1.12: Pressure in Channel 3 with constant power

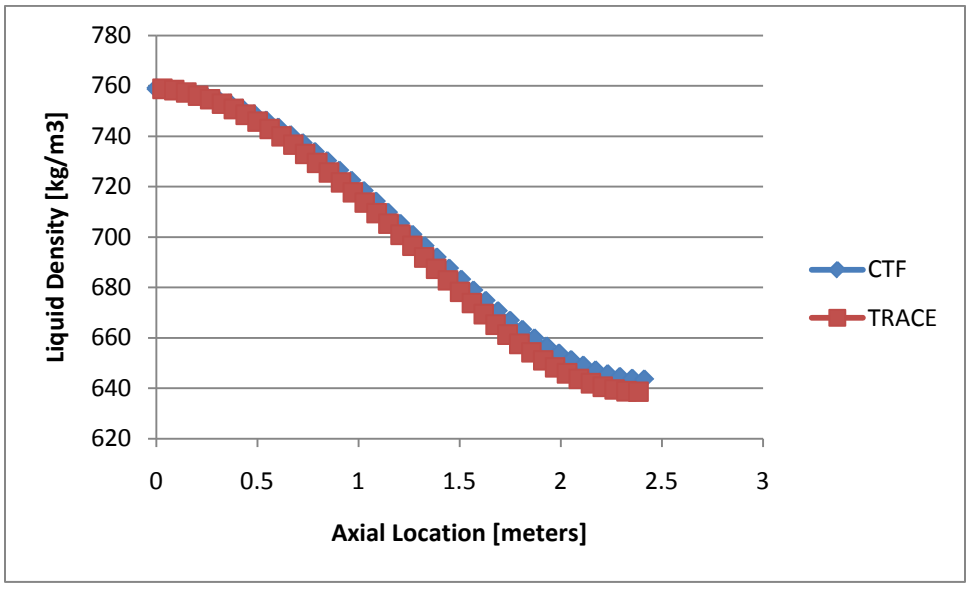

Figure 4.13: Liquid density in Channel 3 with constant power

Figure 4.12 through Figure 4.21 have the constant power option turned on in both codes and the subsequent figures have the point kinetics model for power activated. The purpose of this exercise is to verify thermal-hydraulic models in a standalone mode.

The pressure drop observed in channel 3, which is the hottest channel in the core, shows a good similarity between TRACE and CTF with five sudden drops associated with spacer grids. TRACE, at the moment, does not have the necessary capability to model spacer grids. Even though loss coefficient used in TRACE help, they are still not fully able to predict the drop associate with sudden expansion and contraction of flow through the spacer/structural grids.

Similarly, liquid density shows good agreement between the two codes with slight deviation observed at the upper end of the channel.

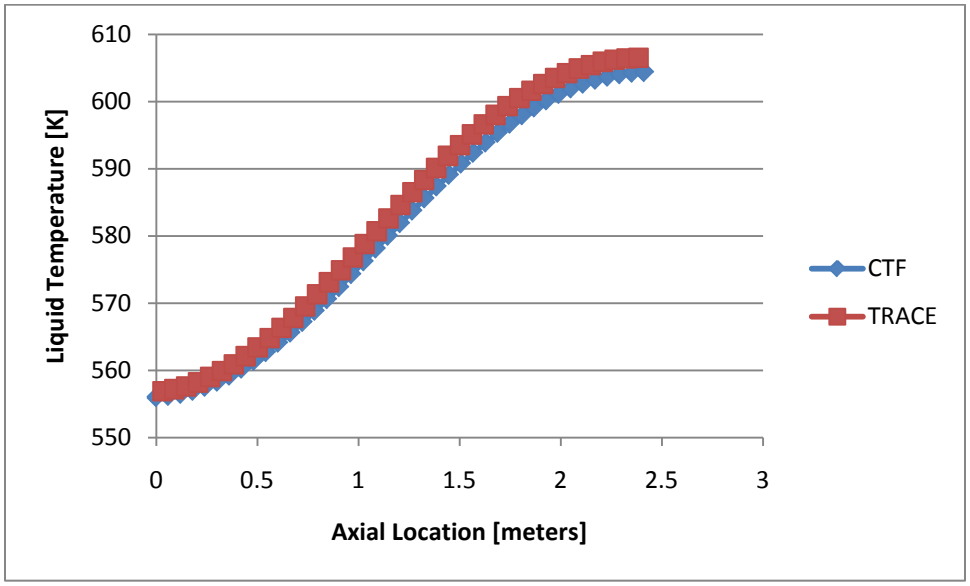

Figure 4.14: Liquid temperature in Channel 3 with constant power

Liquid temperature shows a similar pattern where the deviation is exhibited close 2.413 meters.

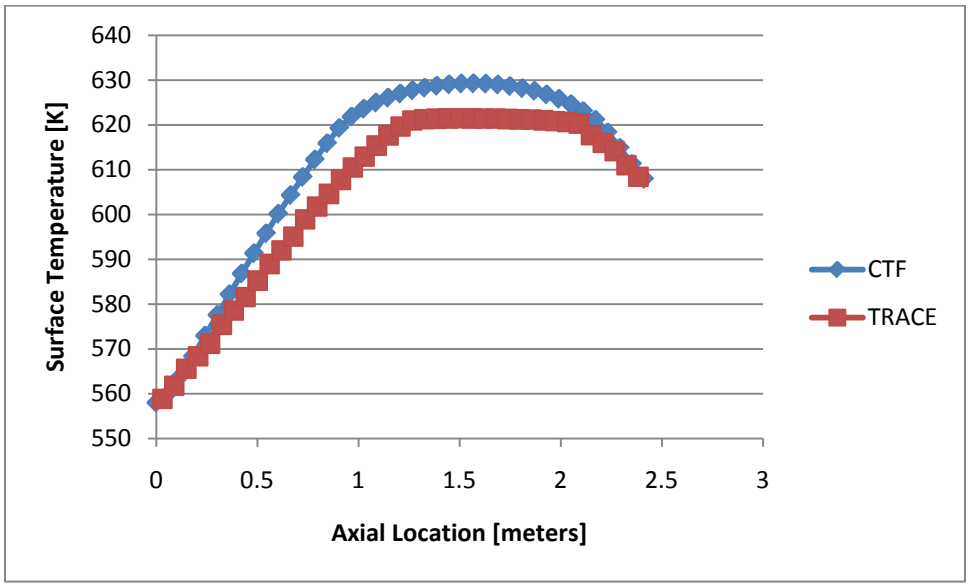

Figure 4.15: Surface/Wall temperature in Channel 3 with constant power

For channel 3, a discrepancy is observed between the two codes when comparing the clad/surface temperatures. It is especially evident in the center axial locations where high flux is expected. It could be attributed to uncertainty associate with heat structure component thickness in TRACE.

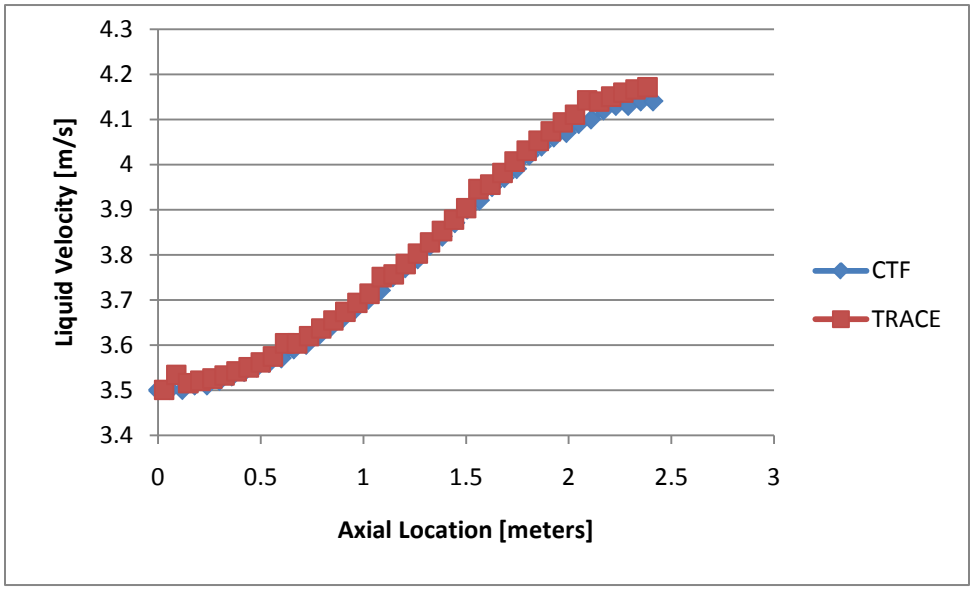

Figure 4.16: Liquid velocity in Channel 3 with constant power

The liquid velocities observed in both codes at constant power are very close with slight jumps at the spacer/structural grid location.

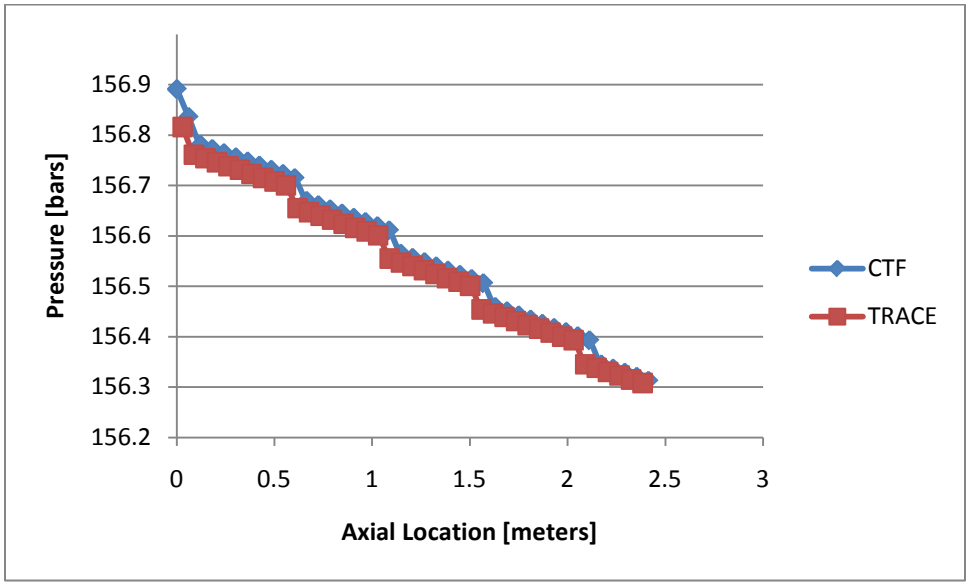

Figure 4.17: Pressure in Channel 9 with constant power

Channel 9 shows a similar pattern as the one observed in channel 3 a notable exception for surface or clad temperature.

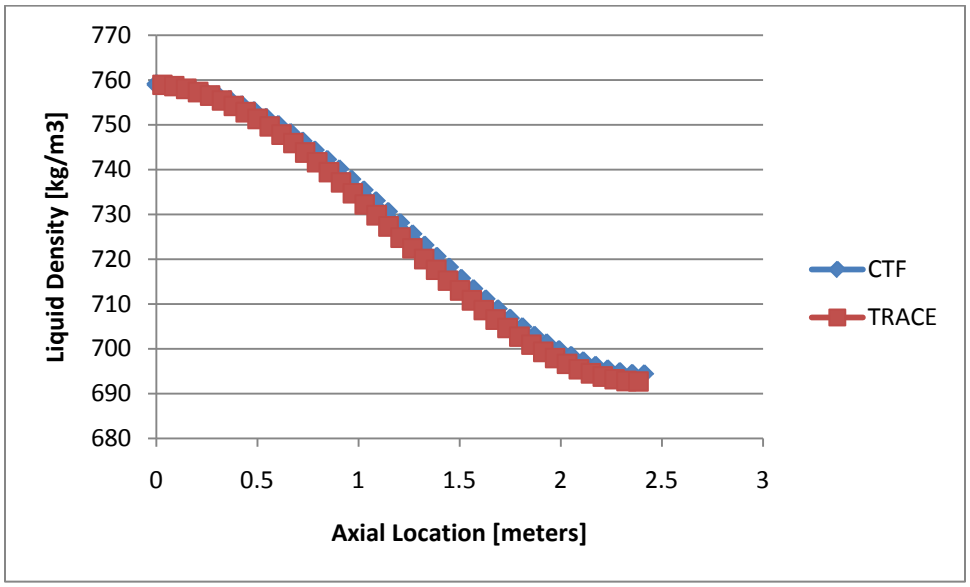

Figure 4.18: Liquid density in Channel 9 with constant power

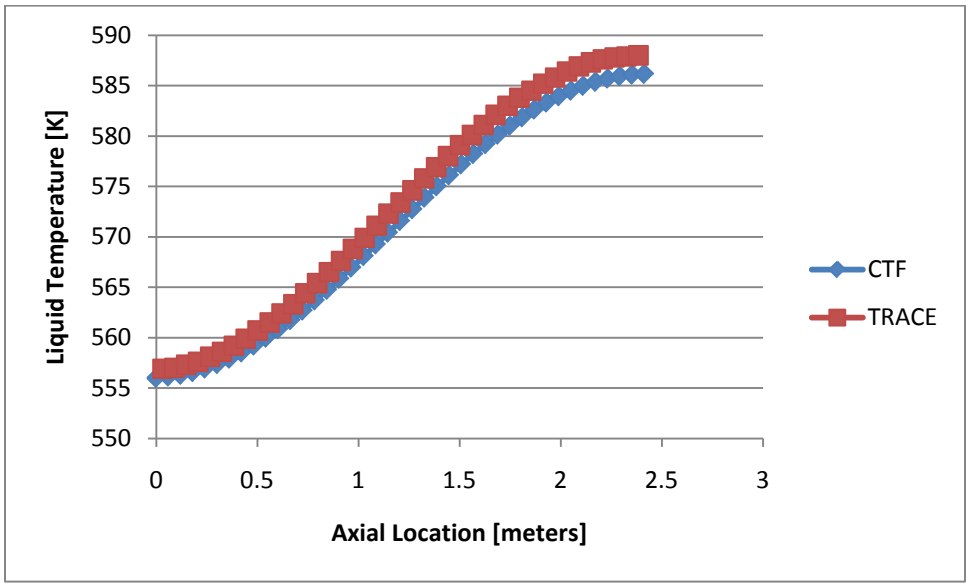

Figure 4.19: Liquid temperature in Channel 9 with constant power

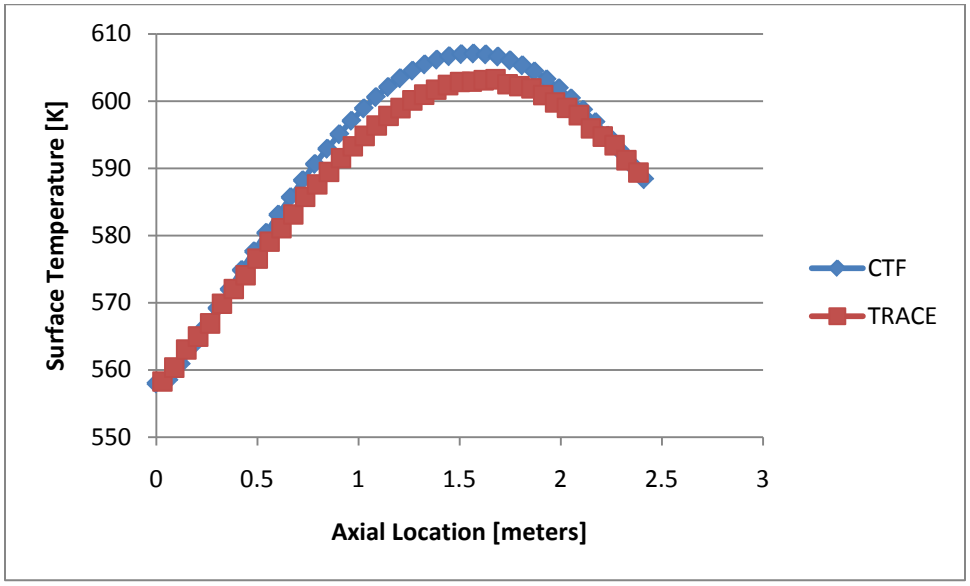

Figure 4.20: Surface/Wall temperature in Channel 9 with constant power

For channel 9, the surface temperature is much closer between the two codes. As noted before, for channel 3, there was a discrepancy observed in the middle of the channel. There is still some deviation in the middle of the subchannel in channel 9, but it is nowhere near as pronounced.

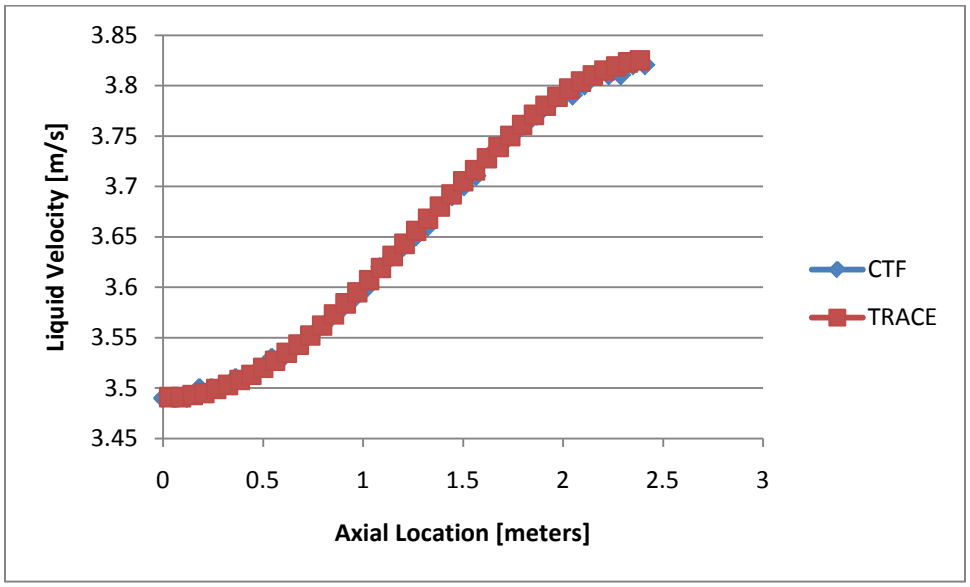

Figure 4.21: Liquid velocity in Channel 9 with constant power

Finally, with constant power, there is great agreement between the two codes for liquid velocities in channel 9. There is no appreciable deviation observed.

In the next set of figures, the point kinetics model has been activated in both codes. Boron feedback reactivity is turned off in both codes.

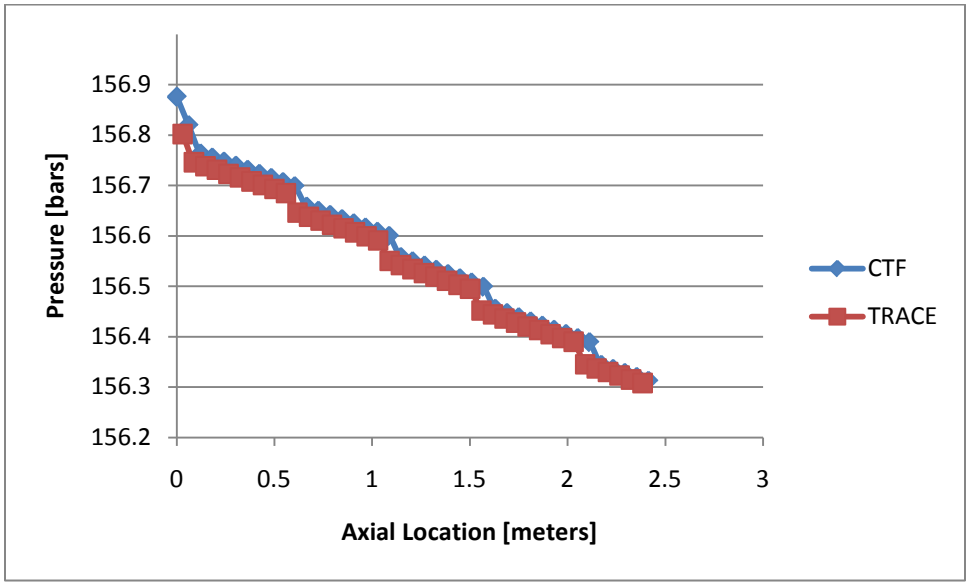

Figure 4.22: Pressure in Channel 3 with PKM activated

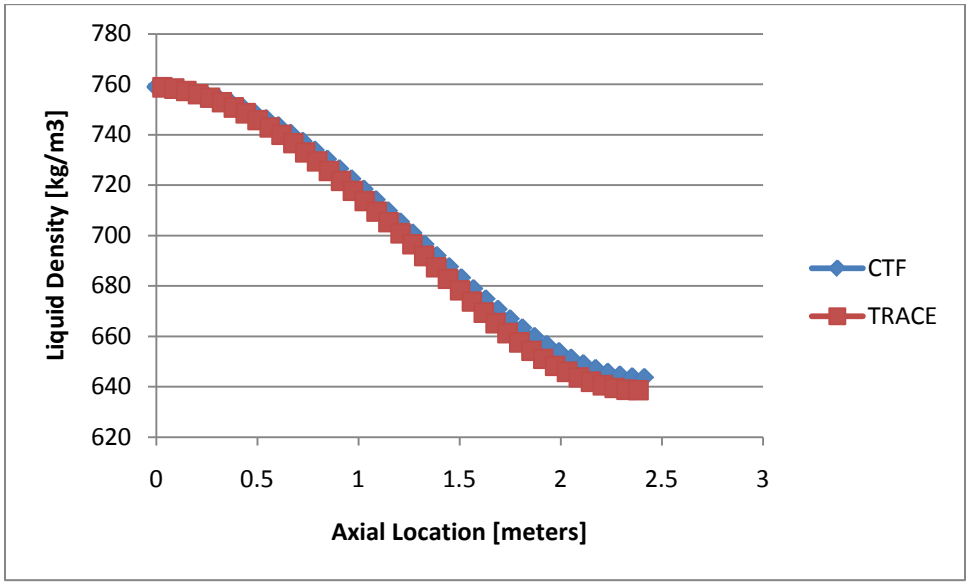

Figure 4.23: Liquid Density in Channel 3 with PKM activated

Both codes predict a similar behavior with PKM activated. CTF is predicting a slightly higher density at the upper elevations.

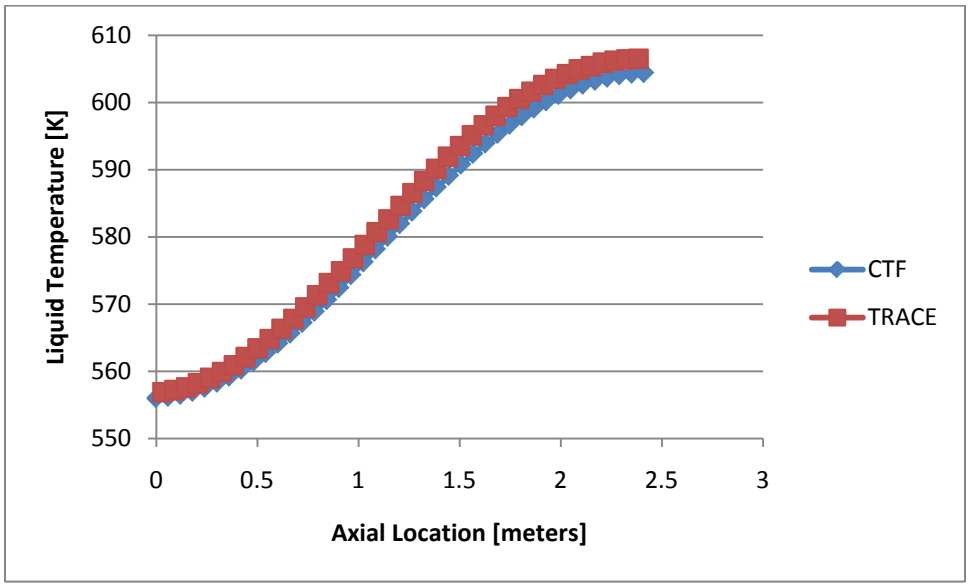

Figure 4.24: Liquid temperature in Channel 3 with PKM activated

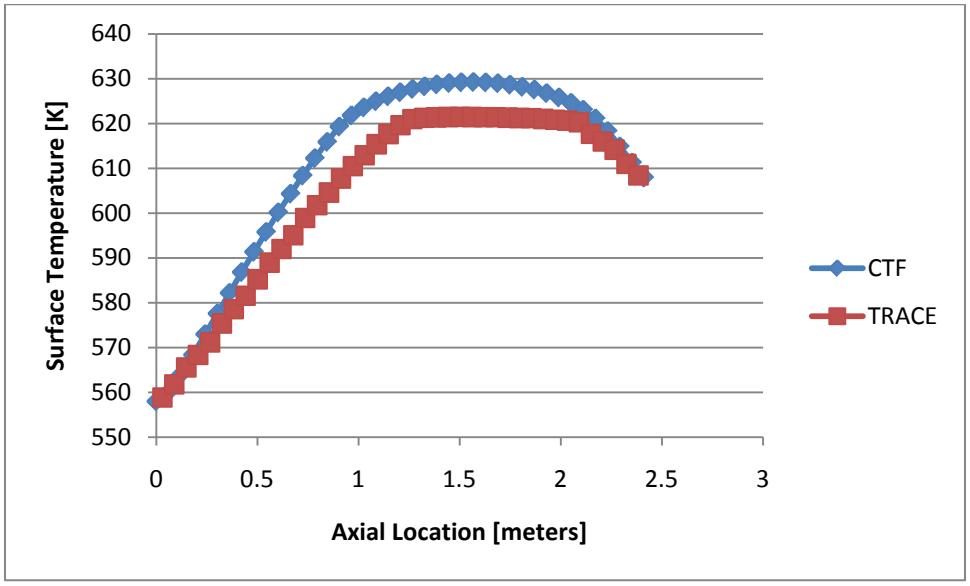

Figure 4.25: Surface/Wall temperature in Channel 3 with PKM activated

TRACE is predicting a flatter temperature profile in the middle of the channel whereas CTF has a more pronounced peak in the middle. Liquid velocities, once again, are spot on.

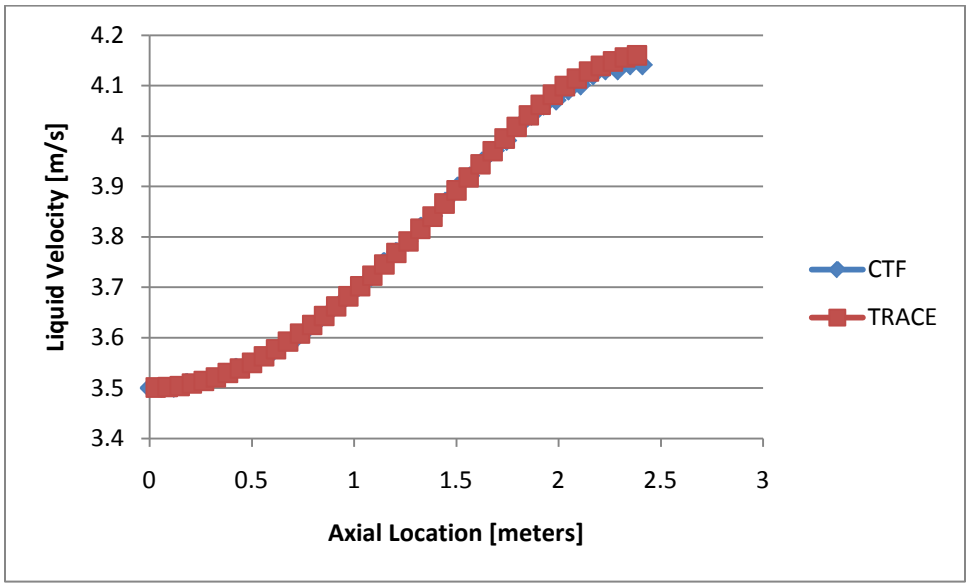

Figure 4.26: Liquid velocity in Channel 3 with PKM activated

Next set of figures contain analysis for channel 9 with PKM activated in both codes.

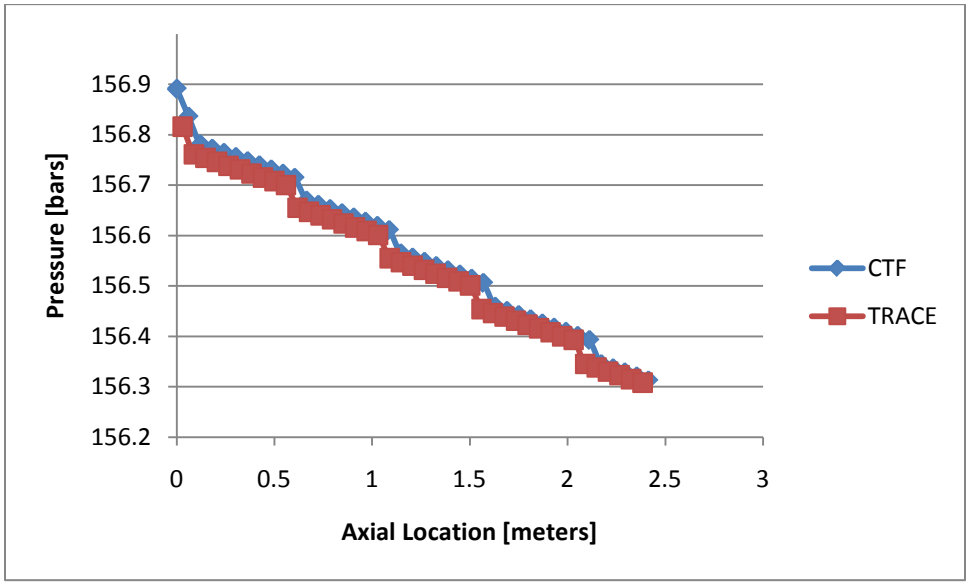

Figure 4.27: Pressure in Channel 9 with PKM activated

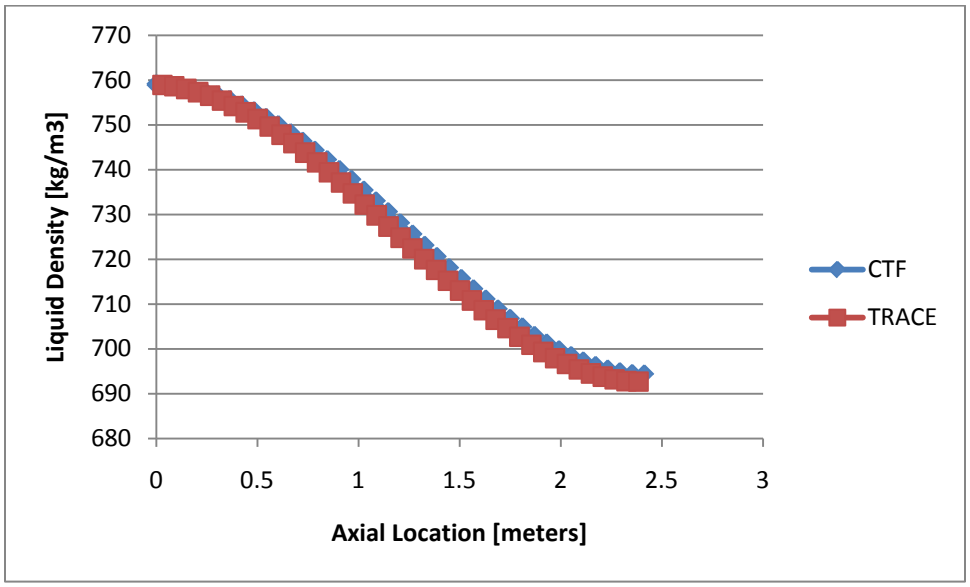

Figure 4.28: Liquid density in Channel 9 with PKM activated

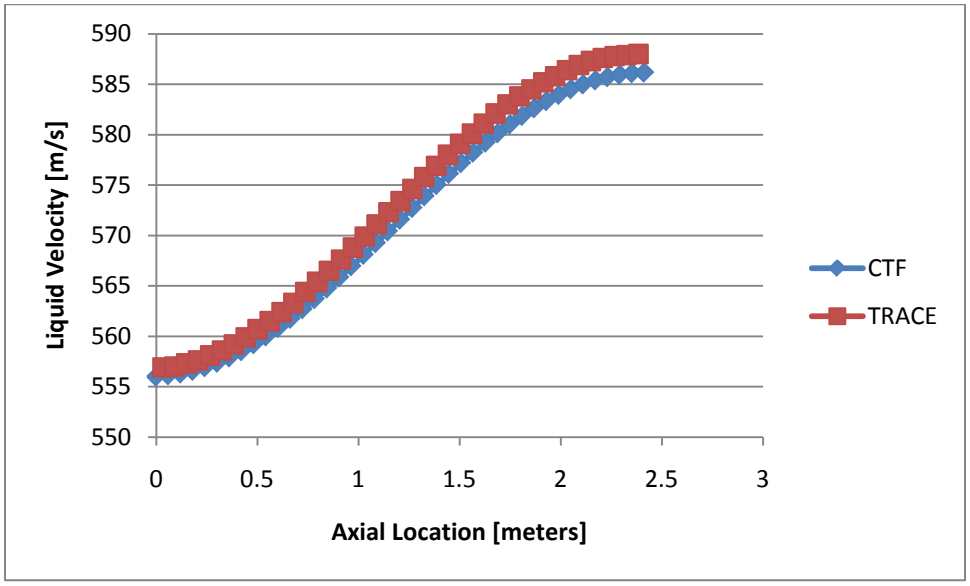

Figure 4.29: Liquid temperature in Channel 9 with PKM activated

For liquid temperature, CTF and TRACE start to deviate from each other at the higher elevation. The difference between the two codes is a bit more pronounced when PKM is used instead of constant power.

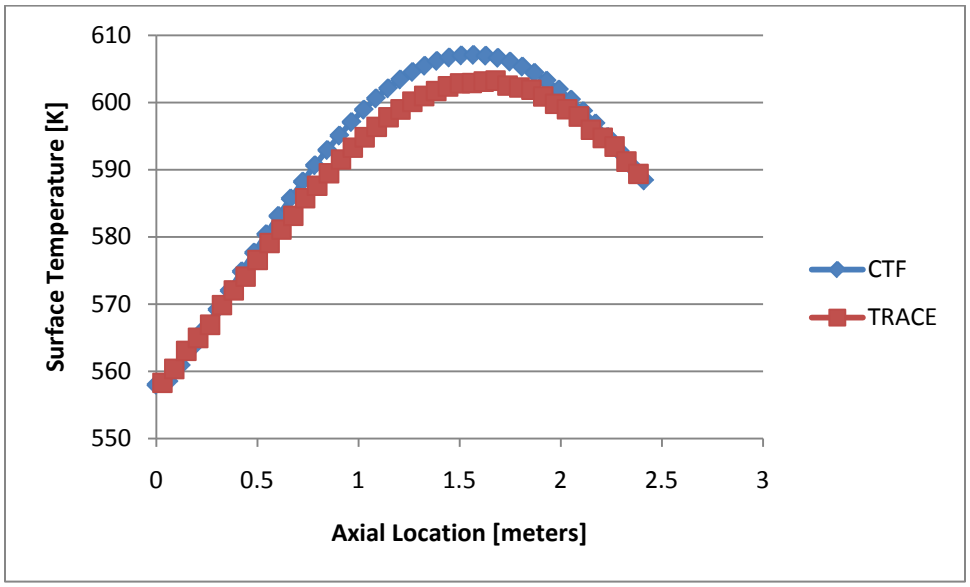

Figure 4.30: Surface/Wall temperature in Channel 9 with PKM activated

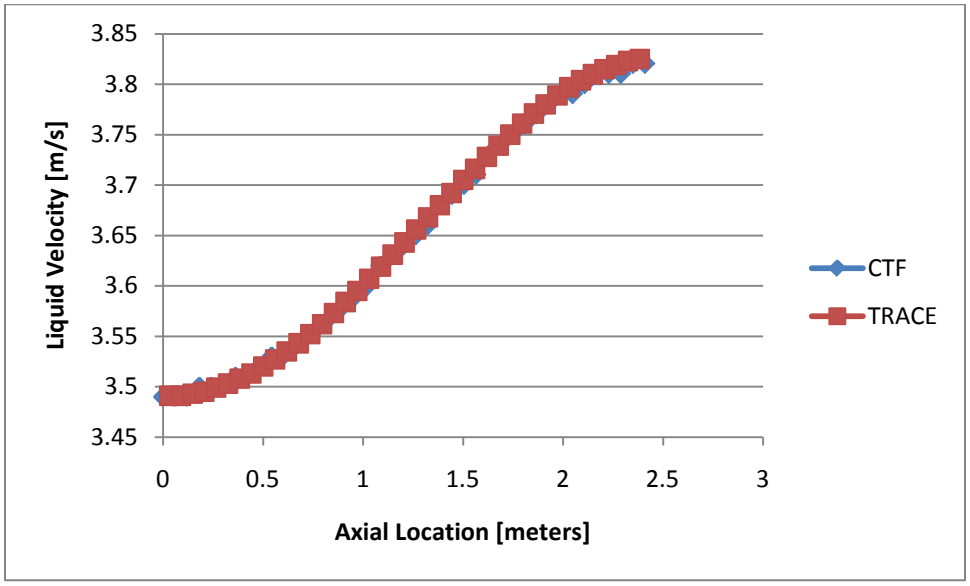

Figure 4.31: Liquid velocity in Channel 9 with PKM activated

With PKM activated in both codes, surface temperature are a little off in the middle of subchannel (power peak in the middle), but they do show good agreement for the rest of the elevations.

4.4.2. **Transient Case**

The transient test case data is collected at the end of the transient, which is 31.8 seconds. COBRA-TF is run as a null transient for 10 seconds and then the transient is run for 21.8 seconds. In TRACE, the “restart dump” option is used to reach a steady state before initiating the transient.

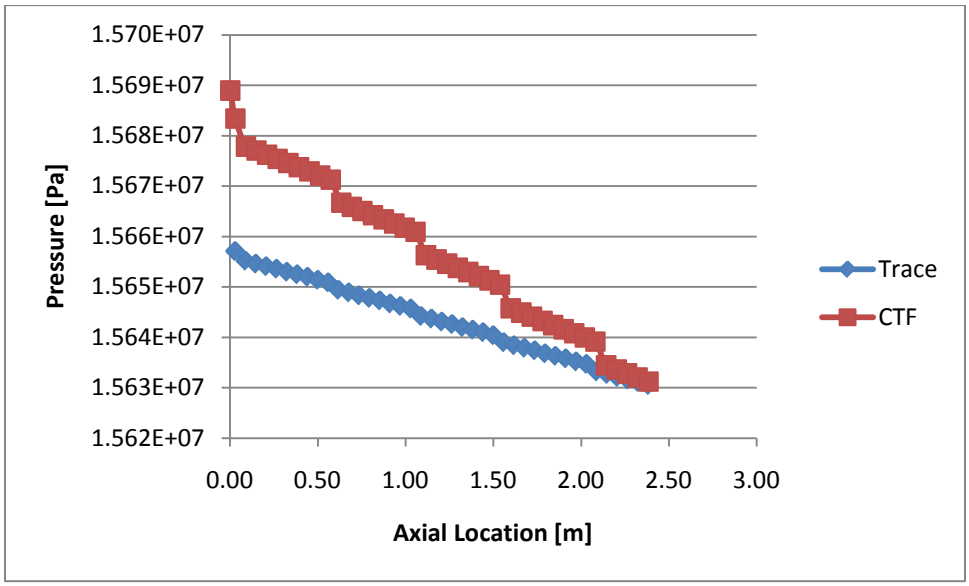

Figure 4.32: Pressure in Channel 3 for Transient Case

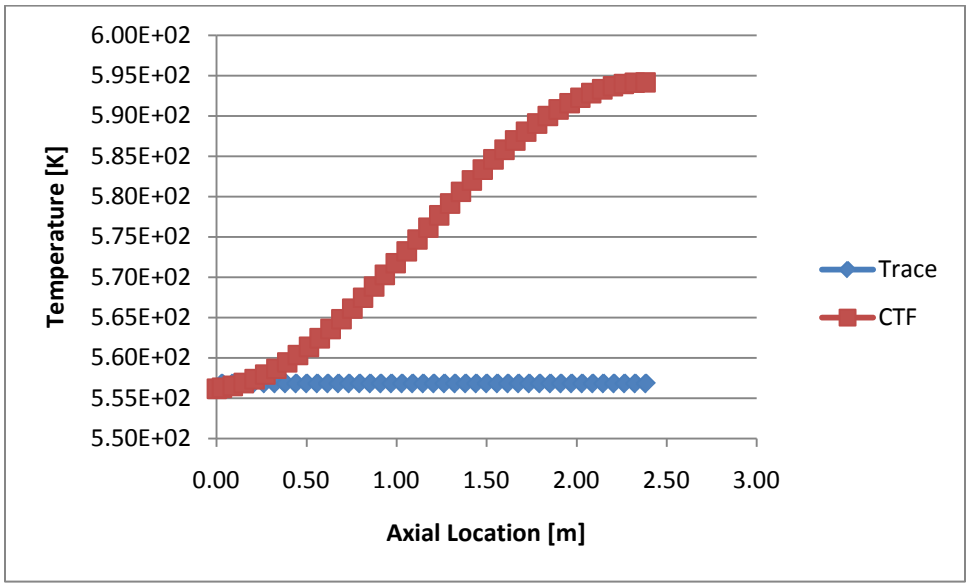

Figure 4.33: Coolant Temperature in Channel 3 for Transient Case

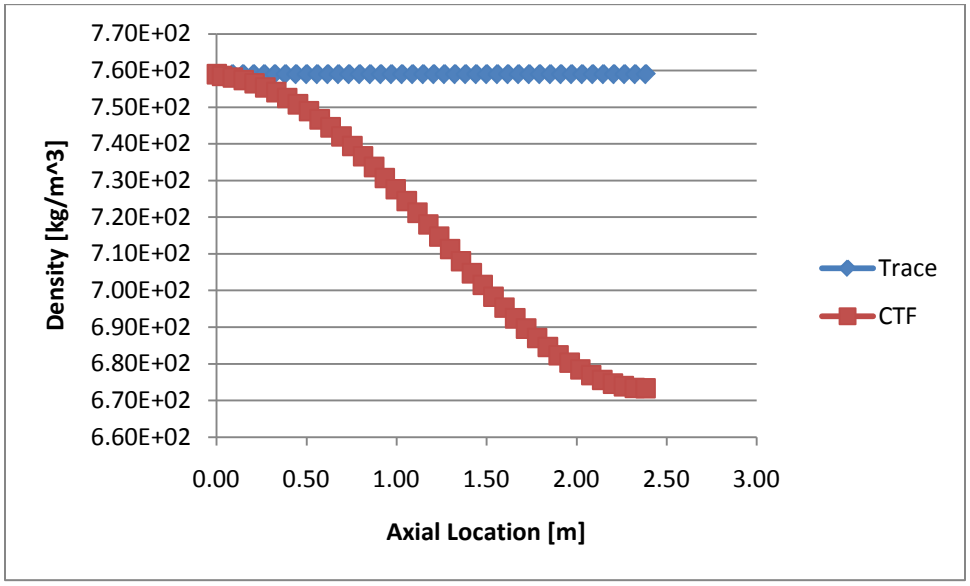

Figure 4.34: Coolant Density in Channel 3 for Transient Case

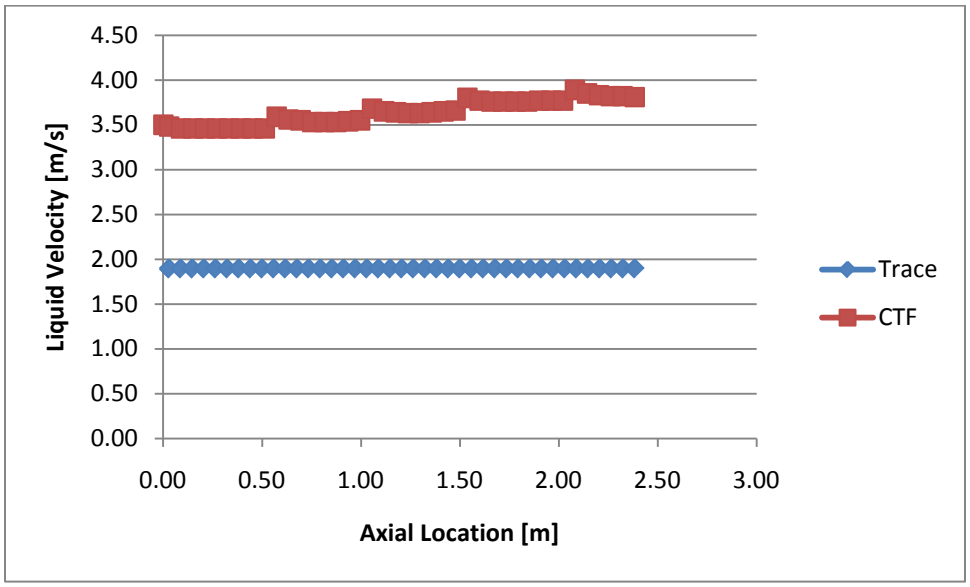

Figure 4.35: Coolant Velocity in Channel 3 for Transient Case

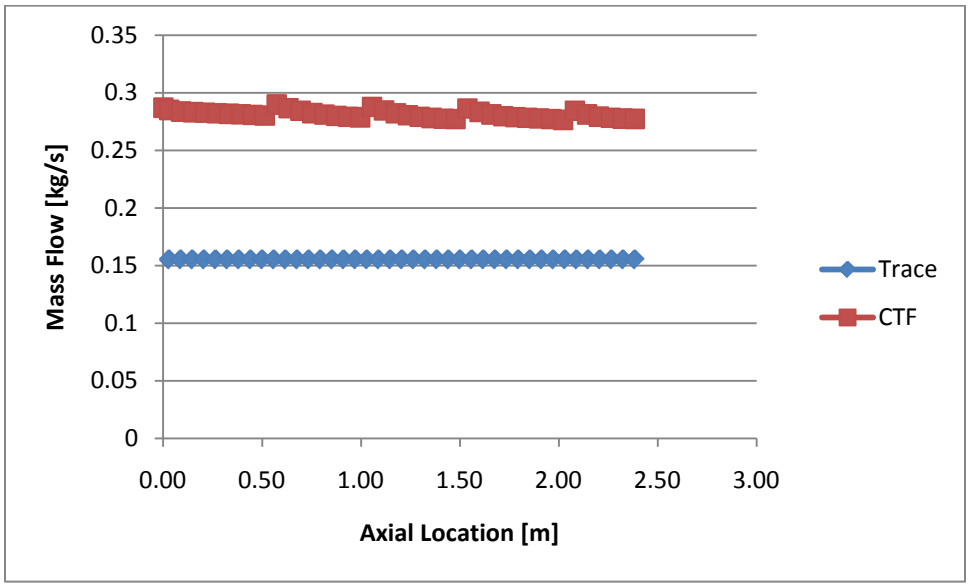

Figure 4.36: Coolant Mass Flow in Channel 3 for Transient Case

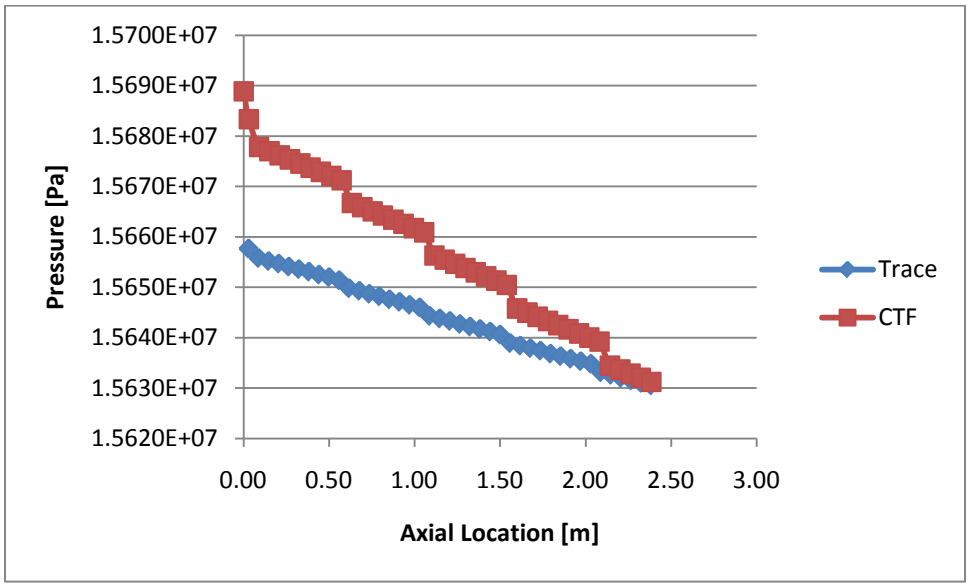

Figure 4.37: Pressure in Channel 9 for Transient Case

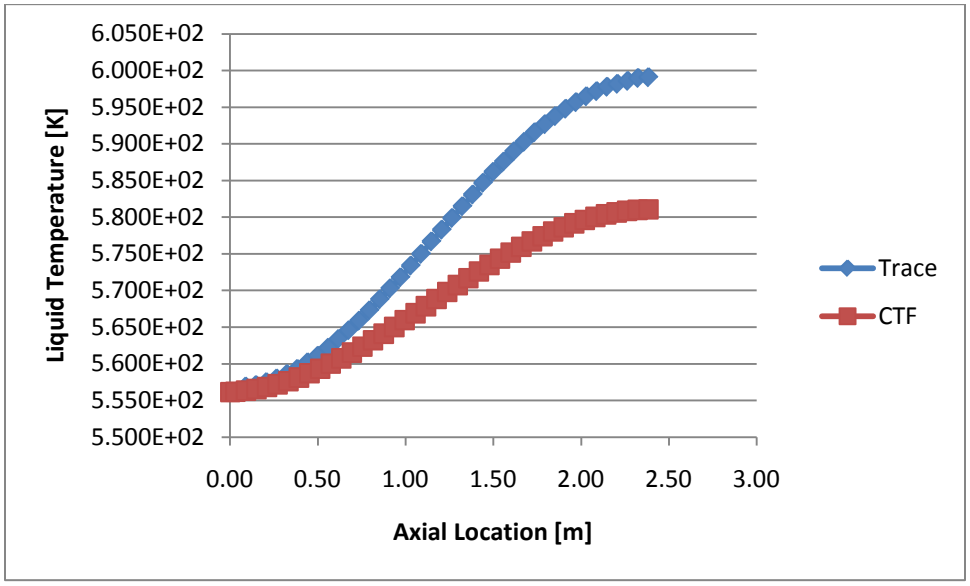

Figure 4.38: Coolant Temperature in Channel 9 for Transient Case

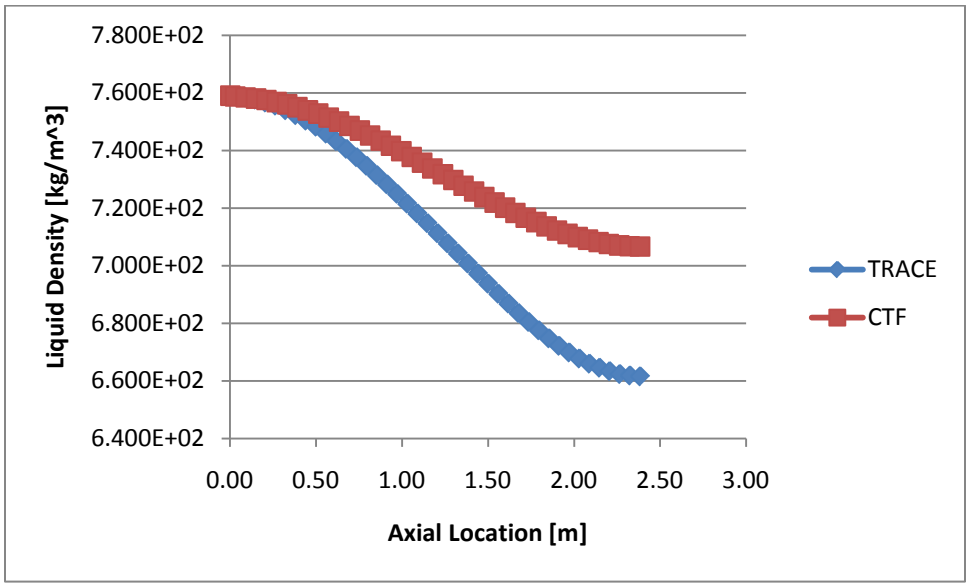

Figure 4.39: Coolant Density in Channel 9 for Transient Case

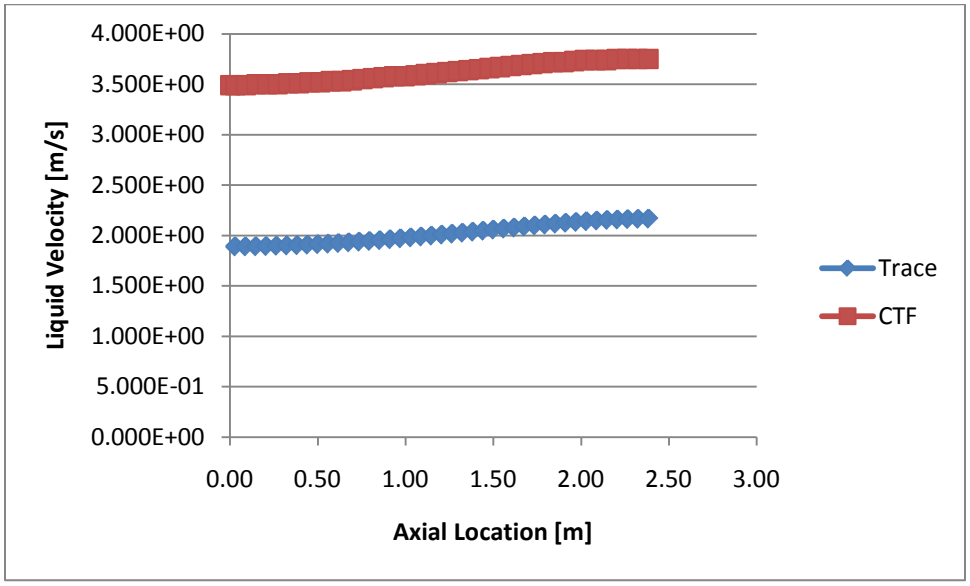

Figure 4.40: Coolant Velocity in Channel 9 for Transient Case

The next set of figures compares one of the more important aspects of the Point Kinetics Model – reactor power response during the transient. Also, the multiplication factor is compared as well between TRACE and COBRA-TF.

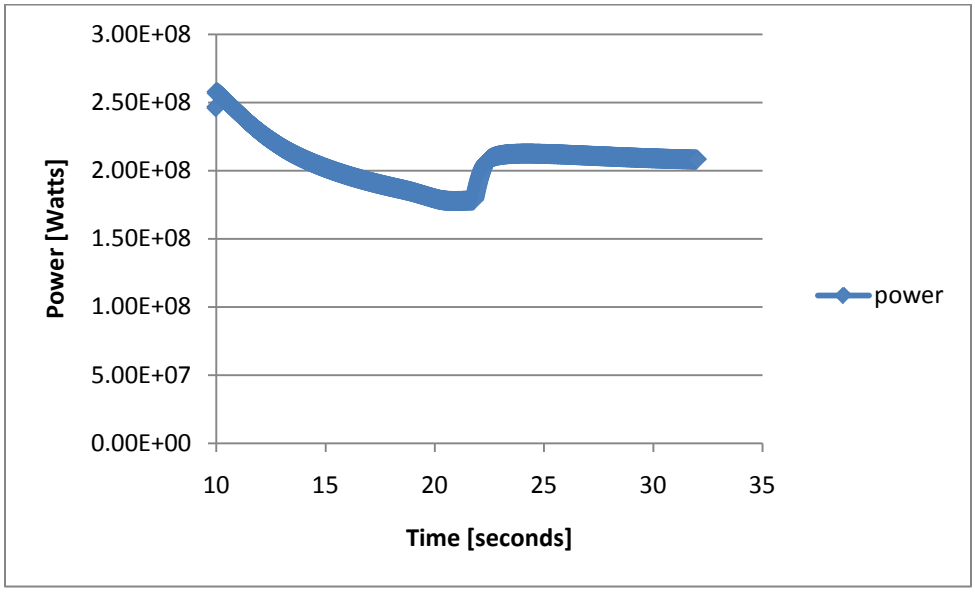

Figure 4.41: Reactor Power using COBRA-TF

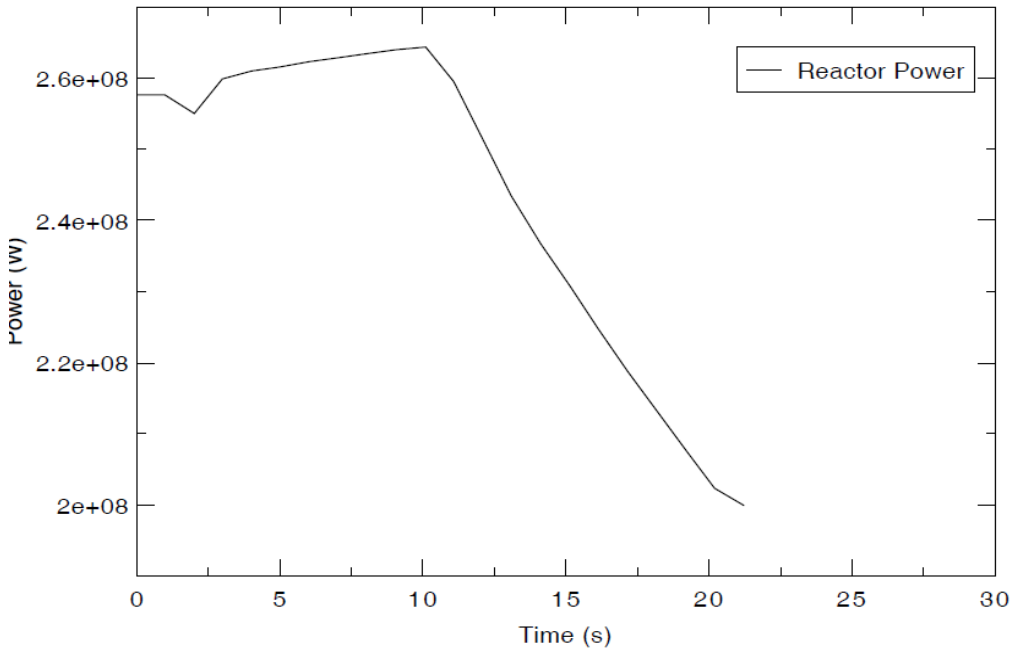

Figure 4.42: Reactor Power using TRACE

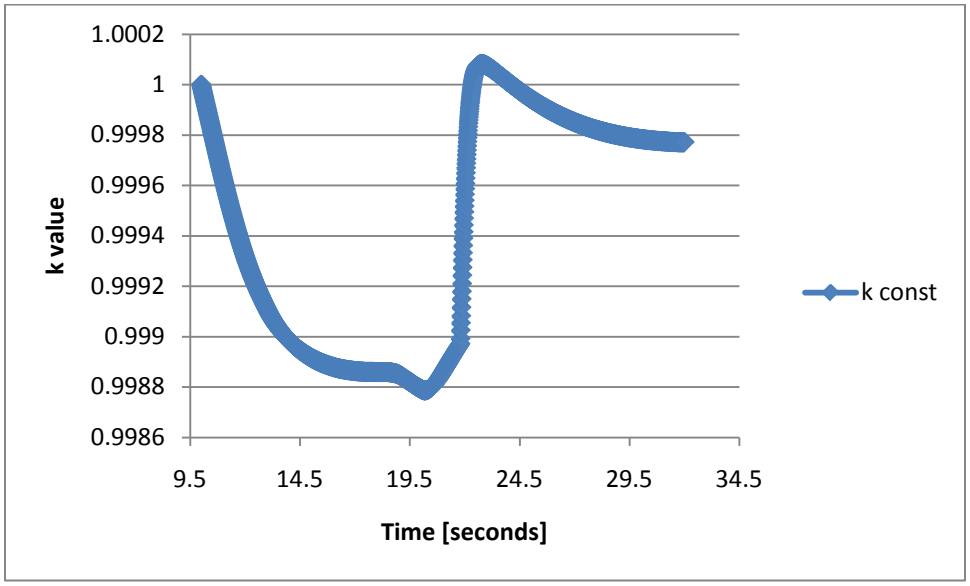

Figure 4.43: Multiplication Factor using COBRA-TF

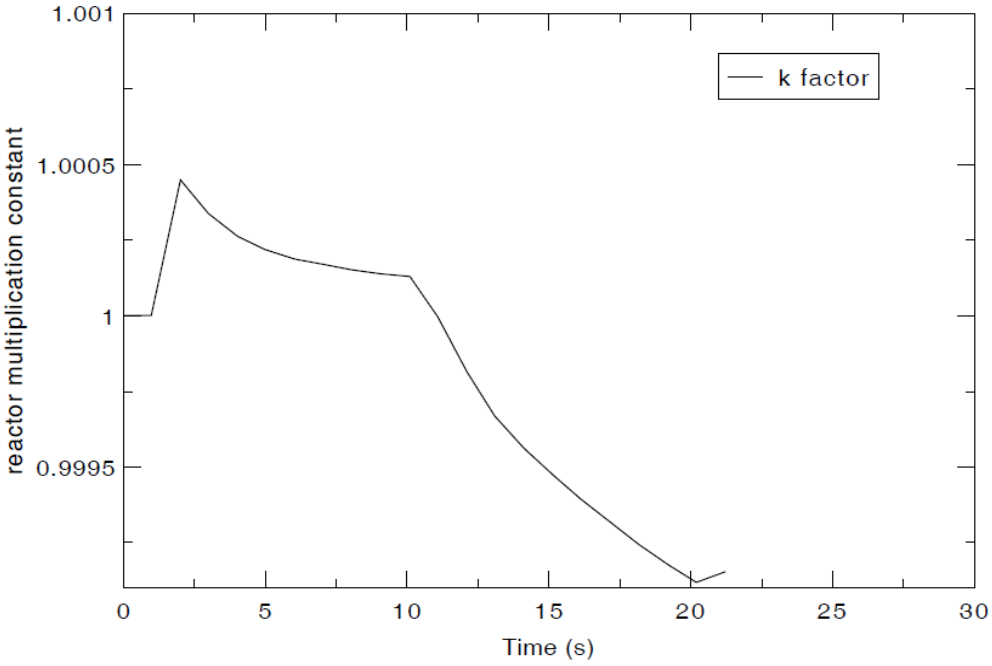

Figure 4.44: Reactor Multiplication Constant using TRACE

Figures 4.41 and 4.43 are plotted in Excel using the data extracted from COBRA-TF output file, whereas, Figures 4.42 and 4.44 are plotted using the built-in AptPlot [5] application in TRACE. Since COBRA-TF is run for 10 seconds as a "null transient" in order to reach steady state, the x-axis starts between 9.5 s and 10 s. The actual transient for COBRA-TF starts at 10 s. For TRACE, the "restart function" is employed in order to allow the test case to reach steady state before initiating the transient. The TRACE transient, when initiated, starts calculations by taking the dump file data generated for steady state and initiates the transient from the end of the file. The TRACE transient is run from time "zero" until the end of the transient. So, in essence, the time scale between TRACE and COBRA-TF do match up despite the values being off by 10 seconds.

Both codes predict a similar behavior for reactor power and multiplication factor when one takes into account the initial time to reach steady state. CTF predicts a decrease in reactor power for the first 10 (adding 10 s for the "null transient") seconds and then there is a drop in power until the end of the transient. TRACE predicts an increase in power for the first 10 seconds and then there is a sharp drop until the transient reaches 31.8 seconds.

4.5. **Conclusion**

TRACE and CTF are compared in order to perform code-to-code verification of the PKM model implemented into CTF. Of the nine channels, two channels are used for analysis; channel 3 being the hottest channel and channel 9 being the largest channel. Different state parameters are analyzed to test how closely the two codes are with either constant power or the PKM option activated in both of them. All the parameters for steady-state test case are very close with a slight discrepancy observed in surface temperature. TRACE over-predicted the results which could be

attributed to the mesh-spacing used for wall thickness. The analysis shows that the PKM routine implanted into CTF gives consistent and accurate results.

For the transient case, though, there are some noticeable discrepancies present especially for channel 3 data. Even though the trend is the same, the values differ by a significant margin. The pressure drop in both channel 3 and channel 9 as predicted by CTF is larger. The initial pressure is higher and then drops with a sharper slope. Liquid temperature in channel 3 as calculated by TRACE stays flat whereas CTF predicts an expected rise in temperature up to 600 K. Density for channel 3 stays the same according to TRACE, which is not what is expected for a transient where the temperature is increasing at steady pace. One should have a steady drop in the coolant density. CTF predicts that accurately. Over all, the transient comparison has predicted less than satisfactory results. This could be attributed to the way the TRACE input deck is created since a modified approach is used to mimic the test problem. Pipes were used as channels along with associated heat structures to model the 9 channels. It is evident that steady state comparison grants good results but the transient case has enough discrepancies to evaluate either the routines implemented in CTF or the TRACE input deck for comparison.

Chapter 5

Conclusion

The information presented in this thesis summarizes and discusses the research and COBRA-TF code modifications for the implementation of point kinetics model (PKM). It also summarizes the basis for introducing PKM in CTF, as well as its coupling with the thermal-hydraulics portion of CTF. Since the PKM being introduced into CTF is based on a similar model as one present in TRACE, a code-to-code verification was performed between TRACE and CTF, using a sample test case with a steady-state and a transient scenario.

Research has resulted in the implementation in CTF of the basic Point Kinetics, Feedback, and Decay Heat Models and using two (1971 and 1979) ANS heat standards for the kinetics, delayed neutron and decay heat data.

The basic CTF model is a computationally efficient model designed for PWR applications, in which the power is determined from the solution of the point reactor kinetics equations. These equations specify the time behavior of the core power level with neutronic reactivity as the driving parameter. The user inputs programmed reactivity thus defining reactivity effects not accounted for by feedback reactivity, such as fuel reactivity and control-rod movement. The model evaluates feedback reactivity based on changes in the core-averaged fuel temperature, coolant temperature, coolant vapor fraction, and boron concentration.

Whereas CTF in its pre-PKM version did not calculate core-averaged fuel temperature, coolant temperature, and coolant vapor fraction, such procedure has now been implemented. In addition, in a companion study, an algorithm has been implemented based on the boron tracking method to evaluate the core-averaged boron concentration. The PK model works in conjunction

with associated models such as Decay Heat and Feedback models. The PK and associated models require data for delayed neutron fractions, decay constants and other decay heat data to be hardwired into the code. The two ANSI/ANS standards for such data are used – 1971 and 1979. For legacy purposes, the older decay heat standard, ANS 1971, has been implemented.

The development of Point Kinetics Model began with a literature review of the point kinetics approach to calculate reactivity and power in a reactor. Even though the COBRA-TF implemented PKM is a coarse approximation of the "Exact" Point Kinetics Model, it gives more than adequate results with minimal computational cost. CTF in its original version only had the so-called constant power model. In this constant power approach, the user input a constant power into the code, and the code calculates the requisite thermal-hydraulics parameters such as void fraction, fuel rod temperatures, moderator temperature, etc. It became clear that there was a need to either couple CTF with a neutronics code or to develop and introduce a brand new power calculation model into CTF. Given the relative ease of programming and testing a new module based on point kinetics approach, it was decided to go with the second option rather than try to couple CTF with any of the available neutronics codes. Initial programming was done to test the feasibility of this option and see how CTF performs with this new approach. Point Kinetics Model works in conjunction with the Decay Heat Model, and Feedback Reactivity Model. These two complimentary models had to be developed and programmed concurrently and then coupled with PKM in the code. The reactivity feedback model is based on the assumption that the multiplication factor is affected by the change in the fuel temperature, moderator temperature, moderator void fraction, and boron concentration. All these factors either alone or in conjunction with each other affect the neutron population in a reactor. Programming was done to have the code calculate the core-averaged values by applying fuel or coolant mass times power times volume weighting factors to the temperature, and power times volume-weighting factor to the coolant vapor fraction and boron concentration. These factors have a direct impact on the adjoint

flux. The adjoint flux is a type of importance function and is used to weight spatially the change in reaction-rate cross sections to estimate their reactivity change. The Feedback Model programming module is then coupled with PKM in the code. Decay Heat Model, which accounts for decay heat in the reactor along with that from the actinides is programmed with the help of decay heat constant formulation. With all that done, all three models intertwine and work in conjunction with each other to take a given power from the user, take into account any feedback and residual heat, and then predict the power for the next time step in CTF. CTF then takes this power and calculates the thermal hydraulic values. These updated values are then forwarded to PKM for the next time step to calculate power. In essence, one calculation follows the other and is then updated after each time step.

The programmed PKM routine has given consistent results, but there was a need to perform some sort of verification and validation of the new models. Since the PKM routine is very similar to what is programmed in TRACE for power calculation, TRACE was chosen for code-to code verification for PKM. A PWR test problem was devised with $1/4^{th}$ of a PWR core including both a steady state and a transient solution. A test input deck was created for CTF. Since TRACE is a component based code, SNAP was used to create a model of the quarter core with the exact same specifications. A few assumptions had to be made to keep the two models, CTF and TRACE, consistent and compatible with each other. TRACE at the moment only has a fuel channels model for Boling Water Reactors, so a novel and somewhat different approach was needed to create the TRACE model. Pipes and attached heat structures were used to model the nine subchannels found in CTF. Each of the nine pipes acts as subchannel and the attached heat structures provide power to these channels.

The results of the code-to-code verification for steady-state case are very positive overall. The steady state parameters predicted by CTF with PKM and the same thing predicted by TRACE are very close with minor discrepancies. The variables that were compared include

pressure, liquid velocity, void fraction, coolant density, coolant temperature, and fuel surface temperature. The values predicted by TRACE and CTF fall within 2 % of each other for the steady state case in both channel 3 and channel 9. A larger than expected discrepancy was observed with fuel surface temperature in channel three. Both codes predict the same surface temperature up to an axial location of 0.5 meters, but between 0.5 meters and 2.1 meters, CTF over predicts by 10 . This was attributed to how the TRACE model was created, using pipes with heat structures that mimic fuel rods. The lack of actual fuel channel structure option necessitates the need to use this approach. A fuel channel in CTF is an open structure with the ability to enable crossflow among neighboring channels. Pipes in TRACE, on the other hand, are enclosed independent structures, which have to be attached to other structures. Although attempts were made to resolve this discrepancy between the two models, the results have remained the same. More work is needed to locate the root cause of this disagreement between the two codes. For the transient test case, discrepancies were observed as well between TRACE and CTF. TRACE calculates an erroneous fuel temperature for channel 3. One should observe a steady increase in temperature but that is not the case. Similarly, other state parameters despite having the same trend show a marked difference between the two codes. It could be attributed to the feedback model programmed into CTF, but that is still a speculation since further testing still continues. One of the more important parameters, reactor power, does show promising agreement between the two codes. Both of the codes predicts a drop in reactor power, and the core power for both cases reaches a value of W at the end of the transient. It would be too hasty to associate the discrepancies with questionable TRACE test case model since steady state comparison between CTF and TRACE does provide promising data. Further work is needed to isolate the issue with the transient portion of the test case.

As it stands, the implementation of PKM and the associated models, Decay Heat Model and Feedback Model, has produced consistent results for steady state calculations. The point

kinetics model with its ability to predict core power based on feedback reactivity from fuel temperature, coolant temperature, coolant density, and boron concentration has made CTF an attractive alternative to system codes such as TRACE.

Finally, given more time, the anomalies and discrepancies in the implementation of the PKM and its associated models can be resolved. It could be accomplished by devising a test problem involving a Boiling Water Reactor core since this would allow one to use the fuel channel option in TRACE. CTF, like any other reactor analysis code, is constantly being updated with newer options to meet the demands of current as well next generation of nuclear reactor safety analysis and as such PKM is one of the steps taken in that direction.

5.1. **Future Work**

There are a handful of issues that need to be resolved with the current implementation of the Point Kinetics Model in COBRA-TF. First and foremost is the slight discrepancy observed between the analytical PK and numerical PK solution. It is postulated to be related to the numerical scheme being used in the algorithm which results in the accumulation of a small error at each time step. Another option being explored is the possibility of truncation error at each time step. Work is currently in progress to debug the source code for COBRA-TF to isolate and ameliorate the issue.

In order to make the PKM more comprehensive, an optional coolant density reactivity feedback is being implemented into COBRA-TF. Currently the PKM takes into account the Doppler feedback, coolant void feedback, and boron concentration feedback. Further testing and research is needed to have this option provide consistent results.

As noted in the transient test case comparison between COBRA-TF and TRACE (please see Figures 32 through 44), there is found to be some inconsistency between the two codes. The

TRACE input deck, which was specifically created for testing purposes, relied on a handful of assumptions in order to replicate the COBRA-TF test problem. Pipe structures were used to mimic core subchannels – 9 subchannels in total. The reason for this novel approach is that TRACE lacks core fuel subchannel option for PWR applications; only BWR fuel subchannel option is available. This approach gives rise to some unexpected results and need to be addressed in order to complete the code-to-code verification of the PKM.

Lastly, options are being explored to introduce a Gamma Smearing Model into COBRA-TF. It is a general energy deposition model. A literature review of the model has been performed and further research will be done to have this model implemented into COBRA-TF.

Appendix A

CTF Input Deck

```
*******************************************************************************
* 9 Channel PWR Core Model; Embedded Mesh Structure                           *
* Loss of Coolant Transient                                                   *
* Ref.: Avramova, M., et al., "Comparative Analysis of PWR Core Wide and Hot  *
*       Channel Calculations",ANS Winter Meeting, 2002                        *
* Maria Avramova, May 2008                                                    *
*******************************************************************************
* Input related to the point kinetics modeling
* I_POWER
      1
* Read number of delay groups, decay groups and history
* IRPWTY    I_PKM    IANS
     1        1      2
*   NDGX     NDHX   NHIST
      6       71       1
*   Q235     Q239     Q238     QAVG     R239PF
  192.5    198.5    193.5    195.0       0.5
* FISPHI     RANS    FP235    FP238
 1.5         1.0     0.95      0.01
* If IRPWTY=2
* NRPWTB
*
* If NRPWTB >0
* Read NRPWTB pairs of entrees for TIMETB and RPWTB
* TIMETB(1) RPWTB(1)  TIMETB(2) RPWTB(2)  ... TIMETB(NRPWTB)  RPWTB(NRPWTB)
*
* End if NRPWTB >0
* End if IRPWTY=2
* Read neutron generation time
*  REACT   TNEUT   RRPWMX
    0.0  1.625e-5 1.00e+20
* RPOWRI   EXTSOU    DTPK      MHI_TIME
 1.161e+9   0.0     1.0e-05     10.0
* If I_PKM = 1
* Read
* IRCJTB(1,1) IRCJTB(2,1) IRCJTB(3,1) IRCJTB(4,1) IBU(1)
          2          1          1          1      0
* IRCJTB(1,2) IRCJTB(2,2) IRCJTB(3,2) IRCJTB(4,2) IBU(2)
          1          2          1          1      0
* IRCJTB(1,3) IRCJTB(2,3) IRCJTB(3,3) IRCJTB(4,3) IBU(3)
          1          1          1          1      0
* IRCJTB(1,4) IRCJTB(2,4) IRCJTB(3,4) IRCJTB(4,4) IBU(4)
          1          1          1          1     -2
* IRCJFM(1) IRCJFM(2) IRCJFM(3) IRCJFM(4) ISNOTB
        1         1         0         0      0
* POWEXP   BPP0   BPP1  BCR0  BCR1
    2.0 2.6e-1 -2.0e-3 1.0e-1 -3.0e+1
* Read IRC(1) entrees for RCTF
*  RCTF  RCTF  RCTF  RCTF ....
 5.5e+02 8.0e+02 0.00e+0 0.00e+0 0.00e+0 -2.928e-05 -2.3334e-05
* Read IRC(2) entrees for RCTC
*  RCTC  RCTC  RCTC  RCTC ....
 0.00e+0 5.5e+02 6.0e+02 0.00e+0 0.00e+0 -9.850e-05 -2.0660e-04
* Read IRC(3) entrees for RCTC
*  RCAL  RCAL  RCAL  RCAL  ....
0.00e+0 0.00e+0 0.00e+0 0.00e+0 -1.2345e-01
```

```
* Read IRC(4) entrees for RCBM
*  RCBM  RCBM  RCBM  RCBM  ....
0.00e+0 5.50e+02 0.00e+0 0.00e+0 -2.321e-05
* End if I_PKM = 1
*
* Read precursor beta's and dcdn's (decay constant) for each delay group
* Read NDGX times BETA_PK
* BETA_PK(1) BETA_PK(2) BETA_PK(3) ... BETA_PK(NDGX)
1.69e-4 8.32e-4 2.64-3 1.22-3 1.38-3 2.47-4
*Read NDGX times DCDN
* DCDN(1) DCDN(2) DCDN(3) ... DCDN(NDGX)
3.87e+0 1.4e+0 3.11e-1 1.15e-1 3.17e-2 1.27e-2
* If NHIS = 0
* Read precursor concentration CDGN NDGX times
* CDGN(1) CDGN(2) CDGN(3) ... CDGN(NDGX)
*
* Read decay-heat concentrations CDHN NDHX times
* CDHN(1) CDHN(2) CDHN(3) ... CDGN(NDHX)
*
* End if NHIS = 0
*
* Read decay group constants DCDH  and energy fractions EDH
* If IANS = 0 read DCDH and EDH NDHX times
* DCDH(1) DCDH(2) DCDH(3) ... DCDH(NDHX)
*
*  EDH(1)  EDH(2)  EDH(3) ...  EDH(NDHX)
*
* End if IANS = 0
*
* If NHIST > 1
* Read THIST and PHIST NHIST times
* THIST(1) PHIST(1) THIST(2) PHIST(2) ...  THIST(NHIST) PHIST(NHIST)
*
* Read FP235 (NHIST-1) times
* FP235(1) FP235(2) ... FP235(NHIST-1)
*
* Read FP239 (NHIST-1) times
* FP239(1) FP239(2) ... FP239(NHIST-1)
*
* End if NHIST > 1
*
* END of Point Kinetics Input
*###############################################################################
* CARD INPUT 1
*      ICOBRA
         1
* CARD INPUT 2
*     INITIAL        DUMPF
          1           0
* CARD INPUT 3
*       EPSO       OITMAX       IITMAX
       0.001           1           40
* CARD COBRA 1
*<-----TEXT------>
   ****9_chan_PWR_core****
*
*******************************************************************************
*GROUP 1 - Calculation Variables and Initial Conditions                       *
*******************************************************************************
*NGRP
   1
* NGAS IRFC EDMD IMIX ISOL NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
   1    2   1    1   0   0   0   0   0   0   0   0   0   0
*    GTOT     AFLUX    DHFRAC
      0. 22.214567       .0
*    PREF      HIN      HGIN    VFRAC1    VFRAC2
  156.304  1251.255   288.233       1.0     .9999
*GTYPE        VFRAC
```

```
air           .0001
*******************************************************************************
* END GROUP 1                                                                  *
*******************************************************************************
*
*
*******************************************************************************
* Group 2 - Channel Description

                                                                               *
*******************************************************************************
*NGRP
   2
*NCHA NDM2 NDM3 NDM4 NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
   9   0   0   0   0   0   0   0   0   0   0   0   0   0
*  I      AN      PW  ABOT ATOP NMGP
  1 0.000094 .036144    0   0   0
  2 0.000094 .036144    0   0   0
  3 0.000108 .033833    0   0   0
  4 0.000094 .036144    0   0   0
  5 0.001186 .424434    0   0   0
  6 0.003606 1.12801    0   0   0
  7 0.012406 4.17068    0   0   0
  8 0.028593 9.38784    0   0   0
  9 0.574773 189.814    0   0   0
*******************************************************************************
* END GROUP 2                                                                  *
*******************************************************************************
*
*
*******************************************************************************
* Group 3 - Transverse Channel Connection (Gap) Data                           *
*******************************************************************************
*NGRP
   3
*  NK NDM2 NDM3 NDM4 NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
  21   0   0   0   0   0   0   0   0   0   0   0   0   0
***** Group 3.2
*K  IK JK     GAPN     LNGT  WKR FWAL IGPB IGPA  FCT IGP  JGP  IGP  JGP IGP JGP
 1  1  2 0.003350 0.014120  1.5  0.0    0   0  1.0  3   6   0   0  0  0
***** Group 3.3
*GMLT ETNR
 1.0  0.0
***** Group 3.2 and 3.3 continued
 2  1  4 0.001885 0.014120  1.5  0.0    0   0  1.0  4   12   0   0  0  0
 1.0  0.0
 3  1  5 0.001885 0.014120  1.5  0.0    0   0  1.0  1   15   0   0  0  0
 1.0  0.0
 4  1  5 0.003350 0.014120  1.5  0.0    0   0  1.0  2   -1   0   0  0  0
 1.0  0.0
 5  2  3 0.003350 0.014120  1.5  0.0    0   0  1.0  7   10   0   0  0  0
 1.0  0.0
 6  2  5 0.001885 0.014120  1.5  0.0    0   0  1.0  1   13   0   0  0  0
 1.0  0.0
 7  2  5 0.001885 0.014120  1.5  0.0    0   0  1.0  5   -1   0   0  0  0
 1.0  0.0
 8  3  4 0.003350 0.014120  1.5  0.0    0   0  1.0  9   11   0   0  0  0
 1.0  0.0
 9  3  5 0.003350 0.014120  1.5  0.0    0   0  1.0  8   13   0   0  0  0
 1.0  0.0
10  3  5 0.003350 0.014120  1.5  0.0    0   0  1.0  5   14   0   0  0  0
 1.0  0.0
11  4  5 0.001885 0.014120  1.5  0.0    0   0  1.0  8   15   0   0  0  0
 1.0  0.0
12  4  5 0.003350 0.014120  1.5  0.0    0   0  1.0  2   14   0   0  0  0
 1.0  0.0
13  5  6 0.013401 0.021440  1.5  0.0    0   0  1.0  6   16   9   16  15  16
 4.0  0.0
```

```
14   5  6 0.009004 0.021440  1.5  0.0    0    0  1.0  10   17   12   17  -1  17
 4.0  0.0
15   5  6 0.011935 0.021440  1.5  0.0    0    0  1.0   3   -1   11   -1  13  -1
 4.0  0.0
16   6  7 0.023970 0.056744  1.5  0.0    0    0  1.0  13   18   -1   18   0   0
 7.0  0.0
17   6  7 0.023970 0.056744  1.5  0.0    0    0  1.0  14   19   -1   19   0   0
 7.0  0.0
18   7  8 0.044069 0.084709  1.5  0.0    0    0  1.0  16   20   -1   20   0   0
13.0  0.0
19   7  8 0.044069 0.084709  1.5  0.0    0    0  1.0  17   21   -1   21   0   0
13.0  0.0
20   8  9 0.071120 0.467360  1.5  0.0    0    0  1.0  18   -1   -1   -1   0   0
20.0  0.0
21   8  9 0.071120 0.467360  1.5  0.0    0    0  1.0  19   -1   -1   -1   0   0
20.0  0.0
*NMGP
   0
*******************************************************************************
* END GROUP 3                                                                  *
*******************************************************************************
*
*
*******************************************************************************
* Group 4 - Vertical Channel Connection Data

      *
*******************************************************************************
*NGRP
   4
* NSEC NSIM IREB NDM4 NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
    1   1   0   0   0   0   0   0   0   0   0   0   0   0
*ISEC NCHN NONO     DXS IVAR
   1   9  40 .060324    0
*   I KCHA KCHA KCHA KCHA KCHA KCHA KCHB KCHB KCHB KCHB KCHB KCHB
   1   1   0   0   0   0   0   1   0   0   0   0   0
   2   2   0   0   0   0   0   2   0   0   0   0   0
   3   3   0   0   0   0   0   3   0   0   0   0   0
   4   4   0   0   0   0   0   4   0   0   0   0   0
   5   5   0   0   0   0   0   5   0   0   0   0   0
   6   6   0   0   0   0   0   6   0   0   0   0   0
   7   7   0   0   0   0   0   7   0   0   0   0   0
   8   8   0   0   0   0   0   8   0   0   0   0   0
   9   9   0   0   0   0   0   9   0   0   0   0   0
*IWDE
   9
*MSIM
  360
*******************************************************************************
* END GROUP 4                                                                  *
*******************************************************************************
*
*
*******************************************************************************
* Group 7 - Local loss coefficient and Grid spacer data
*
*******************************************************************************
*NGRP
   7
* NCD  NGT IFGQ IFSD IFES IFTP GTMP NFBS NDM9 NM10 NM11 NM12 NM13 NM14
  25   0   0   0   0   0   0   0   0   0   0   0   0   0
* CDL    J ICDM ICDM ICDM ICDM ICDM ICDM ICDM ICDM ICDM ICDM ICDM
 1.13   2   1   2   4   0   0   0   0   0   0   0   0
 1.04   2   3   7   0   0   0   0   0   0   0   0   0
 1.10   2   5   0   0   0   0   0   0   0   0   0   0
 0.94   2   6   0   0   0   0   0   0   0   0   0   0
 1.02   2   8   9   0   0   0   0   0   0   0   0   0
*
```

```
 0.91  11   1   2   4   0   0   0   0   0   0   0   0   0
 0.68  11   3   0   0   0   0   0   0   0   0   0   0   0
 0.81  11   5   0   0   0   0   0   0   0   0   0   0   0
 0.85  11   6   0   0   0   0   0   0   0   0   0   0   0
 0.79  11   7   8   9   0   0   0   0   0   0   0   0   0
*
 0.91  19   1   2   4   0   0   0   0   0   0   0   0   0
 0.68  19   3   0   0   0   0   0   0   0   0   0   0   0
 0.81  19   5   0   0   0   0   0   0   0   0   0   0   0
 0.85  19   6   0   0   0   0   0   0   0   0   0   0   0
 0.79  19   7   8   9   0   0   0   0   0   0   0   0   0
*
 0.91  27   1   2   4   0   0   0   0   0   0   0   0   0
 0.68  27   3   0   0   0   0   0   0   0   0   0   0   0
 0.81  27   5   0   0   0   0   0   0   0   0   0   0   0
 0.85  27   6   0   0   0   0   0   0   0   0   0   0   0
 0.79  27   7   8   9   0   0   0   0   0   0   0   0   0
*
 0.91  36   1   2   4   0   0   0   0   0   0   0   0   0
 0.68  36   3   7   0   0   0   0   0   0   0   0   0   0
 0.81  36   5   0   0   0   0   0   0   0   0   0   0   0
 0.85  36   6   0   0   0   0   0   0   0   0   0   0   0
 0.79  36   8   9   0   0   0   0   0   0   0   0   0   0
*
*******************************************************************************
* END GROUP 7                                                                  *
*******************************************************************************
*
*
*******************************************************************************
* Group 8 - Rod and Unheated Conductor Data

     *
*******************************************************************************
*NGRP
   8
*NRRD NSRD   NC NRTB NRAD NLTY NSTA  NXF NCAN RADF   W3 NM12 NM13 NM14
   9   0   1   1   0   0   0   0   0   0   1   0   0   0
***** Group 8.2
*   N IFTY IAXP NRND    DAXMIN     RMULT      HGAP ISEC HTMB TAMB
   1   1   1   0      0.0       1.    5678.3    1  0.0  0.0
***** Group 8.3
*NSCH  PIE NSCH  PIE NSCH  PIE NSCH  PIE NSCH  PIE NSCH  PIE NSCH  PIE NSCH  PIE
   1 .744   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0
***** Group 8.2 and 8.3 continued
   2   1   1   0      0.0       1.    5678.3    1  0.0  0.0
   2 .745   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0
   3   1   1   0      0.0       1.    5678.3    1  0.0  0.0
   3 .989   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0
   4   1   1   0      0.0       1.    5678.3    1  0.0  0.0
   4 .747   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0
   5   1   1   0      0.0      12.    5678.3    1  0.0  0.0
   5 .833   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0
   6   1   1   0      0.0      33.    5678.3    1  0.0  0.0
   6 .962   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0
   7   1   1   0      0.0     120.    5678.3    1  0.0  0.0
   7 .892   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0
   8   1   1   0      0.0     272.    5678.3    1  0.0  0.0
   8 .919   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0
   9   1   1   0      0.0    5488.    5678.3    1  0.0  0.0
   9 .913   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0   0  0.0
***** Group 8.6
*   I NRT1 NST1 NRX1
   1   9   0   2
***** Group 8.7
*IRTB
   1   2   3   4   5   6   7   8   9
***** Group 8.9 Initial heater rod temperature profile
```

```
*   AXIALT    TRINIT
      0.0    280.00
    2.413    280.00
*******************************************************************************
* END GROUP 8                                                                  *
*******************************************************************************
*
*
*******************************************************************************
* Group 9 - Conductor Geometry Description

      *
*******************************************************************************
*NGRP
   9
*NFLT IRLF ICNF IMWR NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
   1   0   0   0   0   0   0   0   0   0   0   0   0   0
*   I FTYP      DROD       DIN NFUL IMTF IMTC  DM8  DM9 DM10 DM11 DM12 DM13 DM14
   1 nucl  0.0118  0.0095   10    0    0    0 0.0 0.00105   0.94    0    0    0
*    1 tube  .0107696  .0107442    1    0    0    0    0    0    0    0    0    0
*NDER MATR     TREG QREG
*    2    1 .0000127  1.0
*******************************************************************************
* END GROUP 9                                                                  *
*******************************************************************************
*
*
*******************************************************************************
* GROUP 10 - Material Properties Tables                                        *
*******************************************************************************
*NGRP
  10
*NMAT NDM2 NDM3 NDM4 NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
   1   0   0   0   0   0   0   0   0   0   0   0   0   0
*   N NTDP     RCOLD                    IMATAN
   1    6   8470.57                Inconel 600
*    TPROP     CPF1      THCF
      -73    0.377     13.40
       93    0.464     15.71
      204    0.485     17.44
      427    0.527     20.90
      649    0.586     24.79
      871    0.623     28.83
*
*******************************************************************************
* END GROUP 10                                                                 *
*******************************************************************************
*
*
*******************************************************************************
* Group 11 - Axial Power Tables and Forcing Functions
                                                                               *
*******************************************************************************
*NGRP
  11
* NQA NAXP MNXN   NQ NGPF  NQR NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
   1   1  30    4    0    1    0    0    0    0    0    0    0    0
*
* Axial Power Distribution/Forcing Functions
*       YQ
      0.0
*   I NAXN
   1   30
*        Y     AXIAL
   1.0000    0.0339
   0.0820    0.1989
   0.1665    0.3617
   0.2485    0.5203
```

```
   0.3330   0.6730
   0.4150   0.8181
   0.4995   0.9538
   0.5815   1.0786
   0.6660   1.1911
   0.7480   1.2901
   0.8325   1.3743
   0.9145   1.4429
   0.9990   1.4951
   1.0810   1.5301
   1.1655   1.5478
   1.2475   1.5478
   1.3320   1.5301
   1.4140   1.4951
   1.4985   1.4429
   1.5805   1.3743
   1.6650   1.2901
   1.7470   1.1911
   1.8315   1.0786
   1.9135   0.9538
   1.9980   0.8181
   2.0800   0.6730
   2.1645   0.5203
   2.2465   0.3617
   2.3310   0.1989
   2.4130   0.0339
*
* Total Power Forcing Functions
*      YQ       FQ       YQ       FQ       YQ       FQ       YQ       FQ
       0.0   0.0000      1.0   0.1000      2.0   1.0000     100.0   1.0000
*      11.8   1.0000     11.9   0.9815     12.0   0.9629     12.1   0.9444
*      12.2   0.9259     12.3   0.9073     12.4   0.8888     12.5   0.8702
*      12.6   0.8517     12.7   0.8303     12.8   0.8096     12.9   0.7908
*      13.0   0.7734     13.1   0.7571     13.2   0.7418     13.3   0.7273
*      13.4   0.7135     13.5   0.7003     13.6   0.6877     13.7   0.6756
*      13.8   0.6639     14.8   0.5638     15.8   0.4835     16.8   0.4146
*      17.8   0.3546     18.8   0.3029     19.8   0.2589     20.8   0.2226
*      21.8   0.2162
* Radial Power Distribution/Forcing Functions
*      YQR
       0.0
*      FQR      FQR      FQR      FRQ      FQR      FRQ      FQR      FRQ
   1.55769   1.55769   1.55769   1.55769   1.52627   1.50229   1.51086   1.51087
   0.95895
*********************************************************************************
* END GROUP 11                                                                  *
*********************************************************************************
*
*
*********************************************************************************
*GROUP 12 - Turbulent mixing data                                               *
*********************************************************************************
*NGRP
  12
* Card 12.1
* AAAK   BETA
  1.0  0.051
*********************************************************************************
* END GROUP 12                                                                  *
*********************************************************************************
*
*
*********************************************************************************
* Group 13 - Boundary Conditions Data

             *
*********************************************************************************
*NGRP
```

```
  13
*NBND NKBD NFUN NGBD NDM5 NDM6 NDM7 NDM8 NDM9 NM10 NM11 NM12 NM13 NM14
  18   0   1   0   0   0   0   0   0   0   0   0   0   0
*NPTS
  32
*
*ABSC    ORDINT ABSC    ORDINT ABSC    ORDINT
 0.0    0.0000  0.1    1.0000 10.0    1.0000 10.1    0.9976 10.2    0.9929
10.3    0.9867 10.4    0.9795 10.5    0.9716 10.6    0.9632 10.7    0.9547
10.8    0.9459 10.9    0.9371 11.0    0.9283 11.1    0.9196 11.2    0.9109
11.3    0.9023 11.4    0.8938 11.5    0.8855 11.6    0.8773 11.7    0.8693
11.8    0.8615 11.9    0.8537 12.0    0.8462 13.0    0.7817 14.0    0.7363
15.0    0.6960 16.0    0.6593 17.0    0.6259 18.0    0.5953 19.0    0.5672
20.0    0.5413 21.8    0.5413
*
* Inlet b.c. ----------------------------------------------------------
*IBD1 IBD2 ISPC N1FN N2FN N3FN  BCVALUE1  BCVALUE2  BCVALUE3   INITGAS
   1    1    2    1    0    0   .24961  1251.255       0.0         1
   2    1    2    1    0    0   .24961  1251.255       0.0         1
   3    1    2    1    0    0   .28699  1251.255       0.0         1
   4    1    2    1    0    0   .24961  1251.255       0.0         1
   5    1    2    1    0    0   3.1446  1251.255       0.0         1
   6    1    2    1    0    0   9.5544  1251.255       0.0         1
   7    1    2    1    0    0   32.872  1251.255       0.0         1
   8    1    2    1    0    0   75.777  1251.255       0.0         1
   9    1    2    1    0    0  1522.97  1251.255       0.0         1
*
* Outlet b.c. ----------------------------------------------------------
*IBD1 IBD2 ISPC N1FN N2FN N3FN  BCVALUE1  BCVALUE2  BCVALUE3
   1   42    1    0    0    0       0.0  1251.255  156.3042         1
   2   42    1    0    0    0       0.0  1251.255  156.3042         1
   3   42    1    0    0    0       0.0  1251.255  156.3042         1
   4   42    1    0    0    0       0.0  1251.255  156.3042         1
   5   42    1    0    0    0       0.0  1251.255  156.3042         1
   6   42    1    0    0    0       0.0  1251.255  156.3042         1
   7   42    1    0    0    0       0.0  1251.255  156.3042         1
   8   42    1    0    0    0       0.0  1251.255  156.3042         1
   9   42    1    0    0    0       0.0  1251.255  156.3042         1
*******************************************************************************
* END GROUP 13                                                                 *
*******************************************************************************
*
*
*******************************************************************************
* Group 14 - Output Options


     *
*******************************************************************************
*NGRP
  14
*  N1 NOU1 NOU2 NOU3 NOU4 IPRP IOPT IRWR NDM9 NM10 NM11 NM12 NM13 NM14
   5   0   0   0   0   0   1   1   0   0   0   0   0   0
*PRCH
*
*PRTG
*
*PRTR
*
*******************************************************************************
* END GROUP 14                                                                 *
*******************************************************************************
*
*
*******************************************************************************
* Group 15 - IME DOMAIN DATA                                                   *
*******************************************************************************
*NGRP
```

```
  15
*   DTMIN  DTMAX  TEND  EDINT  DMPINT  RTWFP
 0.000001   0.01   9.8   5.0   10.0    1.0   0.1
 0.000001   0.001  9.9  0.1    10.0    1.0   0.1
*   DTMIN  DTMAX  TEND  EDINT  DMPINT  RTWFP
 0.000001   0.001  32.0  0.01    44.8    1.0
*   DTMIN (if negative stop)
    -1.0    0.0   0.0    0.0     0.0    0.0
*
*********************************************************************************
* END GROUP 15                                                                  *
*********************************************************************************
```

Appendix B

TRACE input deck

```
free format
*
*
*************
* main data *
*************
*
*       numtcr          ieos         inopt          nmat          id2o
            1             0             1             0             0

*
*
******************
* namelist data *
******************
*
 &inopts
  dtstrt=1.0,
  ikfac=1,
  nosets=1,
  usesjc=3,
  npower=1,
  nhtstr=9
 &end
*
****************
* Model Flags *
****************
*
*        dstep         timet
             0           0.0
*       stdyst        transi         ncomp          njun          ipak
             0             1            37            18             1
*         epso          epss
         1.0E-4        1.0E-4
*       oitmax        sitmax        isolut        ncontr         nccfl
            10            10             0             0             0
*         ntsv          ntcb          ntcf          ntrp          ntcp
             1             0             0             0             0
*
****************************
* component-number data *
****************************
*
*  Component input order (IORDER)
*-- type ---- num ------------ name ----------------  +    jun1   jun2   jun3
* PIPE     *      1 s * Channel 1                    +      1     10
* PIPE     *      2 s * Channel 2                    +      2     11
* PIPE     *      3 s * Channel 3                    +      3     12
* PIPE     *      4 s * Channel 4                    +      4     13
* PIPE     *      5 s * Channel 5                    +      5     14
* PIPE     *      6 s * Channel 6                    +      6     15
* PIPE     *      7 s * Channel 7                    +      7     16
* PIPE     *      8 s * Channel 8                    +      8     17
* PIPE     *      9 s * Channel 9                    +      9     18
* FILL     *     11 s * Channel 1 Fill               +      1
* FILL     *     21 s * Channel 2 Fill               +      2
* FILL     *     31 s * Channel 3 Fill               +      3
```

```
* FILL     *     41 s * Channel 4 Fill              +      4
* FILL     *     51 s * Channel 5 Fill              +      5
* FILL     *     61 s * Channel 6 Fill              +      6
* FILL     *     71 s * Channel 7 Fill              +      7
* FILL     *     81 s * Channel 8 Fill              +      8
* FILL     *     91 s * Channel 9 Fill              +      9
* BREAK    *    101 s * Channel 1 Break              +     10
* BREAK    *    111 s * Channel 2 Fill             +     11
* BREAK    *    121 s * Channel 3 Break              +     12
* BREAK    *    131 s * Channel 4 Fill             +     13
* BREAK    *    141 s * Channel 5 Break              +     14
* BREAK    *    151 s * Channel 6 Break              +     15
* BREAK    *    161 s * Channel 7 Break              +     16
* BREAK    *    171 s * Channel 8 Break              +     17
* BREAK    *    181 s * Channel 9 Break              +     18
* HTSTR    *    191 s * Ch 9 Heat Structure          +
* POWER    *    201 s * Ch 9 Power                 +
* HTSTR    *    211 s * Ch 8 Heat Structure          +
* HTSTR    *    221 s * Ch 7 Heat Structure          +
* HTSTR    *    231 s * Ch 6 Heat Structure          +
* HTSTR    *    241 s * Ch 5 Heat Structure          +
* HTSTR    *    251 s * Ch 4 Heat Structure          +
* HTSTR    *    261 s * Ch 3 Heat Structure          +
* HTSTR    *    271 s * Ch 2 Heat Structure          +
* HTSTR    *    281 e * Ch 1 Heat Structure          +
*
*
**************************************************
* Starting Signal Variable Section of Model      *
**************************************************
*
*       idsv       isvn       ilcn       icn1       icn2
          1          0          0          0          0
**************************************************
* Finished Signal Variable Section of Model      *
**************************************************
*
*
*
*
*******   type         num       userid            component name
pipe                    1          1                  Channel 1
*      ncells       nodes       jun1       jun2       epsw
          41          0          1         10        0.0
*      nsides
          0
*       ichf       iconc     pipetype       ipow       npipes
          1          0          0          0          1
*       radin         th        houtl       houtv       toutl
         0.0         0.0         0.0         0.0         0.0
*       toutv        pwin        pwoff       rpwmx       pwscl
         0.0         0.0         0.0         0.0         0.0
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659e
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
```

```
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6e
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5e
* kfac  *         0.0        1.13        0.0        0.0s
* kfac  *         0.0         0.0        0.0        0.0s
* kfac  *         0.0         0.0       0.91        0.0s
* kfac  *         0.0         0.0        0.0        0.0s
* kfac  *         0.0         0.0       0.91        0.0s
* kfac  *         0.0         0.0        0.0        0.0s
* kfac  *         0.0         0.0       0.91        0.0s
* kfac  *         0.0         0.0        0.0        0.0s
* kfac  *         0.0         0.0        0.0       0.91s
* kfac  *         0.0         0.0        0.0        0.0s
* kfac  *         0.0         0.0e
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0        1.0        1.0s
* grav  *         1.0         1.0e
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403e
* nff   *           1           1          1          1s
* nff   *           1           1          1          1s
* nff   *           1           1          1          1s
* nff   *           1           1          1          1s
* nff   *           1           1          1          1s
* nff   *           1           1          1          1s
* nff   *           1           1          1          1s
* nff   *           1           1          1          1s
* nff   *           1           1          1          1s
* nff   *           1           1          1          1s
* nff   *           1           1e
* alp   *         0.0         0.0        0.0        0.0s
* alp   *         0.0         0.0        0.0        0.0s
* alp   *         0.0         0.0        0.0        0.0s
* alp   *         0.0         0.0        0.0        0.0s
* alp   *         0.0         0.0        0.0        0.0s
* alp   *         0.0         0.0        0.0        0.0s
* alp   *         0.0         0.0        0.0        0.0s
```

```
* alp   *        0.0        0.0        0.0        0.0s
* alp   *        0.0        0.0        0.0        0.0s
* alp   *        0.0        0.0        0.0        0.0s
* alp   *        0.0e
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5        3.5        3.5s
* vl    *       3.5        3.5e
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0e
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77e
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0        0.0        0.0        0.0s
* tv    *       0.0e
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7e
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
```

```
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0e
*
*
*******   type         num       userid              component name
pipe                     2           1                    Channel 2
*      ncells        nodes        jun1        jun2        epsw
         41            0           2          11          0.0
*      nsides
          0
*        ichf        iconc      pipetype        ipow       npipes
          1            0           0           0           1
*       radin           th        houtl        houtv        toutl
         0.0          0.0          0.0          0.0          0.0
*       toutv         pwin        pwoff        rpwmx        pwscl
         0.0          0.0          0.0          0.0          0.0
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659e
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6e
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5e
* kfac  *        0.0        1.13        0.0        0.0s
* kfac  *        0.0         0.0        0.0        0.0s
* kfac  *        0.0         0.0        0.91       0.0s
* kfac  *        0.0         0.0        0.0        0.0s
* kfac  *        0.0         0.0        0.91       0.0s
* kfac  *        0.0         0.0        0.0        0.0s
* kfac  *        0.0         0.0        0.91       0.0s
* kfac  *        0.0         0.0        0.0        0.0s
* kfac  *        0.0         0.0        0.0        0.91s
* kfac  *        0.0         0.0        0.0        0.0s
* kfac  *        0.0         0.0e
* grav  *        1.0         1.0        1.0        1.0s
* grav  *        1.0         1.0        1.0        1.0s
* grav  *        1.0         1.0        1.0        1.0s
* grav  *        1.0         1.0        1.0        1.0s
* grav  *        1.0         1.0        1.0        1.0s
* grav  *        1.0         1.0        1.0        1.0s
* grav  *        1.0         1.0        1.0        1.0s
* grav  *        1.0         1.0        1.0        1.0s
```

```
* grav  *       1.0       1.0       1.0       1.0s
* grav  *       1.0       1.0       1.0       1.0s
* grav  *       1.0       1.0e
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403  0.010403  0.010403s
* hd    *   0.010403  0.010403e
* nff   *        1         1         1         1s
* nff   *        1         1         1         1s
* nff   *        1         1         1         1s
* nff   *        1         1         1         1s
* nff   *        1         1         1         1s
* nff   *        1         1         1         1s
* nff   *        1         1         1         1s
* nff   *        1         1         1         1s
* nff   *        1         1         1         1s
* nff   *        1         1         1         1s
* nff   *        1         1e
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0       0.0       0.0       0.0s
* alp   *      0.0e
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5       3.5       3.5s
* vl    *      3.5       3.5e
* vv    *      0.0       0.0       0.0       0.0s
* vv    *      0.0       0.0       0.0       0.0s
* vv    *      0.0       0.0       0.0       0.0s
* vv    *      0.0       0.0       0.0       0.0s
* vv    *      0.0       0.0       0.0       0.0s
* vv    *      0.0       0.0       0.0       0.0s
* vv    *      0.0       0.0       0.0       0.0s
* vv    *      0.0       0.0       0.0       0.0s
* vv    *      0.0       0.0       0.0       0.0s
* vv    *      0.0       0.0e
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
```

```
* tl    *       556.77e
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0        0.0        0.0        0.0s
* tv    *          0.0e
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7    1.563E7    1.563E7    1.563E7s
* p     *      1.563E7e
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0        0.0        0.0        0.0s
* pa    *          0.0e
*
*
*******   type          num       userid              component name
pipe                      3          1                    Channel 3
*       ncells        nodes        jun1        jun2        epsw
           41           0           3          12          0.0
*       nsides
            0
*          ichf        iconc     pipetype        ipow       npipes
            1           0           0           0           1
*         radin          th        houtl       houtv        toutl
           0.0         0.0         0.0         0.0         0.0
*         toutv        pwin        pwoff       rpwmx        pwscl
           0.0         0.0         0.0         0.0         0.0
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659e
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
```

```
* vol   *    6.3562E-6   6.3562E-6   6.3562E-6   6.3562E-6s
* vol   *    6.3562E-6e
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4     1.08E-4     1.08E-4s
* fa    *     1.08E-4     1.08E-4e
* kfac  *        0.0       1.04        0.0        0.0s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0       0.68        0.0s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0       0.68        0.0s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0       0.68        0.0s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0        0.0       0.68s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0e
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0e
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686   0.0127686   0.0127686s
* hd    *    0.0127686   0.0127686e
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1e
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0e
```

```
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5e
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0e
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77    556.77    556.77    556.77s
* tl    *    556.77e
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0e
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7   1.563E7   1.563E7   1.563E7s
* p     *    1.563E7e
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0e
*
*
```

```
*******   type          num       userid              component name
pipe                    4           1                   Channel 4
*      ncells        nodes        jun1        jun2        epsw
         41            0            4          13          0.0
*      nsides
          0
*        ichf        iconc      pipetype        ipow       npipes
          1            0            0            0            1
*       radin           th         houtl        houtv        toutl
         0.0          0.0          0.0          0.0          0.0
*       toutv         pwin        pwoff        rpwmx        pwscl
         0.0          0.0          0.0          0.0          0.0
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659e
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6  5.53224E-6  5.53224E-6  5.53224E-6s
* vol   *   5.53224E-6e
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5      9.4E-5      9.4E-5s
* fa    *      9.4E-5      9.4E-5e
* kfac  *        0.0        1.13        0.0         0.0s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0        0.91        0.0s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0        0.91        0.0s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0        0.91        0.0s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0        0.0         0.91s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0e
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0e
* hd    *    0.010403    0.010403    0.010403    0.010403s
```

```
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403    0.010403    0.010403    0.010403s
* hd    *    0.010403 0.010940042e
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1e
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0e
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5e
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0e
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77e
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
```

```
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0e
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7    1.563E7    1.563E7    1.563E7s
* p     *     1.563E7e
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0e
*
*
*******   type        num       userid            component name
pipe                   5           1                   Channel 5
*      ncells      nodes       jun1       jun2       epsw
         41          0          5          14         0.0
*      nsides
          0
*       ichf       iconc     pipetype       ipow       npipes
          1          0          0          0          1
*       radin        th        houtl       houtv       toutl
         0.0        0.0        0.0        0.0        0.0
*       toutv       pwin       pwoff       rpwmx       pwscl
         0.0        0.0        0.0        0.0        0.0
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659e
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5  6.98004E-5  6.98004E-5  6.98004E-5s
* vol   *   6.98004E-5e
* fa    *    1.186E-3    1.186E-3    1.186E-3    1.186E-3s
* fa    *    1.186E-3    1.186E-3    1.186E-3    1.186E-3s
```

```
* fa   *   1.186E-3   1.186E-3   1.186E-3   1.186E-3s
* fa   *   1.186E-3   1.186E-3   1.186E-3   1.186E-3s
* fa   *   1.186E-3   1.186E-3   1.186E-3   1.186E-3s
* fa   *   1.186E-3   1.186E-3   1.186E-3   1.186E-3s
* fa   *   1.186E-3   1.186E-3   1.186E-3   1.186E-3s
* fa   *   1.186E-3   1.186E-3   1.186E-3   1.186E-3s
* fa   *   1.186E-3   1.186E-3   1.186E-3   1.186E-3s
* fa   *   1.186E-3   1.186E-3   1.186E-3   1.186E-3s
* fa   *   1.186E-3   1.186E-3e
* kfac *       0.0       1.1       0.0       0.0s
* kfac *       0.0       0.0       0.0       0.0s
* kfac *       0.0       0.0      0.81       0.0s
* kfac *       0.0       0.0       0.0       0.0s
* kfac *       0.0       0.0      0.81       0.0s
* kfac *       0.0       0.0       0.0       0.0s
* kfac *       0.0       0.0      0.81       0.0s
* kfac *       0.0       0.0       0.0       0.0s
* kfac *       0.0       0.0       0.0      0.81s
* kfac *       0.0       0.0       0.0       0.0s
* kfac *       0.0       0.0e
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0       1.0       1.0s
* grav *       1.0       1.0e
* hd   *  0.01117724  0.01117724  0.01117724  0.01117724s
* hd   *  0.01117724  0.01117724  0.01117724  0.01117724s
* hd   *  0.01117724  0.01117724  0.01117724  0.01117724s
* hd   *  0.01117724  0.01117724  0.01117724  0.01117724s
* hd   *  0.01117724  0.01117724  0.01117724  0.01117724s
* hd   *  0.01117724  0.01117724  0.01117724  0.01117724s
* hd   *  0.01117724  0.01117724  0.01117724  0.01117724s
* hd   *  0.01117724  0.01117724  0.01117724  0.01117724s
* hd   *  0.01117724  0.01117724  0.01117724  0.01117724s
* hd   *  0.01117724  0.01117724e
* nff  *         1         1         1         1s
* nff  *         1         1         1         1s
* nff  *         1         1         1         1s
* nff  *         1         1         1         1s
* nff  *         1         1         1         1s
* nff  *         1         1         1         1s
* nff  *         1         1         1         1s
* nff  *         1         1         1         1s
* nff  *         1         1         1         1s
* nff  *         1         1         1         1s
* nff  *         1         1e
* alp  *       0.0       0.0       0.0       0.0s
* alp  *       0.0       0.0       0.0       0.0s
* alp  *       0.0       0.0       0.0       0.0s
* alp  *       0.0       0.0       0.0       0.0s
* alp  *       0.0       0.0       0.0       0.0s
* alp  *       0.0       0.0       0.0       0.0s
* alp  *       0.0       0.0       0.0       0.0s
* alp  *       0.0       0.0       0.0       0.0s
* alp  *       0.0       0.0       0.0       0.0s
* alp  *       0.0       0.0e
* vl   *       3.5       3.5       3.5       3.5s
* vl   *       3.5       3.5       3.5       3.5s
* vl   *       3.5       3.5       3.5       3.5s
* vl   *       3.5       3.5       3.5       3.5s
```

```
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5e
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0e
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77e
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0e
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7   1.563E7   1.563E7   1.563E7s
* p     *     1.563E7e
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0e
*
*
*******   type        num       userid           component name
pipe                  6         1                Channel 6
*      ncells      nodes      jun1       jun2       epsw
         41          0          6         15         0.0
```

```
*      nsides
          0
*        ichf        iconc      pipetype         ipow       npipes
           1            0            0            0            1
*       radin           th         houtl        houtv        toutl
          0.0          0.0          0.0          0.0          0.0
*       toutv         pwin         pwoff        rpwmx        pwscl
          0.0          0.0          0.0          0.0          0.0
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659e
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4  2.12226E-4  2.12226E-4  2.12226E-4s
* vol   *   2.12226E-4e
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3    3.606E-3    3.606E-3s
* fa    *    3.606E-3    3.606E-3e
* kfac  *         0.0        0.94         0.0         0.0s
* kfac  *         0.0         0.0         0.0         0.0s
* kfac  *         0.0         0.0        0.85         0.0s
* kfac  *         0.0         0.0         0.0         0.0s
* kfac  *         0.0         0.0        0.85         0.0s
* kfac  *         0.0         0.0         0.0         0.0s
* kfac  *         0.0         0.0        0.85         0.0s
* kfac  *         0.0         0.0         0.0         0.0s
* kfac  *         0.0         0.0         0.0        0.85s
* kfac  *         0.0         0.0         0.0         0.0s
* kfac  *         0.0         0.0e
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0         1.0         1.0s
* grav  *         1.0         1.0e
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
```

```
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
* hd    *   0.01278712  0.01278712  0.01278712  0.01278712s
* hd    *   0.01278712  0.01278712e
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1e
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0e
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5e
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0e
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77e
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
```

```
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0e
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7e
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0       0.0       0.0       0.0s
* pa    *       0.0e
*
*
*******   type       num     userid          component name
pipe                   7          1                Channel 7
*      ncells      nodes      jun1      jun2      epsw
          41          0          7         16         0.0
*      nsides
           0
*        ichf      iconc    pipetype      ipow      npipes
           1          0          0          0          1
*        radin        th       houtl      houtv      toutl
          0.0        0.0        0.0        0.0        0.0
*        toutv       pwin      pwoff      rpwmx      pwscl
          0.0        0.0        0.0        0.0        0.0
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659e
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4  7.30138E-4  7.30138E-4  7.30138E-4s
* vol   *   7.30138E-4e
* fa    *    0.012406    0.012406    0.012406    0.012406s
* fa    *    0.012406    0.012406    0.012406    0.012406s
* fa    *    0.012406    0.012406    0.012406    0.012406s
* fa    *    0.012406    0.012406    0.012406    0.012406s
* fa    *    0.012406    0.012406    0.012406    0.012406s
* fa    *    0.012406    0.012406    0.012406    0.012406s
```

```
* fa    *    0.012406    0.012406    0.012406    0.012406s
* fa    *    0.012406    0.012406    0.012406    0.012406s
* fa    *    0.012406    0.012406    0.012406    0.012406s
* fa    *    0.012406    0.012406    0.012406    0.012406s
* fa    *    0.012406    0.012406e
* kfac  *       0.0       1.04        0.0         0.0s
* kfac  *       0.0        0.0        0.0         0.0s
* kfac  *       0.0        0.0       0.79         0.0s
* kfac  *       0.0        0.0        0.0         0.0s
* kfac  *       0.0        0.0       0.79         0.0s
* kfac  *       0.0        0.0        0.0         0.0s
* kfac  *       0.0        0.0       0.79         0.0s
* kfac  *       0.0        0.0        0.0         0.0s
* kfac  *       0.0        0.0        0.0        0.79s
* kfac  *       0.0        0.0        0.0         0.0s
* kfac  *       0.0        0.0e
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0        1.0         1.0s
* grav  *       1.0        1.0e
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983   0.0118983   0.0118983s
* hd    *    0.0118983   0.0118983e
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1          1          1s
* nff   *        1          1e
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0        0.0        0.0         0.0s
* alp   *       0.0e
* vl    *       3.5        3.5        3.5         3.5s
* vl    *       3.5        3.5        3.5         3.5s
* vl    *       3.5        3.5        3.5         3.5s
* vl    *       3.5        3.5        3.5         3.5s
* vl    *       3.5        3.5        3.5         3.5s
* vl    *       3.5        3.5        3.5         3.5s
* vl    *       3.5        3.5        3.5         3.5s
* vl    *       3.5        3.5        3.5         3.5s
```

```
* vl    *        3.5        3.5        3.5        3.5s
* vl    *        3.5        3.5        3.5        3.5s
* vl    *        3.5        3.5e
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0e
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77e
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0e
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7e
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0e
*
*
*******   type      num     userid          component name
pipe                  8        1               Channel 8
*     ncells     nodes     jun1      jun2      epsw
        41         0         8        17        0.0
*     nsides
         0
*       ichf      iconc    pipetype     ipow     npipes
         1         0          0          0         1
```

```
*       radin          th        houtl        houtv        toutl
         0.0          0.0          0.0          0.0          0.0
*       toutv         pwin        pwoff        rpwmx        pwscl
         0.0          0.0          0.0          0.0          0.0
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx    *  0.058853659e
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3   1.6828E-3   1.6828E-3   1.6828E-3s
* vol   *    1.6828E-3e
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593    0.028593    0.028593s
* fa    *    0.028593    0.028593e
* kfac  *        0.0        1.02        0.0         0.0s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0       0.79         0.0s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0       0.79         0.0s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0       0.79         0.0s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0        0.0        0.79s
* kfac  *        0.0         0.0        0.0         0.0s
* kfac  *        0.0         0.0e
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0        1.0         1.0s
* grav  *        1.0         1.0e
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
```

```
* hd    *  0.012182994 0.012182994 0.012182994 0.012182994s
* hd    *  0.012182994 0.012182994e
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1         1         1s
* nff   *         1         1e
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0       0.0       0.0       0.0s
* alp   *       0.0e
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5       3.5       3.5s
* vl    *       3.5       3.5e
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0       0.0       0.0s
* vv    *       0.0       0.0e
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77    556.77    556.77    556.77s
* tl    *     556.77e
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0       0.0       0.0       0.0s
* tv    *       0.0e
```

```
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7    1.563E7    1.563E7    1.563E7s
* p    *     1.563E7e
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0        0.0        0.0        0.0s
* pa   *         0.0e
*
*
*******   type          num       userid              component name
pipe                    9           1                    Channel 9
*      ncells       nodes       jun1        jun2        epsw
          41           0           9          18          0.0
*      nsides
           0
*        ichf        iconc     pipetype        ipow       npipes
           1           0           0           0           1
*       radin          th        houtl       houtv       toutl
          0.0         0.0         0.0         0.0         0.0
*       toutv        pwin        pwoff       rpwmx       pwscl
          0.0         0.0         0.0         0.0         0.0
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659 0.058853659 0.058853659 0.058853659s
* dx   *  0.058853659e
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494 0.033827494 0.033827494 0.033827494s
* vol  *  0.033827494e
* fa   *    0.574773    0.574773    0.574773    0.574773s
* fa   *    0.574773    0.574773    0.574773    0.574773s
* fa   *    0.574773    0.574773    0.574773    0.574773s
* fa   *    0.574773    0.574773    0.574773    0.574773s
* fa   *    0.574773    0.574773    0.574773    0.574773s
* fa   *    0.574773    0.574773    0.574773    0.574773s
* fa   *    0.574773    0.574773    0.574773    0.574773s
* fa   *    0.574773    0.574773    0.574773    0.574773s
* fa   *    0.574773    0.574773    0.574773    0.574773s
* fa   *    0.574773    0.574773    0.574773    0.574773s
```

```
* fa    *     0.574773    0.574773e
* kfac  *        0.0       1.02        0.0        0.0s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0       0.79        0.0s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0       0.79        0.0s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0       0.79        0.0s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0        0.0       0.79s
* kfac  *        0.0        0.0        0.0        0.0s
* kfac  *        0.0        0.0e
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0        1.0        1.0s
* grav  *        1.0        1.0e
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123   0.0121123   0.0121123s
* hd    *   0.0121123   0.0121123e
* nff   *         1          1          1          1s
* nff   *         1          1          1          1s
* nff   *         1          1          1          1s
* nff   *         1          1          1          1s
* nff   *         1          1          1          1s
* nff   *         1          1          1          1s
* nff   *         1          1          1          1s
* nff   *         1          1          1          1s
* nff   *         1          1          1          1s
* nff   *         1          1          1          1s
* nff   *         1          1e
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0        0.0        0.0        0.0s
* alp   *       0.0e
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5        3.5        3.5s
* vl    *      3.5        3.5e
* vv    *       0.0        0.0        0.0        0.0s
```

```
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0        0.0        0.0s
* vv    *        0.0        0.0e
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77     556.77     556.77     556.77s
* tl    *     556.77e
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0        0.0        0.0        0.0s
* tv    *        0.0e
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7    1.563E7    1.563E7    1.563E7s
* p     *    1.563E7e
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0        0.0        0.0        0.0s
* pa    *        0.0e
*
*
*******   type        num      userid          component name
fill               11          1            Channel 1 Fill
*       jun1       ifty       ioff
           1          5          0
*       iftr       ifsv       nftb       nfsv       nfrf
           0          1         32          1          0
*      twtold      rfmx      concin      felv
         0.0     1.0E20        0.0        0.0
*       dxin       volin      alpin       vlin       tlin
  0.058853659  5.532244E-6       0.0        0.0     556.77
*        pin        pain     flowin       vvin       tvin
     1.563E7        0.0    0.24961        0.0        0.0
```

```
*       vmscl        vvscl
         1.0          1.0
*
* vmtbv *        0.0        0.0s
* vmtbv *        0.1     0.24961s
* vmtbv *       10.0     0.24961s
* vmtbv *       10.1  0.24901094s
* vmtbv *       10.2  0.24783777s
* vmtbv *       10.3  0.24629019s
* vmtbv *       10.4    0.244493s
* vmtbv *       10.5  0.24252108s
* vmtbv *       10.6  0.24042435s
* vmtbv *       10.7  0.23830267s
* vmtbv *       10.8   0.2361061s
* vmtbv *       10.9  0.23390953s
* vmtbv *       11.0  0.23171296s
* vmtbv *       11.1  0.22954136s
* vmtbv *       11.2  0.22736975s
* vmtbv *       11.3   0.2252231s
* vmtbv *       11.4  0.22310142s
* vmtbv *       11.5  0.22102966s
* vmtbv *       11.6  0.21898285s
* vmtbv *       11.7  0.21698597s
* vmtbv *       11.8  0.21503902s
* vmtbv *       11.9  0.21309206s
* vmtbv *       12.0  0.21121998s
* vmtbv *       13.0  0.19512014s
* vmtbv *       14.0  0.18378784s
* vmtbv *       15.0  0.17372856s
* vmtbv *       16.0  0.16456787s
* vmtbv *       17.0   0.1562309s
* vmtbv *       18.0  0.14859283s
* vmtbv *       19.0  0.14157879s
* vmtbv *       20.0  0.13511389s
* vmtbv *       21.8  0.13511389e
*
*
*******   type          num       userid                component name
fill                    21            1               Channel 2 Fill
*       jun1         ifty         ioff
           2            5            0
*       iftr        ifsv        nftb        nfsv        nfrf
           0            1           32           0           0
*      twtold         rfmx       concin         felv
         0.0       1.0E20          0.0          0.0
*       dxin        volin        alpin        vlin        tlin
  0.058853659   5.532244E-6          0.0          0.0       556.77
*        pin         pain       flowin         vvin         tvin
     1.563E7          0.0      0.24961          0.0          0.0
*       vmscl        vvscl
         1.0          1.0
*
* vmtbv *        0.0        0.0s
* vmtbv *        0.1     0.24961s
* vmtbv *       10.0     0.24961s
* vmtbv *       10.1  0.24901094s
* vmtbv *       10.2  0.24783777s
* vmtbv *       10.3  0.24629019s
* vmtbv *       10.4    0.244493s
* vmtbv *       10.5  0.24252108s
* vmtbv *       10.6  0.24042435s
* vmtbv *       10.7  0.23830267s
* vmtbv *       10.8   0.2361061s
* vmtbv *       10.9  0.23390953s
* vmtbv *       11.0  0.23171296s
* vmtbv *       11.1  0.22954136s
* vmtbv *       11.2  0.22736975s
* vmtbv *       11.3   0.2252231s
```

```
* vmtbv *       11.4  0.22310142s
* vmtbv *       11.5  0.22102966s
* vmtbv *       11.6  0.21898285s
* vmtbv *       11.7  0.21698597s
* vmtbv *       11.8  0.21503902s
* vmtbv *       11.9  0.21309206s
* vmtbv *       12.0  0.21121998s
* vmtbv *       13.0  0.19512014s
* vmtbv *       14.0  0.18378784s
* vmtbv *       15.0  0.17372856s
* vmtbv *       16.0  0.16456787s
* vmtbv *       17.0   0.1562309s
* vmtbv *       18.0  0.14859283s
* vmtbv *       19.0  0.14157879s
* vmtbv *       20.0  0.13511389s
* vmtbv *       21.8  0.13511389e
*
*
*******   type          num       userid              component name
fill                  31           1              Channel 3 Fill
*        jun1         ifty         ioff
           3            5            0
*        iftr         ifsv         nftb         nfsv         nfrf
           0            1           32            0            0
*       twtold         rfmx        concin         felv
         0.0       1.0E20          0.0          0.0
*        dxin         volin        alpin         vlin         tlin
   0.058853659   6.356195E-6          0.0          0.0       556.77
*         pin         pain        flowin         vvin         tvin
       1.563E7          0.0       0.28699          0.0          0.0
*        vmscl        vvscl
           1.0          1.0
*
* vmtbv *        0.0        0.0s
* vmtbv *        0.1    0.28699s
* vmtbv *       10.0    0.28699s
* vmtbv *       10.1  0.28630122s
* vmtbv *       10.2  0.28495237s
* vmtbv *       10.3  0.28317303s
* vmtbv *       10.4   0.2811067s
* vmtbv *       10.5  0.27883948s
* vmtbv *       10.6  0.27642877s
* vmtbv *       10.7  0.27398935s
* vmtbv *       10.8  0.27146384s
* vmtbv *       10.9  0.26893833s
* vmtbv *       11.0  0.26641282s
* vmtbv *       11.1    0.263916s
* vmtbv *       11.2  0.26141919s
* vmtbv *       11.3  0.25895108s
* vmtbv *       11.4  0.25651166s
* vmtbv *       11.5  0.25412964s
* vmtbv *       11.6  0.25177633s
* vmtbv *       11.7  0.24948041s
* vmtbv *       11.8  0.24724188s
* vmtbv *       11.9  0.24500336s
* vmtbv *       12.0  0.24285094s
* vmtbv *       13.0  0.22434008s
* vmtbv *       14.0  0.21131074s
* vmtbv *       15.0  0.19974504s
* vmtbv *       16.0  0.18921251s
* vmtbv *       17.0  0.17962704s
* vmtbv *       18.0  0.17084515s
* vmtbv *       19.0  0.16278073s
* vmtbv *       20.0  0.15534769s
* vmtbv *       21.8  0.15534769e
*
*
*******   type          num       userid              component name
```

```
fill                41          1               Channel 4 Fill
*       jun1        ifty        ioff
          4         5           0
*       iftr        ifsv        nftb        nfsv        nfrf
          0         1           32          0           0
*      twtold        rfmx       concin       felv
         0.0       1.0E20        0.0         0.0
*       dxin        volin       alpin       vlin        tlin
  0.058853659   5.532244E-6       0.0         0.0       556.77
*       pin         pain       flowin       vvin        tvin
     1.563E7          0.0      0.24961        0.0         0.0
*       vmscl       vvscl
         1.0         1.0
*
* vmtbv *        0.0        0.0s
* vmtbv *        0.1     0.24961s
* vmtbv *       10.0     0.24961s
* vmtbv *       10.1  0.24901094s
* vmtbv *       10.2  0.24783777s
* vmtbv *       10.3  0.24629019s
* vmtbv *       10.4    0.244493s
* vmtbv *       10.5  0.24252108s
* vmtbv *       10.6  0.24042435s
* vmtbv *       10.7  0.23830267s
* vmtbv *       10.8   0.2361061s
* vmtbv *       10.9  0.23390953s
* vmtbv *       11.0  0.23171296s
* vmtbv *       11.1  0.22954136s
* vmtbv *       11.2  0.22736975s
* vmtbv *       11.3   0.2252231s
* vmtbv *       11.4  0.22310142s
* vmtbv *       11.5  0.22102966s
* vmtbv *       11.6  0.21898285s
* vmtbv *       11.7  0.21698597s
* vmtbv *       11.8  0.21503902s
* vmtbv *       11.9  0.21309206s
* vmtbv *       12.0  0.21121998s
* vmtbv *       13.0  0.19512014s
* vmtbv *       14.0  0.18378784s
* vmtbv *       15.0  0.17372856s
* vmtbv *       16.0  0.16456787s
* vmtbv *       17.0   0.1562309s
* vmtbv *       18.0  0.14859283s
* vmtbv *       19.0  0.14157879s
* vmtbv *       20.0  0.13511389s
* vmtbv *       21.8  0.13511389e
*
*
*******   type          num       userid              component name
fill                51          1               Channel 5 Fill
*       jun1        ifty        ioff
          5         5           0
*       iftr        ifsv        nftb        nfsv        nfrf
          0         1           32          0           0
*      twtold        rfmx       concin       felv
         0.0       1.0E20        0.0         0.0
*       dxin        volin       alpin       vlin        tlin
  0.058853659   6.980044E-5       0.0         0.0       556.77
*       pin         pain       flowin       vvin        tvin
     1.563E7          0.0       3.1446        0.0         0.0
*       vmscl       vvscl
         1.0         1.0
*
* vmtbv *        0.0        0.0s
* vmtbv *        0.1     3.1446s
* vmtbv *       10.0     3.1446s
* vmtbv *       10.1   3.137053s
* vmtbv *       10.2   3.1222733s
```

```
* vmtbv *       10.3   3.1027768s
* vmtbv *       10.4   3.0801357s
* vmtbv *       10.5   3.0552934s
* vmtbv *       10.6   3.0288787s
* vmtbv *       10.7   3.0021496s
* vmtbv *       10.8   2.9744771s
* vmtbv *       10.9   2.9468047s
* vmtbv *       11.0   2.9191322s
* vmtbv *       11.1   2.8917742s
* vmtbv *       11.2   2.8644161s
* vmtbv *       11.3   2.8373726s
* vmtbv *       11.4   2.8106435s
* vmtbv *       11.5   2.7845433s
* vmtbv *       11.6   2.7587576s
* vmtbv *       11.7   2.7336008s
* vmtbv *       11.8   2.7090729s
* vmtbv *       11.9    2.684545s
* vmtbv *       12.0   2.6609605s
* vmtbv *       13.0   2.4581338s
* vmtbv *       14.0    2.315369s
* vmtbv *       15.0   2.1886416s
* vmtbv *       16.0   2.0732348s
* vmtbv *       17.0   1.9682051s
* vmtbv *       18.0   1.8719804s
* vmtbv *       19.0   1.7836171s
* vmtbv *       20.0    1.702172s
* vmtbv *       21.8    1.702172e
*
*
*******   type          num       userid              component name
fill                61           1              Channel 6 Fill
*       jun1        ifty        ioff
          6           5           0
*       iftr        ifsv        nftb        nfsv        nfrf
          0           1          32           0           0
*      twtold        rfmx       concin        felv
         0.0      1.0E20         0.0         0.0
*       dxin        volin       alpin       vlin        tlin
  0.058853659   2.122263E-4         0.0         0.0      556.77
*        pin        pain       flowin        vvin        tvin
     1.563E7         0.0       9.5544         0.0         0.0
*       vmscl       vvscl
         1.0         1.0
*
* vmtbv *       0.0       0.0s
* vmtbv *       0.1     9.5544s
* vmtbv *      10.0      9.5544s
* vmtbv *      10.1   9.5314694s
* vmtbv *      10.2   9.4865638s
* vmtbv *      10.3   9.4273265s
* vmtbv *      10.4   9.3585348s
* vmtbv *      10.5    9.283055s
* vmtbv *      10.6   9.2027981s
* vmtbv *      10.7   9.1215857s
* vmtbv *      10.8    9.037507s
* vmtbv *      10.9   8.9534282s
* vmtbv *      11.0   8.8693495s
* vmtbv *      11.1   8.7862262s
* vmtbv *      11.2    8.703103s
* vmtbv *      11.3   8.6209351s
* vmtbv *      11.4   8.5397227s
* vmtbv *      11.5   8.4604212s
* vmtbv *      11.6   8.3820751s
* vmtbv *      11.7   8.3056399s
* vmtbv *      11.8   8.2311156s
* vmtbv *      11.9   8.1565913s
* vmtbv *      12.0   8.0849333s
* vmtbv *      13.0   7.4686745s
```

```
* vmtbv *       14.0   7.0349047s
* vmtbv *       15.0   6.6498624s
* vmtbv *       16.0   6.2992159s
* vmtbv *       17.0    5.980099s
* vmtbv *       18.0   5.6877343s
* vmtbv *       19.0   5.4192557s
* vmtbv *       20.0   5.1717967s
* vmtbv *       21.8   5.1717967e
*
*
*******   type          num       userid              component name
fill                  71           1              Channel 7 Fill
*        jun1         ifty         ioff
           7            5            0
*        iftr         ifsv         nftb         nfsv         nfrf
           0            1           32            0            0
*      twtold         rfmx        concin         felv
         0.0      1.0E20          0.0          0.0
*        dxin         volin        alpin         vlin         tlin
  0.058853659   7.301385E-4          0.0          0.0       556.77
*         pin         pain        flowin         vvin         tvin
      1.563E7          0.0        32.872          0.0          0.0
*       vmscl         vvscl
          1.0          1.0
*
* vmtbv *        0.0        0.0s
* vmtbv *        0.1     32.872s
* vmtbv *       10.0     32.872s
* vmtbv *       10.1   32.793107s
* vmtbv *       10.2   32.638609s
* vmtbv *       10.3   32.434802s
* vmtbv *       10.4   32.198124s
* vmtbv *       10.5   31.938435s
* vmtbv *       10.6    31.66231s
* vmtbv *       10.7   31.382898s
* vmtbv *       10.8   31.093625s
* vmtbv *       10.9   30.804351s
* vmtbv *       11.0   30.515078s
* vmtbv *       11.1   30.229091s
* vmtbv *       11.2   29.943105s
* vmtbv *       11.3   29.660406s
* vmtbv *       11.4   29.380994s
* vmtbv *       11.5   29.108156s
* vmtbv *       11.6   28.838606s
* vmtbv *       11.7    28.57563s
* vmtbv *       11.8   28.319228s
* vmtbv *       11.9   28.062826s
* vmtbv *       12.0   27.816286s
* vmtbv *       13.0   25.696042s
* vmtbv *       14.0   24.203654s
* vmtbv *       15.0   22.878912s
* vmtbv *       16.0    21.67251s
* vmtbv *       17.0   20.574585s
* vmtbv *       18.0   19.568702s
* vmtbv *       19.0   18.644998s
* vmtbv *       20.0   17.793614s
* vmtbv *       21.8   17.793614e
*
*
*******   type          num       userid              component name
fill                  81           1              Channel 8 Fill
*        jun1         ifty         ioff
           8            5            0
*        iftr         ifsv         nftb         nfsv         nfrf
           0            1           32            0            0
*      twtold         rfmx        concin         felv
         0.0      1.0E20          0.0          0.0
*        dxin         volin        alpin         vlin         tlin
```

```
  0.058853659   1.682803E-3          0.0          0.0       556.77
*         pin         pain       flowin         vvin         tvin
      1.563E7          0.0        75.777          0.0          0.0
*        vmscl        vvscl
          1.0          1.0
*
* vmtbv *        0.0        0.0s
* vmtbv *        0.1     75.777s
* vmtbv *       10.0     75.777s
* vmtbv *       10.1   75.595135s
* vmtbv *       10.2   75.238983s
* vmtbv *       10.3   74.769166s
* vmtbv *       10.4   74.223572s
* vmtbv *       10.5   73.624933s
* vmtbv *       10.6   72.988406s
* vmtbv *       10.7   72.344302s
* vmtbv *       10.8   71.677464s
* vmtbv *       10.9   71.010627s
* vmtbv *       11.0   70.343789s
* vmtbv *       11.1   69.684529s
* vmtbv *       11.2   69.025269s
* vmtbv *       11.3   68.373587s
* vmtbv *       11.4   67.729483s
* vmtbv *       11.5   67.100534s
* vmtbv *       11.6   66.479162s
* vmtbv *       11.7   65.872946s
* vmtbv *       11.8   65.281886s
* vmtbv *       11.9   64.690825s
* vmtbv *       12.0   64.122497s
* vmtbv *       13.0   59.234881s
* vmtbv *       14.0   55.794605s
* vmtbv *       15.0   52.740792s
* vmtbv *       16.0   49.959776s
* vmtbv *       17.0   47.428824s
* vmtbv *       18.0   45.110048s
* vmtbv *       19.0   42.980714s
* vmtbv *       20.0    41.01809s
* vmtbv *       21.8    41.01809e
*
*
*******   type          num       userid              component name
fill                    91            1              Channel 9 Fill
*         jun1         ifty         ioff
            9            5            0
*         iftr         ifsv         nftb         nfsv         nfrf
            0            1           32            0            0
*       twtold         rfmx        concin         felv
          0.0       1.0E20          0.0          0.0
*         dxin         volin        alpin         vlin         tlin
  0.058853659   0.033827494          0.0          0.0       556.77
*         pin         pain       flowin         vvin         tvin
      1.563E7          0.0       1522.97          0.0          0.0
*        vmscl        vvscl
          1.0          1.0
*
* vmtbv *        0.0        0.0s
* vmtbv *        0.1    1522.97s
* vmtbv *       10.0    1522.97s
* vmtbv *       10.1    1519.31s
* vmtbv *       10.2    1512.16s
* vmtbv *       10.3    1502.71s
* vmtbv *       10.4    1491.75s
* vmtbv *       10.5    1479.72s
* vmtbv *       10.6    1466.92s
* vmtbv *       10.7    1453.98s
* vmtbv *       10.8    1440.58s
* vmtbv *       10.9    1427.18s
* vmtbv *       11.0    1413.77s
```

```
* vmtbv *       11.1     1400.52s
* vmtbv *       11.2     1387.27s
* vmtbv *       11.3     1374.18s
* vmtbv *       11.4     1361.23s
* vmtbv *       11.5     1348.59s
* vmtbv *       11.6      1336.1s
* vmtbv *       11.7     1323.92s
* vmtbv *       11.8     1312.04s
* vmtbv *       11.9     1300.16s
* vmtbv *       12.0     1288.74s
* vmtbv *       13.0     1190.51s
* vmtbv *       14.0     1121.36s
* vmtbv *       15.0     1059.99s
* vmtbv *       16.0     1004.09s
* vmtbv *       17.0      953.23s
* vmtbv *       18.0      906.62s
* vmtbv *       19.0      863.83s
* vmtbv *       20.0      824.38s
* vmtbv *       21.8      824.38e
*
*
*******   type          num       userid              component name
break                   101          1              Channel 1 Break
*       jun1        ibty        isat        ioff      adjpress
         10          0           0           0          0
*       dxin        volin       alpin         tin         pin
  0.058853659   5.532244E-6         0.0       556.77      1.567E7
*       pain       concin         rbmx         poff         belv
        0.0         0.0       1.0E20          0.0          0.0
*
*
*******   type          num       userid              component name
break                   111          1              Channel 2 Fill
*       jun1        ibty        isat        ioff      adjpress
         11          0           0           0          0
*       dxin        volin       alpin         tin         pin
  0.058853659   5.532244E-6         0.0       556.77      1.563E7
*       pain       concin         rbmx         poff         belv
        0.0         0.0       1.0E20          0.0          0.0
*
*
*******   type          num       userid              component name
break                   121          1              Channel 3 Break
*       jun1        ibty        isat        ioff      adjpress
         12          0           0           0          0
*       dxin        volin       alpin         tin         pin
  0.058853659   6.356195E-6         0.0       556.77      1.563E7
*       pain       concin         rbmx         poff         belv
        0.0         0.0       1.0E20          0.0          0.0
*
*
*******   type          num       userid              component name
break                   131          1              Channel 4 Fill
*       jun1        ibty        isat        ioff      adjpress
         13          0           0           0          0
*       dxin        volin       alpin         tin         pin
  0.058853659   5.532244E-6         0.0       556.77      1.563E7
*       pain       concin         rbmx         poff         belv
        0.0         0.0       1.0E20          0.0          0.0
*
*
*******   type          num       userid              component name
break                   141          1              Channel 5 Break
*       jun1        ibty        isat        ioff      adjpress
         14          0           0           0          0
*       dxin        volin       alpin         tin         pin
  0.058853659   6.980044E-5         0.0       556.77      1.563E7
*       pain       concin         rbmx         poff         belv
```

```
        0.0         0.0       1.0E20          0.0         0.0
*
*
*******   type          num       userid              component name
break                   151            1              Channel 6 Break
*       jun1         ibty         isat        ioff      adjpress
         15            0            0           0            0
*       dxin        volin        alpin         tin         pin
  0.058853659   2.122263E-4          0.0      556.77      1.563E7
*       pain       concin         rbmx         poff         belv
        0.0          0.0       1.0E20          0.0          0.0
*
*
*******   type          num       userid              component name
break                   161            1              Channel 7 Break
*       jun1         ibty         isat        ioff      adjpress
         16            0            0           0            0
*       dxin        volin        alpin         tin         pin
  0.058853659   7.301385E-4          0.0      556.77      1.563E7
*       pain       concin         rbmx         poff         belv
        0.0          0.0       1.0E20          0.0          0.0
*
*
*******   type          num       userid              component name
break                   171            1              Channel 8 Break
*       jun1         ibty         isat        ioff      adjpress
         17            0            0           0            0
*       dxin        volin        alpin         tin         pin
  0.058853659   1.682803E-3          0.0      556.77      1.563E7
*       pain       concin         rbmx         poff         belv
        0.0          0.0       1.0E20          0.0          0.0
*
*
*******   type          num       userid              component name
break                   181            1              Channel 9 Break
*       jun1         ibty         isat        ioff      adjpress
         18            0            0           0            0
*       dxin        volin        alpin         tin         pin
  0.058853659   0.033827494          0.0      556.77      1.563E7
*       pain       concin         rbmx         poff         belv
        0.0          0.0       1.0E20          0.0          0.0
*
*
*********************************************
* Starting Heat Structure Section of Model *
*********************************************
*
*d: Channel 9 Heat Structure
*******   type          num       userid              component name
htstr                   191            0           Ch 9 Heat Structure
*      nzhstr         ittc        hscyl         ichf
          41            0            1            1
*    nofuelrod        plane       liqlev       iaxcnd        pdrat
           0            3            1            0          0.0
*       nmwrx          nfci         nfcil         hdri         hdro
           0            1            1          0.0          0.0
*        nhot        nodes         fmon        nzmax      reflood
           0           10            0          460            0
*     dtxht(1)     dtxht(2)        dznht        hgapo
         0.0          0.0       5.0E-3       5678.3
*
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
```

```
*  idbcin *     0     0     0     0s
*  idbcin *     0     0     0     0s
*  idbcin *     0     0     0     0s
*  idbcin *     0e
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
* hcomon2 *     9     1     0     0e
* hcomon2 *     9     2     0     0e
* hcomon2 *     9     3     0     0e
* hcomon2 *     9     4     0     0e
* hcomon2 *     9     5     0     0e
* hcomon2 *     9     6     0     0e
* hcomon2 *     9     7     0     0e
* hcomon2 *     9     8     0     0e
* hcomon2 *     9     9     0     0e
* hcomon2 *     9    10     0     0e
* hcomon2 *     9    11     0     0e
* hcomon2 *     9    12     0     0e
```

```
* hcomon2 *     9     13     0     0e
* hcomon2 *     9     14     0     0e
* hcomon2 *     9     15     0     0e
* hcomon2 *     9     16     0     0e
* hcomon2 *     9     17     0     0e
* hcomon2 *     9     18     0     0e
* hcomon2 *     9     19     0     0e
* hcomon2 *     9     20     0     0e
* hcomon2 *     9     21     0     0e
* hcomon2 *     9     22     0     0e
* hcomon2 *     9     23     0     0e
* hcomon2 *     9     24     0     0e
* hcomon2 *     9     25     0     0e
* hcomon2 *     9     26     0     0e
* hcomon2 *     9     27     0     0e
* hcomon2 *     9     28     0     0e
* hcomon2 *     9     29     0     0e
* hcomon2 *     9     30     0     0e
* hcomon2 *     9     31     0     0e
* hcomon2 *     9     32     0     0e
* hcomon2 *     9     33     0     0e
* hcomon2 *     9     34     0     0e
* hcomon2 *     9     35     0     0e
* hcomon2 *     9     36     0     0e
* hcomon2 *     9     37     0     0e
* hcomon2 *     9     38     0     0e
* hcomon2 *     9     39     0     0e
* hcomon2 *     9     40     0     0e
* hcomon2 *     9     41     0     0e
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659e
*   rdx   *     5012.0e
*  radrd  *       0.0  5.98333E-4  1.19667E-3    1.795E-3  2.39333E-3s
*  radrd  *  2.99167E-3     3.59E-3  4.18833E-3  4.78667E-3    5.385E-3e
*  matrd  *      6      6      6      6 s
*  matrd  *      6      6      6      6 s
*  matrd  *      6 e
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1e
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
```

*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s

```
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77     556.77     556.77s
*    rftn *     556.77     556.77e
* fpuo2 *       0.0e
*   ftd *      0.945e
*  gmix *       1.0       0.0       0.0       0.0s
*  gmix *       0.0       0.0       0.0e
*   gmles *       0.0e
*   pgapt *     1.0E7e
*   plvol *       0.0 e
*   pslen *       0.0 e
*   clenn *       0.0 e
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4e
*
*******   type        num      userid            component name
htstr               211         0          Ch 8 Heat Structure
*      nzhstr        ittc       hscyl        ichf
          41           0          1          1
*    nofuelrod       plane      liqlev      iaxcnd      pdrat
           0          3          1          0        0.0
*        nmwrx        nfci       nfcil       hdri       hdro
           0          1          1        0.0        0.0
*         nhot       nodes       fmon       nzmax      reflood
           0         10          0        460          0
*    dtxht(1)    dtxht(2)      dznht      hgapo
         0.0        0.0      5.0E-3     5678.3
*
*  idbcin *        0        0        0        0s
*  idbcin *        0        0        0        0s
*  idbcin *        0        0        0        0s
*  idbcin *        0        0        0        0s
*  idbcin *        0        0        0        0s
*  idbcin *        0        0        0        0s
*  idbcin *        0        0        0        0s
*  idbcin *        0        0        0        0s
*  idbcin *        0        0        0        0s
*  idbcin *        0        0        0        0s
```

* idbcin *	0e			
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
* hcomon2 *	8	1	0	0e
* hcomon2 *	8	2	0	0e
* hcomon2 *	8	3	0	0e
* hcomon2 *	8	4	0	0e
* hcomon2 *	8	5	0	0e
* hcomon2 *	8	6	0	0e
* hcomon2 *	8	7	0	0e
* hcomon2 *	8	8	0	0e
* hcomon2 *	8	9	0	0e
* hcomon2 *	8	10	0	0e
* hcomon2 *	8	11	0	0e
* hcomon2 *	8	12	0	0e
* hcomon2 *	8	13	0	0e
* hcomon2 *	8	14	0	0e
* hcomon2 *	8	15	0	0e

```
* hcomon2 *      8      16      0      0e
* hcomon2 *      8      17      0      0e
* hcomon2 *      8      18      0      0e
* hcomon2 *      8      19      0      0e
* hcomon2 *      8      20      0      0e
* hcomon2 *      8      21      0      0e
* hcomon2 *      8      22      0      0e
* hcomon2 *      8      23      0      0e
* hcomon2 *      8      24      0      0e
* hcomon2 *      8      25      0      0e
* hcomon2 *      8      26      0      0e
* hcomon2 *      8      27      0      0e
* hcomon2 *      8      28      0      0e
* hcomon2 *      8      29      0      0e
* hcomon2 *      8      30      0      0e
* hcomon2 *      8      31      0      0e
* hcomon2 *      8      32      0      0e
* hcomon2 *      8      33      0      0e
* hcomon2 *      8      34      0      0e
* hcomon2 *      8      35      0      0e
* hcomon2 *      8      36      0      0e
* hcomon2 *      8      37      0      0e
* hcomon2 *      8      38      0      0e
* hcomon2 *      8      39      0      0e
* hcomon2 *      8      40      0      0e
* hcomon2 *      8      41      0      0e
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659e
*   rdx   *      250.0e
*  radrd  *      0.0  5.98333E-4  1.19667E-3    1.795E-3  2.39333E-3s
*  radrd  *  2.99167E-3     3.59E-3  4.18833E-3  4.78667E-3    5.385E-3e
*  matrd  *       6       6       6       6 s
*  matrd  *       6       6       6       6 s
*  matrd  *       6 e
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1e
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
```

*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s

```
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77e
* fpuo2 *        0.0e
*   ftd *      0.945e
*  gmix *        1.0        0.0        0.0        0.0s
*  gmix *        0.0        0.0        0.0e
*   gmles *        0.0e
*   pgapt *      1.0E7e
*   plvol *         0.0 e
*   pslen *         0.0 e
*   clenn *         0.0 e
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4e
*
*******   type          num       userid               component name
htstr                  221           0            Ch 7 Heat Structure
*      nzhstr         ittc        hscyl         ichf
         41            0            1            1
*    nofuelrod        plane       liqlev       iaxcnd        pdrat
          0            3            1            0         0.0
*       nmwrx         nfci        nfcil         hdri         hdro
          0            1            1          0.0          0.0
*        nhot        nodes         fmon        nzmax      reflood
          0           10            0          460           0
*     dtxht(1)     dtxht(2)        dznht        hgapo
        0.0          0.0        5.0E-3       5678.3
*
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *        0e
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
```

```
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2     2     2     2s
*  idbcon *     2e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
*qflxbco1 *    0.0e
* hcomon2 *     7     1     0     0e
* hcomon2 *     7     2     0     0e
* hcomon2 *     7     3     0     0e
* hcomon2 *     7     4     0     0e
* hcomon2 *     7     5     0     0e
* hcomon2 *     7     6     0     0e
* hcomon2 *     7     7     0     0e
* hcomon2 *     7     8     0     0e
* hcomon2 *     7     9     0     0e
* hcomon2 *     7    10     0     0e
* hcomon2 *     7    11     0     0e
* hcomon2 *     7    12     0     0e
* hcomon2 *     7    13     0     0e
* hcomon2 *     7    14     0     0e
* hcomon2 *     7    15     0     0e
* hcomon2 *     7    16     0     0e
* hcomon2 *     7    17     0     0e
* hcomon2 *     7    18     0     0e
```

```
* hcomon2 *     7     19     0     0e
* hcomon2 *     7     20     0     0e
* hcomon2 *     7     21     0     0e
* hcomon2 *     7     22     0     0e
* hcomon2 *     7     23     0     0e
* hcomon2 *     7     24     0     0e
* hcomon2 *     7     25     0     0e
* hcomon2 *     7     26     0     0e
* hcomon2 *     7     27     0     0e
* hcomon2 *     7     28     0     0e
* hcomon2 *     7     29     0     0e
* hcomon2 *     7     30     0     0e
* hcomon2 *     7     31     0     0e
* hcomon2 *     7     32     0     0e
* hcomon2 *     7     33     0     0e
* hcomon2 *     7     34     0     0e
* hcomon2 *     7     35     0     0e
* hcomon2 *     7     36     0     0e
* hcomon2 *     7     37     0     0e
* hcomon2 *     7     38     0     0e
* hcomon2 *     7     39     0     0e
* hcomon2 *     7     40     0     0e
* hcomon2 *     7     41     0     0e
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659e
*   rdx   *     107.75e
*  radrd  *      0.0  5.98333E-4  1.19667E-3    1.795E-3  2.39333E-3s
*  radrd  *  2.99167E-3     3.59E-3  4.18833E-3  4.78667E-3    5.385E-3e
*  matrd  *       6       6       6       6 s
*  matrd  *       6       6       6       6 s
*  matrd  *       6 e
*    nfax *       1       1       1       1s
*    nfax *       1       1       1       1s
*    nfax *       1       1       1       1s
*    nfax *       1       1       1       1s
*    nfax *       1       1       1       1s
*    nfax *       1       1       1       1s
*    nfax *       1       1       1       1s
*    nfax *       1       1       1       1s
*    nfax *       1       1       1       1s
*    nfax *       1       1       1       1s
*    nfax *       1e
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
*    rftn *    556.77    556.77    556.77    556.77s
```

*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s

```
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77e
* fpuo2 *        0.0e
*   ftd *      0.945e
*  gmix *        1.0        0.0        0.0        0.0s
*  gmix *        0.0        0.0        0.0e
*   gmles *        0.0e
*   pgapt *      1.0E7e
*   plvol *         0.0 e
*   pslen *         0.0 e
*   clenn *         0.0 e
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4e
*
*******   type          num       userid              component name
htstr                  231          0            Ch 6 Heat Structure
*      nzhstr         ittc        hscyl         ichf
          41           0            1            1
*    nofuelrod        plane       liqlev       iaxcnd        pdrat
          0            3            1            0          0.0
*       nmwrx         nfci        nfcil         hdri         hdro
          0            1            1          0.0          0.0
*        nhot        nodes         fmon        nzmax       reflood
          0           10            0          460            0
*     dtxht(1)     dtxht(2)       dznht        hgapo
        0.0          0.0       5.0E-3       5678.3
*
*  idbcin *         0         0         0         0s
*  idbcin *         0         0         0         0s
*  idbcin *         0         0         0         0s
*  idbcin *         0         0         0         0s
*  idbcin *         0         0         0         0s
*  idbcin *         0         0         0         0s
*  idbcin *         0         0         0         0s
*  idbcin *         0         0         0         0s
*  idbcin *         0         0         0         0s
*  idbcin *         0         0         0         0s
*  idbcin *         0e
*  idbcon *         2         2         2         2s
*  idbcon *         2         2         2         2s
*  idbcon *         2         2         2         2s
*  idbcon *         2         2         2         2s
*  idbcon *         2         2         2         2s
```

* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
* hcomon2 *	6	1	0	0e
* hcomon2 *	6	2	0	0e
* hcomon2 *	6	3	0	0e
* hcomon2 *	6	4	0	0e
* hcomon2 *	6	5	0	0e
* hcomon2 *	6	6	0	0e
* hcomon2 *	6	7	0	0e
* hcomon2 *	6	8	0	0e
* hcomon2 *	6	9	0	0e
* hcomon2 *	6	10	0	0e
* hcomon2 *	6	11	0	0e
* hcomon2 *	6	12	0	0e
* hcomon2 *	6	13	0	0e
* hcomon2 *	6	14	0	0e
* hcomon2 *	6	15	0	0e
* hcomon2 *	6	16	0	0e
* hcomon2 *	6	17	0	0e
* hcomon2 *	6	18	0	0e
* hcomon2 *	6	19	0	0e
* hcomon2 *	6	20	0	0e
* hcomon2 *	6	21	0	0e

```
* hcomon2 *       6      22       0       0e
* hcomon2 *       6      23       0       0e
* hcomon2 *       6      24       0       0e
* hcomon2 *       6      25       0       0e
* hcomon2 *       6      26       0       0e
* hcomon2 *       6      27       0       0e
* hcomon2 *       6      28       0       0e
* hcomon2 *       6      29       0       0e
* hcomon2 *       6      30       0       0e
* hcomon2 *       6      31       0       0e
* hcomon2 *       6      32       0       0e
* hcomon2 *       6      33       0       0e
* hcomon2 *       6      34       0       0e
* hcomon2 *       6      35       0       0e
* hcomon2 *       6      36       0       0e
* hcomon2 *       6      37       0       0e
* hcomon2 *       6      38       0       0e
* hcomon2 *       6      39       0       0e
* hcomon2 *       6      40       0       0e
* hcomon2 *       6      41       0       0e
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659e
*   rdx   *      31.75e
*  radrd  *        0.0  5.98333E-4  1.19667E-3    1.795E-3  2.39333E-3s
*  radrd  *  2.99167E-3     3.59E-3  4.18833E-3  4.78667E-3    5.385E-3e
*  matrd  *          6           6           6           6 s
*  matrd  *          6           6           6           6 s
*  matrd  *          6 e
*    nfax *          1           1           1           1s
*    nfax *          1           1           1           1s
*    nfax *          1           1           1           1s
*    nfax *          1           1           1           1s
*    nfax *          1           1           1           1s
*    nfax *          1           1           1           1s
*    nfax *          1           1           1           1s
*    nfax *          1           1           1           1s
*    nfax *          1           1           1           1s
*    nfax *          1           1           1           1s
*    nfax *          1e
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
*    rftn *     556.77      556.77      556.77      556.77s
```

*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s

```
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77e
* fpuo2 *         0.0e
*   ftd *       0.945e
*  gmix *         1.0         0.0         0.0         0.0s
*  gmix *         0.0         0.0         0.0e
*   gmles *         0.0e
*   pgapt *       1.0E7e
*   plvol *         0.0 e
*   pslen *         0.0 e
*   clenn *         0.0 e
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4e
*
********   type          num        userid              component name
htstr                  241           0           Ch 5 Heat Structure
*       nzhstr          ittc         hscyl          ichf
             41           0            1            1
*     nofuelrod         plane        liqlev        iaxcnd         pdrat
              0           3            1            0          0.0
*        nmwrx          nfci         nfcil          hdri          hdro
              0           1            1          0.0          0.0
*         nhot         nodes         fmon         nzmax       reflood
              0          10            0          460            0
*     dtxht(1)      dtxht(2)         dznht         hgapo
            0.0           0.0       5.0E-3        5678.3
*
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0e
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
```

* idbcon *	2	2	2	2s
* idbcon *	2	2	2	2s
* idbcon *	2e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
* hcomon2 *	5	1	0	0e
* hcomon2 *	5	2	0	0e
* hcomon2 *	5	3	0	0e
* hcomon2 *	5	4	0	0e
* hcomon2 *	5	5	0	0e
* hcomon2 *	5	6	0	0e
* hcomon2 *	5	7	0	0e
* hcomon2 *	5	8	0	0e
* hcomon2 *	5	9	0	0e
* hcomon2 *	5	10	0	0e
* hcomon2 *	5	11	0	0e
* hcomon2 *	5	12	0	0e
* hcomon2 *	5	13	0	0e
* hcomon2 *	5	14	0	0e
* hcomon2 *	5	15	0	0e
* hcomon2 *	5	16	0	0e
* hcomon2 *	5	17	0	0e
* hcomon2 *	5	18	0	0e
* hcomon2 *	5	19	0	0e
* hcomon2 *	5	20	0	0e
* hcomon2 *	5	21	0	0e
* hcomon2 *	5	22	0	0e
* hcomon2 *	5	23	0	0e
* hcomon2 *	5	24	0	0e

```
* hcomon2 *      5      25      0      0e
* hcomon2 *      5      26      0      0e
* hcomon2 *      5      27      0      0e
* hcomon2 *      5      28      0      0e
* hcomon2 *      5      29      0      0e
* hcomon2 *      5      30      0      0e
* hcomon2 *      5      31      0      0e
* hcomon2 *      5      32      0      0e
* hcomon2 *      5      33      0      0e
* hcomon2 *      5      34      0      0e
* hcomon2 *      5      35      0      0e
* hcomon2 *      5      36      0      0e
* hcomon2 *      5      37      0      0e
* hcomon2 *      5      38      0      0e
* hcomon2 *      5      39      0      0e
* hcomon2 *      5      40      0      0e
* hcomon2 *      5      41      0      0e
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659e
*   rdx   *      10.0e
*  radrd  *      0.0  5.98333E-4  1.19667E-3   1.795E-3  2.39333E-3s
*  radrd  *  2.99167E-3    3.59E-3  4.18833E-3  4.78667E-3   5.385E-3e
*  matrd  *       6       6       6       6 s
*  matrd  *       6       6       6       6 s
*  matrd  *       6 e
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1e
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
```

*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s

```
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
* fpuo2 *        0.0e
*   ftd *       0.945e
*  gmix *         1.0         0.0         0.0         0.0s
*  gmix *         0.0         0.0         0.0e
*   gmles *         0.0e
*   pgapt *       1.0E7e
*   plvol *         0.0 e
*   pslen *         0.0 e
*   clenn *         0.0 e
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4e
*
*******   type           num        userid               component name
htstr                  251           0            Ch 4 Heat Structure
*       nzhstr         ittc         hscyl         ichf
            41            0             1            1
*    nofuelrod        plane        liqlev        iaxcnd         pdrat
             0            3             1             0           0.0
*        nmwrx          nfci         nfcil          hdri          hdro
             0            1             1           0.0           0.0
*         nhot         nodes          fmon         nzmax       reflood
             0           10             0           460             0
*     dtxht(1)      dtxht(2)         dznht         hgapo
          0.0           0.0        5.0E-3        5678.3
*
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0e
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2e
```

*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
* hcomon2 *	4	1	0	0e
* hcomon2 *	4	2	0	0e
* hcomon2 *	4	3	0	0e
* hcomon2 *	4	4	0	0e
* hcomon2 *	4	5	0	0e
* hcomon2 *	4	6	0	0e
* hcomon2 *	4	7	0	0e
* hcomon2 *	4	8	0	0e
* hcomon2 *	4	9	0	0e
* hcomon2 *	4	10	0	0e
* hcomon2 *	4	11	0	0e
* hcomon2 *	4	12	0	0e
* hcomon2 *	4	13	0	0e
* hcomon2 *	4	14	0	0e
* hcomon2 *	4	15	0	0e
* hcomon2 *	4	16	0	0e
* hcomon2 *	4	17	0	0e
* hcomon2 *	4	18	0	0e
* hcomon2 *	4	19	0	0e
* hcomon2 *	4	20	0	0e
* hcomon2 *	4	21	0	0e
* hcomon2 *	4	22	0	0e
* hcomon2 *	4	23	0	0e
* hcomon2 *	4	24	0	0e
* hcomon2 *	4	25	0	0e
* hcomon2 *	4	26	0	0e
* hcomon2 *	4	27	0	0e

```
* hcomon2 *     4     28     0     0e
* hcomon2 *     4     29     0     0e
* hcomon2 *     4     30     0     0e
* hcomon2 *     4     31     0     0e
* hcomon2 *     4     32     0     0e
* hcomon2 *     4     33     0     0e
* hcomon2 *     4     34     0     0e
* hcomon2 *     4     35     0     0e
* hcomon2 *     4     36     0     0e
* hcomon2 *     4     37     0     0e
* hcomon2 *     4     38     0     0e
* hcomon2 *     4     39     0     0e
* hcomon2 *     4     40     0     0e
* hcomon2 *     4     41     0     0e
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659e
*   rdx   *      0.747e
*  radrd  *      0.0  5.98333E-4  1.19667E-3   1.795E-3  2.39333E-3s
*  radrd  *  2.99167E-3    3.59E-3  4.18833E-3  4.78667E-3   5.385E-3e
*  matrd  *      6      6      6      6 s
*  matrd  *      6      6      6      6 s
*  matrd  *      6 e
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1      1      1      1s
*    nfax *      1e
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
*    rftn *   556.77    556.77    556.77    556.77s
```

*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s

```
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77e
* fpuo2 *        0.0e
*   ftd *      0.945e
*  gmix *        1.0         0.0         0.0         0.0s
*  gmix *        0.0         0.0         0.0e
*   gmles *        0.0e
*   pgapt *      1.0E7e
*   plvol *        0.0 e
*   pslen *        0.0 e
*   clenn *        0.0 e
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *     1.54E4e
*
*******   type          num        userid               component name
htstr                   261           0            Ch 3 Heat Structure
*       nzhstr         ittc        hscyl         ichf
           41            0            1            1
*    nofuelrod        plane        liqlev        iaxcnd        pdrat
           0            3            1            0          0.0
*        nmwrx         nfci         nfcil         hdri          hdro
           0            1            1          0.0          0.0
*         nhot        nodes         fmon        nzmax       reflood
           0           10            0          460            0
*     dtxht(1)      dtxht(2)        dznht        hgapo
         0.0          0.0        5.0E-3        5678.3
*
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0          0          0          0s
*  idbcin *         0e
*  idbcon *         2          2          2          2s
*  idbcon *         2          2          2          2s
*  idbcon *         2          2          2          2s
*  idbcon *         2          2          2          2s
*  idbcon *         2          2          2          2s
*  idbcon *         2          2          2          2s
*  idbcon *         2          2          2          2s
*  idbcon *         2          2          2          2s
*  idbcon *         2          2          2          2s
*  idbcon *         2          2          2          2s
*  idbcon *         2e
*qflxbco1 *        0.0e
*qflxbco1 *        0.0e
*qflxbco1 *        0.0e
```

*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
*qflxbco1 *	0.0e			
* hcomon2 *	3	1	0	0e
* hcomon2 *	3	2	0	0e
* hcomon2 *	3	3	0	0e
* hcomon2 *	3	4	0	0e
* hcomon2 *	3	5	0	0e
* hcomon2 *	3	6	0	0e
* hcomon2 *	3	7	0	0e
* hcomon2 *	3	8	0	0e
* hcomon2 *	3	9	0	0e
* hcomon2 *	3	10	0	0e
* hcomon2 *	3	11	0	0e
* hcomon2 *	3	12	0	0e
* hcomon2 *	3	13	0	0e
* hcomon2 *	3	14	0	0e
* hcomon2 *	3	15	0	0e
* hcomon2 *	3	16	0	0e
* hcomon2 *	3	17	0	0e
* hcomon2 *	3	18	0	0e
* hcomon2 *	3	19	0	0e
* hcomon2 *	3	20	0	0e
* hcomon2 *	3	21	0	0e
* hcomon2 *	3	22	0	0e
* hcomon2 *	3	23	0	0e
* hcomon2 *	3	24	0	0e
* hcomon2 *	3	25	0	0e
* hcomon2 *	3	26	0	0e
* hcomon2 *	3	27	0	0e
* hcomon2 *	3	28	0	0e
* hcomon2 *	3	29	0	0e
* hcomon2 *	3	30	0	0e

```
* hcomon2 *    3    31    0    0e
* hcomon2 *    3    32    0    0e
* hcomon2 *    3    33    0    0e
* hcomon2 *    3    34    0    0e
* hcomon2 *    3    35    0    0e
* hcomon2 *    3    36    0    0e
* hcomon2 *    3    37    0    0e
* hcomon2 *    3    38    0    0e
* hcomon2 *    3    39    0    0e
* hcomon2 *    3    40    0    0e
* hcomon2 *    3    41    0    0e
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659e
*   rdx   *     0.989e
*  radrd  *      0.0  5.98333E-4  1.19667E-3    1.795E-3  2.39333E-3s
*  radrd  *  2.99167E-3     3.59E-3  4.18833E-3  4.78667E-3    5.385E-3e
*  matrd  *      6      6      6      6 s
*  matrd  *      6      6      6      6 s
*  matrd  *      6 e
*   nfax *      1      1      1      1s
*   nfax *      1      1      1      1s
*   nfax *      1      1      1      1s
*   nfax *      1      1      1      1s
*   nfax *      1      1      1      1s
*   nfax *      1      1      1      1s
*   nfax *      1      1      1      1s
*   nfax *      1      1      1      1s
*   nfax *      1      1      1      1s
*   nfax *      1      1      1      1s
*   nfax *      1e
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
```

*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s

```
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77e
* fpuo2 *         0.0e
*    ftd *       0.945e
*   gmix *         1.0         0.0         0.0         0.0s
*   gmix *         0.0         0.0         0.0e
*   gmles *        0.0e
*   pgapt *       1.0E7e
*   plvol *        0.0 e
*   pslen *        0.0 e
*   clenn *        0.0 e
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*  burn *      1.54E4e
*
******** type          num        userid              component name
htstr                 271           0           Ch 2 Heat Structure
*       nzhstr         ittc         hscyl         ichf
            41            0            1            1
*     nofuelrod         plane        liqlev        iaxcnd         pdrat
             0            3            1            0           0.0
*        nmwrx          nfci         nfcil          hdri          hdro
             0            1            1          0.0           0.0
*         nhot         nodes          fmon         nzmax       reflood
             0           10            0          460             0
*     dtxht(1)      dtxht(2)         dznht         hgapo
           0.0           0.0        5.0E-3        5678.3
*
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0          0          0          0s
*  idbcin *          0e
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2          2          2          2s
*  idbcon *          2e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
```

```
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
* hcomon2 *      2      1      0      0e
* hcomon2 *      2      2      0      0e
* hcomon2 *      2      3      0      0e
* hcomon2 *      2      4      0      0e
* hcomon2 *      2      5      0      0e
* hcomon2 *      2      6      0      0e
* hcomon2 *      2      7      0      0e
* hcomon2 *      2      8      0      0e
* hcomon2 *      2      9      0      0e
* hcomon2 *      2     10      0      0e
* hcomon2 *      2     11      0      0e
* hcomon2 *      2     12      0      0e
* hcomon2 *      2     13      0      0e
* hcomon2 *      2     14      0      0e
* hcomon2 *      2     15      0      0e
* hcomon2 *      2     16      0      0e
* hcomon2 *      2     17      0      0e
* hcomon2 *      2     18      0      0e
* hcomon2 *      2     19      0      0e
* hcomon2 *      2     20      0      0e
* hcomon2 *      2     21      0      0e
* hcomon2 *      2     22      0      0e
* hcomon2 *      2     23      0      0e
* hcomon2 *      2     24      0      0e
* hcomon2 *      2     25      0      0e
* hcomon2 *      2     26      0      0e
* hcomon2 *      2     27      0      0e
* hcomon2 *      2     28      0      0e
* hcomon2 *      2     29      0      0e
* hcomon2 *      2     30      0      0e
* hcomon2 *      2     31      0      0e
* hcomon2 *      2     32      0      0e
* hcomon2 *      2     33      0      0e
```

```
* hcomon2 *      2      34      0      0e
* hcomon2 *      2      35      0      0e
* hcomon2 *      2      36      0      0e
* hcomon2 *      2      37      0      0e
* hcomon2 *      2      38      0      0e
* hcomon2 *      2      39      0      0e
* hcomon2 *      2      40      0      0e
* hcomon2 *      2      41      0      0e
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659e
*   rdx   *      0.745e
*  radrd  *       0.0  5.98333E-4  1.19667E-3    1.795E-3  2.39333E-3s
*  radrd  *  2.99167E-3     3.59E-3  4.18833E-3  4.78667E-3    5.385E-3e
*  matrd  *        6        6        6        6 s
*  matrd  *        6        6        6        6 s
*  matrd  *        6 e
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1       1       1       1s
*   nfax *       1e
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
*   rftn *    556.77    556.77    556.77    556.77s
```

*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s

```
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77      556.77      556.77s
*    rftn *      556.77      556.77e
* fpuo2 *         0.0e
*    ftd *       0.945e
*   gmix *         1.0         0.0         0.0         0.0s
*   gmix *         0.0         0.0         0.0e
*    gmles *         0.0e
*    pgapt *       1.0E7e
*    plvol *          0.0 e
*    pslen *          0.0 e
*    clenn *          0.0 e
*   burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*   burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*   burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*   burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*   burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*   burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*   burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*   burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*   burn *      1.54E4      1.54E4      1.54E4      1.54E4s
*   burn *      1.54E4e
*
********   type          num        userid               component name
htstr                  281           0           Ch 1 Heat Structure
*       nzhstr         ittc        hscyl         ichf
             41           0           1           1
*     nofuelrod        plane        liqlev        iaxcnd         pdrat
              0           3           1            0          0.0
*         nmwrx          nfci         nfcil          hdri          hdro
              0           1           1          0.0          0.0
*         nhot         nodes          fmon         nzmax       reflood
              0          10           0          460            0
*      dtxht(1)      dtxht(2)         dznht         hgapo
           0.0           0.0       5.0E-3        5678.3
*
*   idbcin *          0          0          0          0s
*   idbcin *          0          0          0          0s
*   idbcin *          0          0          0          0s
*   idbcin *          0          0          0          0s
*   idbcin *          0          0          0          0s
*   idbcin *          0          0          0          0s
*   idbcin *          0          0          0          0s
*   idbcin *          0          0          0          0s
*   idbcin *          0          0          0          0s
*   idbcin *          0          0          0          0s
*   idbcin *          0e
*   idbcon *          2          2          2          2s
*   idbcon *          2          2          2          2s
*   idbcon *          2          2          2          2s
*   idbcon *          2          2          2          2s
*   idbcon *          2          2          2          2s
*   idbcon *          2          2          2          2s
*   idbcon *          2          2          2          2s
*   idbcon *          2          2          2          2s
*   idbcon *          2          2          2          2s
*   idbcon *          2          2          2          2s
*   idbcon *          2e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
```

```
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
*qflxbco1 *     0.0e
* hcomon2 *      1      1      0      0e
* hcomon2 *      1      2      0      0e
* hcomon2 *      1      3      0      0e
* hcomon2 *      1      4      0      0e
* hcomon2 *      1      5      0      0e
* hcomon2 *      1      6      0      0e
* hcomon2 *      1      7      0      0e
* hcomon2 *      1      8      0      0e
* hcomon2 *      1      9      0      0e
* hcomon2 *      1     10      0      0e
* hcomon2 *      1     11      0      0e
* hcomon2 *      1     12      0      0e
* hcomon2 *      1     13      0      0e
* hcomon2 *      1     14      0      0e
* hcomon2 *      1     15      0      0e
* hcomon2 *      1     16      0      0e
* hcomon2 *      1     17      0      0e
* hcomon2 *      1     18      0      0e
* hcomon2 *      1     19      0      0e
* hcomon2 *      1     20      0      0e
* hcomon2 *      1     21      0      0e
* hcomon2 *      1     22      0      0e
* hcomon2 *      1     23      0      0e
* hcomon2 *      1     24      0      0e
* hcomon2 *      1     25      0      0e
* hcomon2 *      1     26      0      0e
* hcomon2 *      1     27      0      0e
* hcomon2 *      1     28      0      0e
* hcomon2 *      1     29      0      0e
* hcomon2 *      1     30      0      0e
* hcomon2 *      1     31      0      0e
* hcomon2 *      1     32      0      0e
* hcomon2 *      1     33      0      0e
* hcomon2 *      1     34      0      0e
* hcomon2 *      1     35      0      0e
* hcomon2 *      1     36      0      0e
```

```
* hcomon2 *     1     37     0     0e
* hcomon2 *     1     38     0     0e
* hcomon2 *     1     39     0     0e
* hcomon2 *     1     40     0     0e
* hcomon2 *     1     41     0     0e
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659 0.058853659 0.058853659 0.058853659s
* dhtstrz * 0.058853659e
*   rdx   *     0.744e
*  radrd  *     0.0  5.98333E-4  1.19667E-3   1.795E-3  2.39333E-3s
*  radrd  *  2.99167E-3    3.59E-3  4.18833E-3  4.78667E-3    5.385E-3e
*  matrd  *     6     6     6     6 s
*  matrd  *     6     6     6     6 s
*  matrd  *     6 e
*    nfax *     1     1     1     1s
*    nfax *     1     1     1     1s
*    nfax *     1     1     1     1s
*    nfax *     1     1     1     1s
*    nfax *     1     1     1     1s
*    nfax *     1     1     1     1s
*    nfax *     1     1     1     1s
*    nfax *     1     1     1     1s
*    nfax *     1     1     1     1s
*    nfax *     1     1     1     1s
*    nfax *     1e
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
*    rftn *   556.77   556.77   556.77   556.77s
```

*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77	556.77	556.77s
*	rftn *	556.77	556.77e		

```
* fpuo2 *        0.0e
*   ftd *      0.945e
*  gmix *        1.0        0.0        0.0        0.0s
*  gmix *        0.0        0.0        0.0e
*   gmles *        0.0e
*   pgapt *      1.0E7e
*   plvol *         0.0 e
*   pslen *         0.0 e
*   clenn *         0.0 e
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4     1.54E4     1.54E4     1.54E4s
*  burn *     1.54E4e
********************************************
* Finished Heat Structure Section of Model *
********************************************
*
*
*
*
*****************************************
*     Starting Power Components         *
*****************************************
*
*******   type          num        userid               component name
power                  201           1                   Ch 9 Power
*       numpwr       chanpow
           9            0
* htnum *        191        211        221        231        241 s
* htnum *        251        261        271        281 e
*       irpwty         ndgx         ndhx         nrts         nhist
          11            6           71          100            1
*        q235         q239         q238         qavg        r239pf
        192.5        198.5        193.5        195.0          0.5
*       fisphi         rans        fp235        fp238
          1.5          1.0         0.95         0.01
*       izpwtr       izpwsv       nzpwtb       nzpwsv       nzpwrf
           0            1            1            0            0
*       ipwrad       ipwdep      promheat     decaheat     wtbypass
           0            0          0.0          0.0          0.0
*        nzpwz        nzpwi       nfbpwt        nrpwr        nrpwi
          30            0            0            1            0
*        react        tneut       rpwoff       rrpwmx       rpwscl
          0.0          0.0          0.0       1.0E20          1.0
*       rpowri        zpwin       zpwoff       rzpwmx
   2.576426E8          0.0      -1.0E19       1.0E20
*       extsou         pldr        pdrat       fucrac
          0.0          0.0          1.0          1.0
*   ircjtb 1,0   ircjtb 2,0   ircjtb 3,0   ircjtb 4,0        ibu 0
           2            1            1            1            0
*   ircjtb 1,1   ircjtb 2,1   ircjtb 3,1   ircjtb 4,1        ibu 1
           1            2            1            1            0
*   ircjtb 1,2   ircjtb 2,2   ircjtb 3,2   ircjtb 4,2        ibu 2
           1            1            1            1            0
*   ircjtb 1,3   ircjtb 2,3   ircjtb 3,3   ircjtb 4,3        ibu 3
           1            1            1            1           -2
*      ircjfm1      ircjfm2      ircjfm3      ircjfm4       isnotb
           1            1            0            0            1
*       powexp         bpp0         bpp1         bpp2         bpp3
          2.0         0.26     -2.0E-3          0.1        -30.0
* rdpwr *        1.0        1.0        1.0        1.0        1.0s
```

```
* rdpwr *        1.0        1.0        1.0        1.0        1.0e
* cpowr *  0.99975893 0.078569729 0.033863331  9.92169E-3  3.17482E-3s
* cpowr *  2.42041E-4  3.20454E-4  2.41393E-4  2.41069E-4e
* zpwzt *        0.0      0.082      0.1665      0.2485      0.333s
* zpwzt *      0.415      0.4995      0.5815      0.666      0.748s
* zpwzt *     0.8325      0.9145      0.999      1.081      1.1655s
* zpwzt *     1.2475      1.332      1.414      1.4985      1.5805s
* zpwzt *      1.665      1.747      1.8315      1.9135      1.998s
* zpwzt *       2.08      2.1645      2.2465      2.331      2.413e
* zpwtb1*        0.0s
* zpwtb1*     0.0339      0.1989      0.3617      0.5203      0.673s
* zpwtb1*     0.8181      0.9538      1.0786      1.1911      1.2901s
* zpwtb1*     1.3743      1.4429      1.4951      1.5301      1.5478s
* zpwtb1*     1.5478      1.5301      1.4951      1.4429      1.3743s
* zpwtb1*     1.2901      1.1911      1.0786      0.9538      0.8181s
* zpwtb1*      0.673      0.5203      0.3617      0.1989      0.0339e
* rctf  *      550.0      800.0        0.0        0.0        0.0s
* rctf  *   -2.928E-5  -2.3334E-5e
* rctc  *        0.0      550.0      600.0        0.0        0.0s
* rctc  *    -9.85E-5   -2.066E-4e
* rcal  *        0.0        0.0        0.0        0.0    -0.12345e
* rcbm  *        0.0        0.0        0.0        0.0   -2.321E-5e
* beta  *    1.69E-4    8.32E-4    2.64E-3    1.22E-3    1.38E-3s
* beta  *    2.47E-4e
* lamda *       3.87        1.4      0.311      0.115      0.0317s
* lamda *     0.0127e
*****************************************
*     Finished Power Components         *
*****************************************
*
*
*
end
*
******************
* Timestep Data *
******************
*       dtmin        dtmax         tend        rtwfp
       1.0E-6          0.1         21.8          1.0
*       edint        gfint       dmpint       sedint
       100.0          1.0        100.0          1.0
*
*      endflag
         -1.0
```

REFERENCES

1. M. Avramova, D. Cuervo, K. Ivanov, “Improvements and Applications of COBRA-TF for Stand-Alone and Coupled LWR Safety Analyses”, *Proc. of PHYSOR-2006 Conference*, Vancouver, Canada (2006).
2. M. Avramova, K. Ivanov, L. Hochreiter, "Analysis of Steady State and Transient Void Distribution Predictions for Phase I of the OECD/NRC BFBT Benchmark using CTF/NEM," NURETH-12 Conference, Pittsburgh, Pennsylvania, U.S.A. September 30-October 4 (2007).
3. D. Cuervo, M. Avramova, K. Ivanov, R. Miro, "Evaluation and Enhancement of COBRA-TF Efficiency for LWR Calculations," Annals of Nuclear Energy, Volume 33, pp. 837-847 (2006).
4. A. Henry, 1975, Nuclear Reactor Analysis, The MIT Press, Cambridge, Mass. and London England.
5. Glasstone, S. and Sesonske, A., 1967, Nuclear Reactor Engineering, Van Nostrand, New York.
6. TRACE V5.0 Theory Manual, US NRC, 2008.
7. J. W. Spore et al., TRAC-PF1/MOD2: Volume I. Theory Manual. 1993.
8. ANSI/ANS 5.1-1979, 1978, “Decay Heat in Light Water Reactors,” American Nuclear Society.
9. ANSI/ANS-5.1-1994: Decay Heat in Light Water Reactors

10. L. Hochreiter, et al., 1988, “Westinghouse Large Break LOCA Best Estimate Methodology, Volume 1: Model Description and Validation, Addendum 2: Revised Appendix B: Heat Source Models,” WCAP-10924-P-A, Revision 1.
11. ANSI/ANS-5.1-2005: Decay Heat Power in Light Water Reactors.
12. M. Avramova, K. Ivanov, L. Hochreiter, S. Balzus and R. Mueller “Comparative Analysis of PWR Core Wide Hot Channel Calculations”, Transactions from 2002 ANS Meeting, Washington DC, November 2002, Vol 87, pp 208-210.