The Pennsylvania State University
The Graduate School

DEVELOPMENT OF A FULLY IMPLICIT STEADY STATE SOLUTION FOR TRACE AND PARCS

A Thesis in
Nuclear Engineering
by
Matthew Shawn Ellis

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2012

The thesis of Matthew Shawn Ellis was reviewed and approved* by the following:

Kostadin Ivanov
Distinguished Professor of Nuclear Engineering
Thesis Advisor

Maria Avramova
Assistant Professor of Nuclear Engineering

Justin Watson
Research Associate and Assistant Professor of Nuclear Engineering

Arthur Motta
Professor of Nuclear Engineering and Materials Science and Engineering
Chair of Nuclear Engineering

*Signatures are on file in the Graduate School.

Abstract

This thesis describes the implementation of an implicit steady state solution method in the coupled TRAC/RELAP Advanced Computational Engine (TRACE) thermal-hydraulics system code and Purdue Advanced Reactor Core Simulator (PARCS) code with the goal of improving solution stability and efficiency. The implicit steady state solution method has been implemented within the framework of the existing pseudo-transient solution method in TRACE and includes time-dependent thermal-hydraulic and heat transfer equations and time-independent criticality neutron diffusion equations. The implicit steady state solution was first evaluated using two different meshes overlaid on a two-phase pipe model closely matching a boiling water reactor hydraulic channel. The two-phase pipe model showed that the implicit solution method reproduces the correct steady state solution for varying time step sizes for each mesh. A four-channel reactor model was created to further evaluate the performance of the implicit steady state solution. The results from the four-channel reactor model show that the computer runtime required in the implicit solution method is much greater than the required runtime for the well-developed and optimized explicit solution method. These results direct future development of the implicit solution method towards optimization strategies to reduce computer runtime.

Table of Contents

List of Figures

List of Tables

List of Symbols

Independent Variables

t Time

x Cartesian coordinate direction

y Cartesian coordinate direction

z Cartesian coordinate direction

Thermal-Hydraulic Variables

P Pressure

α Void fraction

T Temperature

V Velocity

f Force per unit volume

ρ Density

Γ Interfacial mass-transfer rate (positive from liquid to gas)

$\overrightarrow{g}$ Gravity vector

e Internal energy

q Heat transfer rate per unit volume

C Shear coefficient

Γ^- Minimum of Γ and zero

Γ^+ Maximum of Γ and zero

h Heat transfer coefficient

$h_{l'}$ Liquid enthalpy of the bulk liquid if the liquid is vaporizing or the liquid saturation enthalpy if vapor is condensing

$h_{v'}$ Vapor enthalpy of the bulk vapor if the vapor is condensing or the vapor saturation enthalpy if vapor is vaporizing‘

h_{sg} Gas saturation enthalpy

Thermal-Hydraulic Subscripts

w Wall

l Liquid

g Gas

i Interfacial

wl Wall-to-liquid

wg Wall-to-gas

dg Direct-to-gas

dl Direct-to-liquid

$wsat$ Wall-to-saturated mixture

ig Interfacial-to-gas

j Cell centered index

Thermal-Hydraulic Superscripts

n Current-time quantity

$n+1$ New-time quantity

Heat Conduction Variables

k Thermal conductivity

q''' Volumetric heat-generation rate

T Temperature

ρ Density

C_p Specific heat

Δr Volume length in the radial direction

Δz Volume length in the z direction

h Heat Transfer Coefficient

Heat Conduction Subscripts

i Nodal index in the radial direction

j Nodal index in the z direction

l Liquid

g Gas

w Wall

Heat Conduction Superscripts

n Current-time quantity

$n+1$ New-time quantity

p Predicted new-time values

Reactor Physics Variables

Σ Macroscopic cross section

ϕ Neutron flux

a Neutron leakage coefficient

k_{eff} Effective multiplication factor

k_s Shifted effective multiplication factor

v Neutron speed

h Neutronics node length

χ	Fraction of delayed or prompt neutrons
β	Delayed-neutron fraction
Q	Total model (core) rated thermal power
M	Neutronics migration matrix
F	Neutronics fission matrix
Ψ	Fission source
nr	Number of heat structure nodes in given axial plane
κ	Average energy release per fission
ν	Average number of neutrons produced per fission
C	Delayed neutron precursor concentration
J	Neutron current
λ	Delayed neutron precursor decay constant or eigenvalue
D	Neutron diffusion coefficient
$\hat{D}$	Nodal coupling correction coefficient
$\tilde{D}$	Nodal coupling coefficient
Δu	Neutronics volume length in the u direction
R	Neutron flux scaling factor
W_i	Power deposition weighting factor
P	Pressure
ρ	Density
T	Temperature
α	Void Fraction
q'''	Volumetric heat generation
V	Volume

Reactor Physics Superscripts

m	Neutronics node number
i	Heat Structure node number
j	Hydraulic node number
$m+$	Neighboring neutronics node number in the positive direction
$m-$	Neighboring neutronics node number in the negative direction
$j+$	Neighboring hydraulic node number in the positive direction
$j-$	Neighboring hydraulic node number in the negative direction
$i+$	Neighboring heat structure node number in the positive direction
$i-$	Neighboring heat structure node number in the negative direction
$m \pm l_u$	Neighboring neutronics node number either the positive or negative direction u
M	All other neutronics nodes other than cell m
J	All other hydraulic nodes other than cell j
I	All other heat structure nodes other than cell i
i'	All other heat structure nodes coinciding with cell m other than i
n	Current time level
$n+1$	New time level
k	Current iterate
$k+1$	Next iterate

Reactor Physics Subscripts

a	Noncondensable-gas component
d	Delayed
f	Fuel

g	Gas field
g	Neutron energy group
g'	Neutron energy group other than neutron energy group g
$g'g$	From energy group g' to energy group g
l	Liquid field
k	Delayed neutron precursor group
p	Prompt
u	Any coordinate direction
m	Mixture

Acknowledgments

I would like to thank my family for their support throughout my undergraduate and graduate education at The Pennsylvania State University. Without their continued support many of the opportunities that defined my journey through college would not have been possible.

I would also like to acknowledge Dr. Justin Watson and Dr. Kostadin Ivanov for their technical contributions and mentoring throughout my graduate education. Their extensive knowledge of reactor analysis was invaluable during the development of this work, and their eagerness to teach has made working on this project an enjoyable and worthwhile endeavor.

Furthermore, I would like to thank The Applied Research Laboratory (ARL) for their financial support through the Walker Graduate Assistantship. Also, the commitment of faculty and staff at the Garfield Thomas Water Tunnel to graduate student education has made working at ARL a uniquely rewarding and enjoyable experience.

Finally, I would like to thank my fellow graduate students for their friendship over the last two years. Their unyielding humor, passion, and curiosity have had a tremendously positive impact on me. I wish them all the best in their future endeavors.

Chapter 1

Introduction

The ability to simulate nuclear reactor systems during steady state and transient conditions is necessary to ensure that nuclear reactors operate safely and efficiently. The complexity of reactor systems necessitates the use of numerical methods instead of an analytical approach, and there is a wide range of software packages that model various aspects of reactor systems. Of particular interest in this investigation is the coupled reactor safety analysis package used by the Nuclear Regulatory Commission (NRC). This reactor analysis package includes the thermal-hydraulics code TRAC/RELAP Advanced Computation Engine (TRACE) and the reactor physics code Purdue Advanced Reactor Core Simulator (PARCS).

The simulation of nuclear reactor systems is a particularly challenging task because of the multiple physical processes occurring simultaneously within a reactor. Earlier methods used to model the multiphysic characteristics of nuclear reactors relied on splitting the different physical processes into separate computational units, a technique referred to as Operator Splitting (OS) [1]. In reactor analysis the OS technique typically consists of a hydraulic component used for calculating core flow, heat transfer component used for determining structure temperatures, and a reactor physics component used for calculating the neutron population in the core. The separate computational units communicate through either an external or internal method which allows for a coupled calculation to be performed. The external coupling method is the simplest to implement because it relies on file input and output that is native to each program to share information. The internal coupling method meshes the different computational units into a single program

and typically uses a general message passing interface to share information.

Unfortunately, the OS technique is inherently nonlinearly inconsistent and is inferior to numerical methods that resolve the nonlinearities of the physical system through more rigorous methods [1]. The limitations of OS techniques were investigated with an analytical approach for several nonlinear equation sets by Knoll *et al.*, and it was shown that OS methods can lead to a loss of solution accuracy when large time steps are utilized [2]. The loss of solution accuracy and resulting need for smaller time steps is the main catalyst for investigating improved OS methods and nonlinearly consistent methods.

As it applies to time-dependent problems, OS is typically synonymous with explicit time-integration techniques where the field of unknowns is reduced and linearized through the use of previous time step solutions for nonlinear coefficients and principle unknowns. Unfortunately, explicit time-integration techniques are not unconditionally stable and are limited to smaller time steps because of stability constraints and first-order temporal accuracy [3]. However, even with the aforementioned time step size restrictions, explicit time integration techniques are commonly used to solve reactor analysis problems [1].

The inherent time step size limitation and potential error of the explicit time integration technique is the primary impetus for including fully implicit time integration methods in reactor analysis codes. In implicit time integration techniques all of the independent parameters coupling the multiple physical processes are solved for simultaneously at the time step of interest instead of the separated convergence indicative of OS methods. Early investigations that pertained to the development of fully implicit two-phase flow equations showed improved time step size capability and increased accuracy compared to existing operator-splitting and explicit time-integration techniques [4, 5, 6]. Other investigations that sought to develop implicit solution techniques for entire sets of hydraulic, heat transfer, and neutronics equations reported varied, but promising, improvements in solution accuracy and maximum time step sizes or an increase in solution efficiency for a steady state solution [7, 8, 9, 1, 3].

The trend in the literature concerning implicit models in reactor analysis has been an evolution from simplified fully implicit fluid models, to fully implicit sets of reactor equations, to implementation of fully implicit solution methods in well-

established core analysis tools. The work presented in this thesis is an extension of this trend and is related to the implicit coupling method for transient solutions previously implemented in the thermal hydraulic code TRACE and reactor physics code PARCS [3]. The purpose of the work presented in this document is to implement an implicit steady state solution algorithm in TRACE and PARCS using the existing implicit transient solution methodology recently developed [3].

Chapter 2 in this document will describe the basic numerical methods that are utilized in TRACE and PARCS. Chapter 3 in this document will describe the explicit solution technique traditionally used in TRACE and PARCS to solve steady state reactor analysis problems and the structure of the recently implemented implicit steady state solution method. Chapter 4 will include the equations necessary to implement an implicit steady state solution in the existing implicit transient solution structure. Chapter 5 will present the results of the new implicit steady state solution in TRACE and PARCS. Chapter 6 and Chapter 7 will discuss and summarize these results.

Chapter 2

Numerical Methods for Reactor Analysis

Solving the partial differential equations (PDEs) that describe nuclear reactor systems relies on methods that are applicable to the solution of many different PDEs. These solution methods rely on discretizing the domain and using numerical approximations to evaluate integral and differential quantities. The contents of this chapter are meant to provide a brief overview of the numerical methods that are pertinent to the solution of the governing equations in TRACE and PARCS. Additionally, the discrete form of the governing equations used in the existing explicit solution method in TRACE and PARCS will be presented in this chapter.

2.1 Finite Difference Method

The finite difference method is the oldest numerical method for solving PDEs, and it is the easiest method to implement for simple geometries [10]. The finite difference method can be directly applied to equations in differential form, and it is the underlying method from which the discretized steady state neutron diffusion equations in PARCS and discretized heat transfer equations in TRACE are derived.

The first step in applying the finite difference approach to a set of PDEs is to discretize the domain. A rudimentary example of a spatial discretization for a one-dimensional domain is shown in Figure 2.1 [10]. In the interest of condensing equations that will be introduced later in this section, shorthand notation for the

spacing between the computational nodes is shown in equation (2.1) and equation (2.2). Only one-dimensional domains such as the domain in Figure 2.1 will be used to illustrate the basic theory of finite difference methods because the added complexity of additional dimensions does not provided any meaningful insight into the framework of the finite difference method.

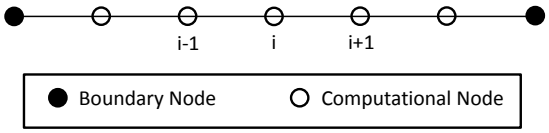


Figure 2.1: One-dimensional finite difference discretized domain

$$\Delta x_{i+1} = x_{i+1} - x_i \tag{2.1}$$

$$\Delta x_i = x_i - x_{i-1} \tag{2.2}$$

To solve PDEs using the finite difference method, it is necessary to be able to approximate derivatives at any of the computational nodes in the domain [10]. For illustrative purposes the computational node (i) in the one-dimensional domain shown in Figure 2.1 will be considered. Assuming that the quantity of interest on the domain (i.e. $\phi(x)$) is a continuous function, a Taylor series expansion for the value of $\phi(x)$ at the points $(i{+}1)$ and $(i{-}1)$ can be formulated about the point (i). The aforementioned Taylor series expansions of $\phi(x)$ at points $(i{+}1)$ and $(i{-}1)$ are shown in equation (2.3) and equation (2.4) [10]. The Taylor series expansions can then be rearranged as shown in equation (2.5) and equation (2.6) to yield the forward-difference scheme (FDS) and backward-difference scheme (BDS), respectively, for approximating the derivative at point (i) [10]. In practice, only the first term in equation (2.5) and equation (2.6) is used to approximate the derivative, and the discarded terms are collectively referred to as the truncation error. The FDS and BDS are both first-order accurate because the largest term in the truncation error scales linearly with the mesh spacing.

$$\phi\left(x_{i+1}\right) = \phi\left(x_i\right) + \left(\Delta x_{i+1}\right)\left(\frac{\partial \phi}{\partial x}\right)\Bigg|_i + \frac{\left(\Delta x_{i+1}\right)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)\Bigg|_i \tag{2.3}$$

$$+\frac{(\Delta x_{i+1})^3}{3!}\left(\frac{\partial^3\phi}{\partial x^3}\right)\Bigg|_i+\ldots+\frac{(\Delta x_{i+1})^n}{n!}\left(\frac{\partial^n\phi}{\partial x^n}\right)\Bigg|_i+H$$

$$\begin{aligned}\phi(x_{i-1})=\phi(x_i)-(\Delta x_i)\left(\frac{\partial\phi}{\partial x}\right)\Bigg|_i+\frac{(\Delta x_i)^2}{2!}\left(\frac{\partial^2\phi}{\partial x^2}\right)\Bigg|_i \\ -\frac{(\Delta x_i)^3}{3!}\left(\frac{\partial^3\phi}{\partial x^3}\right)\Bigg|_i+\ldots+\frac{(-1^n)(\Delta x_i)^n}{n!}\left(\frac{\partial^n\phi}{\partial x^n}\right)\Bigg|_i+H\end{aligned} \tag{2.4}$$

$$\left(\frac{\partial\phi}{\partial x}\right)\Bigg|_i=\frac{\phi(x_{i+1})-\phi(x_i)}{(\Delta x_{i+1})}-\frac{(\Delta x_{i+1})}{2!}\left(\frac{\partial^2\phi}{\partial x^2}\right)\Bigg|_i-H \tag{2.5}$$

$$\left(\frac{\partial\phi}{\partial x}\right)\Bigg|_i=\frac{\phi(x_i)-\phi(x_{i-1})}{(\Delta x_i)}+\frac{(\Delta x_i)}{2!}\left(\frac{\partial^2\phi}{\partial x^2}\right)\Bigg|_i\mp H \tag{2.6}$$

The Taylor series expansions in equation (2.3) and equation (2.4) can be combined to create another approximation of the derivative of ϕ at point (i). For a uniform grid or a uniformly changing grid the central-difference scheme (CDS) shown in equation (2.7) from the combination of equation (2.3) and equation (2.4) is second-order accurate [10].

$$\frac{\partial\phi}{\partial x}\Bigg|_i=\frac{\phi_{i+1}-\phi_{i-1}}{\Delta x_{i+1}+\Delta x_i}-\frac{(\Delta x_{i+1})^2-(\Delta x_i)^2}{2(\Delta x_{i+1}+\Delta x_i)}\left(\frac{\partial^2\phi}{\partial x^2}\right)\Bigg|_i\pm H \tag{2.7}$$

In many cases an approximation for the second derivative of a dependent variable with respect to a spatial variable is needed to solve the differential form of a conservation equation. The second derivatives also can be constructed by manipulating Taylor series expansions of $\phi(x)$ at points around (i) [10]. However, as shown in the following sections, in the case of the steady state neutron diffusion equation used in PARCS and the heat conduction equation used in TRACE, only an approximation for the first derivative is needed.

In this section the example domain used to illustrate the formulation of the FDS, BDS, and CDS approximations was described in terms of spatial dimensions. However, the finite difference methods discussed in this section are applicable to the approximation of derivatives with respect to time. In the discrete form of the

thermal-hydraulics, heat conduction, and neutronics equations presented in the following sections, the FDS is used to approximation all of the derivatives with respect to time.

2.1.1 Discretized Steady State Neutron Diffusion Equation

The steady state neutron diffusion equation is derived from a neutron continuity equation which includes the rate of production of neutrons through fission and the rate of loss of neutrons through absorption, scattering, and leakage [11]. A multiplication factor (k_{eff}) is included in the steady state neutron diffusion equation to indicate the overall balance between the production and loss of neutrons. In a critical reactor k_{eff} is equal to one which indicates that the rate of loss of neutrons in the reactor is exactly balanced by the rate of production of neutrons. If k_{eff} is less than one then the reactor is labeled as subcritical and the rate of loss of neutrons is greater than the rate of production of neutrons. Conversely, if k_{eff} is greater than one then the reactor is considered supercritical and the rate of production of neutrons is greater than the rate of loss of neutrons.

The numerical method for solving the steady state neutron diffusion equation in PARCS is known as the Coarse Mesh Finite Difference (CMFD) method [12]. The method is considered to utilize a coarse mesh because the distance between numerical computation points in the finite difference method is larger than the characteristic diffusion length of neutrons in the reactor [13]. However, even with the coarse mesh approach, accurate results are obtained through the use of current correction factors that reproduce the node-wise reaction rates and global eigenvalue that would be obtained from a higher order nodal solution [14]. The current correction factors will be discussed later in this section but are mentioned here to explain the reasoning behind the coarse mesh approach utilized in PARCS.

The formulation of the numerical solution method used in PARCS begins with the differential form of the steady state neutron diffusion equation shown in equation (2.8) [15]. The differential form of the neutron diffusion equation shown in equation (2.8) includes the Fickian diffusion approximation which will be addressed later in this section. Also, the continuous energy spectrum has already been discretized in the formulation of equation (2.8). A more detailed description of the

use of the Fickian diffusion approximation and the energy discretization can be found in several introductory nuclear engineering textbooks [11, 16].

$$-\nabla \cdot D_g(r)\nabla\phi_g(r) + \Sigma_{t,g}(r)\phi_g(r) = \sum_{g'\neq g}\Sigma_{g'g}(r)\phi_{g'}(r) + \frac{1}{k_{eff}}\sum_{g'}\nu(r)\chi_g(r)\Sigma_{f,g'}(r)\phi_{g'}(r) \tag{2.8}$$

To facilitate the formulation of the discrete neutron diffusion equation used in PARCS, a one-dimensional domain will be considered for simplicity and is shown in Figure 2.2. It is important, however, to distinguish between the domain shown in Figure 2.2 and the domain that is seen in literature detailing the formulation of the finite difference solution of the neutron diffusion equation [15, 17]. In some earlier derivations of the discrete form of the neutron diffusion equation, a "corner-mesh" nodalization was used where the computational nodes are located on the boundaries between homogeneous material volumes [17]. In the finite difference method used in PARCS, however, the computational nodes are at the center of the homogeneous material volume. Despite this difference in computational node locations, the same methodology in both cases is used to formulate the discrete form of the neutron diffusion equation [17]. More specifically, Taylor series expansions are used to expand fluxes and approximate derivatives in a small volume and the continuity of flux and current at computational volume boundaries is imposed [17].

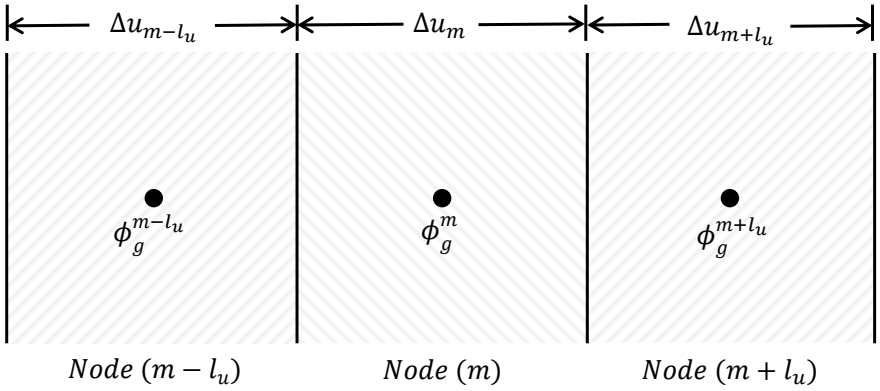


Figure 2.2: One-dimensional finite difference domain for the discretization of the steady state neutron diffusion equation

To derive the discrete form of equation 2.8 that is used in PARCS, first consider the computational node (m). Equation (2.8) can be integrated from $\left(m - \frac{\Delta u_m}{2}\right)$ to $\left(m + \frac{\Delta u_m}{2}\right)$ with the aim of reducing the second order derivative in equation (2.8) to a first order derivative. The integrated equation for the one dimensional domain shown in Figure (2.2) is shown in equation (2.9). Using the approximation that the material composition, and therefore cross sections and diffusion coefficients, are constant within node m, the integrals in equation 2.9 can be simplified as shown in equation (2.10).

$$\int_{m-\frac{\Delta u_m}{2}}^{m+\frac{\Delta u_m}{2}} \frac{d}{dx}\left(-D_g \frac{d\phi_g}{dx}\right) dx + \int_{m-\frac{\Delta u_m}{2}}^{m+\frac{\Delta u_m}{2}} \Sigma_{t,g}(r)\,\phi_g(r)\,dx = \tag{2.9}$$
$$\int_{m-\frac{\Delta u_m}{2}}^{m+\frac{\Delta u_m}{2}} \sum_{g' \neq g} \Sigma_{g'g}(r)\,\phi_{g'}(r)\,dx + \int_{m-\frac{\Delta u_m}{2}}^{m+\frac{\Delta u_m}{2}} \frac{1}{k_{eff}} \sum_{g'} \nu(r)\,\chi_g(r)\,\Sigma_{f,g'}(r)\,\phi_{g'}(r)\,dx$$

$$\left(-D_g \frac{d\phi_g}{dx}\right)\Bigg|_{m-\frac{\Delta u_m}{2}}^{m+\frac{\Delta u_m}{2}} + \Sigma_{t,g}^m \phi_g^m \cdot \Delta u_m \tag{2.10}$$
$$= \sum_{g' \neq g} \Sigma_{g'g}^m \phi_{g'}^m \cdot \Delta u_m + \frac{1}{k_{eff}} \sum_{g'} \nu^m \chi_g^m \Sigma_{f,g'}^m \phi_{g'}^m \cdot \Delta u_m$$

The integration transformation shown in equation (2.9) and equation (2.10) is

not consistent with a strict definition of the finite difference method presented in contemporary textbooks [10, 18]. Additionally, when the CMFD is written for a three-dimensional domain, it becomes readily apparent that the resulting equations are more consistent with the finite volume method described in Section 2.2 than the finite difference method. However, the literature describing the formulation of the CMFD method describes this integration step while still considering the solution approach as a finite difference method, and as a result, the CMFD method will be considered a finite difference approach in order to remain consistent with the nomenclature in the foundational literature [15, 17, 19]. A similar argument is used to include the discretized heat conduction equations as a finite difference method in Section 2.1.2.

The final step before obtaining the CMFD equations utilized in PARCS is to expand the derivative at the material boundaries in equation (2.10) as a function of the unknown fluxes and the known material properties of each region. The subtlety of this expansion is that a continuity of current must be enforced for the cells on the left and right of the material boundary [17]. To illustrate this point, the Fickian diffusion approximation to estimate neutron current is shown in equation (2.11). The neutron current approximation shown in equation (2.11) matches the form of the surface derivatives in equation (2.10) that must be evaluated at the boundary between material compositions. To evaluate the surface neutron current between node m and node $m + l_u$, a BDS can be used to approximate the derivative at the material interface using the flux and material properties from node m. Similarly if the neutron diffusion equation were integrated over the volume surrounding node $m + l_u$ then the neutron current between node m and node $m + l_u$ could be written using a FDS approximation for the surface derivative using the flux and material properties from node $m + l_u$. The estimation of the surface neutron current using BDS and FDS methods are shown in equation (2.12) and equation (2.13), respectively.

$$J = -D\nabla\phi \tag{2.11}$$

$$J_{BDS} = -2D_g^m \frac{\phi_g^m - \phi_g^{m+\frac{\Delta u_m}{2}}}{\Delta u_m} \tag{2.12}$$

$$J_{FDS} = -2D_g^{m+l_u} \frac{\phi_g^{m+l_u} - \phi_g^{m+\frac{\Delta u_m}{2}}}{\Delta u_{m+l_u}} \tag{2.13}$$

Because the flux must be a continuous function, the neutron current should also be a continuous function. This means that the current calculated from the left of the boundary should be equal to the current that is calculated from the right of the boundary [20]. Equation (2.12) can be manipulated to yield an expression for the flux at the boundary which can be substituted into equation (2.13). With this manipulation and substitution an expression for the current at the surface that is consistent for both cells on either side of the material interface is obtained and is shown in equation (2.14)

$$J_{gu}^{m+} = \frac{-D_g^{m+l_u} D_g^m}{D_g^{m+l_u} \Delta u_m + D_g^m \Delta u_{m+l_u}} \left(\phi_g^{m+l_u} - \phi_g^m\right) \tag{2.14}$$

The average diffusion coefficient in equation (2.14) is referred to in PARCS as the base nodal coupling coefficient [12]. This coefficient is augmented by a corrective nodal coupling coefficient which forces the finite difference surface neutron current to match the value that is obtained for a higher-order transport calculation [12]. The definition of the base nodal coupling coefficient and the surface neutron current with the inclusion of the corrective nodal coupling coefficient are shown in equation (2.15) and equation (2.16), respectively [12].

$$\tilde{D}_{gu}^{m\pm} = \frac{2D_g^{m\pm l_u} D_g^m}{D_g^{m\pm l_u} \Delta u_m + D_g^m \Delta u_{m\pm l_u}} \tag{2.15}$$

$$J_{gu}^{m\pm} = \mp \tilde{D}_{gu}^{m\pm} \left(\phi_g^{m\pm l_u} - \phi_g^m\right) - \hat{D}_{gu}^{m\pm} \left(\phi_g^{m\pm l_u} + \phi_g^m\right) \tag{2.16}$$

The expression shown in equation (2.16) can be substituted back into equation (2.10) to complete the derivation of CMFD form of the neutron diffusion equation that is utilized in PARCS. The general form of this equation is shown in equation (2.17). Note, however, that discussion of the methods for solving equation (2.17) are left for Chapter 4.

$$0 = \frac{1}{k_{eff}} \chi_g \sum_g \nu_g \Sigma_{fg}^m \phi_g^m + \sum_{g'} \Sigma_{g'g}^m \phi_{g'}^m - \sum_{u=x,y,z} \frac{1}{h_u^m} \left(J_{gu}^{m+} - J_{gu}^{m-}\right) - \Sigma_{tg}^m \phi_g^m \tag{2.17}$$

In the interest of simplicity when presenting expressions included in the implicit steady state solution method, the coefficients below are used to rearrange the neutron leakage terms shown in equation (2.17).

$$a_{gu}^{m-} = \frac{-1}{h_u^m}\left(\widetilde{D}_{gu}^{m-} - \hat{D}_{gu}^{m-}\right) \tag{2.18}$$

$$a_{gu}^{m} = \frac{1}{h_u^m}\left(\widetilde{D}_{gu}^{m+} + \widetilde{D}_{gu}^{m-} - \hat{D}_{gu}^{m+} + \hat{D}_{gu}^{m-}\right) \tag{2.19}$$

$$a_{gu}^{m+} = \frac{-1}{h_u^m}\left(\widetilde{D}_{gu}^{m+} + \hat{D}_{gu}^{m+}\right) \tag{2.20}$$

Using equation (2.18), equation (2.19), and equation (2.20), the steady state nodal neutron diffusion equation is rewritten in a form that will be used throughout the rest of this document as the analytic Jacobian is constructed. This form of the steady state nodal neutron diffusion equation is shown in (2.21).

$$0 = \lambda\chi_g \sum_g \nu_g \Sigma_{fg}^m \phi_g^m + \sum_{g'} \Sigma_{g'g}^m \phi_{g'}^m - \sum_{u=x,y,z}\left[a_{gu}^{m-}\phi_{gu}^{m-} + a_{gu}^{m}\phi_{gu}^{m} + a_{gu}^{m+}\phi_{gu}^{m+}\right] - \Sigma_{tg}^m \phi_g^m \tag{2.21}$$

2.1.2 Discretized Heat Transfer Equations

The heat conduction solution in TRACE is derived from a differential form of the heat conduction equation. Using Fourier's law of conduction, the differential form of the heat conduction equation can be written as shown in equation (2.22) [21].

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + q''' \tag{2.22}$$

Like in the derivation of the CMFD equations used in PARCS, the differential form of the heat conduction equations are integrated to reduce the order of the derivative of the temperature field. Because of the similarity of this integration step to the integration operation shown in the previous section, it will not be shown here. However, it is worth noting that the computational stencil of the heat conduction solution differs slightly from neutronics stencil. More specifically, in the heat structure computational stencil the computational nodes lie on the intersection between homogeneous material volumes in the radial direction [21]. Because

of the position of the computational nodes, the discrete heat conduction equation includes terms that account for the likely occurrence of varying material properties on either side of the computational node. As an example, the computational mesh for a heated rod is shown in Figure 2.3 [21].

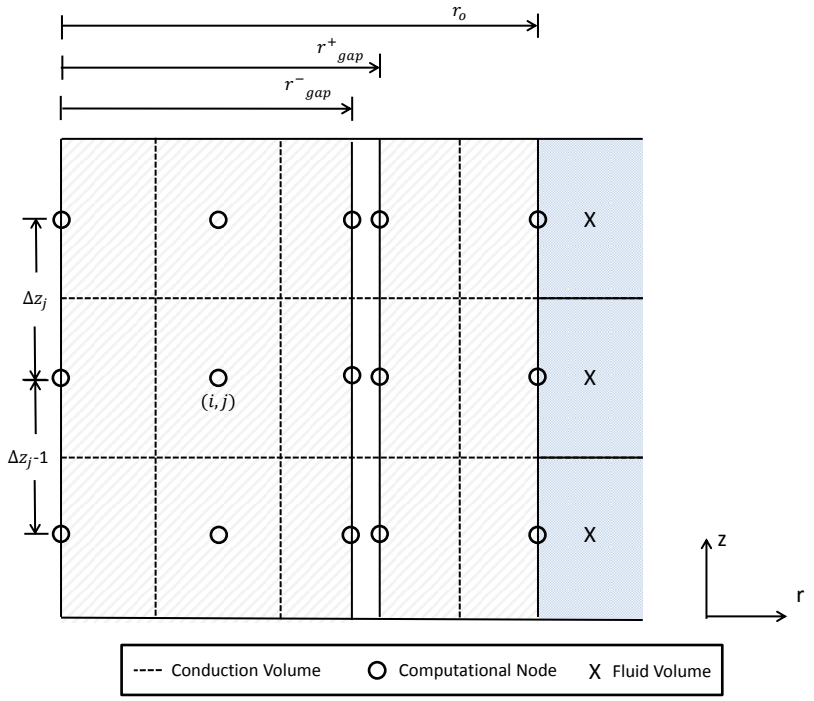


Figure 2.3: Fuel pin finite difference discretization for the heat conduction solution in TRACE

The discrete form of the heat conduction equation for an interior node that would be used to solve for the temperature distribution in Figure 2.3 is shown in equation (2.23). In equation (2.23) temporal superscripts are used for completeness even though time-integration techniques have not yet been discussed. The implications of the choice of temporal superscripts will be discussed in later sections; however it is worth noting that in equation (2.23) the axial conduction contributions to the equation are treated explicitly while the radial conduction contributions are treated implicitly. Also, material properties that are needed in equation (2.23) at computational volume boundaries or average values over the volume are determined using a spatial weighting or averaging procedures described

in the TRACE theory manual [21].

$$\left[\left(\rho c_p\right)_{ij}\frac{T_{ij}^{n+1}-T_{ij}^{n}}{\Delta t}-q_{ij}'''\right]\frac{1}{2}\left[\left(r_i\Delta r_i+\frac{\Delta r_i^2}{4}\right)+\left(r_i\Delta r_{i-1}-\frac{\Delta r_{i-1}^2}{4}\right)\right] \tag{2.23}$$
$$\times\left[\frac{\Delta z_j+\Delta z_{j-1}}{2}\right]=\left[r_{i+\frac{1}{2}}k_{i+\frac{1}{2},j}\left(\frac{T_{i+1,j}^{n+1}-T_{ij}^{n+1}}{\Delta r_i}\right)+r_{i-\frac{1}{2}}k_{i-\frac{1}{2},j}\left(\frac{T_{i-1,j}^{n+1}-T_{ij}^{n+1}}{\Delta r_{i-1}}\right)\right]$$
$$\times\left[\frac{\Delta z_j+\Delta z_{j-1}}{2}\right]+\left[k_{i,j+\frac{1}{2}}\left(\frac{T_{i,j+1}^{n}-T_{ij}^{n}}{\Delta z_j}\right)+k_{i,j-\frac{1}{2}}\left(\frac{T_{i,j-1}^{n}-T_{ij}^{n}}{\Delta z_{j-1}}\right)\right]$$
$$\times\frac{1}{2}\left[\left(r_i\Delta r_i+\frac{\Delta r_i^2}{4}\right)+\left(r_i\Delta r_{i-1}-\frac{\Delta r_{i-1}^2}{4}\right)\right]$$

The discrete form of the heat conduction equation shown in equation (2.23) is applicable to an interior computational node. For a node that lies on the centerline or fluid interface, a symmetry boundary condition or convective heat transfer boundary condition must be enforced, respectively. Of particular interest to the objectives of this research is the convective heat transfer boundary condition because it couples the heat transfer equations to the thermal-hydraulic equations. Equation (2.24) shows the formulation of the heat flux that is used to couple the conduction solution to the energy equation of the thermal-hydraulic solution [21]. Like before, superscripts related to the time-integration method in TRACE have been included for completeness. These temporal superscripts show that the TRACE solution treats heat transfer coefficients explicitly while treating the wall, liquid, and gas temperatures implicitly.

$$q_{total}^{n+1}=h_l^n\left(T_w^p-T_l^n\right)+h_g^n\left(T_w^p-T_g^n\right)+\left[h_l^n\left(\frac{\partial T_w}{\partial T_l}-1\right)+h_g^n\frac{\partial T_w}{\partial T_l}\right]\left(T_l^{n+1}-T_l^n\right) \tag{2.24}$$
$$+\left[h_g^n\left(\frac{\partial T_w}{\partial T_g}-1\right)+h_l^n\frac{\partial T_w}{\partial T_g}\right]\left(T_g^{n+1}-T_g^n\right)$$

2.2 Finite Volume Method

The finite volume method was first introduced by McDonald (1971) and MacCormack and Paullay (1972) to solve the two-dimensional time-dependent Euler Equations [18, 22, 23]. Rizze and Inouye (1973) extended the finite volume method to a

three-dimensional formulation in order to model blunt-body flow [18, 24]. Since its initial introduction, the finite volume method has become a very popular solution method in engineering applications such as Computational Fluid Dynamics (CFD) because of its adaptability to complex geometries [18]. The finite volume method is important to the work presented in this document because it is the numerical method used in TRACE to solve the thermal-hydraulic equations.

The first step in applying the finite volume method to a particular problem is to divide the domain into control volumes with the centroid of the control volume being the point at which principle unknowns are evaluated. The grid that is used to define control volume surfaces can be structured or unstructured which allows for complex geometries to be discretized [18]. For each control volume in the domain the integral form of the conservation law is solved by using numerical approximations for volume and surface integrals. The simplest and most common approximation for surface integral is the midpoint rule which is the product of the cell-face center value and the surface area. This approximation of the surface integral is second-order accurate [10]. Similarly, volume integrals are often approximated as the product of the volume-centered value of the variable and the volume. This approximation for the volume integral is second-order accurate [10].

One of the limitations of the finite volume method stems from the combined approximation of surface integrals being evaluated with a discrete number of points on the volume face with a subsequent interpolation scheme to express the face values in terms of control volume centered values [18]. This two-step numerical approximation makes it more difficult to implement discretization schemes more accurate than a second order approximation because both the surface integral approximation and interpolation method must have the same order of accuracy. However, this is only a small disadvantage compared to the robustness of the finite volume method [18].

2.2.1 Finite Volume Thermal-Hydraulic Equations

The finite volume method used in TRACE differs slightly from the finite volume method that one would see in most contemporary thermal-hydraulic solution methods. In modern thermal-hydraulic solutions it is common to have all of the

variables colocated on the same mesh. However, an older staggered mesh technique is used in TRACE to stagger the mesh for the momentum conservation equation relative to the mesh that is used for the mass and energy conservation equations. An example of the staggered mesh is shown in Figure 2.4 [21]. The principle advantage of a staggered arrangement for the equation set is improved coupling between velocities and pressure [10]. Without the staggered mesh arrangement, it is possible to observe convergence issues related oscillations in pressure and velocity fields [10]. The staggered mesh arrangement, however, is difficult to implement with complex geometries in an arbitrary coordinate system, and improved methods for coupling the velocity and pressure fields has made the colocated method generally preferred over the staggered approach.

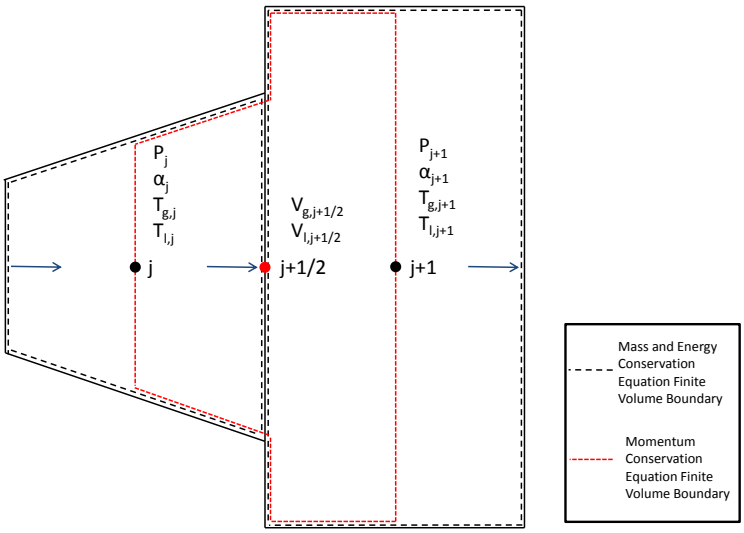


Figure 2.4: Staggered mesh diagram for finite volume method used in TRACE

The mesh shown in Figure 2.4 is an example of a staggered finite volume discretization given in the TRACE theory manual [21]. In this particular example the principle unknowns in the mass and energy conservation equations are evaluated at the volume centered locations (j) and (j+1), and the principle unknown for the momentum equation is evaluated at the volume centered location (j+1/2). Note that the faces associated with the mass and energy conservation equations finite volumes are located at the same point as the center of the momentum conservation equation finite volume. Similarly, the faces of the momentum conservation equa-

tion finite volume are colocated with the center of the mass and energy conservation equations finite volume centers.

The staggered mesh approach directly impacts the interpolation schemes that are necessary in the discretized thermal-hydraulic conservation equations. More specifically, in the mass and energy conservation equations the flux terms across the finite volume boundaries would normally require interpolation to approximate the velocity at the surface if a colocated mesh is used. However, in the staggered mesh approach the velocity variable coincides with the surface of the mass or energy conservation equation volume and interpolation is not needed. When state variables are needed for the surface integrals in the mass and energy conservation equations, the values are used from the nearest upstream volume center as determined by the fluid velocity contained in the surface integral [21]. The use of volume center state variable as an approximation for surface values on the boundaries of the finite volume is a first order approximation [21].

The development of the discretized momentum equations is more difficult than the process used for the discretized mass and energy conservation equations. The first additional complexity introduced into the discrete momentum is the ability in TRACE to have discontinuities in flow area on the edges of mass and energy conservation equation finite volumes which coincides with the center of momentum conservation equations as shown in Figure 2.4 [21]. An additional complexity that is added by the staggered finite volume discretization is that there are different thermodynamic state variables on either side of the center of the momentum volume because the thermodynamic properties are considered constant over the mass and energy conservation equation finite volumes [21]. These complexities are handled in part by manipulating mass conservation equations for the left and right halves of the momentum conservation equation finite volume and keeping constant volumetric flow on either side of the momentum finite volume center [21]. A detailed description of the derivation of the discrete momentum equations is provided in the TRACE theory manual [21].

The discrete form of the thermal-hydraulic equations used in TRACE will not be derived in this document because it would unnecessarily burden the reader with details that will not be helpful in later sections of this document. It is worthwhile, however, to present the six-equation thermal-hydraulic model that is

implemented in TRACE. The equations are presented in differential form to be consistent with the TRACE theory manual. However, as stated in the theory manual, Gauss' Theorem is first utilized to transform the equations into integral form before writing the discrete form of the finite volume equations [21].

The liquid and gas mass equations are given in equation (2.25) and (2.26), respectively. The liquid and gas momentum conservation equations are shown in equation (2.27) and equation (2.28), respectively. The mixture and gas energy equations are shown in equation (2.29) and equation (2.30), respectively. Finally, a seventh equation can be added for non-condensible gases, and it is shown in equation (2.31)

$$\frac{\partial\left(1-\alpha\right)\rho_l}{\partial t}+\nabla\cdot\left(\left(1-\alpha\right)\rho_l\overrightarrow{V}_l\right)=-\Gamma_i \tag{2.25}$$

$$\frac{\partial\alpha\rho_g}{\partial t}+\nabla\cdot\left(\alpha\rho_g\overrightarrow{V}_g\right)=\Gamma_i \tag{2.26}$$

$$\begin{aligned}\left(1-\alpha\right)\rho_l\left(\frac{\partial\overrightarrow{V}_l}{\partial t}+\overrightarrow{V}_l\cdot\nabla\overrightarrow{V}_l\right)=&-\left(1-\alpha\right)\nabla P+\left(1-\alpha\right)\rho_l\overrightarrow{g}\\&+f_i+f_{wl}-\Gamma\left(\overrightarrow{V}_i-\overrightarrow{V}_l\right)\end{aligned} \tag{2.27}$$

$$\alpha\rho_g\left(\frac{\partial\overrightarrow{V}_g}{\partial t}+\overrightarrow{V}_g\cdot\nabla\overrightarrow{V}_g\right)=-\alpha\nabla P+\alpha\rho_g\overrightarrow{g}-f_i+f_{wg}-\Gamma\left(\overrightarrow{V}_g-\overrightarrow{V}_i\right) \tag{2.28}$$

$$\begin{aligned}&\frac{\partial\left(\alpha\rho_g e_g+\left(1-\alpha\right)\rho_l e_l\right)}{\partial t}+\nabla\cdot\left[\alpha\rho_g e_g\overrightarrow{V}_g+\left(1-\alpha\right)\rho_l e_l\overrightarrow{V}_l\right]\\&=-P\nabla\cdot\left[\alpha\overrightarrow{V}_g+\left(1-\alpha\right)\overrightarrow{V}_l\right]+q_{wg}+q_{wl}+q_{dg}+q_{dl}+q_{wsat}\end{aligned} \tag{2.29}$$

$$\begin{aligned}&\frac{\partial\left(\alpha\rho_g e_g\right)}{\partial t}+\nabla\cdot\left(\alpha\rho_g e_g\overrightarrow{V}_g\right)\\&=-P\left[\frac{\partial\alpha}{\partial t}+\nabla\cdot\left(\alpha\overrightarrow{V}_g\right)\right]+\Gamma h_{v'}+q_{wg}+q_{ig}+q_{dg}\end{aligned} \tag{2.30}$$

$$\frac{\partial \alpha \rho_a}{\partial t} + \nabla \cdot \left(\alpha \rho_a \overrightarrow{V}_g \right) = 0 \tag{2.31}$$

The discrete thermal-hydraulic equations are derived in detail in the TRACE theory manual [21]. The semi-implicit form of the discrete thermal-hydraulic equations are important in this research, and as a result, they will be presented below. An alternate set of the discrete thermal-hydraulic equations can be used if the SETS solution scheme in TRACE is utilized; however the SETS equations were not implemented in the implicit steady state solution methods, and therefore, will not be presented in this document. The discrete from of equation (2.25) through equation (2.31) is shown in equation (2.32) through equation (2.38), respectively.

$$\frac{\left(1-\alpha_j^{n+1}\right)\rho_{lj}^{n+1} - \left(1-\alpha_j^{n}\right)\rho_{lj}^{n}}{\Delta t} + \nabla_j \cdot \left[\left(1-\alpha^n\right)\rho_l^n V_l^{n+1} \right] = -\Gamma^{n+1} \tag{2.32}$$

$$\frac{\alpha_j^{n+1}\rho_{gj}^{n+1} - \alpha_j^{n}\rho_{gj}^{n}}{\Delta t} + \nabla_j \cdot \left[\alpha^n \rho_g^n V_g^{n+1} \right] = \Gamma^{n+1} \tag{2.33}$$

$$\frac{V_l^{n+1} - V_l^n}{\Delta t} + V_l^{n+1} \nabla_{j+\frac{1}{2}} V_l^n + \frac{C_i^n \left| V_l^n - V_g^n \right|}{\left\langle \left(1-\alpha\right)\rho_l \right\rangle_{j+\frac{1}{2}}^n} \left[2\left(V_l^{n+1} - V_g^{n+1} \right) - \left(V_l^n - V_g^n \right) \right] \tag{2.34}$$
$$+ \frac{1}{\left\langle \rho_l \right\rangle_{j+\frac{1}{2}}^n} \frac{P_{j+1}^{n+1} - P_j^{n+1}}{\Delta x_{j+\frac{1}{2}}} + \frac{\Gamma_{j+\frac{1}{2}}^{-n}}{\left\langle \left(1-\alpha\right)\rho_l \right\rangle_{j+\frac{1}{2}}^n} \left(V_l^{n+1} - V_g^{n+1} \right)$$
$$+ \frac{C_{wl}}{\left\langle \left(1-\alpha\right)\rho_l \right\rangle_{j+\frac{1}{2}}^n} \left(2V_l^{n+1} - V_l^n \right) \left| V_l^n \right| + g \times cos\left(\theta\right) = 0$$

$$\frac{V_g^{n+1} - V_g^n}{\Delta t} + V_g^{n+1} \nabla_{j+\frac{1}{2}} V_g^n + \frac{C_i^n \left| V_g^n - V_l^n \right|}{\left\langle \alpha \rho_g \right\rangle_{j+\frac{1}{2}}^n} \left[2\left(V_g^{n+1} - V_l^{n+1} \right) - \left(V_g^n - V_l^n \right) \right] \tag{2.35}$$
$$+ \frac{1}{\left\langle \rho_g \right\rangle_{j+\frac{1}{2}}^n} \frac{P_{j+1}^{n+1} - P_j^{n+1}}{\Delta x_{j+\frac{1}{2}}} + \frac{\Gamma_{j+\frac{1}{2}}^{+n}}{\left\langle \alpha \rho_g \right\rangle_{j+\frac{1}{2}}^n} \left(V_g^{n+1} - V_l^{n+1} \right)$$
$$+ \frac{C_{wg}}{\left\langle \alpha \rho_g \right\rangle_{j+\frac{1}{2}}^n} \left(2V_g^{n+1} - V_g^n \right) \left| V_g^n \right| + g \times cos\left(\theta\right) = 0$$

$$\frac{\left[\alpha_j^{n+1}\rho_{gj}^{n+1}e_{gj}^{n+1}+\left(1-\alpha_j^{n+1}\right)\rho_{lj}^{n+1}e_{lj}^{n+1}\right]-\alpha_j^n\rho_{gj}^n e_{gj}^n+\left(1-\alpha_j^n\right)\left(\rho_{lj}^n e_{lj}^n\right)}{\Delta t} \tag{2.36}$$
$$+\nabla_j\cdot\left[\alpha^n\rho_g^n e_g^n V_g^{n+1}+\left(1-\alpha^n\right)\rho_l^n e_l^n V_l^{n+1}\right]$$
$$+P^{n+1}\nabla_j\cdot\left[\left(1-\alpha^n\right)V_l^{n+1}+\alpha^n V_g^{n+1}\right]=q_{wg}^{n+1}+q_{wsat}^{n+1}+q_{dl}^n+q_{dg}^n+q_{wl}^{n+1}$$

$$\frac{\left[\alpha_j^{n+1}\rho_{gj}^{n+1}e_{gj}^{n+1}-\alpha_j^n\rho_{gj}^n e_{gj}^n\right]}{\Delta t}+\nabla_j\cdot\left[\alpha^n\rho_g^n e_g^n\overrightarrow{V}_g^{n+1}\right] \tag{2.37}$$
$$+P^{n+1}\left[\frac{\alpha^{n+1}-\alpha^n}{\Delta t}+\nabla_j\cdot\left(\alpha^n\overrightarrow{V}_g^{n+1}\right)\right]=q_{wg}^{n+1}+q_{dg}^n+q_{ig}^{n+1}+\Gamma^{n+1}h_{sg}^{n+1}$$

$$\frac{\alpha_j^{n+1}\rho_{aj}^{n+1}-\alpha_j^n\rho_{aj}^n}{\Delta t}+\nabla_j\cdot\left[\alpha^n\rho_a^n V_g^{n+1}\right]=0 \tag{2.38}$$

The interfacial mass transfer rate that is used in equation (2.32), equation (2.33), and equation (2.36) is defined in equation (2.39). The wall heat transfer to the fluid and gas phases is defined in equation (2.40) and equation (2.41), respectively. Also, the interfacial heat transfer is defined in equation (2.42) and equation (2.43), respectively. The wall heat transfer to the liquid is particularly important to the implicit solution method because it is the coupling mechanism between the heat conduction solution and the thermal-hydraulics solution.

$$\Gamma^{n+1}=\frac{-\left(q_{ig}^{n+1}+q_{il}^{n+1}\right)}{\left(h_v'\right)^{n+1}-\left(h_l'\right)^{n+1}} \tag{2.39}$$

$$q_{wl}^{n+1}=h_{wl}^n a_w\left(T_w^{n+1}-T_l^{n+1}\right) \tag{2.40}$$

$$q_{wg}^{n+1}=h_{wg}^n a_w\left(T_w^{n+1}-T_g^{n+1}\right) \tag{2.41}$$

$$q_{il}^{n+1}=h_{il}^n a_i^n\left(T_{sat}^{n+1}-T_l^{n+1}\right) \tag{2.42}$$

$$q_{ig}^{n+1}=h_{ig}^n a_i^n\left(T_{sat}^{n+1}-T_g^{n+1}\right) \tag{2.43}$$

Chapter 3

TRACE and PARCS Solution Method

The discrete system of equations for the thermal-hydraulic, heat conduction, and reactor physics calculations in TRACE and PARCS are a tightly coupled set of nonlinear equations. The existing explicit time-integration method used in TRACE and PARCS to solve the nonlinear set of equations will be described in Section 3.1. Using the implicit transient solution as a template, the framework of the implicit steady state solution method will be described in Section 3.2. In Section 3.3 a TRACE model used to evaluate the proposed implicit steady state solution is described. The description of the TRACE model will be useful in Chapter 4 when the connectedness of the different computational meshes impacts the derivation of the implicit equation set.

3.1 Explicit Steady State Solution Method

The thermal-hydraulic equations in TRACE are formulated in a semi-implicit manner with an optional stability-enhancing two-step (SETS) solution method [21]. The SETS method builds on the semi-implicit thermal-hydraulic equations by adding motion, mass, and energy stabilizing equations to reduce inherent time step restrictions in the semi-implicit formulation [21]. However, the SETS method introduces numerical diffusion into the hydraulic solution, and as a result, is not ideal for every analysis [21]. The semi-implicit thermal-hydraulic equation set is

used in the coupled implicit solution method, and therefore, further mention of the explicit solution in TRACE/PARCS will refer to the basic semi-implicit formulation.

In the TRACE/PARCS explicit solution the thermal-hydraulic equations are coupled to the heat conduction equations via the heat flux from the heat structures to the fluid. At the beginning of the time step in TRACE, the heat conduction equations are solved, and the resulting heat structure temperatures are written as a function of the change in surface heat flux. After each hydraulic solution the wall heat flux is used to calculate new heat structure nodal temperatures. It is worth noting that the heat transfer coefficients that are used in the heat structure and fluid coupling are evaluated explicitly at the beginning of the calculation in the TRACE transient solver.

After the thermal-hydraulic solution has converged in TRACE, the general information passing interface is used to pass the thermal-hydraulic information to PARCS for a reactor physics calculation. The thermal-hydraulic solution is used to generate cross sections, and a power-method iteration scheme is used to solve the discrete form of the steady state neutron diffusion equation. The general form of the neutron diffusion equation used in the power method in PARCS is shown in equation (3.1).

$$\left(M - \frac{1}{k_{eff}^{k}}\right)\phi^{k+1} = \left(\frac{1}{k_{eff}^{k}} - \frac{1}{k_{s}^{k}}\right)F\phi^{k} \tag{3.1}$$

The shifted eigenvalue value in equation (3.1) is calculated with user-defined eigenvalue shift as shown in equation (3.2), and the eigenvalue update for the shifted eigenvalue is shown in equation (3.3) and equation (3.4).

$$k_{s}^{k} = k_{eff}^{k} + \delta k \tag{3.2}$$

$$k_{eff}^{k+1} = \left[\frac{\gamma}{k_{eff}^{k}} + \frac{1-\gamma}{k_{s}^{k}}\right]^{-1} \tag{3.3}$$

$$\gamma = \frac{\langle \Psi^{n+1}, \Psi^{n} \rangle}{\langle \Psi^{n+1}, \Psi^{n+1} \rangle} \tag{3.4}$$

The power method iterations are performed for a user-specified number of it-

erations or until the flux and eigenvalue have converged. After either of the aforementioned conditions is met, updates for the next set of PARCS calculations are performed. These updates include, but are not limited to, xenon and samarium number density updates and corrective nodal coupling coefficient updates. When these updates are complete, the power distribution is passed to TRACE through the general information passing interface for the next thermal-hydraulic and heat transfer solution.

Each time step of the solution is composed of a sequential solution of thermal-hydraulic and heat transfer equations in TRACE and the the steady state neutron diffusion equation in PARCS as described in this section. The null-transient calculation is terminated after TRACE determines that the system has reached steady state conditions or the number of time steps has reached the user-specified maximum value. TRACE determines if the model is at steady state conditions by comparing the maximum fractional change per second of the total pressure, liquid velocity, steam velocity, steam volume fraction, liquid temperature, steam temperature, and noncondensable-gas pressure in each finite volume of the model to a user-specified convergence value every fifth time-step [25].

3.2 Implicit Steady State Solution Method

The explicit and implicit steady state solution methods have the same sequence of TRACE and PARCS program executions, but the purpose of each program execution is much different in the implicit method. At the beginning of the time step in the implicit solution, TRACE solves the semi-implicit form of the thermal-hydraulic equations; however, the heat structure equations and steady state neutron diffusion equations are also solved at the same time as the thermal-hydraulic equations. The three sets of equations are solved simultaneously using Newton's method with an analytic construction of the required Jacobian matrix and direct linear solver for the resulting system of equations. Details of the analytic expressions used in the Jacobian matrix are presented in Chapter 4.

The Newton's method iterations in TRACE are terminated for the time step when the user-specified convergence criteria for the thermal-hydraulic variables and the neutronics variables are both satisfied. After the implicit calculation has

converged, the thermal-hydraulic solution is sent to PARCS using the general information passing interface. The thermal-hydraulic information is used to generate a new set of cross sections, cross section derivatives, coupling coefficients, and coupling coefficient derivatives that are used in the analytic Jacobian matrix. Unlike the explicit solution, the neutron diffusion equation is not solved in PARCS during a time-step; and as a result, the power profile passed from PARCS to TRACE through the general information passing interface is identical to the power profile that had been calculated in TRACE. However, the coefficients that would have been used in the power method solution of the steady state neutron diffusion equation are stored in arrays accessible in TRACE through module files to be used as the first iterate values in the next coupled calculation. Module files are used to share information between TRACE and PARCS instead of a general information passing interface in the implicit solution because of efficiency concerns regarding packing and unpacking coefficient and coefficient derivative for each neutronics node in the model.

Although the implicit method uses the same semi-implicit formulation of the thermal-hydraulic equations as the explicit method, the implicit transient solution scheme uses a different coupling mechanism for the hydraulic and heat transfer equations. More specifically, instead of using surface heat fluxes to couple the equation sets, wall temperatures are used to couple the two equation sets [3]. The wall temperature coupling mechanism that will be used in the implicit steady state solution is unchanged from the method introduced in the implicit transient solution [3].

The general structure of the implicit solution method presented in this section was implemented as part of the transient implicit solution method in TRACE and PARCS [3]. However, there is a significant difference in the set of neutronics equations that must be solved, and as a byproduct, the coupling between the heat transfer and the neutronics equations in the implicit steady state solution also differs from the implicit transient solution. The aforementioned differences are addressed in more detail in Chapter 4 when the implicit equation set is presented.

3.3 Model Description

The results and comparisons presented in this thesis will be based partly on a heated pipe model modeled in TRACE to resemble a BWR channel. The model was built using the Peach Bottom Turbine Trip (PBTT) benchmark initial channel lattice dimensions [26]. Additionally, the PBTT cross section files were utilized as input data for the PARCS cross section subroutines. A simplified graphical representation of the heated pipe structure built in TRACE is shown in Figure 3.1. Note that throughout this document the model shown in Figure 3.1 will be referred to as the "BWR channel model" or "channel model" which is different from the channel component in TRACE.

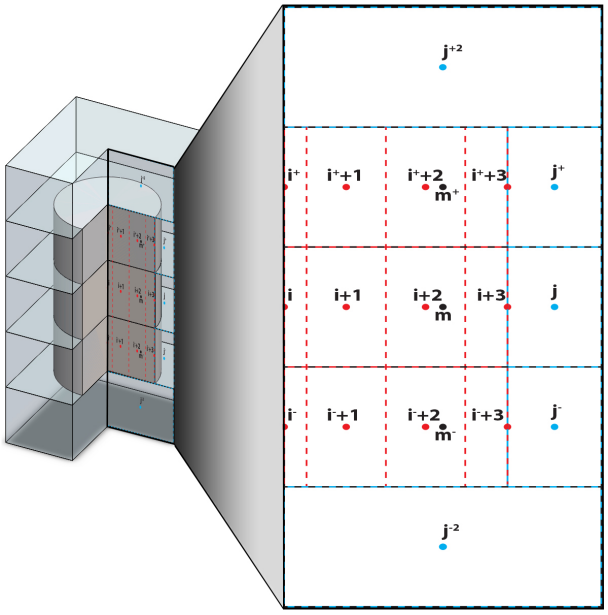


Figure 3.1: TRACE pipe component based on PBTT benchmark initial channel lattice dimensions for implicit steady state solution evaluation.

The BWR channel model depicted in Figure 3.1 consists of a pipe, heat structure, power, fill, and break component in the TRACE input deck. The fill component is positioned at the bottom of the BWR channel and is used to specify the mass flow boundary condition while the break component is positioned at the top of the BWR channel and is used to specify the pressure boundary condition [21]. In this model the mass flow was set to 12.65 kg/s for the fill component, and the pressure was set to 6.864 MPa. The pipe component is discretized into five axial

cells with the three center cells coupled to the heat structure and power components. The heat structure component consists of three axial cells with each axial cell discretized into four radial cells. The power component which comprises the neutronics domain also consists of three axial cells which overlay the heat structure and pipe component cells.

The schematic of the BWR channel mesh shown in Figure 3.1 will become particularly important in later sections when the spatial connectedness of the meshes directly impacts the derivative expressions used in the analytic Jacobian. More specifically, it is important to realize that each neutronics node overlaps a hydraulics node and more than one heat structure node. Also, it will be important to distinguish in later sections those cells that are adjacent to a cell of interest from the rest of the cells in the computational domain. In this document the neutronics cell of interest is denoted by m, and its neighboring cell in the positive and negative directions are denoted by $m+$ and $m-$, respectively. In this particular model the neutronics mesh is one dimensional; however, in the equation sets discussed in later sections a subscript u is used for the general case when the neutronics mesh is two or three dimensional.

A similar convention for nodal labeling is used for hydraulic and heat transfer meshes. In the neutronics equations presented in Chapter 4 a hydraulic volume coupled to the neutronics node of interest m will be denoted by j. The heat structure nodes directly coupled to the neutronics cell will be denoted by i, $i+1$, ... $i+nr$ starting from the fuel centerline and ending at the interface between the fuel and fluid. Also, the same positive and negative superscript notation to indicate adjacent cells to the cell of interest that was used for the neutronics cells will be used for the heat structure and fluid cells.

In Figure 3.1 the labels for each node are written relative to the second axial neutronics node. When presenting the implicit equation set in Section 4, it will be important to distinguish those cells that are coincident with the neutronics cell m from all other cells in the domain. Throughout this document neutronics cells other than m will be referred to generically as M. Similarly, hydraulic and heat transfer cells not coincident with the neutronics cell m will be referred to using J and I, respectively. In the mesh shown in Figure 3.1 the hydraulic nodes labeled j^{-2}, j^{-}, j^{+}, and j^{+2} would be contained in the set of hydraulic cells J because they

are not coincident with the neutronics node m. Similarly, all heat structures nodes other than i, $i+1$, $i+2$, and $i+3$ are part of the set of heat structure cells I because they are not coincident with the neutronics node m. The same justification is used to group the neutronics nodes labeled as m^+ and m^- into the set of neutronics cells M. It is important to note that the nodes contained in the sets I, J, and M is dependent on which neutronics node is being referenced.

In Chapter 4 it will also be important to distinguish between fuel temperatures coinciding with the same neutronics volume. For example, if heat structure node $i+1$ is the volume of interest, then the heat structure nodes i, $i+2$, and $i+3$ will be reference generically by the set i'. Like the other complimentary sets discussed above, the nodes contained in i' will depend on particular heat structure volume of interest.

In addition to the five-node problem shown in Figure 3.1, a fourteen-node model was also created using the same geometry and boundary conditions. To create this model the three center hydraulic cells, axial heat structure cells, and neutronics cells were split equally into a total of twelve cells. This model was created to assist in evaluating the scaling capabilities of the proposed implicit steady state solution. A four-channel reactor core model will be introduced in Section 5.2.1 that also tests the scaling capabilities of the implicit steady state solution method. However, the four-channel core model requires significantly more CPU runtime than the fourteen-node model, and as a result, some of the scaling features of the implicit solution are better addressed with the fourteen-node model.

Chapter 4

Implicit Solution Method

An essential characteristic of a fully implicit time integration technique is that all of the coefficients and variables, aside from those that might be introduced by a finite difference approximation of a time derivative, are evaluated at the future time step level. With regards to the steady state neutron diffusion equation shown in equation (2.21), this means that the cross sections, diffusion coefficients, base nodal coupling coefficient, corrective nodal coupling coefficient, fluxes, and eigenvalue are all updated and converged within a time step. In Section 4.1 Newton's method will be presented because it is used to solve the implicit set of equations that will be derived later in the chapter. The material presented in Section 4.1 will help motivate a large portion of the material contained in this chapter. The remainder of this chapter will focus on the derivation of the implicit form of the neutronics and heat conduction equations that are a unique contribution of the implicit steady state solution. In Section 4.2 the formulation of the cross section and cross section derivatives used in the implicit solution will be presented. Information regarding the approximation of the cross section derivatives will be pertinent to the derivation of the discrete form of the implicit steady state neutron diffusion equation presented in Section 4.3. In Section 4.3.1 the coupling of the implicit steady state neutron diffusion equation to the heat conduction equations will be addressed.

4.1 Newton's Method for Solving the Implicit Steady State Solution

Newton's method is a popular iterative method for solving nonlinear systems of equations. In its most elementary form Newton's methods is used to find the root of real valued functions [27]. Newton's method for finding the root of a real valued function can be derived by considering the Taylor series expansion for an arbitrary nonlinear function $f(x)$ as shown in equation (4.1) [27]. In Newton's method all nonlinear terms in the Taylor series approximation are discarded, and with this approximation finding the root of $f\left(x^{(0)}+\delta x\right)$ implies that the expression in equation (4.2) must be true [27]. Successive solutions of equation (4.2) to approximate the root of $f(x)$ yields the iterative nature of Newton's method as shown in equation (4.3) [27].

$$f\left(x^{(0)}+\delta x\right)=f\left(x^{(0)}\right)+f'\left(x^{0}\right)\delta x+\frac{1}{2}f''\left(x^{(0)}\right)\delta x^{2}+\ldots \tag{4.1}$$

$$\delta x=\frac{-f\left(x^{(0)}\right)}{f'\left(x^{0}\right)\delta x} \tag{4.2}$$

$$x^{(k+1)}=x^{(k)}-\frac{f\left(x^{(k)}\right)}{f'\left(x^{k}\right)} \tag{4.3}$$

In the implicit steady state solution a large system of equations must be solved. To accomplish this, Newton's method can be formulated to accommodate a vector-valued function $F(x)$, where x is a vector of length n, $x=[x_1,....,x_n]^T$ [27]. A Taylor series expansion for $F(x)$ about an arbitrary vector $x^{(k)}$ is shown in equation (4.4) [27]. Again, a linear approximation is used for the expression in equation (4.4), and the first derivative of $F(x)$ is replaced by the Jacobian matrix as shown in equation (4.5) and equation (4.6).

$$F\left(x^{(k)}+\delta x\right)=F\left(x^{(k)}\right)+F'\left(x^{(k)}\right)\delta x+\frac{1}{2}F''\left(x^{(k)}\right)\delta x^{2}+\ldots \tag{4.4}$$

$$F\left(x^{(k)}+\delta x\right)=F\left(x^{(k)}\right)+J\left(x^{(k)}\right)\delta x \tag{4.5}$$

$$\tag{4.6}$$

$$J\left(x^{(k)}\right) = \begin{bmatrix} \frac{\partial F_1\left(x^{(k)}\right)}{\partial x_1} & \frac{\partial F_1\left(x^{(k)}\right)}{\partial x_2} & \frac{\partial F_1\left(x^{(k)}\right)}{\partial x_3} & \cdots & \frac{\partial F_1\left(x^{(k)}\right)}{\partial x_n} \\ \frac{\partial F_2\left(x^{(k)}\right)}{\partial x_1} & \frac{\partial F_2\left(x^{(k)}\right)}{\partial x_2} & \cdots & \cdots & \frac{\partial F_2\left(x^{(k)}\right)}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_n\left(x^{(k)}\right)}{\partial x_1} & \frac{\partial F_n\left(x^{(k)}\right)}{\partial x_2} & \cdots & \cdots & \frac{\partial F_n\left(x^{(k)}\right)}{\partial x_n} \end{bmatrix}$$

To find the root of equation (4.5), the expression in equation (4.7) must be true [27]. In equation (4.7) the vector δx is determined using a direct or iterative linear system solver [27]. After obtaining the vector δx that satisfies equation (4.7), an estimate of the root of the system is given by the vector shown in equation (4.8) [27]. The repeated solving of equation (4.7) with continuous updates of the root vector forms the framework of Newton's method for a system of linear equations [27].

$$J\left(x^{(k)}\right)\delta x = -F\left(x^{(k)}\right) \tag{4.7}$$

$$x^{(k+1)} = x^{(k)} + \delta x^{(k)} \tag{4.8}$$

The Jacobian matrix defined in equation (4.6) can be calculated numerically using finite difference approximations outlined in Section 2.1 or analytically if the system of equations is easily differentiable. In the implicit steady state solution method presented in this document, the Jacobian matrix is calculated analytically. In the literature for the implicit transient solution the Jacobian entries for the thermal-hydraulic entries and heat conduction derivatives were presented, and these expressions will remain largely unchanged [3]. The unique contribution of this work is the formulation of the Jacobian entries for the steady state neutron diffusion equation and corresponding fission source in the conduction equations.

4.2 Cross Section Derivative Calculation

In the formulation of the implicit steady state neutron diffusion equation, it will be necessary to express the change in the cross section as a function of the change in the primary thermal-hydraulic and heat conduction unknowns. More specifically, the implicit steady state neutron diffusion equations will be formulated in a way that the derivative of macroscopic cross sections with respect to fuel temperature and mixture density will be needed. However, since the cross sections are deter-

mined from tabulated values, analytical expressions for the cross section derivatives are not available and numerical approximations for the derivatives must be made.

In the MSLB and PBTT cross section database files cross sections are tabulated for each type of interaction as a function of fuel temperature and mixture density. To determine the value of a cross sections for a given fuel temperature and mixture density, a two-step interpolation calculation is needed. An example of the computational stencil for the interpolation process is shown in Figure 4.1. To obtain the cross section of interest at the fuel temperature T_f and mixture density ρ_m, the cross sections for the two fuel temperatures neighboring the fuel temperature of interest are first each multiplied by a weighting factor to produce intermediate cross sections ($\Sigma\left(T_{f'}, \rho_{m,1}\right)$ and $\Sigma\left(T_{f'}, \rho_{m,2}\right)$). The weighting factors for this first linear interpolation are produced by assuming a linear behavior between the two neighboring cross sections tabulated at the same mixture density but different fuel temperature. The weighting factors used to calculate the intermediate cross section values are shown in equation (4.9) and equation (4.10).

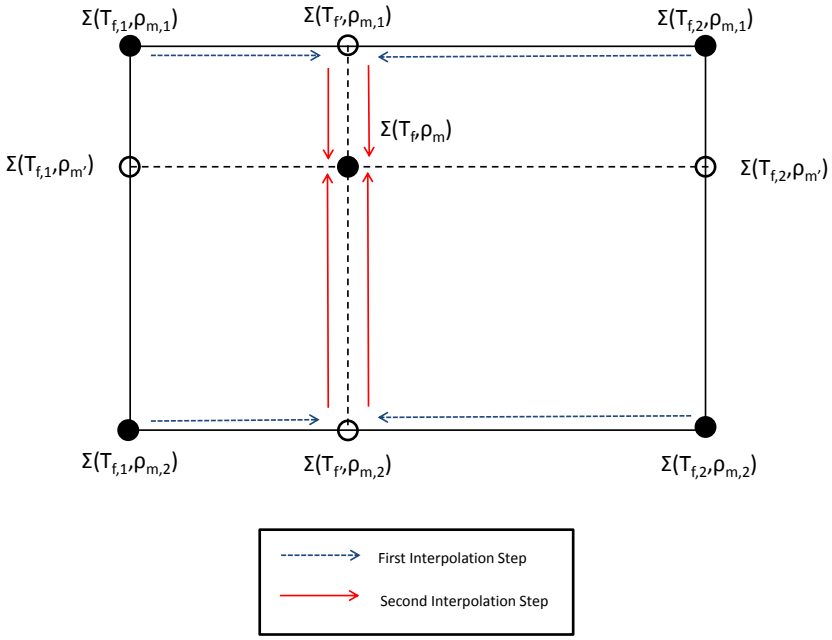


Figure 4.1: Cross section interpolation grid for the PBTT and MSLB cross section libraries

$$\Sigma\left(T_{f,'},\rho_{m,1}\right)=\frac{T_{f,2}-T_f}{T_{f,2}-T_{f,1}}\Sigma\left(T_{f,1},\rho_{m,1}\right)+\frac{T_f-T_{f,1}}{T_{f,2}-T_{f,1}}\Sigma\left(T_{f,2},\rho_{m,1}\right) \tag{4.9}$$

$$\Sigma\left(T_{f,'},\rho_{m,2}\right)=\frac{T_{f,2}-T_f}{T_{f,2}-T_{f,1}}\Sigma\left(T_{f,1},\rho_{m,2}\right)+\frac{T_f-T_{f,1}}{T_{f,2}-T_{f,1}}\Sigma\left(T_{f,2},\rho_{m,2}\right) \tag{4.10}$$

With the intermediate cross sections determined, the final cross section of interest can be determined with a second interpolation step. This time the interpolation is with respect to the mixture density values. Weighting factors are used again to interpolate the intermediate cross sections to the correct mixture density value. Again the weighting factors are determined by assuming a linear behavior between the two neighboring mixture density table values. The expression for the final cross section is shown in equation (4.11).

$$\Sigma\left(T_f,\rho_m\right)=\frac{\rho_{m,2}-\rho_m}{\rho_{m,2}-\rho_{m,1}}\Sigma\left(T_{f,'},\rho_{m,1}\right)+\frac{\rho_m-\rho_{m,1}}{\rho_{m,2}-\rho_{m,1}}\Sigma\left(T_{f,'},\rho_{m,2}\right) \tag{4.11}$$

The calculation of the cross sections from the MSLB and PBTT cross section files was not a contribution of this research, but it informs the calculation of the cross section derivatives which is a necessary component of the fully implicit solution. The cross section derivatives for a particular fuel temperature and mixture density are calculated in a similar fashion to the interpolated cross section of interest. As an example, the calculation of the fuel temperature derivative will be shown. A similar process is used for the calculation of the cross section derivative with respect to mixture density.

An approximation for the cross section derivative with respect to fuel temperature is first obtained by estimating the fuel temperature derivative at the neighboring mixture density values using a finite difference approximation as explained in Section 2.1. For clarity, the derivative approximations are shown in equation (4.12) and equation (4.13). The derivatives shown in equation (4.12) and equation (4.13) are then multiplied by weighting factors based on the same linear interpolation scheme that was used for the determination of the cross section shown in equation (4.11). The final calculation of the cross section derivative with respect

to fuel temperature is shown in equation (4.14).

$$\left.\frac{\partial \Sigma}{\partial T_f}\right|_{\rho_{m,1}} = \frac{\Sigma\left(T_{f,2}, \rho_{m,1}\right) - \Sigma\left(T_{f,1}, \rho_{m,1}\right)}{T_{f,2} - T_{f,1}} \tag{4.12}$$

$$\left.\frac{\partial \Sigma}{\partial T_f}\right|_{\rho_{m,2}} = \frac{\Sigma\left(T_{f,2}, \rho_{m,2}\right) - \Sigma\left(T_{f,1}, \rho_{m,2}\right)}{T_{f,2} - T_{f,1}} \tag{4.13}$$

$$\left.\frac{\partial \Sigma}{\partial T_f}\right|_{\rho_m} = \left(\frac{\rho_{m,2} - \rho_m}{\rho_{m,2} - \rho_{m,1}}\right) \left.\frac{\partial \Sigma}{\partial T_f}\right|_{\rho_{m,1}} + \left(\frac{\rho_m - \rho_{m,1}}{\rho_{m,2} - \rho_{m,1}}\right) \left.\frac{\partial \Sigma}{\partial T_f}\right|_{\rho_{m,2}} \tag{4.14}$$

The method described above for calculating the cross section derivative with respect to fuel temperature and mixture density was already implemented as part of the implicit transient solution method [3]. A unique contribution of this research was to add the capability to utilize the Purdue Macroscopic Cross Section (PMAXS) library format in addition to the PBTT and MSLB cross section files. The extension of the implicit solution method to include the use of PMAXS files is important because most reactor simulations used in PARCS use this file format for storing cross section data.

Compared to the PBTT and MSLB cross section file format, the PMAXS format contains a more extensive tree structure for cross section and cross section derivatives. The tree structure in the PMAXS file contains branches characterized by the independent variables described in the PMAXS manual: control rod position, coolant density, soluble poison concentration, fuel temperature, and coolant temperature [28]. In a PMAXS file a macroscopic cross section for each type of neutron interaction is tabulated for a single reference state. These tabulated macroscopic cross sections are known collectively as reference cross sections. In addition to reference cross sections, derivatives for each type of cross section with respect to each independent variable are tabulated for different variable states [28]. The reference cross section and cross section derivatives are used to calculate the cross section of interest.

Because the cross section derivatives with respect to state variables are stored in PMAXS files, analogous calculations like those shown in equation (4.12) and equation (4.13) are not necessary. However, weighting factors still must be used in a first-order approximation to calculate derivatives for the correct fuel temper-

ature and mixture density. Because the PMAXS file tree structure has effectively five independent variables instead of the two in the PBTT and MSLB cross section library files, the expressions for calculating the derivatives of interest become lengthy. For this reason, the complete expressions are not presented here. However, the exact interpolation step shown in equation (4.14) is extended to all of the independent variables in the PMAXS tree structure.

4.3 Implicit Form of the Steady State Neutron Diffusion Equation

In the framework of Newton's method for solving a system of nonlinear equations, each of the new time step $(n+1)$ quantities are equal to the sum of the previous iterate (k) estimate of the new time value and corresponding residual. For the flux and eigenvalue the aforementioned iterative strategy is shown in equation (4.15) and equation (4.16).

$$\phi_g^{m,n+1} = \phi_g^{m,k} + \delta\phi_g^m \tag{4.15}$$

$$\lambda^{n+1} = \lambda^k + \delta\lambda \tag{4.16}$$

The same iterative strategy that is shown in equation (4.15) and equation (4.16) is used to update cross sections and diffusion coefficients; however, because cross sections are not a principle unknown in the system of equations, an expansion is necessary in order to maintain consistency with the chosen set of thermal-hydraulic and heat transfer principle unknowns. To do this, the cross section delta value is expanded as a function of fuel temperature and mixture density which is then manipulated into an expression including the hydraulic and heat transfer principle unknowns. This expansion was originally chosen during the implicit transient solution development in order to match the organization of the Peach Bottom Turbine Trip (PBTT) and Three Mile Island Main Steam Line Break (MSLB) benchmark cross section libraries which use mixture density and fuel temperature for tabulating cross sections [3]. This same treatment of cross sections also appears in the development of another implicit method mentioned in Chapter 1 [8]. As previously discussed, functionality was added to the implicit solution method to

enable the use of the general PMAXS file format.

The process for expressing the cross sections as a function of fuel temperature and mixture density is shown in equation (4.17), equation (4.18), and equation (4.19). A similar manipulation can be performed for the nodal diffusion coefficients, but the manipulation is not shown here because it would be needlessly redundant.

$$\Sigma_x^{m,n+1} = \Sigma_x^{m,k} + \delta\Sigma_x^m \tag{4.17}$$

$$\Sigma_x^{m,n+1}\left(T_f^i, \rho_m^j\right) = \Sigma_x^{m,k}\left(T_f^i, \rho_m^j\right) + \sum_i \left(\frac{\partial\Sigma_x^m}{\partial T_f^i}\right)^k \delta T_f^i + \left(\frac{\partial\Sigma_x^m}{\partial\rho_m^j}\right)^k \delta\rho_m^j \tag{4.18}$$

$$\begin{aligned}\Sigma_x^{m,n+1}\left(T_f^i, \rho_m^j\right) = \Sigma_x^{m,k}\left(T_f^i, \rho_m^j\right) &+ \sum_i \left(\frac{\partial\Sigma_x^m}{\partial T_f^i}\right)^n \delta T_f^i \\ &+ \left(\frac{\partial\Sigma_x^m}{\partial\rho_m^j}\right)^n \left[\left(\frac{\partial\rho_m^j}{\partial P^j}\right)^k \delta P^j + \left(\frac{\partial\rho_m^j}{\partial\alpha^j}\right)^k \delta\alpha^j \right. \\ &\left. + \left(\frac{\partial\rho_m^j}{\partial T_{\mathrm{g}}^j}\right)^k \delta T_{\mathrm{g}}^j + \left(\frac{\partial\rho_m^j}{\partial T_l^j}\right)^k \delta T_l^j + \left(\frac{\partial\rho_m^j}{\partial P_a^j}\right)^k \delta P_a^j \right]\end{aligned} \tag{4.19}$$

In the transition from equation (4.18) to equation (4.19) the cross section derivatives were changed from iterate values to old time step values. This simplification is motivated by practical programming considerations and the numeric nature of the cross section derivatives. In the PBTT and MSLB cross section libraries cross sections are provided at specified values of the mixture density and fuel temperature, and nodal cross sections are calculated by linearly interpolating between the tabulated cross sections surrounding the nodal fuel temperature and mixture density, as explained in Section 4.2. Cross section derivatives are calculated using the same multidimensional linear interpolation scheme, and as a result, the derivatives are constant within the boundary defined by the temperature and density statepoints if the cross derivative $(\partial\Sigma_x)/(\partial T_f \partial\rho_m)$ term is ignored. Neglecting the cross derivative is a reasonable approximation since cross sections themselves are only first order accurate from the linear interpolation scheme.

Like the cross sections and diffusion coefficients, the nodal coupling correction coefficient should be updated every outer iteration in order to be consistent with

a fully implicit time integration scheme. However, unlike the cross sections and diffusion coefficients, there is not a simple expression for the derivative of the nodal coupling correction coefficient with respect to the principle unknowns. Therefore, derivatives of the nodal coupling correction coefficient are not included in the Jacobian matrix, but the correction coefficient is updated at the end of each outer iteration.

The exclusion of the nodal coupling correction coefficient significantly simplifies the contribution of the leakage coefficients defined in equation (2.18), equation (2.19), and equation (2.20) to the Jacobian matrix. The only special consideration required for the leakage coefficients now is related to the base nodal coupling coefficient. The same process that was used to expand the cross sections in terms of fuel temperature and hydraulic variables can be used to expand the leakage coefficients defined in equation (2.18), equation (2.19), and equation (2.20). A partial expansion will only be shown for equation (2.19), but the expansion for equation (2.18) and equation (2.20) follows the same method. The expansion is shown in equation (4.20) and equation (4.21).

$$a_{gu}^{m,n+1} = a_{gu}^{m,k} + \delta a_{gu}^{m} \tag{4.20}$$

$$\begin{aligned} a_{gu}^{m,k} + \delta a_{gu}^{m} = a_{gu}^{m,k} &+ \sum_{i} \frac{\partial a_{gu}^{m}}{\partial T_f^i} \delta T_f^i + \sum_{i^+} \frac{\partial a_{gu}^{m}}{\partial T_f^{i+}} \delta T_f^{i+} + \sum_{i^-} \frac{\partial a_{gu}^{m}}{\partial T_f^{i-}} \delta T_f^{i-} \\ &+ \frac{\partial a_{gu}^{m}}{\partial \rho_m^j} \delta \rho_m^j + \frac{\partial a_{gu}^{m}}{\partial \rho_m^{j+}} \delta \rho_m^{j+} + \frac{\partial a_{gu}^{m}}{\partial \rho_m^{j-}} \delta \rho_m^{j-} \end{aligned} \tag{4.21}$$

Unlike the iterative expansion for the cross sections, the iterative update of the leakage coefficients for cell m is dependent on the change in each neighboring cell's principle unknowns. This spatial connectedness is shown in equation (4.21) and directly stems from the definition of the base nodal coupling coefficient in equation (2.15). The two-node functionality of the base nodal coupling coefficient in equation (2.15) leads to a very lengthy expression when the mixture density derivatives are expanded into an expression including only principle unknowns. For this reason the full expression is not shown in this section; however, the derivatives of the leakage coefficients with respect to mixture density are shown in equation

(4.22), equation (4.23), and equation (4.24). Note that similar expressions for the derivatives of the leakage coefficients with respect to fuel temperature can be derived but are not shown here.

$$\frac{\partial a_{gu}^{m}}{\partial \rho_{m}^{j}} = \left[\frac{1}{h_{u}^{m}} \frac{2\left(D_{g}^{m+l_u}\right)^{2} \Delta u_{m} \frac{\partial D_{g}^{m}}{\partial \rho_{m}^{j}}}{\left(D_{g}^{m+l_u} \Delta u_{m} + D_{g}^{m} \Delta u_{m+l_u}\right)^{2}} \right] + \left[\frac{1}{h_{u}^{m}} \frac{2\left(D_{g}^{m-l_u}\right)^{2} \Delta u_{m} \frac{\partial D_{g}^{m}}{\partial \rho_{m}^{j}}}{\left(D_{g}^{m-l_u} \Delta u_{m} + D_{g}^{m} \Delta u_{m-l_u}\right)^{2}} \right] \tag{4.22}$$

$$\frac{\partial a_{gu}^{m}}{\partial \rho_{m}^{j+}} = \left[\frac{1}{h_{u}^{m}} \frac{2\left(D_{g}^{m}\right)^{2} \Delta u_{m+l_u} \frac{\partial D_{g}^{m+l_u}}{\partial \rho_{m}^{j+}}}{\left(D_{g}^{m+l_u} \Delta u_{m} + D_{g}^{m} \Delta u_{m+l_u}\right)^{2}} \right] \tag{4.23}$$

$$\frac{\partial a_{gu}^{m}}{\partial \rho_{m}^{j-}} = \left[\frac{1}{h_{u}^{m}} \frac{2\left(D_{g}^{m}\right)^{2} \Delta u_{m-l_u} \frac{\partial D_{g}^{m-l_u}}{\partial \rho_{m}^{j-}}}{\left(D_{g}^{m-l_u} \Delta u_{m} + D_{g}^{m} \Delta u_{m-l_u}\right)^{2}} \right] \tag{4.24}$$

4.3.1 Eigenvalue Linearization Scheme

If the eigenvalue is to be included in the Jacobian matrix as a principle unknown, an ancillary equation is needed to prevent the matrix from being row-rank deficient. A formulation for a general ancillary equation was presented by Peters (1979) for use in solving a general eigenvalue problem using Newton's method. This general method relies on the eigenvector being normalized according to its 2 norm. Gill (2011) adapted this ancillary equation to solve for the eigenvalue in the multi-group neutron diffusion equation using Newton's method. The ancillary equation utilized by Gill (2011) is shown in equation (4.25).

$$\frac{1}{2}\phi^{T}\phi = \frac{1}{2} \tag{4.25}$$

A similar technique was utilized by Knoll *et al.* (2011) to formulate an ancillary equation necessary to solve the k-eigenvalue problem using Newton's method. The method presented by Knoll *et al.* (2011) begins with the power iteration method equation for updating the eigenvalue shown in equation (4.26) and relies on constraining the magnitude of the previous eigenvalue and fission source normalization fraction to unity as shown in equation (4.27).

$$k^{n+1} = \frac{k^{n} \sum_{g} \int dV \nu \Sigma_{fg} \phi_{g}^{n+1}}{\sum_{g} \int dV \nu \Sigma_{fg} \phi_{g}^{n}} \tag{4.26}$$

$$\frac{k^n}{\sum_g \left(\nu\Sigma_f\phi^n\right)} = 1 \tag{4.27}$$

Using the normalization constraint shown in equation (4.27), the ancillary equation necessary to use Newton's method to solve the eigenvalue problem as presented by Knoll (2011) *et al.* is shown in equation (4.28).

$$k^{n+1} - \sum \left(\nu\Sigma_f\phi^{n+1}\right) = 0 \tag{4.28}$$

Another ancillary equation for the solution of the eigenvalue problem in the neutron diffusion equation using Newton's method was proposed by Kastanya (2003) and is based on the physics of the steady state neutronics problem. To solve the steady state neutron diffusion equation using Newton's method, Kastanya (2003) introduced a power constraint equation into the set of equations. The power constraint equation is shown in equation (4.29).

$$\sum_m \sum_g \kappa\Sigma_{fg}^m \phi_g^m V^m = Q \tag{4.29}$$

The power constraint principle used to formulate equation (4.29) is already used in TRACE and PARCS to scale the flux vector solution to the core power for use in the heat transfer equations. Effectively, the ancillary equation introduced by Kastanya (2003) reproduces this feature and scales the eigenvector to a magnitude that produces the desired steady state power level. For the ancillary equations utilized by Knoll *et al.* (2011) and Gill (2011), the eigenvector is not scaled to the desired steady state power level, and as a result, treatment of the scaling factor used to scale the eigenvector for inclusion in the heat transfer equations requires special consideration. The treatment of the flux scaling factor in the implicit steady state solution is addressed in the proceeding section.

The ancillary equations shown in equation (4.25), equation (4.28), and equation (4.29) were included in the implicit solution as optional ancillary equations. Additionally, all of the fluxes and cross sections in equation (4.25), equation (4.28), and equation (4.29) were manipulated to be included in the Jacobian as shown in equation (4.15), equation (4.16), and equation (4.19).

4.4 Implicit Power Generation Term

The general unsteady heat conduction equation in TRACE used to calculate the temperature profile in heat structure components is shown in equation (4.30) [21].

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + q''' \tag{4.30}$$

The discretization scheme for the time derivative and spatial derivatives in equation (4.30) are shown in the TRACE theory manual [21]. Additionally, the coupling of the heat transfer equations via wall and fluid temperatures to the thermal-hydraulic equations is described in the documentation for the implicit transient solution and will not be described here [3]. However, the heat generation term in equation (4.30) requires special consideration because it couples the nodal heat transfer equations to the steady state neutronics solution through the nodal flux solutions.

The eigenvectors in an eigenvalue solution are determined up to a multiplicative constant. Therefore, while the distribution of the flux taken directly from the eigenvector is correct, the magnitude of the nodal fluxes are unknown by a multiplicative constant. The total core power is used to scale the eigenvector to flux values that yield the correct total power. The scaling factor is shown in equation (4.31).

$$R = \frac{Q}{\sum_m \sum_g \left(\kappa \Sigma_f\right)_g^m \phi_g^m V^m} \tag{4.31}$$

The volumetric heat generation source in a heat conduction node can then be written using the scaling factor shown in equation (4.31) and is shown in equation (4.32)

$$q_i''' = W_i R \sum_g \left(\kappa \Sigma_f\right)_g^m \phi_g^m V^m \tag{4.32}$$

The weighting factor W_i in equation (4.32) is used to split the total nodal power produced from the nodal fission source among the heat structure nodes that are contained in the neutronics node. The weighting factor W_i is calculated based on a volume-weighting process and a user-defined pin power profile for the heat structure nodes, and W_j remains constant throughout the entire steady state calculation.

The complexity of including equation (4.32) in the Jacobian matrix stems from the dependence of the scaling factor R on all nodal flux values, nodal thermal-hydraulic principle unknowns, and heat structure principle unknowns. The necessity of including the derivative of the scaling factor R will be evaluated in a later sections; nonetheless the derivative of R is included in the expansion of the volumetric heat generation term for inclusion in the Jacobian matrix as an implicit solution option. The volumetric heat generation term is expanded to be consistent with the principle unknowns of the system of equations, and the expansion is shown in equation (4.33) and equation (4.34).

$$q_i'''^{,n+1} = q_i'''^{,k} + \delta q_i''' \tag{4.33}$$

$$\begin{aligned} q_i'''^{,k} + \delta q_i''' = q_i'''^{,k} &+ \sum_i \left(\frac{\partial q_i'''}{\partial T_f^i}\right)^k \delta T_f^i + \sum_{i'} \left(\frac{\partial q_i'''}{\partial T_f^{i'}}\right)^k \delta T_f^{i'} + \sum_I \left(\frac{\partial q_i'''}{\partial T_f^I}\right)^k \delta T_f^I \\ &+ \sum_g \left(\frac{\partial q_i'''}{\partial \phi_g^m}\right)^k \delta \phi_g^m + \sum_M \sum_g \left(\frac{\partial q_i'''}{\partial \phi_g^M}\right)^k \delta \phi_g^M \\ &+ \sum_j \left(\frac{\partial q_i'''}{\partial \rho_m^j}\right)^k \left[\left(\frac{\partial \rho_m^j}{\partial P^j}\right)^k \delta P^j + \left(\frac{\partial \rho_m^j}{\partial \alpha^j}\right)^k \delta \alpha^j + \left(\frac{\partial \rho_m^j}{\partial T_{\mathrm{g}}^j}\right)^k \delta T_{\mathrm{g}}^j \right. \\ &\left. + \left(\frac{\partial \rho_m^j}{\partial T_l^j}\right)^k \delta T_l^j + \left(\frac{\partial \rho_m^j}{\partial P_a^j}\right)^k \delta P_a^j \right] \\ &+ \sum_J \left(\frac{\partial q_i'''}{\partial \rho_m^J}\right)^k \left[\left(\frac{\partial \rho_m^J}{\partial P^J}\right)^k \delta P^J + \left(\frac{\partial \rho_m^J}{\partial \alpha^J}\right)^k \delta \alpha^J \right. \\ &\left. + \left(\frac{\partial \rho_m^J}{\partial T_{\mathrm{g}}^J}\right)^k \delta T_{\mathrm{g}}^J + \left(\frac{\partial \rho_m^J}{\partial T_l^J}\right)^k \delta T_l^J + \left(\frac{\partial \rho_m^J}{\partial P_a^J}\right)^k \delta P_a^J \right] \end{aligned} \tag{4.34}$$

Unlike in the previous expansions, the expression for the delta term in equation (4.34) includes the summation of the derivatives from all the other nodes in the model. The summation is a direct result of the scaling factor R being a function of all the principle unknowns in the system. An additional level of complexity is added by the derivatives of the volumetric heat generation term with respect

to fuel temperature and moderator density varying based on whether or not the fuel temperature and moderator density are included in the same finite volume as the nodal volumetric heat generation term. The derivative of the volumetric heat generation term with respect to the fuel temperature within the heat structure node coinciding with the volumetric heat generation is shown in equation (4.35).

$$\left(\frac{\partial q_i'''}{\partial T_f^i}\right)^k = W_i R^k \sum_g \left(\frac{\partial \left(\kappa\Sigma_f\right)_g^m}{\partial T_f^i}\right)^n \phi_g^{m,k} V^m \left(1 - \frac{R^k}{Q} \sum_g \left(\kappa\Sigma_f\right)_g^{m,k} \phi_g^{m,k} V^m\right) \tag{4.35}$$

The derivative of the scaling factor is included in equation (4.35) through the expression shown in equation (4.36).

$$\frac{R^k}{Q} = \left(\frac{\partial R}{\partial T_f^i}\right)^k \left(-R^k \sum_g \left(\frac{\partial \left(\kappa\Sigma_f\right)_g^m}{\partial T_f^i}\right)^n \phi_g^{m,k} V^m\right)^{-1} \tag{4.36}$$

The expression in equation (4.35) can be adapted slightly to calculate the derivative of the volumetric heat generation term with respect to a fuel temperature other than T_f^i that coincides with the neutronics node that contains the volumetric heat generation term. The derivative of the volumetric heat generation term with respect to $T_f^{i'}$ is shown in equation (4.37).

$$\left(\frac{\partial q_i'''}{\partial T_f^{i'}}\right)^k = W_i R^k \sum_g \left(\frac{\partial \left(\kappa\Sigma_f\right)_g^m}{\partial T_f^{i'}}\right)^n \phi_g^{m,k} V^m \left(1 - \frac{R^k}{Q} \sum_g \left(\kappa\Sigma_f\right)_g^{m,k} \phi_g^{m,k} V^m\right) \tag{4.37}$$

The derivative of the scaling factor is included in equation (4.37) through the expression shown in equation (4.38).

$$\frac{R^k}{Q} = \left(\frac{\partial R}{\partial T_f^{i'}}\right)^k \left(-R^k \sum_g \left(\frac{\partial \left(\kappa\Sigma_f\right)_g^m}{\partial T_f^{i'}}\right)^n \phi_g^{m,k} V^m\right)^{-1} \tag{4.38}$$

The derivative of the volumetric heat generation term with respect to a fuel temperature not coinciding with the neutronics cell containing q_i''' is shown in equation

(4.39).

$$\left(\frac{\partial q_i'''}{\partial T_f^I}\right)^k = -W_j \frac{(R^2)^k}{Q}\left[\sum_g \left(\frac{\partial \left(\kappa\Sigma_f\right)_g^M}{\partial T_f^I}\right)^n \phi_g^{M,k} V^M\right]\left(\sum_g \left(\kappa\Sigma_f\right)_g^{m,k} \phi_g^{m,k} V^m\right) \tag{4.39}$$

The derivative of the volumetric heat generation term with respect to mixture density that is directly coupled with the neutronics cell containing q_i''' can calculated by replacing the derivatives with respect to fuel temperature in equation (4.35) with derivatives with respect to mixture density. Similarly, the derivative of the volumetric heat generation term with respect to a nodal mixture density not coinciding with the neutronics cell containing q_i''' can be calculated by replacing the derivatives with respect to fuel temperature in equation (4.39) with derivatives with respect to mixture density. An additional step is needed to expand the mixture density derivatives in terms of thermal-hydraulic principle unknowns. The expansion of the mixture density derivative was shown in equation (4.19) for cross sections and the same method was used for the volumetric heat generation derivatives with respect to mixture density.

The derivative of the volumetric heat generation term with respect to nodal group flux is not formed as easily using a simple substitution as described for the derivatives with respect to mixture density. The derivative of the volumetric heat generation term with respect to nodal group flux that is directly coupled with the neutronics cell containing q_i''' is shown in equation (4.40).

$$\left(\frac{\partial q_i'''}{\partial \phi_g^m}\right)^k = W_i R^k \left(\kappa\Sigma_f\right)_g^{m,k} V^m \left(1 - \frac{R^k}{Q}\sum_g \left(\kappa\Sigma_f\right)_g^m \phi_g^{m,k} V^m\right) \tag{4.40}$$

The derivative of the scaling factor with respect to the nodal group flux is included in equation (4.40) through the expression shown in equation (4.41).

$$\frac{R^k}{Q} = \left(\frac{\partial R}{\partial \phi_g^m}\right)^k \left(-R^k \left(\kappa\Sigma_f\right)_g^{m,k} V^m\right)^{-1} \tag{4.41}$$

The derivative of the volumetric heat generation term with respect to a nodal group flux not coinciding with the neutronics cell containing q_i''' is shown in equa-

tion (4.42).

$$\left(\frac{\partial q_i'''}{\partial \phi_g^M}\right)^k = -W_i \frac{\left(R^2\right)^k}{Q} \left(\kappa\Sigma\right)_g^{M,k} V^M \left(\sum_g \left(\kappa\Sigma_f\right)_g^{m,k} \phi_g^{m,k} V^m\right) \tag{4.42}$$

Neglecting the derivative of the scaling factor simplifies the derivatives in the heat structure equations and reduces the number of nonzero values that must be stored in memory. If the derivative of the scaling factor is neglected then equation (4.39) and equation (4.42) would be equal to zero. Additionally, the contributions to equation (4.35), equation (4.37), and equation (4.40) from equation (4.36), equation (4.38), and equation (4.41), respectively, would be equal to zero because the derivative of the scaling factor would equal to zero.

To quantify the significance of excluding the scaling factor derivative, a finite difference method was used to evaluate the neutronics equation derivative with respect to nodal pressure. The nodal pressure for the third axial hydraulic volume of the five-node channel model was changed from 6.8417E6 Pa to 6.9500E6 Pa, and the derivative of the nodal volumetric heat generation term with respect to the pressure in the third axial hydraulic volume was calculated for each fuel temperature node in the system. The results are shown in Table 4.1.

Heat Structure Cell	Finite Difference Approximation $\left[\frac{W}{cm^3 \cdot Pa}\right]$	Analytic Derivative with $\frac{\partial R}{\partial P} = 0$ $\left[\frac{W}{cm^3 \cdot Pa}\right]$	Analytic Derivative with $\frac{\partial R}{\partial P} \neq 0$ $\left[\frac{W}{cm^3 \cdot Pa}\right]$
(1,1)	-0.06087	0.00000	-0.06038
(2,1)	-0.06087	0.00000	-0.06038
(3,1)	-0.06087	0.00000	-0.06038
(4,1)	0.00000	0.00000	0.00000
(1,2)	0.08289	0.11516	0.08226
(2,2)	0.08289	0.11516	0.08226
(3,2)	0.08289	0.11516	0.08226
(4,2)	0.00000	0.00000	0.00000
(1,3)	-0.02202	0.00000	-0.02187
(2,3)	-0.02202	0.00000	-0.02187
(3,3)	-0.02202	0.00000	-0.02187
(4,3)	0.00000	0.00000	0.00000

Table 4.1: Volumetric heat generation derivative with respect to pressure in third axial hydraulic volume

In Table 4.1 it can be seen that the analytic derivatives that include the derivative of the scaling factor are very close to the finite difference estimate. However, the the analytic derivative calculation that treats the scaling factor as a constant differs by 32.6% from the finite difference approximation. While the error between the two analytic expressions will vary based on the magnitude of the scaling factor derivative, the results in Table 4.1 show that a significant amount of error in approximating the power derivative is possible by neglecting the scaling factor derivative.

The numerical error of the analytic derivative with constant scaling factor can be understood by the physical inconsistency introduced by treating the scaling factor as a constant. More specifically, the finite difference approximation and analytic derivative with non-constant scaling factor sum to zero across all of the nodes. The aforementioned behavior reflects the steady state power constraint that is used to scale the fluxes. The power constraint implies that a change in the pressure in axial channel cell three should change the distribution of power in the model without changing the total power. Conversely, the analytic derivatives with constant scaling factor do not sum to zero which is not as accurate of an approximation of the power constraint placed on the system.

The data shown in Table 4.1 and proceeding comments are directly related to the choice of the ancillary equation for Jacobian matrix as presented in Section 4.3.1. For those ancillary equations that do not scale the eigenvector to the desired steady state power level, the scaling factor R must be included in the heat transfer equations, and as a result, inclusion of the scaling factor derivative must be considered. The results in Chapter 5 will be used to comment on the necessity of including the scaling factor derivative.

Chapter 5

Results

As discussed in Section 4.3.1, three different ancillary equations for the solution of the steady state eigenvalue problem using Newton's method have been presented in related literature [29, 30, 31]. For comparison purposes all three ancillary equations shown in Section 4.3.1 were included as separate ancillary equation options in the implicit solution. When comparing these methods in figures and tables, the method presented by Gill and Azmy (2011) which relies on the 2-norm will be denoted by TN, the method presented by Knoll *et al.* (2011) which relies on a normalized power iteration equation will be denoted by NPI, and the method presented by Kastanya (2003) which uses a power constraint equation will be denoted by PC.

In the following sections the three eigenvalue linearziation methods will be compared to the standard power iteration method used by PARCS. This comparison is motivated by the foresight that the implicit solution method will require some optimization. To gain more insight into the behavior of the implicit solution method, an additional implicit steady state solution option was added that utilizes the power iteration method with Wielandt shift to converge the eigenvalue. For the power iteration method in the implicit solution the eigenvalue is not included as a principle unknown in the Jacobian matrix but is updated at the end of each outer iteration in a manner consistent with the update scheme used in PARCS which is shown in equation (3.1). Even with the power iteration method used to update the eigenvalue, the cross sections, diffusion coefficients, nodal coupling coefficients, fluxes, and power generation terms are all treated implicitly as shown in Chapter 4. Therefore, the solution method is still implicit, but the eigenvalue

is not included as a principle unknown in the Jacobian matrix. In plots and tables this implicit steady state solution method will be denoted by the abbreviation PI. The standard explicit steady state solution used in PARCS will also be included in tables and figures for comparative purposes.

In Section 5.1 results from the five-node and fourteen-node channel models will be presented for each of the aforementioned implicit steady state solution methods. In Section 5.2 a TRACE model that is more representative of a full core model will be presented. The results in Section 5.1 and Section 5.2 will be used in later sections to evaluate the overall performance and viability of the implicit steady state solution method.

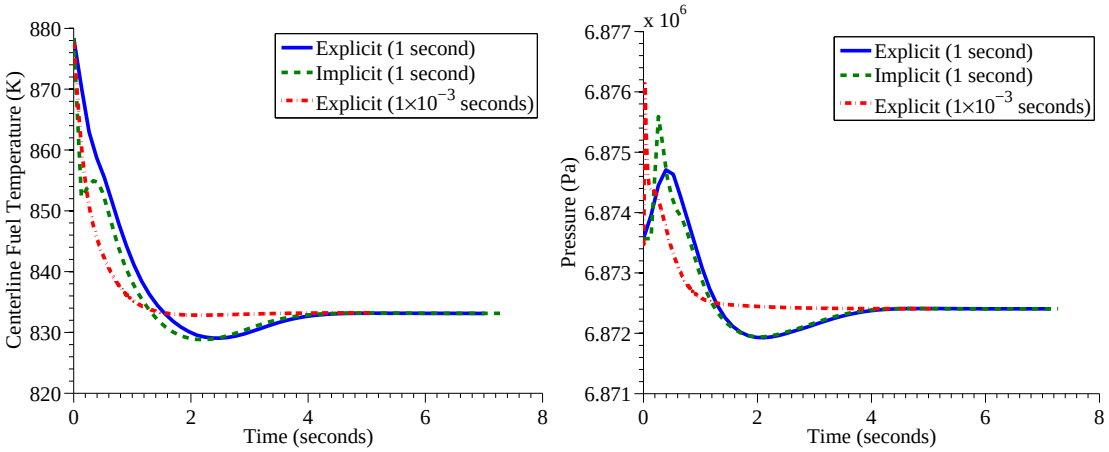
5.1 Solution Comparison for the Five-Node and Fourteen-Node Channel Models

The implicit steady state solution was compared against the explicit steady state solution to verify that the implicit neutronics, heat transfer, and thermal-hydraulic equations produce the correct steady state values. The eigenvalue was chosen as the primary criterion for comparison because it is a global principle unknown that is dependent on the solution of all of the local principle unknowns. Table 5.1 shows the steady state k_{eff} values generated by each of the implicit solution methods compared to the steady state k_{eff} value calculated with the explicit solution method at varying maximum time step values. Despite being implemented in TRACE as an eigenvalue linearization ancillary equation, the PC linearization method is not included in Table 5.1 because of instabilities observed in the solution. More specifically, at time step sizes similar to those shown in Table 5.1, flow reversals were seen within a time step which forced TRACE to restart the time step which ultimately led to a failure of the program. Only when the implicit solution with the PC eigenvalue linearization was run at a very small time step did the solution converge. The inclusion of the PC linearization method is not necessary for the conclusions that will ultimately be presented in this work.

Maximum Time Step Size (s)	Explicit	TN	NPI	PI
1.0	0.997845	0.997844	0.997844	0.997844
0.1	0.997845	0.997845	0.997845	0.997845
0.01	0.997848	0.997848	0.997848	0.997848
0.001	0.997849	0.997848	0.997848	0.997848

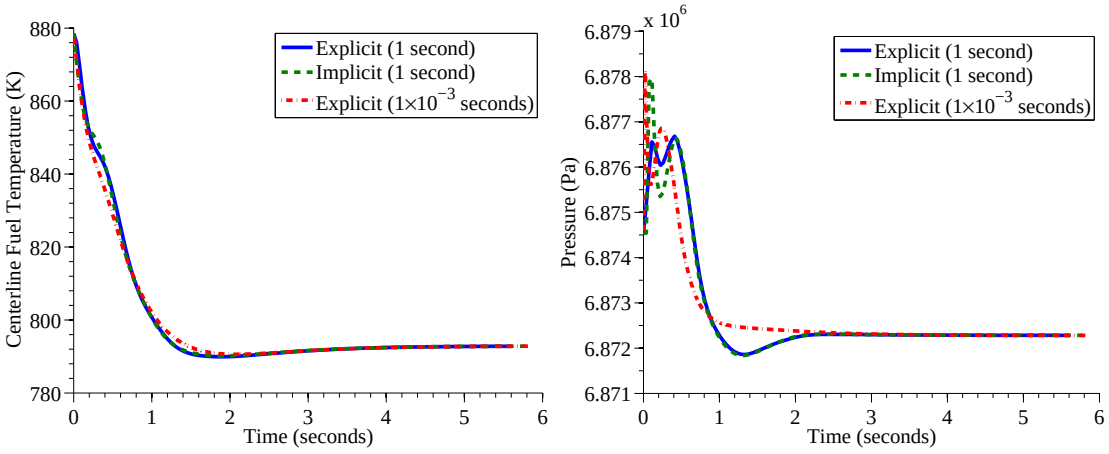
Table 5.1: Comparison of k-eff values calculated with implicit and explicit time-integration methods for the five-node channel model

The results in Table 5.1 show that the implicit solution is capable of reproducing the same solution as the explicit solution technique using the TN and NPI ancillary equations. While Table 5.1 only shows a global comparison between the methods, Figure 5.1a and Figure 5.1b show the fuel centerline temperature of the second axial heat structure cell and the pressure in the third axial hydraulic cell for the five-node channel model, respectively. The first two sets of local quantities shown in Figure 5.1a and Figure 5.1b were obtained with a maximum allowable time step of one second specified in the TRACE input files in order to allow the time step size to be controlled by the internal logic in TRACE. The last set of the local quantities shown in Figure 5.1a and Figure 5.1b were obtained with the explicit solution run with a maximum allowable time step size of 1×10^{-3}. This solution will be used as a reference solution throughout this section. A similar set of local parameter data is shown in Figure 5.2a and Figure 5.2b for the fourteen-node channel model. In Figure 5.2a and Figure 5.2b the centerline fuel temperature was taken from heat structure axial level six and the thermal-hydraulic axial level seven. The choice of the axial levels for both figures was made such that data would be presented for coinciding volumes near the axial center of each model.



(a) (b)

Figure 5.1: Temporal behavior of the fuel centerline temperature of the second axial conduction node (a) and the pressure of the third axial thermal-hydraulic node (b) in the five-node channel model



(a) (b)

Figure 5.2: Temporal behavior of the fuel centerline temperature of the sixth axial conduction node (a) and the pressure of the seventh axial thermal-hydraulic node (b) in the fourteen-node channel model

To provide a more quantitative view of the local convergence behavior between the implicit and explicit steady state solutions, Table 5.2 and Table 5.3 below show the linear power rate calculation for the five-node and fourteen-node channel models as a function of axial position, respectively. Table 5.2 and Table 5.3 show good

agreement between the linear power distributions which indicates satisfactory local convergence of the implicit solution method. The nonzero percent differences in Table 5.2 and Table 5.3 are not necessarily cause for concern because, as shown in Figure 5.1 and Figure 5.2, the implicit and explicit solution methods do not necessarily have the same temporal history leading up to a steady state solution. As a result, there may be variation in the solution within the specified steady state convergence criterion in TRACE which determines steady state convergence by comparing fractional changes in thermal-hydraulic variables [21]. It is difficult, however, to be certain as to the source of the differences because of the fundamentally different approach that the implicit and explicit methods use to solve the equation set.

Axial Node Number (s)	Explicit Linear Power(W/m)	Implicit Linear Power (W/m)	Percent Difference
1	3.5444E+04	3.5444E+04	0.0000E+00
2	1.8380E+04	1.8381E+04	5.4407E-03
3	9.0065E+03	9.0063E+03	2.2206E-03

Table 5.2: Steady state axial linear power distribution per rod for the five-node channel model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of one second

Axial Node Number (s)	Explicit Linear Power (W/m)	Implicit Linear Power (W/m)	Percent Difference
1	5.9094E+04	5.9094E+04	0.00000
2	4.6899E+04	4.6898E+04	-2.1322E-03
3	3.6152E+04	3.6152E+04	0.00000
4	2.7584E+04	2.7584E+04	0.00000
5	2.0975E+04	2.0976E+04	4.7676E-03
6	1.5998E+04	1.5999E+04	6.2508E-03
7	1.2265E+04	1.2265E+04	0.00000
8	9.4872E+03	9.4874E+03	2.1081E-03
9	7.4525E+03	7.4527E+03	2.6837E-03
10	6.0049E+03	6.0051E+03	3.3306E-03
11	5.0297E+03	5.0298E+03	1.9882E-03
12	4.3813E+03	4.3814E+03	2.2824E-03

Table 5.3: Steady state axial linear power distribution per rod for the fourteen-node channel model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of one second

To make sure that the nonzero percent difference is the result of the time step size that was used to generate the values in Table 5.2 and Table 5.3, the five-node and fourteen-node channel models were run with a maximum time step size of 1×10^{-3} seconds with the implicit and explicit solution methods. Both methods again produced identical eigenvalues. However, differences were again seen in the axial linear power distribution per rod calculated by the implicit and explicit solution methods. Table 5.4 and Table 5.5 summarize the results.

Axial Node Number (s)	Explicit Linear Power (W/m)	Implicit Linear Power (W/m)	Percent Difference
1	3.5440E+04	3.5444E+04	1.1287E-02
2	1.8388E+04	1.8380E+04	-4.3507E-02
3	9.0039E+03	9.0070E+03	3.4430E-02

Table 5.4: Steady state axial linear power distribution per rod for the five-node channel model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of 10^{-3} seconds

Axial Node Number (s)	Explicit Linear Power (W/m)	Implicit Linear Power (W/m)	Percent Difference
1	5.9086E+04	5.9093E+04	1.1847E-02
2	4.6903E+04	4.6898E+04	-1.0660E-02
3	3.6155E+04	3.6152E+04	-8.2976E-03
4	2.7586E+04	2.7584E+04	-7.2501E-03
5	2.0977E+04	2.0976E+04	-4.7671E-03
6	1.5999E+04	1.5999E+04	0.00000
7	1.2265E+04	1.2265E+04	0.00000
8	9.4870E+03	9.4874E+03	4.2163E-03
9	7.4523E+03	7.4528E+03	6.7093E-03
10	6.0045E+03	6.0052E+03	1.1658E-02
11	5.0292E+03	5.0300E+03	1.5907E-02
12	4.3810E+03	4.3816E+03	1.3696E-02

Table 5.5: Steady state axial linear power distribution per rod for the fourteen-node channel model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of 10^{-3} seconds

Along with comparing the linear power results, it is often useful to compare the normalized axial power distribution. Table 5.6 and Table 5.7 show a comparison of the normalized axial power distribution for the five-node channel model with a maximum time step size of one second and 1×10^{-3} seconds, respectively. Similarly, Table 5.8 and Table 5.9 show a comparison of the normalized axial power distribution for the fourteen-node channel model with a maximum time step size of one second and 1×10^{-3} seconds, respectively. Figures for the normalized axial power distribution would normally be provided, however, because the values are nearly all identical, a visual comparison is not useful.

Axial Node Number (s)	Explicit Normalized Power Fraction	Implicit Normalized Power Fraction	Percent Difference
1	1.692	1.692	0.000
2	0.878	0.878	0.000
3	0.430	0.430	0.000

Table 5.6: Steady state axial normalized power distribution for the five-node channel model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of 1 second

Axial Node Number (s)	Explicit Normalized Power Fraction	Implicit Normalized Power Fraction	Percent Difference
1	1.692	1.692	0.000
2	0.878	0.878	0.000
3	0.430	0.430	0.000

Table 5.7: Steady state axial normalized power distribution for the five-node channel model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of 1×10^{-3} second

Axial Node Number (s)	Explicit Normalized Power Fraction	Implicit Normalized Power Fraction	Percent Difference
1	2.822	2.822	0.000
2	2.239	2.239	0.000
3	1.726	1.726	0.000
4	1.317	1.317	0.000
5	1.002	1.002	0.000
6	0.764	0.764	0.000
7	0.586	0.586	0.000
8	0.453	0.453	0.000
9	0.356	0.356	0.000
10	0.287	0.287	0.000
11	0.240	0.240	0.000
12	0.209	0.209	0.000

Table 5.8: Steady state normalized axial power distribution for the fourteen-node channel model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of 1 second

Axial Node Number (s)	Explicit Normalized Power Fraction	Implicit Normalized Power Fraction	Percent Difference
1	2.821	2.822	0.035
2	2.239	2.239	0.000
3	1.726	1.726	0.000
4	1.317	1.317	0.000
5	1.002	1.002	0.000
6	0.764	0.764	0.000
7	0.586	0.586	0.000
8	0.453	0.453	0.000
9	0.356	0.356	0.000
10	0.287	0.287	0.000
11	0.240	0.240	0.000
12	0.209	0.209	0.000

Table 5.9: Steady state normalized axial power distribution for the fourteen-node channel model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of 1×10^{-3} seconds

An important factor in the solution efficiency is the time step size at which the steady state calculation can be performed. Figure 5.3a and Figure 5.3b show the time step sizes determined in TRACE for the implicit and explicit solution methods for the five-node and fourteen-node channel models, respectively. The time step sizes shown in Figure 5.3a and Figure 5.3b for the implicit and explicit methods is determined by the material Courant limit imposed by the semi-implicit formulation of the thermal-hydraulic equations. There is one exception, however, in Figure 5.3a. The decrease in the time step size of the implicit method below 0.08 seconds was caused by a rate of change in the void-fraction that exceeded the limits included in the TRACE time step size selection subroutines. This decrease in time step size is not believed to be an inherent limitation of or error in the implicit solution method because the temporal accuracy of the implicit solution agrees well with the temporal behavior of the reference solution as shown in Figure 5.1a. The decrease in the time step is a reflection of the constraints of the physics of the pseudo-transient solution. A similar reduction was not seen for the fourteen-node problem which also suggests that it is not an error in the code but a byproduct of the pseudo-transient temporal path for the five-node channel model.

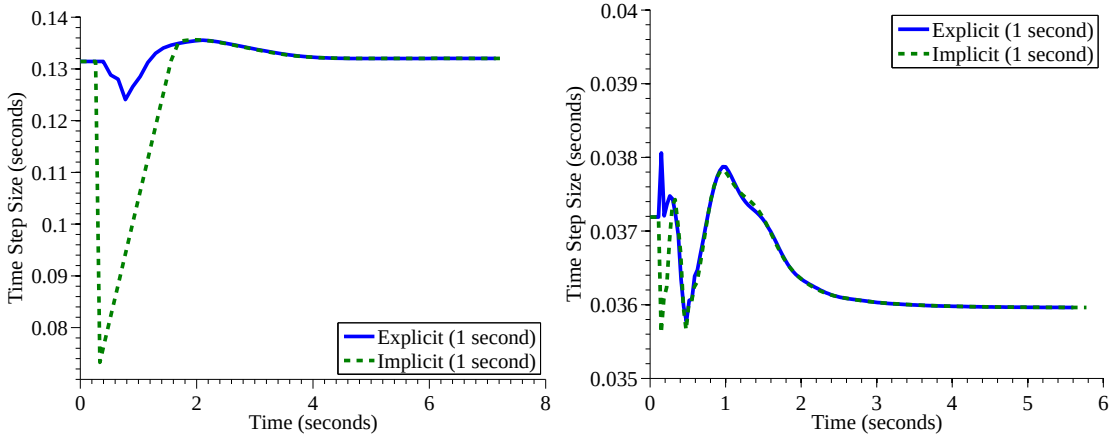


Figure 5.3: Time step size comparison for the implicit and explicit solution methods utilized in the steady state calculation for the five-node (a) and fourteen-node (b) channel models

5.1.1 Implicit Steady State Solution Efficiency for the Five-node and Fourteen-node Channel Models

As discussed in Section 4.4, the derivative of the power generation term in the heat transfer equations adds a significant amount of nonzero entries to the Jacobian matrix if the scaling factor R is not treated as constant in a given outer iteration. However, it was shown in Table 4.1 that for a given set of system parameters the derivative value can change by up to 32.6% if the scaling factor R is neglected from the power generation derivative. To determine if the computational effort required to calculate the derivatives of the scaling factor R is worthwhile, the total number of implicit outer iterations, CPU time, and total number of coupled iterations were tracked for the TN and NPI implicit solution methods with and without the scaling factor derivatives. Additionally, the aforementioned parameters were tracked for comparison purposes for the PI implicit method and standard explicit solution. Table 5.10 and Table 5.11 summarize the aforementioned data for the five-node and fourteen-node TRACE models, respectively. For each of the implicit solution methods shown in Table 5.10 and Table 5.11, no increase or decrease in local and global convergence accuracy was seen when scaling factor derivatives were included or excluded. The same level of differences that were reported in the

previous section were also seen when the scaling factor derivatives were excluded.

Solution Method	Total Number of Outer Iterations	Total Number of Coupled Iterations	Average Total CPU Time (s)
TN $\left(\frac{\partial R}{\partial X} = 0\right)$	255	59	1.0171 ± 0.0202
TN $\left(\frac{\partial R}{\partial X} \neq 0\right)$	246	59	1.0070 ± 0.0180
NPI $\left(\frac{\partial R}{\partial X} = 0\right)$	282	59	1.0312 ± 0.0169
NPI $\left(\frac{\partial R}{\partial X} \neq 0\right)$	272	59	1.0171 ± 0.0209
PI	745	59	1.3159 ± 0.0214
Explicit	TRACE: 59 PARCS: 121	55	0.8198 ± 0.0131

Table 5.10: Summary of runtime statistics for five-node two-phase TRACE channel model

Solution Method	Total Number of Outer Iterations	Total Number of Coupled Iterations	Average Total CPU Time (s)
TN $\left(\frac{\partial R}{\partial X} = 0\right)$	478	160	1.8548 ± 0.0151
TN $\left(\frac{\partial R}{\partial X} \neq 0\right)$	493	160	2.4983 ± 0.0281
NPI $\left(\frac{\partial R}{\partial X} = 0\right)$	553	160	2.0062 ± 0.0267
NPI $\left(\frac{\partial R}{\partial X} \neq 0\right)$	550	160	2.6582 ± 0.0397
PI	1167	155	4.2190 ± 0.0433
Explicit	TRACE: 162 PARCS: 311	155	1.0241 ± 0.0173

Table 5.11: Summary of runtime statistics for the fourteen-node two-phase TRACE channel model

Figure 5.4a and Figure 5.4b show the impact of the scaling factor on the structure of and the number of nonzero entries in the Jacobian matrix for the TN linearization solution scheme applied to the fourteen-node channel model. In Figure 5.4a there are 1301 nonzero entries and in Figure 5.4b there are 4889 nonzero entries. The same trend can be seen for the NPI linearization method but with added nonzero entries in the last row for derivatives of cross sections with respect to principle thermal-hydraulic and heat conduction variables in the eigenvalue linearization equation. With the data in Table 5.10 and Table 5.11, it is clear that the additional nonzero entries added by the scaling factor R derivatives will be a significant computational burden for large problems.

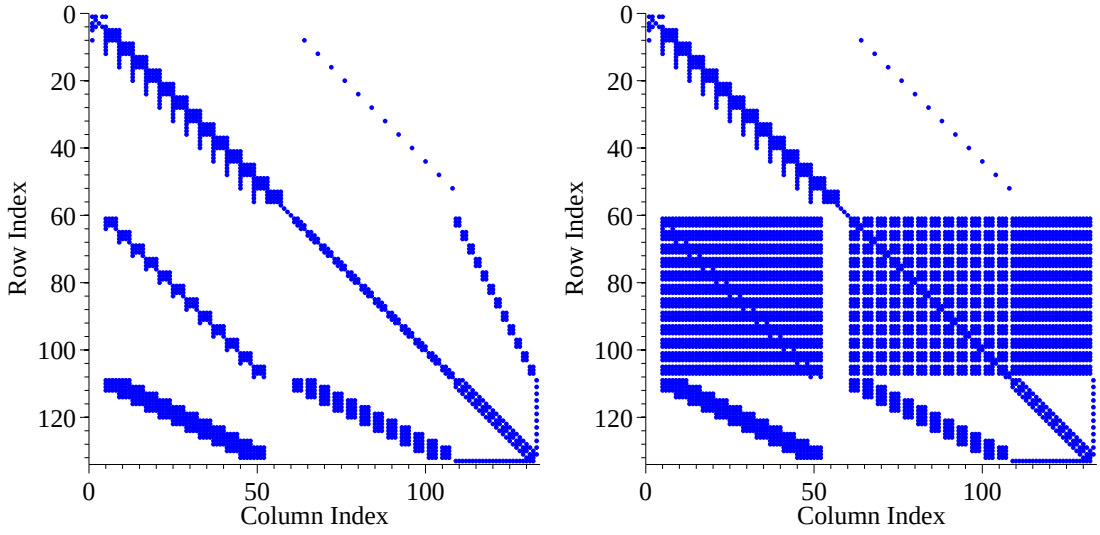


Figure 5.4: Jacobian matrix structure for the implicit solution method without the inclusion of power scaling factor derivatives (a) and with the inclusion of scaling factor derivatives for the fourteen-node channel model

5.2 Solution of a Scaled Reactor Core Model

In Section 3.3 the five-node and fourteen-node channel models were introduced as a basic test to determine if the implicit steady state solution method had been implemented correctly. These models, however, are not adequate to fully assess the viability of using the implicit steady state solution methods for a full core analysis. In a full core analysis the vessel and channel components in TRACE are used to build a full reactor system model which significantly increases the complexity of the discrete thermal-hydraulic, heat conduction, and reactor physics equations. To test if the implicit steady state solution method that has been implemented is capable of solving these more complex multi-component models, a multi-channel core model was created to further assess the implicit steady state solution method. This model will be referred to throughout this document generically as the core model or four-channel core model.

The core model consists a reactor vessel component with three axial finite volume regions. Each axial finite volume region has four azimuthal sectors and two radial regions, producing a total of eight finite volumes per axial level in the

vessel component. The outer radial region models the downcomer which is coupled to four identical inlet pipes at the top of the reactor vessel. The inner radial region contains the active fuel region in the center of the core. The active fuel region is composed of four channel components.

Each channel component in the model by default contains a neutronics reflector region at the top and bottom of the component. In between the the reflector region is an active fuel region which is coupled to a heat structure and neutronics mesh. In the core model presented in this section each of the four channel components consists of 27 axial thermal-hydraulic volumes which are coupled to 27 neutronics volumes and and 25 heat structure volumes. The 25 axial heat structure volumes are each composed of eight conduction volumes (9 computational nodes) which model the fuel pin. Unlike the primitive channel model introduced in Section 3.3, the fuel element in this model includes a gas gap between the fuel and cladding which more accurately represents a the structure of a fuel pin.

The dimensions of the reactor vessel and channel components were approximated using dimensions given in the PBTT benchmark [26]. However, because the vessel only contains four fuel bundles, not all of the dimensions in the TRACE model match the PBTT benchmark because the size of the reactor vessel was significantly reduced. Additionally, the complexity of the reactor vessel was greatly reduced when it was decided that the vessel component would contain only three axial volumes. Differences between the TRACE core model and the PBTT benchmark specifications are not a cause for concern because the object of this analysis is to compare the implicit and explicit steady state solution methods and not to necessarily reproduce PBTT benchmark results. Even with the reduced complexity of the vessel geometry, the four-channel core model used in this analysis will illuminate key challenges to using the implicit steady state solution method for full core analyses like the PBTT benchmark.

The challenge of implementing the implicit solution method to the core model stems primarily from the significant increase in model complexity compared to the five-node and fourteen-node channel models. In contrast to the pipe component used in the channel model shown in Section 3.3, the reactor vessel is a three-dimensional component in TRACE which is coupled to the one-dimensional channel components. The three-dimensional nature of the vessel component sig-

nificantly increases the complexity of the thermal-hydraulic equations; however, this complexity is already handled by the implicit transient solution. More specifically, because the semi-implicit formulation of the thermal-hydraulic equations is used from the explicit solution method, the construction of the discrete form of the thermal-hydraulics equations is not a particularly challenging task. Similarly, the construction of the heat conduction equations and steady state neutron diffusion equations is a task that is largely accomplished by existing programming in the explicit and implicit solution methods. However, the algorithms written to calculate and arrange Jacobian matrix values related to thermal hydraulic, heat conduction, and neutron flux derivatives in the implicit steady state solution had only been applied to a simple one-dimensional geometry shown in Section 3.3. As a result, the extension of the implicit steady state solution algorithms to a significantly more complicated three-dimensional geometry with a wider range of TRACE components was not a trivial task.

5.2.1 Implicit Steady State Results for the Four-Channel Core Model

The explicit and implicit solution methods were used to calculate a steady state solution to the model described above. Both of the methods were run with a maximum time step size of one second in order to leave the determination of an appropriate time step size to the internal logic of TRACE. The implicit solution method was run using the TN eigenvalue linearization scheme without the scaling factor derivatives. The choice of the eigenvalue linearization scheme was made entirely on being able to solve the core model in the least amount of time.

The implicit solution method produced a k_{eff} value of 1.033193 for the core model, and the explicit solution method produced a k_{eff} value of 1.033196. The implicit solution method required a CPU time of 16777 seconds (4.66 hours) compared to the explicit solution method that only required 728.8 seconds (12.1 minutes). The temporal behavior of the fuel centerline temperature for axial heat structure node twelve and the fluid pressure for axial thermal-hydraulic volume thirteen of the second fuel bundle are shown in Figure 5.5a and Figure 5.5b, respectively. The implicit and explicit solution methods illustrate nearly identical

behavior at the center of channel two. The steady state pressure difference for the thermal-hydraulic axial volume thirteen in channel was one Pascal. The steady state fuel centerline temperature difference for the heat conduction axial volume twelve was 0.057 Kelvin.

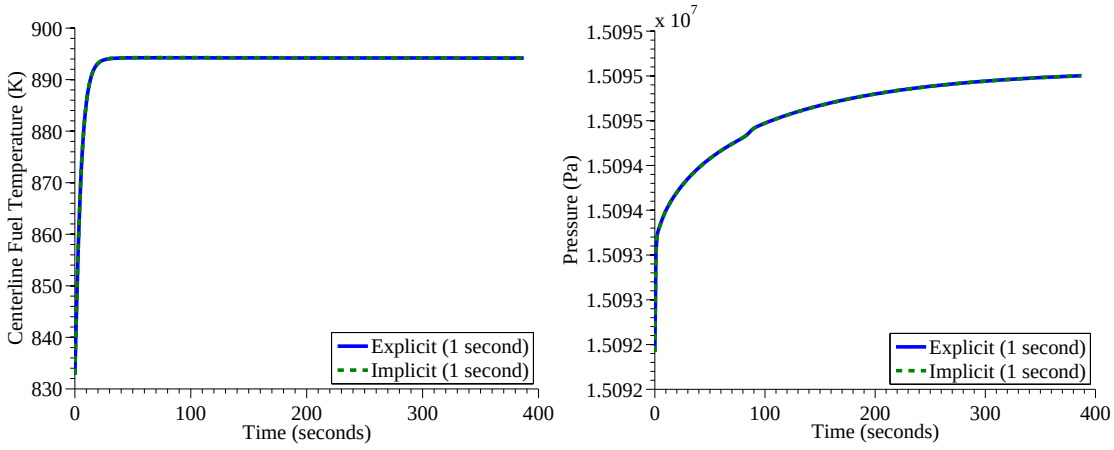


Figure 5.5: Temporal behavior of the fuel centerline temperature of axial conduction node twelve (a) and the pressure of the axial thermal-hydraulic node thirteen (b) in channel two of the four-channel core model

As shown the previous five-node and fourteen-node channel models, it is also important to compare locally converged quantities. Table 5.12 compares the linear power distribution per rod for channel two of the core model. Only channel two is shown because all of the channels show nearly identical results because of the symmetry maintained in the core model. Unlike the five-node and fourteen-node models shown in the previous section, the core model shows slightly more variation in the linear power rate reported in the TRACE output file near the bottom of the core. The results, however, still seem reasonable because there is no apparent anomalies in the temporal behavior of the solution as shown in Figure 5.5 that would indicative of a significant error in the implementation of the implicit solution method.

Axial Node Number (s)	Explicit Linear Power (W/m)	Implicit Linear Power (W/m)	Percent Difference
1	7.6820E+02	7.7337E+02	6.7300E-01
2	2.0580E+03	2.0623E+03	2.0894E-01
3	4.3000E+03	4.2829E+03	-3.9767E-01
4	6.5737E+03	6.5606E+03	-1.9928E-01
5	8.6070E+03	8.5962E+03	-1.2548E-01
6	1.0413E+04	1.0405E+04	-7.6827E-02
7	1.1973E+04	1.1967E+04	-5.0113E-02
8	1.3278E+04	1.3274E+04	-3.0125E-02
9	1.4329E+04	1.4326E+04	-2.0937E-02
10	1.5131E+04	1.5131E+04	0.0000E+00
11	1.5695E+04	1.5696E+04	6.3715E-03
12	1.6031E+04	1.6033E+04	1.2476E-02
13	1.6147E+04	1.6150E+04	1.8579E-02
14	1.6051E+04	1.6054E+04	1.8690E-02
15	1.5749E+04	1.5753E+04	2.5398E-02
16	1.5247E+04	1.5252E+04	3.2793E-02
17	1.4547E+04	1.4552E+04	3.4371E-02
18	1.3651E+04	1.3657E+04	4.3953E-02
19	1.2560E+04	1.2565E+04	3.9809E-02
20	1.1276E+04	1.1281E+04	4.4342E-02
21	9.8023E+03	9.8069E+03	4.6928E-02
22	8.1495E+03	8.1536E+03	5.0310E-02
23	6.3327E+03	6.3360E+03	5.2110E-02
24	4.3756E+03	4.3781E+03	5.7135E-02
24	2.3269E+03	2.3283E+03	6.0166E-02

Table 5.12: Steady state axial linear power distribution per rod for channel two of the four-channel core model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of one second

A comparison of the normalized axial power distribution for the core model is shown in Table 5.13 for the implicit and explicit methods run with a maximum allowable time step of one second. Again, the normalized axial power distribution was not plotted because the values are close enough that the power distribution curves are almost indistinguishable. The differences in the normalized axial power distribution for the core model are larger than those seen for the five-node and fourteen-node channel models. However, the differences are still small enough

to be confident that the implicit steady state solution method was successfully expanded for three-dimensional geometries typically analyzed for full core models.

Axial Node Number (s)	Explicit Normalized Power Fraction	Implicit Normalized Power Fraction	Percent Difference
1	0.072	0.073	1.3889
2	0.194	0.194	0.0000
3	0.405	0.403	-0.4938
4	0.619	0.618	-0.1616
5	0.811	0.810	-0.1233
6	0.981	0.980	-0.1019
7	1.128	1.127	-0.0887
8	1.251	1.251	0.0000
9	1.350	1.350	0.0000
10	1.425	1.425	0.0000
11	1.479	1.479	0.0000
12	1.510	1.510	0.0000
13	1.521	1.521	0.0000
14	1.512	1.512	0.0000
15	1.484	1.484	0.0000
16	1.436	1.437	0.0696
17	1.370	1.371	0.0730
18	1.286	1.287	0.0778
19	1.183	1.184	0.0845
20	1.062	1.063	0.0942
21	0.923	0.924	0.1083
22	0.768	0.768	0.0000
23	0.597	0.597	0.0000
24	0.412	0.412	0.0000
24	0.219	0.219	0.0000

Table 5.13: Steady state normalized axial power distribution for channel two of the four-channel core model as calculated by the explicit and implicit solution methods for a maximum allowable time step size of one second

Chapter 6

Discussion

In this chapter an analysis and discussion will be presented for the results found in Chapter 5. In Section 6.1 the discussion will focus on the accuracy of the implicit solution method. In Section 6.2 the analysis and discussion will pertain to the efficiency of the implicit solution. Section 6.2 will also include a discussion of possible methods for improving the implicit steady state solution method.

6.1 Implicit Solution Accuracy

The most important evaluation criterion for the implicit solution method is whether or not calculated steady state values are correct. The results presented in Chapter 5 offer an encouraging outlook for the implicit solution method with respect to accuracy. Specifically, the data in Table 5.1 shows that the implicit solution method produces the same eigenvalue results for the five-node channel model at varying maximum time step values. The agreement in the eigenvalue indicates an equivalent global convergence, but it is important to also have local convergence in each finite volume in the discretized domain. Figure 5.1 and Figure 5.2 show that the implicit steady state solution exhibits favorable local convergence qualities. The data in Table 5.2 and Table 5.3 show that the axial linear power distribution for the implicit and explicit solutions are in close agreement. Additionally, Table 5.6 through Table 5.9 show that the normalized axial power distribution for the five-node and fourteen-node channel models agree nearly exactly. There is, however, a nonzero discrepancy between the implicit and explicit axial linear power distribu-

tions for the five-node and fourteen-node channel models, and it is important to acknowledge that the source of this error is not completely known. The error could be the result of different temporal paths for the pseudo-transient simulation in the explicit and implicit solution methods, or the discrepancy could be the result of an error in the implementation of the implicit solution method. The difference might also be the manifestation of the fundamentally different iterative methods that are employed for the explicit and implicit methods.

The global and local converged steady state values for the core were slightly larger than the relatively small differences seen in the five-node and fourteen-node channel models. Additionally, while the normalized axial power distributions for the five-node and fourteen node channel models agreed almost exactly, there was a larger difference in the results for the core model. Like before, no systematic error in the implementation of the implicit steady state solution could be found, and at this time it is believed that the differences are most likely related to the fundamentally different approach of the implicit and explicit solution methods.

6.2 Implicit Solution Efficiency

The data in Table 5.10 and Table 5.11 show that the inclusion of the scaling factor derivatives had a mixed impact on the number of total outer iterations in the implicit solution. However, for the larger 14-node problem, the inclusion of the scaling factor derivatives had a significant impact on the total CPU time. More specifically, the TN and NPI implicit solution methods were slower by 34.7% and 32.5% with the scaling factor derivatives for the 14-node model, respectively. For these particular test problems the aforementioned data suggests that for realistic meshes such as the 14-node problem the increased computational expense of calculating and storing the additional scaling factor derivatives outweighs any gains that might be realized by a reduction in total number of outer iterations. In fact, Table 5.11 shows that the inclusion of the scaling factor derivative does not guarantee a reduction in the total number of outer iterations. This outcome may or may not be applicable to every problem set, but it is encouraging from a practical viewpoint that the exclusion of the complicated scaling factor derivatives appears to be the more efficient solution method.

The data in Table 5.10 and Table 5.11 also shows that it is advantageous to include the eigenvalue as a principle unknown in the Jacobian matrix. For the fourteen-node channel model runtime data shown in Table 5.11, the inclusion of the eigenvalue as a principle unknown in the Jacobian matrix reduced the number of outer iterations required in the implicit solution by at least 53%. This reduction of outer iterations had an equally significant effect on overall runtime by reducing the CPU runtime for the implicit solution method by at least 37%.

Despite the advantages of including the eigenvalue in the Jacobian as a principle unknown, the data in Table 5.10 and Table 5.11 shows that the implicit method requires more CPU time than the explicit method to converge to the same steady state solution. There are several contributing factors to the longer runtime of the implicit method, some of which can be mitigated as the implicit solution method is optimized. One of the contributions to the extended runtime of the implicit solution is the convergence criteria used in the implicit solution. In the implicit solution all of the thermal-hydraulic, heat transfer, and neutronics principle unknowns are required to converge for each coupled calculation to the user-specified criteria in the TRACE and PARCS input decks. However, the explicit solution allows for a thermal-hydraulic calculation to be performed without requiring that the flux and eigenvalue solution from PARCS is fully converged. This significantly reduces the total number of outer iterations performed by PARCS in the explicit calculation, but more importantly, increases the number of outer iterations required in the implicit calculation. To emphasize this point, the calculated values for the explicit convergence criteria in PARCS were divided by the user-specified convergence criteria and plotted for each coupled calculation. The aforementioned convergence criteria includes k_{eff}, L_2, and L^{∞}, and the results are plotted in Figure 6.1.

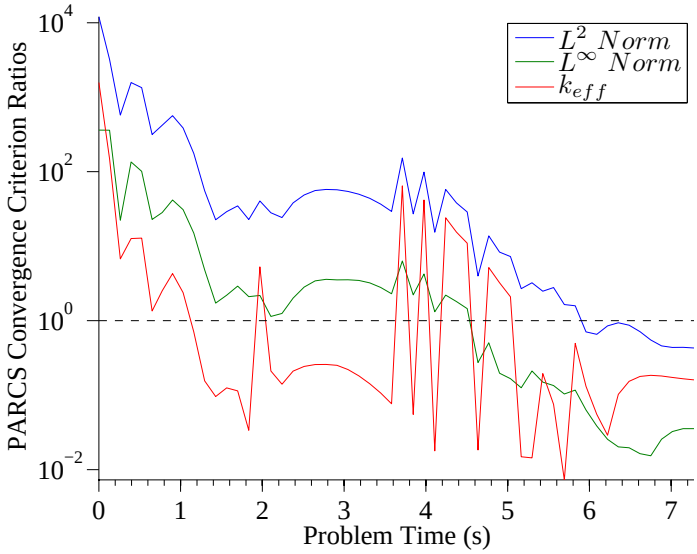


Figure 6.1: Convergence criterion ratios for the five-node two-phase TRACE channel model

Each of the convergence criterion shown in Figure 6.1 is orders of magnitude above the user-specified convergence criteria at the beginning of the steady state calculation. More specifically, the L^∞-norm has a maximum value 0.18 during the steady state calculation which is well above the specified convergence criterion of 5×10^{-4}. Similarly, the the L^2-norm has a maximum value of 0.12 which is well above the user specified criterion of 1×10^{-5}. Also, the k-effective convergence criterion has a maximum value of 2.4×10^{-3} which is well above the 1×10^{-6} user specified convergence criterion. It is also worth noting that the ratio of the calculated convergence criteria to the user-specified convergence criteria does not go below a value of one until late in the steady state solution.

This lack of complete convergence during the explicit PARCS calculation is tolerable because temporal accuracy is not important during the beginning of the steady state calculation. However, with the required convergence of all of the parameters in the implicit solution methods, the implicit solution method is at a disadvantage because more outer iterations must be performed to meet all of the convergence criteria. Relaxing the convergence criteria in a manner similar to the explicit method for some or all of the parameters in the implicit method is currently being investigated in order to reduce the runtime of the implicit calculation.

The primary source of inefficiency in the implicit solution is the construction

and solution of the Jacobian matrix. In the explicit solution in TRACE and PARCS, the linear systems that must be solved are much smaller than the implicit system of equations. The disparity in the size of the linear systems is particularly relevant to the scaling potential of the implicit method. The data in Table 5.10 and Table 5.11 show that the five-node implicit calculation is 22% slower than the explicit calculation and the fourteen-node implicit calculation is 81% slower than the explicit calculation. The results of the core model presented in Chapter 5 show that in its current form the implicit steady state solution is not practical for use with large three-dimensional models because of the large amounts of time needed to solve the Jacobian matrix equation using a direct solver. To prevent the runtime from becoming a limiting quality that determines the effectiveness of the implicit steady state solution, an iterative solver should be used instead of a direct solver to solve the linear system associated with each Newton's method iteration. Additionally, reducing the overall size of the matrix by using a method similar to the implicit heat conduction and thermal-hydraulic coupling in TRACE should be considered for future work.

The introduction of an iterative solver will most likely still not be enough to make the implicit solution method faster than the explicit solution method. In order for the implicit solution method to be beneficial with respect to CPU runtime, it must run at a larger time step size than the explicit solution method during the pseudo-transient solution. Because the implicit solution method uses the semi-implicit formulation of the thermal-hydraulic equations, the disparity in time step size would have to come from time step size limitations related to the OS methodology used for the explicit solution. In the small set of test problems used to evaluate the implicit steady state solution, the explicit solution method was not limited by the OS technique. In fact, in almost every time step history both the implicit and explicit solution methods were limited by the material Courant number. However, before beginning to develop a test base of problems that exploits potential limitations of the OS method, the computational inefficiency of solving the Jacobian matrix must be addressed because it prevents the implicit steady state solution method from being used to solve a larger set of problems where OS limitations can be investigated.

Chapter 7

Conclusions and Future Work

In this document the methodology used to implement an implicit steady state solution into the framework of the implicit transient solution in TRACE and PARCS was presented. For the one-dimensional models the implicit steady state solution produces the correct steady state eigenvalue solution for varying specified time-step sizes but showed small nonzero variations in local axial power values. For a multi-channel three-dimensional model that was used to test the implicit steady state solution, CPU runtime was prohibitively large and the local and global steady state values differed slightly from those obtained with the explicit solution method. The magnitude of these differences, however, are not indicative of a fundamental error in the implicit solution method but are most likely the byproduct of the fundamental difference in the iterative schemes of the implicit and explicit solution methods.

The results in this document show that as the models become more complex and representative of models that are used in full reactor system analyses, the required CPU time for the implicit solution method in its current form will be prohibitive. However, the implicit steady state solution has yet to be optimized. Improvements to the implicit steady state solution methodology that could reduce CPU runtime include, but are not limited to, reducing coupled calculation convergence criterion when temporal accuracy is not required, using a more efficient linear solver for each iteration of Newton's method, and reducing the overall size of the matrix by implementing an implicit, coupled solution strategy that is similar to the implementation of the heat conduction solution in TRACE.

Future work regarding the implicit steady state solution in TRACE and PARCS should first focus on reducing the overall runtime. The easiest method for reducing the runtime will be the replacement of the direct solver currently used for Newton's method with an iterative solver that scales more favorably with the number of nonzero entries in the Jacobian matrix. The implementation of an iterative scheme for the implicit equation set that is similar to the current heat conduction solution in TRACE should also be strongly considered as an immediate continuation of the research in this document because it has the potential to benefit both the steady state implicit solution and the transient implicit solution. The large amount of work that has been devoted to developing the analytic expressions for the Jacobian matrix in the implicit solution method will prove very useful in the implementation of an iterative scheme for the implicit equation set that is similar to the current heat conduction solution in TRACE. If the runtime of the implicit steady state solution method can be reduced enough through these methods to make full core analyses more practical, larger problem sets should be developed in the future to investigate the limitations of the operator splitting method and potential advantages of the implicit steady state solution.

Bibliography

[1] Ragusa, J. C. and V. S. Mahadevan (2009) "Consistent and accurate schemes for coupled neutronics thermal-hydraulics reactor analysis," *Nuclear Engineering and Design*, **239**(3), pp. 566 – 579.

[2] Knoll, D. A., L. Chacon, L. G. Margolin, and V. A. Mousseau (2003) "On balanced approximations for time integration of multiple time scale systems," *Journal of Computational Physics*, **185**(2), pp. 583 – 611.

[3] Watson, J. K. (2010) *Implicit Time-Integration Method For Simultaneous Solution of a Coupled Non-Linear System*, Ph.D. thesis, The Pennsylvania State University.

[4] Mousseau, V. A. (2004) "Implicitly balanced solution of the two-phase flow equations coupled to nonlinear heat conduction," *Journal of Computational Physics*, **200**(1), pp. 104 – 132.

[5] Frepoli, C., J. H. Mahaffy, and K. Ohkawa (2003) "Notes on the implementation of a fully-implicit numerical scheme for a two-phase three-field flow model," *Nuclear Engineering and Design*, **225**(2-3), pp. 191 – 217.

[6] Mousseau, V. A. (2005) "A Fully Implicit Hybrid Solution Method for a Two-Phase Thermal-Hydraulic Model," *Journal of Heat Transfer*, **127**(5), pp. 531–539.

[7] Kastanya, D. Y. F. and P. J. Turinsky (2005) "Development and Implementation of a Newton-BICGSTAB Iterative Solver in the FORMOSA-B BWR Core Simulator Code," *Nuclear Science and Engineering*, **150**(1), pp. 56–71.

[8] Mousseau, V. A. (2006) "A Fully Implicit, Second Order in Time, Simulation of a Nuclear Reactor Core," in *Proceedings of ICONE 14*, Miami, Florida.

[9] Mousseau, V. A. and M. A. Pope (2007) "Accuracy and Efficiency of a Coupled Neutronics and Thermal Hydraulics Model," in *The 12th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-12)*, Sheraton Station Square, Pittsburgh, Pennsylvania, U.S.A.

[10] Ferziger, J. and M. Peric (2002) *Computational Methods for Fluid Dynamics*, Springer.

[11] Lamarsh, J. R. (1975) *Introduction to nuclear engineering*, Addison-Wesley Publishing Company.

[12] Downar, T., Y. Xu, and V. Seker (2009) *PARCS v3.0 U.S. NRC Core Neutronics Simulator Theory Manual*, Department of Nuclear Engineering and Radiological Sciences, University of Michigan Ann Arbor, MI 48109.

[13] Sutton, T. M. and B. N. Aviles (1996) "Diffusion theory methods for spatial kinetics calculations," *Progress in Nuclear Energy*, **30**(2), pp. 119 – 182.

[14] Lee, D., T. J. Downar, and Y. Kim (2003) "Convergence Analysis of the Nonlinear Coarse Mesh Finite Difference Method," in *Nuclear Mathematical and Computational Sciences: A century in Review, A Century Anew*, American Nuclear Society, Gatlinburg, Tennessee.

[15] Wachspress, E. L. (1966) *Iterative Solution of Elliptic Systems And Applications to the Neutron Diffusion Equations of Reactor Physics*, Prentice-Hall, Inc.

[16] Duderstadt, J. J. and L. J. Hamilton (1976) *Nuclear Reactor Analysis*, 2nd ed., John Wiley & Sons.

[17] Adams, C. (1977) *Current Trends in Methods for Neutron Diffusion Calculations*, *Tech. rep.*, Argonne National Laboratory.

[18] Tu, J., G. H. Yeoh, and C. Liu (2008) *Computational Fluid Dynamics*, Elsevier.

[19] Sutton, T. (1988) "Wielandt Iteration as Applied to the Nodal Expansion Method," *Nuclear Science and Engineering*, **98**, pp. 169–173.

[20] Chao, Y. (2000) "Coarse mesh finite difference methods and applications," in *Proc. Int. Conf. on Physics of Reactors (PHYSOR2000)*, pp. 7–12.

[21] Bajorek, S. *et. al.* (2007) *TRACE V5.0 Theory Manual Patch 2- Field Equations, Solution Methods, and Physical Models*, U. S. Nuclear Regulatory Commission, Washington, DC, 1st ed.

[22] McDonald, P. W. (1971) "The Computation of Transonic Flow through Two-Dimensional Gas Turbine Cascades," *ASME Paper 71-GT-89*, Gas Turbine Conference and Products Show, Houston, Texas.

[23] MacCormack, R. W. and A. Paullay (1972) "Computational Efficiency Achieved by Time Splitting of Finite Difference Operators," *AIAA Paper 72-154*, San Diego, CA.

[24] Rizzi, A. W. and M. Inuoye (1973) "Time Split Finite Volume Method for 3D Blunt Body Flows," *AIAA Journal*, **11**(11), pp. 1478–1485.

[25] Bajorek, S. *et. al.* (2007) *TRACE V5.0 User's Manual - Volume 1: Input Specification*, U. S. Nuclear Regulatory Commission, Washington, DC, 1st ed.

[26] Larsen, N. H. (1978) *Core Design and Operating Data for Cycle 1 and 2 of Peach Bottom 2, Tech. rep.*, Electric Power Research Institute (EPRI).

[27] Gill, D. F. (2009) *Newton-Krylov Methods for the Solution of the k-eigenvalue problem in Multigroup Neutronics Calculations*, Ph.D. thesis, The Pennyslvania State University.

[28] Xu, Y. and T. Downar (2006) *GenPMAXS Code for Generating the PARCS Cross Section Interface File PMAXS*, Purdue University School of Nuclear Engineering.

[29] Gill, D. F. and Y. Y. Azmy (2011) "Newton's Method for Solving k-Eigenvalue Problems in Neutron Diffusion Theory," *Nuclear Science and Engineering*, **167**(2), pp. 141–153.

[30] Knoll, D. A., H. Park, and C. Newman (2011) "Acceleration of k-Eigenvalue/Criticality Calculations Using the Jacobian-Free Newton-Krylov Method," *Nuclear Science and Engineering*, **167**(2), pp. 133–140.

[31] Kastanya, D. F. (2003) *Implementation of a Newton-Krylov iterative method to address strong non-linear feedback effects in FORMOSA-B BWR core simulator*, Ph.D. thesis, North Carolina State University.

Appendix A

Input Files for TRACE and PARCS Five-Node Heated Pipe Model

The input files for the five-node test problem are included in this appendix. Section A.1 shows the standalone input file which specifies the hydraulic and heat transfer component geometry and boundary conditions. Section A.2 shows the TRACE restart file for the five-node channel model. To change the eigenvalue linearization scheme for the implicit steady state solution, the *namelist* variable *ss_couple* must be added. Also, the *namelist* variable *n_couple* must be included to activate the implicit solution method. Section A.3 shows the PARCS restart file for the five-node channel model.

A.1 Standalone TRACE Input File

```
free format
*
*
*************
* main data *
*************
*
*       numtcr       ieos       inopt       nmat       id2o
            2          0          1          0          0
*
*
*****************
* namelist data *
*****************
*
```

```
 &inopts
  cpuflg=1,
  itdmr=0,
  nosets=1,
  nsolver=1,
  usesjc=3,
  npower=1,
  nhtstr=1
 &end
*
***************
* Model Flags *
***************
*
*         dstep          timet
              0            0.0
*        stdyst         transi          ncomp           njun          ipak
              1              0              5              2             1
*          epso           epss
         1.0E-3        1.0E-10
*        oitmax         sitmax         isolut         ncontr        nccfl
             10             10              0              0             0
*          ntsv           ntcb           ntcf           ntrp          ntcp
              1              0              0              1             0
*
************************
* component-number data *
************************
*
*  Component input order (IORDER)
*-- type ---- num ------------- name -------------  + jun1 jun2 jun3
* FILL      *       1 s * velocity bc               +    1
* PIPE      *       2 s * subcooled liquid channel  +    1    3
* BREAK     *       3 s * pressure bc               +    3
* HTSTR     *       4 s * powered-rod conductor     +
* POWER     *     901 e * power data input test1    +
*
*
************************************************
* Starting Signal Variable Section of Model     *
************************************************
*
*          idsv           isvn           ilcn          icn1          icn2
               1              0              0             0             0
************************************************
* Finished Signal Variable Section of Model     *
************************************************
*
*
*
*
*********************************
* Starting Trip Section of Model  *
*********************************
*
*          Trip Storage Count Card
```

```
*          ntse          ntct          ntsf          ntdp          ntsd
             0             0             0             0             0
*
*trip                     -2
*          idtp          isrt          iset          itst          idsg
            -2             2             0             1             1
*       setp(1)       setp(2)
             0.0           0.1
*       dtsp(1)       dtsp(2)
             0.0           0.0
*       ifsp(1)       ifsp(2)
             0             0
*
*********************************
* Finished Trips: Starting Extras*
*********************************
*
*********************************
* Finished Trip Section of Model *
*********************************
*
*
*
*
******* type          num       userid          component name
fill                    1            1             velocity bc
*         jun1        ifty        ioff
             1           2           0
*       twtold        rfmx       concin        felv
             0.0         0.0         0.0         0.0
*         dxin        volin        alpin        vlin         tlin
             1.0     0.019079         0.0         0.0        548.0
*          pin         pain       flowin        vvin         tvin
     6.9684E6          0.0        12.65         0.0        548.0
*
*
******* type          num       userid          component name
pipe                    2            1    subcooled liquid channel
*         ncells      nodes        jun1          jun2          epsw
              5          0           1             3           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     *           1.0       1.2192       1.2192       1.2192s
* dx     *           1.0e
* vol    *      0.019079 0.013666013 0.013666013 0.013666013s
* vol    *      0.019079e
* fa     *      0.019079     0.011209     0.011209     0.011209s
* fa     *      0.011209     0.019079e
* fric   *           0.0          0.0          0.0          0.0s
* fric   *           0.0          0.0e
```

```
* grav   *          1.0          1.0          1.0          1.0s
* grav   *          1.0          1.0e
* hd     *   0.15585935   0.17867779    0.1194644    0.1194644s
* hd     *   0.17867779   0.15585935e
* nff    *            0            0            0           0s
* nff    *            0           0e
* alp    *          0.0         0.01         0.01         0.01s
* alp    *         0.18e
* vl     *          0.0          0.0          0.0          0.0s
* vl     *          0.0          0.0e
* vv     *          0.0          0.0          0.0          0.0s
* vv     *          0.0          0.0e
* tl     *        548.0        558.0        558.0        558.0s
* tl     *        558.0e
* tv     *        558.0        558.0        558.0        558.0s
* tv     *        558.0e
* p      *       6.92E6       6.92E6       6.92E6       6.92E6s
* p      *       6.92E6e
* pa     *          0.0          0.0          0.0          0.0s
* pa     *          0.0e
*
*
*******  type          num         userid          component name
break                    3              1             pressure bc
*         jun1         ibty         isat         ioff     adjpress
            3            0            0            0            0
*         dxin        volin        alpin         tin          pin
          1.0     0.019079          1.0        558.0      6.864E6
*         pain        concin         rbmx        poff         belv
          0.0          0.0          0.0          0.0          0.0
*
*
*****************************************
* Starting Heat Structure Section of Model *
*****************************************
*
*******   type            num       userid            component name
htstr                      4            0        powered-rod conductor
*        nzhstr          ittc        hscyl         ichf
             3             0            1            1
*     nofuelrod         plane       liqlev       iaxcnd
             0             3            0            0
*         nmwrx          nfci         nfcil         hdri         hdro
             0             0            0          0.0          0.0
*          nhot         nodes         fmon        nzmax     reflood
             0             4            0            3            0
*      dtxht(1)      dtxht(2)        dznht        hgapo
           0.0           0.0        100.0       4600.0
*
*  idbcin *          0           0           0e
*  idbcon *          2           2           2e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
*qflxbco1 *         0.0e
* hcomon2 *          2           2           0           0e
* hcomon2 *          2           3           0           0e
```

```
* hcomon2 *          2           4           0           0e
* dhtstrz *        1.4         1.4         1.4e
*   rdx   *       49.0e
*  radrd  *        0.0   3.6347E-3   6.0579E-3   7.1501E-3e
*  matrd  *          1           1           2 e
*    nfax *          1           1           1e
*    rftn *      450.0       450.0       450.0       450.0s
*    rftn *      450.0       450.0       450.0       450.0s
*    rftn *      450.0       450.0       450.0       450.0e
* fpuo2 *        0.0e
*   ftd  *        1.0e
*  gmix * f       0.0e
*   gmles *       0.0e
*   pgapt *       0.0e
*   plvol *       0.0 e
*   pslen *       0.0 e
*   clenn *       0.0 e
*  burn *         0.0         0.0         0.0e
*****************************************
* Finished Heat Structure Section of Model *
*****************************************
*
*
*
*
***************************************
*     Starting Power Components       *
***************************************
*
******  type          num        userid          component name
power                 901             1   power data input test1
*      numpwr      chanpow
            1            0
* htnum *          4 e
*      irpwty        ndgx        ndhx        nrts        nhist
            5           0           0           5           0
*      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
            0           0           0           1           0
*       react        tneut       rpwoff      rrpwmx       rpwscl
          0.0          0.0         0.0         0.0         1.0
*      rpowri        zpwin       zpwoff      rzpwmx
     4.3102E6          0.0         0.0         0.0
*       extsou        pldr        pdrat       fucrac
          0.0          0.0         1.3         0.7
* rdpwr *        1.0         1.0         1.0         0.0e
* cpowr *        1.0e
* zpwtb1*        1.0s
* zpwtb1*        1.0         1.0         1.0e
***************************************
*     Finished Power Components       *
***************************************
*
```

```
*
*
end
*
*****************
* Timestep Data *
*****************
*         dtmin          dtmax           tend          rtwfp
         1.0E-5            1.0         1000.0           10.0
*         edint          gfint         dmpint         sedint
          100.0           10.0            1.0           50.0
*
*       endflag
            -1.0
```

A.2 TRACE Restart Input File

```
free format
*
*
*************
* main data *
*************
*
*       numtcr       ieos       inopt        nmat        id2o
            2          0           1           0           0
*
*
*
*****************
* namelist data *
*****************
*
 &inopts
  cpuflg=0,
  itdmr=1,
  nsolver=1,
  usesjc=3,
  npower=1,
  nhtstr=1,
  graphlevel="full",
  nosets=1
 &end
*
***************
* Model Flags *
***************
*
*         dstep          timet
             -1            0.0
*        stdyst         transi          ncomp           njun         ipak
             1              0              5              2            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              1          0
*
************************
* component-number data *
************************
*
*  Component input order (IORDER)
- type ---- num ------------ name ------------  + jun1 jun2 jun3
* FILL      *      1 s * velocity bc            +    1
* PIPE      *      2 s * subcooled liquid channel +  1    3
* BREAK     *      3 s * pressure bc            +    3
* HTSTR     *      4 s * powered-rod conductor  +
* POWER     *    901 e * power data input test1 +
*
*
************************************************
* Starting Signal Variable Section of Model     *
************************************************
*
*          idsv           isvn           ilcn          icn1       icn2
             1              0              0             0          0
************************************************
* Finished Signal Variable Section of Model     *
************************************************
*
*
*
*
*********************************
* Starting Trip Section of Model  *
*********************************
*
*          Trip Storage Count Card
*          ntse           ntct           ntsf          ntdp       ntsd
             0              0              0             0          0
*
*trip                       -2
*          idtp           isrt           iset          itst       idsg
            -2              2              0             1          1
*       setp(1)        setp(2)
            0.0            0.1
*       dtsp(1)        dtsp(2)
            0.0            0.0
*       ifsp(1)        ifsp(2)
              0              0
*
*********************************
* Finished Trips: Starting Extras*
*********************************
*
*********************************
* Finished Trip Section of Model *
*********************************
*
```

```
*************************************
*     Finished Power Components     *
*************************************
*
*
*
end
*
*****************
* Timestep Data *
*****************
*         dtmin          dtmax           tend          rtwfp
          1.0E-5           1.0         1000.0           10.0
*         edint          gfint         dmpint         sedint
           1.0           1.0E-5           1.0         1.0E-5
*
*       endflag
             -1.0
```

A.3 PARCS Restart Input File

```
!************************************************************
CASEID ss          Small Test Problem
!************************************************************
CNTL
      core_type  BWR
      core_power 100.00000  ! Initial core power level in %
      ppm        0.0        ! Initial boron concentration in ppm
!                bank1 (480-out,0-in)
      bank_pos   480.0
      th_fdbk    T
      xe_sm      0 0
!                0 : no xe, 1 : eq. xe   2 : tr. xe  3 : given xe
      decay_heat T
      ROT_ADF    T
!     {ROT_ADF is the assembly rotation option, T or F}
!     for LPRM
      pin_power  F
!     {Pin power reconstruction option, T or F
      ext_th     T    ../maptab_1ch      TRAC    1    1
      transient  F
      restart    F   ss_1.rst   0
!                    input  iteration     planar              adj
!                     edit      table      power      pin      reac
      print_opt        T          F          T        F        F
!                     fdbk       flux     planar
!                      rho     precurs       flux       Xe      T/H
      print_opt        F          F          F        F        T
!!                    oneD     Radial       assy
!!                   const      Shape      const
!     print_opt        F          F          F
!
!                     oneD       PKRE     Radial   Radial     assy
!                    const       Data      Shape    Shape    const
      print_opt        F          F          F        T        F
```

```
      oned_kin    F    SA1D
!************************************************************
PARAM !table 6
      n_iters     5       500
      conv_ss     1.0e-6   1.e-5   5.e-4     0.001
!                 keff,    epsl2, epslinf, tempf
      wielandt    0.04     0.1      1.0
      nodal_kern HYBRID
      nlupd_ss    3 3 1
!      nlupd_ss    100 100 100
      eps_anm     0.005
      eps_erf     0.005
      decusp      0
      init_guess 0
!************************************************************
XSEC  !tabl 7
      func_type   13 435
      dnp_ngrp    6
      EFIL_XS_R   'PBTT_xsec_rodded'
      EFIL_XS_UR 'PBTT_xsec_unrodded'
      kin_comp    1  1 -435
! pbtt specs
      dnp_beta    0.000167 0.001134 0.001022 0.002152 0.000837
         0.000214
      dnp_lambda 0.012813 0.031536 0.124703 0.328273 1.405280
         3.844728
!************************************************************
GEOM  !table 8
      file        ../geom
!************************************************************
TH    !table 9
!fix assume all assembiles are 7x7, no guide/instumentation tubes
      n_pingt     49 0                          !npin,ngt
!fix by Yunlin Xu: 3293/764=4.310209      3293*61.6459% = 2030MW
      FA_POWPIT   4.31021 15.24
!                 assembly power(Mw) and pitch(cm)
!
      pin_dim     5.0000 6.0000 0.900 6.0000
!                 pin radii, rs,rw,tw, and rgt in mm
!fix: 555-273=282, 9585/764=12.548
      flow_cond   282.0 12.548
!                 tin,cmfrfa(Kg/sec)
!
! pbtt spec
      CDC_DED     1 1
!     format:  CDC_DED   CDC_op      DED_op
!     CDC_op:  integer  for coolant density correction option:
!              =0 no correction(old models),
!              =1 correct with by pass density(PBTT)
!              =2 correct with  bypass, and water rod
!                 densities(ESBWR)
!     DED_op:  integer  for direct energy deposition:option:
!              =0 one  whole core factor for coolant(old models),
!              =1 two whole core factors for  coolant, bypass(PBTT)
!              =2 three whole core factors for  coolant,
!                 bypass, water rod ( can be used in ESBWR)
```

```
!              =3 three node-wise coolant factors,  bypass,
!                 water rod ( ESBWR)
      gamma_frac 0.00  0.000   !direct heating fraction
      CDC_DAT    0.761  0.401
      hgap       10000.        !hgap(w/M^2-C)
      n_ring     6             !number of meshes in pellet
      thmesh_x   1*1           !Number of T/H Nodes per FA in X-dir
      thmesh_y   1*1           !Number of T/H Nodes per FA in Y-dir
      thmesh_z   1 2 3         !Junction Locations
      freeze_tf  F    800.0
      freeze_dm  F    300.0
      write_fbv  T  ss3d.fbv
      read_dopl  F  ss3d.fbv
      read_tmdm  F  ss3d.fbv
!************************************************************
!PFF  table 10, power form function
PFF
 npin_side 7
 pff_comp 1    1 -435
 pff_unrodd  1  !group 1 of set 1
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 pff_unrodd  2  !group 2 of set 1
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 pff_rodded  1  !group 1 of set 1
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 pff_rodded  2  !group 2 of set 1
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
!************************************************************
!PLOT
!    file sscoupl.plc
!     file trcoupl.plc
!************************************************************
TRAN  !table 11
     time_step  5.0 0.00001
     expo_opt   F     F
     Scram      F 95.0 0.0 143.
     pin_freq   1                        !control lprm freq too
! The above values need checking
     theta      1.0   0.5   0.5
     conv_tr    0.001
     nlupd_tr   5 1 5 10
     eps_xsec   0.01
     rst_freq   1
.
```

Appendix B

Input Files for TRACE and PARCS Fourteen-Node Heated Pipe Model

The input files for the fourteen-node test problem are included in this appendix. Section B.1 shows the standalone input file which specifies the hydraulic and heat transfer component geometry and boundary conditions. Section B.2 shows the TRACE restart file for the fourteen-node channel model. To change the eigenvalue linearization scheme for the implicit steady state solution, the *namelist* variable *ss_couple* must be added. Also, the *namelist* variable *n_couple* must be included to activate the implicit solution method. Section B.3 shows the PARCS restart file for the fourteen-node channel model.

B.1 Standalone TRACE Input File

```
free format
*
*
*************
* main data *
*************
*
*       numtcr       ieos       inopt       nmat       id2o
             2          0           1          0          0
*
*
*****************
* namelist data *
*****************
*
```

```
 &inopts
  cpuflg=1,
  itdmr=0,
  nosets=1,
  nsolver=1,
  usesjc=3,
  npower=1,
  nhtstr=1
 &end
*
***************
* Model Flags *
***************
*
*         dstep          timet
              0            0.0
*        stdyst         transi          ncomp           njun          ipak
              1              0              5              2             1
*          epso           epss
         1.0E-3        1.0E-10
*        oitmax         sitmax         isolut         ncontr        nccfl
             10             10              0              0             0
*          ntsv           ntcb           ntcf           ntrp          ntcp
              1              0              0              1             0
*
************************
* component-number data *
************************
*
*  Component input order (IORDER)
*-- type ---- num ------------- name -------------  + jun1 jun2 jun3
* FILL      *       1 s * velocity bc               +    1
* PIPE      *       2 s * subcooled liquid channel  +    1    3
* BREAK     *       3 s * pressure bc               +    3
* HTSTR     *       4 s * powered-rod conductor     +
* POWER     *     901 e * power data input test1    +
*
*
************************************************
* Starting Signal Variable Section of Model     *
************************************************
*
*          idsv           isvn           ilcn          icn1          icn2
               1              0              0             0             0
************************************************
* Finished Signal Variable Section of Model     *
************************************************
*
*
*
*
*********************************
* Starting Trip Section of Model  *
*********************************
*
*          Trip Storage Count Card
```

```
*          ntse          ntct          ntsf          ntdp          ntsd
             0             0             0             0             0
*
*trip                     -2
*          idtp          isrt          iset          itst          idsg
            -2             2             0             1             1
*       setp(1)       setp(2)
           0.0           0.1
*       dtsp(1)       dtsp(2)
           0.0           0.0
*       ifsp(1)       ifsp(2)
             0             0
*
*********************************
* Finished Trips: Starting Extras*
*********************************
*
*********************************
* Finished Trip Section of Model *
*********************************
*
*
*
*
******* type          num       userid          component name
fill                    1            1             velocity bc
*        jun1         ifty         ioff
             1            2            0
*      twtold         rfmx       concin         felv
           0.0          0.0          0.0          0.0
*         dxin         volin        alpin         vlin         tlin
           1.0     0.019079          0.0          0.0        548.0
*          pin         pain       flowin         vvin         tvin
     6.9684E6          0.0        12.65          0.0        548.0
*
*
******* type          num       userid          component name
pipe                    2            1   subcooled liquid channel
*        ncells       nodes        jun1         jun2         epsw
            14           0            1            3          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     *           1.0        0.3048       0.3048       0.3048s
* dx     *        0.3048        0.3048       0.3048       0.3048s
* dx     *        0.3048        0.3048       0.3048       0.3048s
* dx     *        0.3048          1.0e
* vol    *      0.019079     3.4165E-3    3.4165E-3    3.4165E-3s
* vol    *     3.4165E-3     3.4165E-3    3.4165E-3    3.4165E-3s
* vol    *     3.4165E-3     3.4165E-3    3.4165E-3    3.4165E-3s
* vol    *     3.4165E-3      0.019079e
```

```
* fa     *      0.019079     0.011209     0.011209     0.011209s
* fa     *      0.011209     0.011209     0.011209     0.011209s
* fa     *      0.011209     0.011209     0.011209     0.011209s
* fa     *      0.011209     0.011209     0.019079e
* fric   *           0.0          0.0          0.0          0.0s
* fric   *           0.0          0.0          0.0          0.0s
* fric   *           0.0          0.0          0.0          0.0s
* fric   *           0.0          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          1.0e
* hd     *    0.15585935   0.17867779   0.17867779   0.17867779s
* hd     *    0.17867779    0.1194644    0.1194644    0.1194644s
* hd     *     0.1194644    0.1194644    0.1194644    0.1194644s
* hd     *     0.1194644   0.17867779   0.15585935e
* nff    *             0            0            0            0s
* nff    *             0            0            0            0s
* nff    *             0            0            0            0s
* nff    *             0            0            0e
* alp    *           0.0         0.01         0.01         0.01s
* alp    *          0.01         0.01         0.01         0.01s
* alp    *          0.01         0.01         0.01         0.01s
* alp    *          0.01         0.18e
* vl     *           0.0          0.0          0.0          0.0s
* vl     *           0.0          0.0          0.0          0.0s
* vl     *           0.0          0.0          0.0          0.0s
* vl     *           0.0          0.0          0.0e
* 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          0.0e
* tl     *         548.0        558.0        558.0        558.0s
* tl     *         558.0        558.0        558.0        558.0s
* tl     *         558.0        558.0        558.0        558.0s
* tl     *         558.0        558.0e
* tv     *         558.0        558.0        558.0        558.0s
* tv     *         558.0        558.0        558.0        558.0s
* tv     *         558.0        558.0        558.0        558.0s
* tv     *         558.0        558.0e
* p      *        6.92E6       6.92E6       6.92E6       6.92E6s
* p      *        6.92E6       6.92E6       6.92E6       6.92E6s
* p      *        6.92E6       6.92E6       6.92E6       6.92E6s
* p      *        6.92E6       6.92E6e
* 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          0.0e
*
*
******* type          num         userid          component name
break                   3              1             pressure bc
*         jun1         ibty         isat         ioff     adjpress
              3            0            0            0            0
*         dxin        volin        alpin         tin          pin
            1.0     0.019079          1.0        558.0      6.864E6
```

```
*         pain       concin          rbmx          poff          belv
            0.0          0.0           0.0           0.0           0.0
*
*
****************************************
* Starting Heat Structure Section of Model *
****************************************
*
*******    type           num       userid            component name
htstr                       4            0       powered-rod conductor
*        nzhstr          ittc         hscyl          ichf
              12             0             1             1
*     nofuelrod         plane        liqlev        iaxcnd
               0             3             0             0
*         nmwrx          nfci         nfcil          hdri          hdro
               0             0             0           0.0           0.0
*          nhot         nodes          fmon         nzmax      reflood
               0             4             0            30             0
*      dtxht(1)      dtxht(2)         dznht          hgapo
             0.0           0.0         100.0        4600.0
*
*  idbcin *           0             0             0             0s
*  idbcin *           0             0             0             0s
*  idbcin *           0             0             0             0e
*  idbcon *           2             2             2             2s
*  idbcon *           2             2             2             2s
*  idbcon *           2             2             2             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
* 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
* dhtstrz *        0.35          0.35          0.35          0.35s
* dhtstrz *        0.35          0.35          0.35          0.35s
* dhtstrz *        0.35          0.35          0.35          0.35e
*    rdx   *        49.0e
*   radrd  *         0.0    3.6347E-3     6.0579E-3     7.1501E-3e
```

```
*   matrd   *           1           1           2  e
*     nfax  *           1           1           1           1s
*     nfax  *           1           1           1           1s
*     nfax  *           1           1           1           1e
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0s
*     rftn  *       450.0       450.0       450.0       450.0e
*  fpuo2  *         0.0e
*    ftd  *         1.0e
*   gmix  * f       0.0e
*    gmles  *         0.0e
*    pgapt  *         0.0e
*    plvol  *           0.0  e
*    pslen  *           0.0  e
*    clenn  *           0.0  e
*   burn  *         0.0         0.0         0.0         0.0s
*   burn  *         0.0         0.0         0.0         0.0s
*   burn  *         0.0         0.0         0.0         0.0e
******************************************
* Finished Heat Structure Section of Model *
******************************************
*
*
*
*
****************************************
*      Starting Power Components          *
****************************************
*
*******   type            num         userid            component name
power                     901              1    power data input test1
*        numpwr         chanpow
             1               0
* htnum  *             4  e
*        irpwty          ndgx          ndhx          nrts          nhist
              5             0             0             5              0
*        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
              0             0             0             1              0
*         react         tneut        rpwoff        rrpwmx         rpwscl
            0.0           0.0           0.0           0.0            1.0
*        rpowri         zpwin        zpwoff        rzpwmx
      4.3102E6            0.0           0.0           0.0
*        extsou          pldr         pdrat        fucrac
```

```
          0.0          0.0          1.3          0.7
* rdpwr *          1.0          1.0          1.0          0.0e
* cpowr *          1.0e
* zpwtb1*          1.0s
* zpwtb1*          1.0          1.0          1.0          1.0          1.0
    s
* zpwtb1*          1.0          1.0          1.0          1.0          1.0
    s
* zpwtb1*          1.0          1.0e
****************************************
*     Finished Power Components        *
****************************************
*
*
*
end
*
*****************
* Timestep Data *
*****************
*        dtmin          dtmax           tend          rtwfp
        1.0E-5            1.0         1000.0           10.0
*        edint          gfint         dmpint         sedint
         100.0           10.0            1.0           50.0
*
*       endflag
           -1.0
```

B.2 TRACE Restart Input File

```
free format
*
*
*************
* main data *
*************
*
*       numtcr       ieos       inopt       nmat       id2o
            2          0           1          0          0
*
*
*
*****************
* namelist data *
*****************
*
 &inopts
  cpuflg=0,
  itdmr=1,
  nsolver=1,
  usesjc=3,
  npower=1,
  nhtstr=1,
  graphlevel="full",
  nosets=1
```

```
 &end
*
***************
* Model Flags *
***************
*
*         dstep          timet
             -1            0.0
*        stdyst         transi          ncomp           njun          ipak
              1              0              5              2             1
*          epso           epss
         1.0E-4         1.0E-4
*        oitmax         sitmax         isolut         ncontr         nccfl
           300            300              0              0             0
*          ntsv           ntcb           ntcf           ntrp          ntcp
              1              0              0              1             0
*
************************
* component-number data *
************************
*
*  Component input order (IORDER)
- type ---- num ------------ name ------------  + jun1 jun2 jun3
* FILL      *      1 s * velocity bc             +    1
* PIPE      *      2 s * subcooled liquid channel +    1    3
* BREAK     *      3 s * pressure bc             +    3
* HTSTR     *      4 s * powered-rod conductor   +
* POWER     *    901 e * power data input test1  +
*
*
************************************************
* Starting Signal Variable Section of Model     *
************************************************
*
*          idsv           isvn           ilcn          icn1          icn2
              1              0              0             0             0
************************************************
* Finished Signal Variable Section of Model     *
************************************************
*
*
*
*
*********************************
* Starting Trip Section of Model  *
*********************************
*
*          Trip Storage Count Card
*          ntse           ntct           ntsf          ntdp          ntsd
             0              0              0             0             0
*
*trip                       -2
*          idtp           isrt           iset          itst          idsg
            -2              2              0             1             1
*       setp(1)        setp(2)
           0.0            0.1
```

```
*       dtsp(1)         dtsp(2)
            0.0             0.0
*       ifsp(1)         ifsp(2)
              0               0
*
*********************************
* Finished Trips: Starting Extras*
*********************************
*
*********************************
* Finished Trip Section of Model *
*********************************
*
****************************************
*      Finished Power Components        *
****************************************
*
*
*
end
*
*****************
* Timestep Data *
*****************
*         dtmin           dtmax           tend           rtwfp
          1.0E-5            1.0         1000.0            10.0
*         edint           gfint         dmpint         sedint
            1.0         1.0E-5            1.0         1.0E-5
*
*       endflag
             -1.0
```

B.3 PARCS Restart Input File

```
!************************************************************
CASEID ss          Small Test Problem
!************************************************************
CNTL
      core_type  BWR
      core_power 100.00000  ! Initial core power level in %
      ppm        0.0        ! Initial boron concentration in ppm
!               bank1 (480-out,0-in)
      bank_pos   480.0
      th_fdbk    T
      xe_sm      0 0
!     xe_sm      0 0
!                0 : no xe, 1 : eq. xe   2 : tr. xe  3 : given xe
      decay_heat T
      ROT_ADF     T
!     {ROT_ADF is the assembly rotation option, T or F}
!     for LPRM
      pin_power  F
!     {Pin power reconstruction option, T or F
      ext_th     T    ../maptab_1ch      TRAC    1    1
      transient  F
```

```
      restart    F   ss_1.rst   0
!                    input  iteration      planar                adj
!                     edit      table       power      pin       reac
      print_opt         T          F           T         F         F
!                     fdbk       flux       planar
!                      rho     precurs        flux       Xe       T/H
      print_opt         F          F           F         F         T
!!                    oneD     Radial        assy
!!                   const      Shape       const
!     print_opt         F          F           F
!
! Jean LPRM
!                     oneD       PKRE      Radial   Radial      assy
!                    const       Data       Shape    Shape     const
      print_opt         F          F           F        T         F
      oned_kin   F   SA1D
!************************************************************
PARAM !table 6
      n_iters    5      500
      conv_ss    1.0e-6  1.e-5  5.e-4    0.001
!                keff,   epsl2, epslinf, tempf
      wielandt   0.04   0.1    1.0
      nodal_kern HYBRID
      nlupd_ss    200 200 1
!      nlupd_ss     3 3 1
      eps_anm    0.005
      eps_erf    0.005
      decusp     0
      init_guess 0
!************************************************************
XSEC  !tabl 7
      func_type  13 435
      dnp_ngrp   6
      EFIL_XS_R  'PBTT_xsec_rodded'
      EFIL_XS_UR 'PBTT_xsec_unrodded'
      kin_comp   1  1 -435
! pbtt specs
      dnp_beta   0.000167 0.001134 0.001022 0.002152 0.000837
         0.000214
      dnp_lambda 0.012813 0.031536 0.124703 0.328273 1.405280
         3.844728
!************************************************************
GEOM  !table 8
      file       ../geom
!************************************************************
TH    !table 9
!fix assume all assembiles are 7x7, no guide/instumentation tubes
      n_pingt    49 0                        !npin,ngt
!fix by Yunlin Xu: 3293/764=4.310209     3293*61.6459% = 2030MW
      FA_POWPIT  4.31021 15.24             !assembly power(Mw) and
         pitch(cm)
!
      pin_dim    5.0000 6.0000 0.900 6.0000 !pin radii, rs,rw,tw, and
          rgt in mm
!fix: 555-273=282, 9585/764=12.548
      flow_cond  282.0 12.548               !tin,cmfrfa(Kg/sec)
```

```
!
! pbtt spec
      CDC_DED    1 1
!     format:  CDC_DED   CDC_op     DED_op
!     CDC_op:  integer  for coolant density correction option:
!              =0 no correction(old models),
!              =1 correct with by pass density(PBTT)
!              =2 correct with  bypass, and water rod
!                 densities(ESBWR)
!     DED_op:  integer  for direct energy deposition:option:
!              =0 one  whole core factor for coolant(old models),
!              =1 two whole core factors for  coolant, bypass(PBTT)
!              =2 three whole core factors for  coolant,
!                 bypass, water rod ( can be used in ESBWR)
!              =3 three node-wise coolant factors,  bypass,
!                 water rod ( ESBWR)
      gamma_frac 0.00  0.000    !direct heating fraction
      CDC_DAT    0.761  0.401
      hgap       10000.         !hgap(w/M^2-C)
      n_ring     6              !number of meshes in pellet
      thmesh_x   1*1            !Number of T/H Nodes per FA in X-dir
      thmesh_y   1*1            !Number of T/H Nodes per FA in Y-dir
      thmesh_z   1 2 3 4 5 6 7 8 9 10 11 12 !junction locations
      freeze_tf  F    800.0
      freeze_dm  F    300.0
      write_fbv  T  ss3d.fbv
      read_dopl  F  ss3d.fbv
      read_tmdm  F  ss3d.fbv
!*************************************************************
!PFF  table 10, power form function
PFF
 npin_side 7
 pff_comp 1    1 -435
 pff_unrodd  1  !group 1 of set 1
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 pff_unrodd  2  !group 2 of set 1
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 pff_rodded  1  !group 1 of set 1
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 pff_rodded  2  !group 2 of set 1
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
 1. 1. 1. 1.
!*************************************************************
!PLOT
!    file sscoupl.plc
```

```
!     file trcoupl.plc
!*************************************************************
TRAN  !table 11
      time_step   5.0  0.00001
      expo_opt    F      F
      Scram       F  95.0  0.0  143.
      pin_freq    1                          !control lprm freq too
! The above values need checking
      theta       1.0    0.5    0.5
      conv_tr     0.001
      nlupd_tr    5  1  5  10
      eps_xsec    0.01
      rst_freq    1
.
```

Appendix C

Input Files for TRACE and PARCS Four-Channel Core Model

The input files for the core model are included in this appendix. Section C.1 shows the standalone input file which specifies the hydraulic and heat transfer component geometry and boundary conditions. Section C.2 shows the TRACE restart file for the core model. To change the eigenvalue linearization scheme for the implicit steady state solution, the *namelist* variable *ss_couple* must be added. Also, the *namelist* variable *n_couple* must be included to activate the implicit solution method. Section C.3 shows the PARCS restart file for the core model.

C.1 Standalone TRACE Input File

```
free format
*m: SNAP: Symbolic Nuclear Analysis Package,  Version 2.1.3, June 12,
    2012
*m: PLUGIN:TRACE Version 3.2.3
*m: CODE:TRACE V 5.0 Patch 3
*m: DATE:11/12/12
*
*
*************
* main data *
*************
*
*     numtcr        ieos       inopt        nmat        id2o
          1           0           1           1           0
      Single Channel
*
```

```
*
*****************
* namelist data *
*****************
*
 &inopts
  dtstrt=1.0E-4,
  icflow=2,
  ikfac=1,
  ipowr=1,
  itdmr=0,
  nosets=1,
  nsolver=1,
  usesjc=3,
  npower=1
 &end
*
***************
* Model Flags *
***************
*
*      dstep       timet
           0         0.0
*     stdyst      transi      ncomp        njun        ipak
           1           0         22          24           1
*       epso        epss
      1.0E-4      1.0E-4
*     oitmax      sitmax     isolut      ncontr       nccfl
          100          10          0           0           0
*       ntsv        ntcb       ntcf        ntrp        ntcp
           7           7          0           0           0
*
************************
* component-number data *
************************
*
*  Component input order (IORDER)
*-- type ---- num ---------- name ------- +     jun1    jun2    jun3
* CHAN      *       1 s * 8x8                +       10    2101
* CHAN      *       2 s * 8x8                +     2151    2161
* CHAN      *       3 s * 8x8                +     2131    2141
* CHAN      *       4 s * 8x8                +     2111    2121
* FILL      *       5 s * fill               +        4
* FILL      *       6 s * fill               +     2281
* FILL      *       7 s * fill               +     2271
* FILL      *       8 s * fill               +     2261
* BREAK     *       9 s * Break              +        5
* BREAK     *      10 s * Break              +     2201
* BREAK     *      11 s * Break              +     2211
* BREAK     *      12 s * Break              +     2221
* PIPE      *      13 s * outlet pipe        +        7       5
* PIPE      *      14 s * outlet pipe        +     2171    2201
* PIPE      *      15 s * outlet pipe        +     2181    2211
* PIPE      *      16 s * outlet pipe        +     2191    2221
* PIPE      *      17 s * feed water pipe    +     2261    2231
* PIPE      *      18 s * feed water pipe    +     2271    2241
```

```
* PIPE     *     19 s * feed water pipe      +       4      6
* PIPE     *     20 s * feed water pipe      +    2281   2251
* VESSEL   *     21 s * Reactor Vessel       +
* POWER    *     22 e * power com            +
*
**********************
* material properties *
**********************
*
*   matb*          50 e
*  ptbln*           1 e
*  User Defined Material : 50
*
*
*  prptb        temp        rho         cp       cond        emis
*  prptb*      200.0     7820.0      530.0    1.0E-20         0.0 e
*
*
*
************************************************
* Starting Signal Variable Section of Model     *
************************************************
*
*         idsv        isvn         ilcn         icn1         icn2
              1           0            0            0            0
*
*        idsv         isvn         ilcn         icn1         icn2
              2          69           19            3            0
*
*        idsv         isvn         ilcn         icn1         icn2
              3         105           19            3            0
*
*        idsv         isvn         ilcn         icn1         icn2
              4          69           13            3            0
*
*        idsv         isvn         ilcn         icn1         icn2
              5         105           13            3            0
*
*        idsv         isvn         ilcn         icn1         icn2
              6          69            1           27            0
*
*        idsv         isvn         ilcn         icn1         icn2
              7         105            1           27            0
************************************************
* Finished Signal Variable Section of Model     *
************************************************
*
*
*
*
*****************************************
* Starting Control System Section of Model  *
*****************************************
*
****** Control Blocks ******
*
```

```
*       idcb        icbn        icb1        icb2        icb3
        -601          39           2           3           0
*     cbgain      cbxmin      cbmax      cbcon1      cbcon2
          1.0    -1.0E20     1.0E20         0.0         0.0
*
*
*       idcb        icbn        icb1        icb2        icb3
        -602          39           4           5           0
*     cbgain      cbxmin      cbmax      cbcon1      cbcon2
          1.0    -1.0E20     1.0E20         0.0         0.0
*
*
*       idcb        icbn        icb1        icb2        icb3
        -603          39           6           3           0
*     cbgain      cbxmin      cbmax      cbcon1      cbcon2
          1.0    -1.0E20     1.0E20         0.0         0.0
*
*
*       idcb        icbn        icb1        icb2        icb3
        -604          39           6           7           0
*     cbgain      cbxmin      cbmax      cbcon1      cbcon2
          1.0    -1.0E20     1.0E20         0.0         0.0
*
*
*       idcb        icbn        icb1        icb2        icb3
        -605          54        -602        -601           0
*     cbgain      cbxmin      cbmax      cbcon1      cbcon2
          1.0    -1.0E20     1.0E20         0.0         0.0
*
*
*       idcb        icbn        icb1        icb2        icb3
        -606          54        -604        -603           0
*     cbgain      cbxmin      cbmax      cbcon1      cbcon2
          1.0    -1.0E20     1.0E20          0.         0.0
*
*
*       idcb        icbn        icb1        icb2        icb3
        -607          54        -605        -606           0
*     cbgain      cbxmin      cbmax      cbcon1      cbcon2
          1.0    -1.0E20     1.0E20         0.0         1.0
*
*****************************************
* Finished Control System Section of Model *
*****************************************
*
*
*
*
******* type          num       userid          component name
chan                    1            0                   8x8
*       ncell       nodes         jun1        jun2        epsw
           27           2           10        2101      3.0E-4
*      nsides
            0
*        ichf       iconc      iaxcnd      liqlev       nhcom
            1           0           0           0          21
```

```
*       width          th       houtl       houtv       toutl
        0.5456         0.2         0.0         0.0         0.0
*       toutv      advbwrf     quadsym    numwrods     nvfrays
          0.0
*        ngrp       nchans      nodesr        nrow        ncrz
            1            1           9           8          25
*        icrnk        icrlh       nmwrx        nfci       nfcil
            1            1           0           0           0
*        fmon      reflood       nzmax      nzmaxw       ibeam
            0            0          26          28           1
*        dznht       dznhtw      dtxht1      dtxht2
          0.1          0.1         3.0        10.0
*        hgapo        pdrat        pldr      fucrac       norad
       8515.5       1.2898         0.0         0.0           0
*       emcif1       emcif2      emcif3       noani
          0.67          0.0         0.0           0
*       emcof1       emcof2      emcof3
          0.67          0.0         0.0
* dx     *          0.304       0.1472      0.1472      0.1472s
* dx     *         0.1472       0.1472      0.1472      0.1472s
* dx     *         0.1472       0.1472      0.1472      0.1472s
* dx     *         0.1472       0.1472      0.1472      0.1472s
* dx     *         0.1472       0.1472      0.1472      0.1472s
* dx     *         0.1472       0.1472      0.1472      0.1472s
* dx     *         0.1472       0.1472       0.414e
* vol    *      3.002E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3    1.4536E-3   4.0883E-3e
* fa     *    9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3     9.875E-3    9.875E-3    9.875E-3e
* kfac   *         0.0          0.0         0.0         0.0s
* kfac   *         0.0          0.0         0.0         0.0s
* kfac   *         0.0          0.0         0.0         0.0s
* kfac   *         0.0          0.0         0.0         0.0s
* kfac   *         0.0          0.0         0.0         0.0s
* kfac   *         0.0          0.0         0.0         0.0s
* kfac   *         0.0          0.0         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         1.0e
* hd     *      0.0118       0.0118      0.0118      0.0118s
* hd     *      0.0118       0.0118      0.0118      0.0118s
* hd     *      0.0118       0.0118      0.0118      0.0118s
```

```
* hd     *        0.0118        0.0118        0.0118        0.0118s
* hd     *        0.0118        0.0118        0.0118        0.0118s
* hd     *        0.0118        0.0118        0.0118        0.0118s
* hd     *        0.0118        0.0118        0.0118        0.0118e
* icflg  *             0             0             0             0s
* icflg  *             0             0             0             0s
* icflg  *             0             0             0             0s
* icflg  *             0             0             0             0s
* icflg  *             0             0             0             0s
* icflg  *             0             0             0             0s
* icflg  *             0             0             0             0e
* nff    *          -100          -100          -100          -100s
* nff    *          -100          -100          -100          -100s
* nff    *          -100          -100          -100          -100s
* nff    *          -100          -100          -100          -100s
* nff    *          -100          -100          -100          -100s
* nff    *          -100          -100          -100          -100s
* nff    *          -100          -100          -100          -100e
* 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           0.0e
* vl     *           5.0           5.0           5.0           5.0s
* vl     *           5.0           5.0           5.0           5.0s
* vl     *           5.0           5.0           5.0           5.0s
* vl     *           5.0           5.0           5.0           5.0s
* vl     *           5.0           5.0           5.0           5.0s
* vl     *           5.0           5.0           5.0           5.0s
* vl     *           5.0           5.0           5.0           5.0e
* vv     *           5.0           5.0           5.0           5.0s
* vv     *           5.0           5.0           5.0           5.0s
* vv     *           5.0           5.0           5.0           5.0s
* vv     *           5.0           5.0           5.0           5.0s
* vv     *           5.0           5.0           5.0           5.0s
* vv     *           5.0           5.0           5.0           5.0s
* vv     *           5.0           5.0           5.0           5.0e
* tl     *         300.0         301.0         562.0         562.0s
* tl     *         562.0         562.0         562.0         562.0s
* tl     *         562.0         562.0         562.0         562.0s
* tl     *         562.0         562.0         562.0         562.0s
* tl     *         562.0         562.0         562.0         562.0s
* tl     *         562.0         562.0         562.0         562.0s
* tl     *         562.0         562.0         562.0e
* tv     *         559.6         559.6         562.0         562.0s
* tv     *         562.0         562.0         562.0         562.0s
* tv     *         562.0         562.0         562.0         562.0s
* tv     *         562.0         562.0         562.0         562.0s
* tv     *         562.0         562.0         562.0         562.0s
* tv     *         562.0         562.0         562.0         562.0s
* tv     *         562.0         562.0         562.0e
* p      *         1.5E7         1.5E7         1.5E7         1.5E7s
* p      *         1.5E7         1.5E7         1.5E7         1.5E7s
* p      *         1.5E7         1.5E7         1.5E7         1.5E7s
```

```
* p      *         1.5E7        1.5E7        1.5E7        1.5E7s
* p      *         1.5E7        1.5E7        1.5E7        1.5E7s
* p      *         1.5E7        1.5E7        1.5E7        1.5E7s
* p      *         1.5E7        1.5E7        1.5E7e
* 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          0.0e
*  qppp  * f 0.0e
* mat    *            50 e
* tw     *      300.01       301.47       301.04       301.47       302.51s
* tw     *      301.48       304.41       301.48       306.74       301.48s
* tw     *      309.46       301.49       312.55       301.49       315.96s
* tw     *       301.5       319.66       301.51        323.6       301.51s
* tw     *      327.73       301.52        332.0       301.53       336.35s
* tw     *      301.54       340.73       301.55       345.08       301.55s
* tw     *      349.36       301.56       353.49       301.57       357.44s
* tw     *      301.58       361.15       301.59       364.58       301.59s
* tw     *      367.68        301.6       370.41        301.6       372.75s
* tw     *      301.61       374.67       301.61       376.14       301.61s
* tw     *      377.18       301.61       377.18       301.61e
* idrod  *           1e
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2e
*  rdx   *        64.0e
* radrd  *         0.0    1.044E-3    2.088E-3    3.132E-3    4.176E-3s
* radrd  *     5.22E-3    5.325E-3    5.725E-3    6.125E-3e
* matrd  *           1            1            1            1 s
* matrd  *           1            3            2            2 e
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0e
*  rftn  *      317.37       317.12       316.36        315.1s
*  rftn  *      313.35        311.1       308.87        308.3s
*  rftn  *      307.77       334.23       333.73       332.22s
*  rftn  *      329.73       326.25       321.82       317.51s
*  rftn  *      316.43       315.41        352.0       351.23s
*  rftn  *      348.92       345.09       339.77       333.01s
*  rftn  *      326.59       324.99        323.5       369.26s
*  rftn  *      368.22       365.09       359.91       352.74s
*  rftn  *      343.67       335.23       333.15        331.2s
*  rftn  *      385.83       384.51       380.57       374.07s
*  rftn  *      365.08       353.74       343.41       340.88s
*  rftn  *      338.52       401.47       399.88       395.16s
*  rftn  *      387.37       376.64       363.14       351.07s
```

```
*   rftn *       348.14        345.4       415.95       414.12s
*   rftn *       408.66       399.67        387.3       371.78s
*   rftn *       358.16       354.87       351.79       429.06s
*   rftn *       427.01       420.88       410.79       396.95s
*   rftn *       379.62       364.63       361.03       357.67s
*   rftn *       440.66       438.41       431.69       420.65s
*   rftn *       405.53       386.63       370.49       366.64s
*   rftn *       363.04       450.61       448.19       440.99s
*   rftn *       429.17       412.98       392.78       375.74s
*   rftn *       371.69       367.91       458.79       456.25s
*   rftn *       448.68       436.25       419.26       398.08s
*   rftn *       380.38       376.19       372.28       465.14s
*   rftn *       462.51        454.7       441.87       424.34s
*   rftn *       402.52       384.41       380.15       376.17s
*   rftn *       469.56        466.9       458.97       445.97s
*   rftn *        428.2       406.07       387.84       383.56s
*   rftn *       379.56       472.03       469.37       461.48s
*   rftn *       448.52       430.81       408.75       390.65s
*   rftn *       386.41       382.45       472.56       469.96s
*   rftn *       462.23       449.54       432.18       410.56s
*   rftn *       392.85       388.72       384.85       471.18s
*   rftn *       468.68       461.25       449.04       432.34s
*   rftn *        411.5       394.46       390.48       386.76s
*   rftn *       467.96       465.61        458.6       447.09s
*   rftn *       431.31       411.59       395.46        391.7s
*   rftn *       388.18       462.99       460.82       454.35s
*   rftn *       443.72       429.13       410.85       395.86s
*   rftn *       392.37       389.11       456.38       454.42s
*   rftn *        448.6       439.01       425.83       409.28s
*   rftn *       395.65       392.48       389.52       448.24s
*   rftn *       446.53       441.44       433.04       421.47s
*   rftn *        406.9       394.83       392.02        389.4s
*   rftn *       438.74       437.31       433.01       425.91s
*   rftn *        416.1       403.72        393.4       390.99s
*   rftn *       388.74       428.06        426.9       423.45s
*   rftn *       417.73       409.81       399.79       391.35s
*   rftn *       389.38       387.54       416.37       415.51s
*   rftn *       412.93       408.64        402.7       395.14s
*   rftn *       388.72       387.22       385.82       403.89s
*   rftn *       403.32       401.62       398.79       394.86s
*   rftn *       389.84       385.53       384.52       383.57s
*   rftn *       391.44       391.16       390.29       388.85s
*   rftn *       386.84       384.27       382.03       381.51s
*   rftn *       381.02e
* rdpwr *          1.0          1.0          1.0          1.0         1.0s
* rdpwr *          1.0          0.0          0.0          0.0e
* cpowr *    0.015625e
* radpw *      0.18568      0.35797      0.53312      0.70045     0.85753s
* radpw *        1.002       1.1319       1.2451       1.3401      1.4155s
* radpw *       1.4701       1.5033       1.5144       1.5033      1.4702s
* radpw *       1.4155       1.3401       1.2451       1.1319       1.002s
* radpw *      0.85753      0.70046      0.53312      0.35798     0.18568e
* fpuo2 *          0.0e
*   ftd *         0.95e
*  gmix * f        0.0e
* pgapt *       7.0E6e
```

```
*  burn *    1.9647E4    1.9647E4    1.9647E4    1.9647E4s
*  burn *    1.9647E4    1.9647E4    1.9647E4    1.9647E4s
*  burn *    1.9647E4    1.9647E4    1.9647E4    1.9647E4s
*  burn *    1.9647E4    1.9647E4    1.9647E4    1.9647E4s
*  burn *    1.9647E4    1.9647E4    1.9647E4    1.9647E4s
*  burn *    1.9647E4    1.9647E4    1.9647E4    1.9647E4s
*  burn *    1.9647E4e
*viewgrp *     0.83459     0.16541      0.7467      0.2533e
*beamlen *   5.0662E-3    0.012984    0.012984    0.024143e
*
*
******* type          num       userid          component name
chan                    2            0                   8x8
*       ncell       nodes         jun1         jun2         epsw
           27           2         2151         2161      3.0E-4
*      nsides
            0
*        ichf       iconc       iaxcnd       liqlev       nhcom
            1           0            0            0          21
*       width          th       houtl       houtv       toutl
       0.5456         0.2          0.0          0.0         0.0
*       toutv      advbwrf     quadsym     numwrods     nvfrays
          0.0
*        ngrp       nchans      nodesr         nrow         ncrz
            1           1            9            8          25
*       icrnk       icrlh       nmwrx         nfci        nfcil
            1           1            0            0            0
*        fmon      reflood       nzmax       nzmaxw       ibeam
            0           0           26           28           1
*       dznht      dznhtw      dtxht1      dtxht2
          0.1         0.1         3.0         10.0
*       hgapo       pdrat        pldr       fucrac       norad
      8515.5      1.2898         0.0          0.0           0
*      emcif1      emcif2      emcif3        noani
          0.67        0.0         0.0           0
*      emcof1      emcof2      emcof3
          0.67        0.0         0.0
* dx     *         0.304      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472       0.414e
* vol    *     3.002E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *    1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *    1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *    1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *    1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *    1.4536E-3    1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *    1.4536E-3    1.4536E-3   4.0883E-3e
* fa     *     9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *     9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *     9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *     9.875E-3     9.875E-3    9.875E-3    9.875E-3s
* fa     *     9.875E-3     9.875E-3    9.875E-3    9.875E-3s
```

* fa	*	9.875E–3	9.875E–3	9.875E–3	9.875E–3s
* fa	*	9.875E–3	9.875E–3	9.875E–3	9.875E–3e
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	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	1.0e
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118e
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0e
* nff	*	−100	−100	−100	−100s
* nff	*	−100	−100	−100	−100s
* nff	*	−100	−100	−100	−100s
* nff	*	−100	−100	−100	−100s
* nff	*	−100	−100	−100	−100s
* nff	*	−100	−100	−100	−100s
* nff	*	−100	−100	−100	−100e
* 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	0.0e	
* vl	*	5.0	5.0	5.0	5.0s
* vl	*	5.0	5.0	5.0	5.0s
* vl	*	5.0	5.0	5.0	5.0s
* vl	*	5.0	5.0	5.0	5.0s
* vl	*	5.0	5.0	5.0	5.0s
* vl	*	5.0	5.0	5.0	5.0s
* vl	*	5.0	5.0	5.0	5.0e
* vv	*	5.0	5.0	5.0	5.0s
* vv	*	5.0	5.0	5.0	5.0s
* vv	*	5.0	5.0	5.0	5.0s
* vv	*	5.0	5.0	5.0	5.0s
* vv	*	5.0	5.0	5.0	5.0s

```
* vv     *          5.0          5.0          5.0          5.0s
* vv     *          5.0          5.0          5.0          5.0e
* tl     *        300.0        301.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0e
* tv     *        559.6        559.6        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0e
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7e
* 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          0.0e
*  qppp  * f 0.0e
* mat    *         50 e
* tw     *      300.01       301.47       301.04       301.47       302.51s
* tw     *      301.48       304.41       301.48       306.74       301.48s
* tw     *      309.46       301.49       312.55       301.49       315.96s
* tw     *       301.5       319.66       301.51        323.6       301.51s
* tw     *      327.73       301.52        332.0       301.53       336.35s
* tw     *      301.54       340.73       301.55       345.08       301.55s
* tw     *      349.36       301.56       353.49       301.57       357.44s
* tw     *      301.58       361.15       301.59       364.58       301.59s
* tw     *      367.68        301.6       370.41        301.6       372.75s
* tw     *      301.61       374.67       301.61       376.14       301.61s
* tw     *      377.18       301.61       377.18       301.61e
* idrod  *          1e
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2e
*  rdx   *        64.0e
* radrd  *         0.0    1.044E-3    2.088E-3    3.132E-3    4.176E-3s
* radrd  *     5.22E-3    5.325E-3    5.725E-3    6.125E-3e
* matrd  *           1            1            1            1 s
* matrd  *           1            3            2            2 e
*  nfax  *           0            0            0           0s
```

*	nfax	*	0	0	0	0s
*	nfax	*	0	0	0	0s
*	nfax	*	0	0	0	0s
*	nfax	*	0	0	0	0s
*	nfax	*	0	0	0	0s
*	nfax	*	0e			
*	rftn	*	317.37	317.12	316.36	315.1s
*	rftn	*	313.35	311.1	308.87	308.3s
*	rftn	*	307.77	334.23	333.73	332.22s
*	rftn	*	329.73	326.25	321.82	317.51s
*	rftn	*	316.43	315.41	352.0	351.23s
*	rftn	*	348.92	345.09	339.77	333.01s
*	rftn	*	326.59	324.99	323.5	369.26s
*	rftn	*	368.22	365.09	359.91	352.74s
*	rftn	*	343.67	335.23	333.15	331.2s
*	rftn	*	385.83	384.51	380.57	374.07s
*	rftn	*	365.08	353.74	343.41	340.88s
*	rftn	*	338.52	401.47	399.88	395.16s
*	rftn	*	387.37	376.64	363.14	351.07s
*	rftn	*	348.14	345.4	415.95	414.12s
*	rftn	*	408.66	399.67	387.3	371.78s
*	rftn	*	358.16	354.87	351.79	429.06s
*	rftn	*	427.01	420.88	410.79	396.95s
*	rftn	*	379.62	364.63	361.03	357.67s
*	rftn	*	440.66	438.41	431.69	420.65s
*	rftn	*	405.53	386.63	370.49	366.64s
*	rftn	*	363.04	450.61	448.19	440.99s
*	rftn	*	429.17	412.98	392.78	375.74s
*	rftn	*	371.69	367.91	458.79	456.25s
*	rftn	*	448.68	436.25	419.26	398.08s
*	rftn	*	380.38	376.19	372.28	465.14s
*	rftn	*	462.51	454.7	441.87	424.34s
*	rftn	*	402.52	384.41	380.15	376.17s
*	rftn	*	469.56	466.9	458.97	445.97s
*	rftn	*	428.2	406.07	387.84	383.56s
*	rftn	*	379.56	472.03	469.37	461.48s
*	rftn	*	448.52	430.81	408.75	390.65s
*	rftn	*	386.41	382.45	472.56	469.96s
*	rftn	*	462.23	449.54	432.18	410.56s
*	rftn	*	392.85	388.72	384.85	471.18s
*	rftn	*	468.68	461.25	449.04	432.34s
*	rftn	*	411.5	394.46	390.48	386.76s
*	rftn	*	467.96	465.61	458.6	447.09s
*	rftn	*	431.31	411.59	395.46	391.7s
*	rftn	*	388.18	462.99	460.82	454.35s
*	rftn	*	443.72	429.13	410.85	395.86s
*	rftn	*	392.37	389.11	456.38	454.42s
*	rftn	*	448.6	439.01	425.83	409.28s
*	rftn	*	395.65	392.48	389.52	448.24s
*	rftn	*	446.53	441.44	433.04	421.47s
*	rftn	*	406.9	394.83	392.02	389.4s
*	rftn	*	438.74	437.31	433.01	425.91s
*	rftn	*	416.1	403.72	393.4	390.99s
*	rftn	*	388.74	428.06	426.9	423.45s
*	rftn	*	417.73	409.81	399.79	391.35s
*	rftn	*	389.38	387.54	416.37	415.51s

```
*   rftn *       412.93       408.64        402.7       395.14s
*   rftn *       388.72       387.22       385.82       403.89s
*   rftn *       403.32       401.62       398.79       394.86s
*   rftn *       389.84       385.53       384.52       383.57s
*   rftn *       391.44       391.16       390.29       388.85s
*   rftn *       386.84       384.27       382.03       381.51s
*   rftn *       381.02e
* rdpwr *          1.0          1.0          1.0          1.0          1.0s
* rdpwr *          1.0          0.0          0.0          0.0e
* cpowr *    0.015625e
* radpw *      0.18568      0.35797      0.53312      0.70045      0.85753s
* radpw *        1.002       1.1319       1.2451       1.3401       1.4155s
* radpw *       1.4701       1.5033       1.5144       1.5033       1.4702s
* radpw *       1.4155       1.3401       1.2451       1.1319        1.002s
* radpw *      0.85753      0.70046      0.53312      0.35798      0.18568e
* fpuo2 *          0.0e
*    ftd *         0.95e
*  gmix * f        0.0e
* pgapt *         7.0E6e
*   burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*   burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*   burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*   burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*   burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*   burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*   burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*   burn *     1.9647E4e
*viewgrp *      0.83459      0.16541       0.7467       0.2533e
*beamlen *    5.0662E-3     0.012984     0.012984     0.024143e
*
*
******* type          num       userid          component name
chan                  3            0                       8x8
*       ncell       nodes         jun1         jun2         epsw
           27           2         2131         2141      3.0E-4
*      nsides
            0
*        ichf       iconc       iaxcnd       liqlev        nhcom
            1           0            0            0           21
*       width          th       houtl        houtv       toutl
       0.5456         0.2         0.0          0.0         0.0
*       toutv      advbwrf     quadsym     numwrods     nvfrays
         0.0
*        ngrp       nchans      nodesr         nrow         ncrz
            1           1            9            8           25
*       icrnk        icrlh       nmwrx         nfci        nfcil
            1           1            0            0            0
*        fmon      reflood       nzmax        nzmaxw        ibeam
            0           0           26           28            1
*       dznht       dznhtw      dtxht1       dtxht2
         0.1          0.1         3.0         10.0
*       hgapo        pdra         pldr       fucrac        norad
      8515.5       1.2898          0.0          0.0            0
*      emcif1       emcif2      emcif3        noani
         0.67          0.0         0.0            0
*      emcof1       emcof2      emcof3
         0.67          0.0         0.0
```

* dx	*	0.304	0.1472	0.1472	0.1472s
* dx	*	0.1472	0.1472	0.1472	0.1472s
* dx	*	0.1472	0.1472	0.1472	0.1472s
* dx	*	0.1472	0.1472	0.1472	0.1472s
* dx	*	0.1472	0.1472	0.1472	0.1472s
* dx	*	0.1472	0.1472	0.1472	0.1472s
* dx	*	0.1472	0.1472	0.414e	
* vol	*	3.002E-3	1.4536E-3	1.4536E-3	1.4536E-3s
* vol	*	1.4536E-3	1.4536E-3	1.4536E-3	1.4536E-3s
* vol	*	1.4536E-3	1.4536E-3	1.4536E-3	1.4536E-3s
* vol	*	1.4536E-3	1.4536E-3	1.4536E-3	1.4536E-3s
* vol	*	1.4536E-3	1.4536E-3	1.4536E-3	1.4536E-3s
* vol	*	1.4536E-3	1.4536E-3	1.4536E-3	1.4536E-3s
* vol	*	1.4536E-3	1.4536E-3	4.0883E-3e	
* fa	*	9.875E-3	9.875E-3	9.875E-3	9.875E-3s
* fa	*	9.875E-3	9.875E-3	9.875E-3	9.875E-3s
* fa	*	9.875E-3	9.875E-3	9.875E-3	9.875E-3s
* fa	*	9.875E-3	9.875E-3	9.875E-3	9.875E-3s
* fa	*	9.875E-3	9.875E-3	9.875E-3	9.875E-3s
* fa	*	9.875E-3	9.875E-3	9.875E-3	9.875E-3s
* fa	*	9.875E-3	9.875E-3	9.875E-3	9.875E-3e
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	0.0	0.0s
* kfac	*	0.0	0.0	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	1.0e
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118s
* hd	*	0.0118	0.0118	0.0118	0.0118e
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0s
* icflg	*	0	0	0	0e
* nff	*	-100	-100	-100	-100s
* nff	*	-100	-100	-100	-100s
* nff	*	-100	-100	-100	-100s
* nff	*	-100	-100	-100	-100s
* nff	*	-100	-100	-100	-100s
* nff	*	-100	-100	-100	-100s
* nff	*	-100	-100	-100	-100e

```
* 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          0.0e
* vl     *          5.0          5.0          5.0          5.0s
* vl     *          5.0          5.0          5.0          5.0s
* vl     *          5.0          5.0          5.0          5.0s
* vl     *          5.0          5.0          5.0          5.0s
* vl     *          5.0          5.0          5.0          5.0s
* vl     *          5.0          5.0          5.0          5.0s
* vl     *          5.0          5.0          5.0          5.0e
* vv     *          5.0          5.0          5.0          5.0s
* vv     *          5.0          5.0          5.0          5.0s
* vv     *          5.0          5.0          5.0          5.0s
* vv     *          5.0          5.0          5.0          5.0s
* vv     *          5.0          5.0          5.0          5.0s
* vv     *          5.0          5.0          5.0          5.0s
* vv     *          5.0          5.0          5.0          5.0e
* tl     *        300.0        301.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0        562.0        562.0e
* tv     *        559.6        559.6        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0        562.0        562.0e
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7        1.5E7        1.5E7e
* 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          0.0e
*  qppp  * f  0.0e
* mat    *           50  e
* tw     *      300.01       301.47       301.04       301.47       302.51s
* tw     *      301.48       304.41       301.48       306.74       301.48s
* tw     *      309.46       301.49       312.55       301.49       315.96s
* tw     *       301.5       319.66       301.51        323.6       301.51s
* tw     *      327.73       301.52        332.0       301.53       336.35s
```

```
* tw     *     301.54      340.73      301.55      345.08      301.55s
* tw     *     349.36      301.56      353.49      301.57      357.44s
* tw     *     301.58      361.15      301.59      364.58      301.59s
* tw     *     367.68       301.6      370.41       301.6      372.75s
* tw     *     301.61      374.67      301.61      376.14      301.61s
* tw     *     377.18      301.61      377.18      301.61e
* idrod  *          1e
* nhcel  *          2           2           2           2           2s
* nhcel  *          2           2           2           2           2s
* nhcel  *          2           2           2           2           2s
* nhcel  *          2           2           2           2           2s
* nhcel  *          2           2           2           2           2s
* nhcel  *          2           2e
*  rdx   *       64.0e
* radrd  *        0.0    1.044E-3    2.088E-3    3.132E-3    4.176E-3s
* radrd  *    5.22E-3    5.325E-3    5.725E-3    6.125E-3e
* matrd  *          1           1           1           1 s
* matrd  *          1           3           2           2 e
*  nfax  *          0           0           0           0s
*  nfax  *          0           0           0           0s
*  nfax  *          0           0           0           0s
*  nfax  *          0           0           0           0s
*  nfax  *          0           0           0           0s
*  nfax  *          0           0           0           0s
*  nfax  *          0e
*  rftn  *      317.37      317.12      316.36       315.1s
*  rftn  *      313.35       311.1      308.87       308.3s
*  rftn  *      307.77      334.23      333.73      332.22s
*  rftn  *      329.73      326.25      321.82      317.51s
*  rftn  *      316.43      315.41       352.0      351.23s
*  rftn  *      348.92      345.09      339.77      333.01s
*  rftn  *      326.59      324.99       323.5      369.26s
*  rftn  *      368.22      365.09      359.91      352.74s
*  rftn  *      343.67      335.23      333.15       331.2s
*  rftn  *      385.83      384.51      380.57      374.07s
*  rftn  *      365.08      353.74      343.41      340.88s
*  rftn  *      338.52      401.47      399.88      395.16s
*  rftn  *      387.37      376.64      363.14      351.07s
*  rftn  *      348.14       345.4      415.95      414.12s
*  rftn  *      408.66      399.67       387.3      371.78s
*  rftn  *      358.16      354.87      351.79      429.06s
*  rftn  *      427.01      420.88      410.79      396.95s
*  rftn  *      379.62      364.63      361.03      357.67s
*  rftn  *      440.66      438.41      431.69      420.65s
*  rftn  *      405.53      386.63      370.49      366.64s
*  rftn  *      363.04      450.61      448.19      440.99s
*  rftn  *      429.17      412.98      392.78      375.74s
*  rftn  *      371.69      367.91      458.79      456.25s
*  rftn  *      448.68      436.25      419.26      398.08s
*  rftn  *      380.38      376.19      372.28      465.14s
*  rftn  *      462.51       454.7      441.87      424.34s
*  rftn  *      402.52      384.41      380.15      376.17s
*  rftn  *      469.56       466.9      458.97      445.97s
*  rftn  *       428.2      406.07      387.84      383.56s
*  rftn  *      379.56      472.03      469.37      461.48s
*  rftn  *      448.52      430.81      408.75      390.65s
```

```
*   rftn *       386.41       382.45       472.56       469.96s
*   rftn *       462.23       449.54       432.18       410.56s
*   rftn *       392.85       388.72       384.85       471.18s
*   rftn *       468.68       461.25       449.04       432.34s
*   rftn *        411.5       394.46       390.48       386.76s
*   rftn *       467.96       465.61        458.6       447.09s
*   rftn *       431.31       411.59       395.46        391.7s
*   rftn *       388.18       462.99       460.82       454.35s
*   rftn *       443.72       429.13       410.85       395.86s
*   rftn *       392.37       389.11       456.38       454.42s
*   rftn *        448.6       439.01       425.83       409.28s
*   rftn *       395.65       392.48       389.52       448.24s
*   rftn *       446.53       441.44       433.04       421.47s
*   rftn *        406.9       394.83       392.02        389.4s
*   rftn *       438.74       437.31       433.01       425.91s
*   rftn *        416.1       403.72        393.4       390.99s
*   rftn *       388.74       428.06        426.9       423.45s
*   rftn *       417.73       409.81       399.79       391.35s
*   rftn *       389.38       387.54       416.37       415.51s
*   rftn *       412.93       408.64        402.7       395.14s
*   rftn *       388.72       387.22       385.82       403.89s
*   rftn *       403.32       401.62       398.79       394.86s
*   rftn *       389.84       385.53       384.52       383.57s
*   rftn *       391.44       391.16       390.29       388.85s
*   rftn *       386.84       384.27       382.03       381.51s
*   rftn *       381.02e
* rdpwr *          1.0          1.0          1.0          1.0          1.0s
* rdpwr *          1.0          0.0          0.0          0.0e
* cpowr *     0.015625e
* radpw *      0.18568      0.35797      0.53312      0.70045      0.85753s
* radpw *        1.002       1.1319       1.2451       1.3401       1.4155s
* radpw *       1.4701       1.5033       1.5144       1.5033       1.4702s
* radpw *       1.4155       1.3401       1.2451       1.1319        1.002s
* radpw *      0.85753      0.70046      0.53312      0.35798      0.18568e
* fpuo2 *          0.0e
*   ftd *         0.95e
*  gmix * f        0.0e
* pgapt *        7.0E6e
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4
*viewgrp *      0.83459      0.16541       0.7467       0.2533e
*beamlen *    5.0662E-3     0.012984     0.012984     0.024143e
*
*
******* type          num       userid          component name
chan                    4            0                   8x8
*       ncell       nodes         jun1         jun2         epsw
           27           2         2111         2121       3.0E-4
*      nsides
            0
*        ichf       iconc       iaxcnd       liqlev        nhcom
```

```
            1           0           0           0          21
*       width          th       houtl       houtv       toutl
       0.5456         0.2         0.0         0.0         0.0
*       toutv      advbwrf     quadsym    numwrods     nvfrays
          0.0
*        ngrp       nchans      nodesr        nrow        ncrz
            1           1           9           8          25
*       icrnk       icrlh       nmwrx        nfci       nfcil
            1           1           0           0           0
*        fmon      reflood       nzmax      nzmaxw       ibeam
            0           0          26          28           1
*       dznht       dznhtw      dtxht1      dtxht2
          0.1         0.1         3.0        10.0
*       hgapo       pdrat       pldr      fucrac       norad
       8515.5      1.2898         0.0         0.0           0
*       emcif1      emcif2      emcif3       noani
          0.67         0.0         0.0           0
*       emcof1      emcof2      emcof3
          0.67         0.0         0.0
* dx     *         0.304      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472      0.1472      0.1472s
* dx     *        0.1472      0.1472       0.414e
* vol    *     3.002E-3   1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3   1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3   1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3   1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3   1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3   1.4536E-3   1.4536E-3   1.4536E-3s
* vol    *   1.4536E-3   1.4536E-3   4.0883E-3e
* fa     *    9.875E-3    9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3    9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3    9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3    9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3    9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3    9.875E-3    9.875E-3    9.875E-3s
* fa     *    9.875E-3    9.875E-3    9.875E-3    9.875E-3e
* kfac   *         0.0         0.0         0.0         0.0s
* kfac   *         0.0         0.0         0.0         0.0s
* kfac   *         0.0         0.0         0.0         0.0s
* kfac   *         0.0         0.0         0.0         0.0s
* kfac   *         0.0         0.0         0.0         0.0s
* kfac   *         0.0         0.0         0.0         0.0s
* kfac   *         0.0         0.0         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         1.0e
* hd     *      0.0118      0.0118      0.0118      0.0118s
* hd     *      0.0118      0.0118      0.0118      0.0118s
```

```
* hd     *      0.0118      0.0118      0.0118      0.0118s
* hd     *      0.0118      0.0118      0.0118      0.0118s
* hd     *      0.0118      0.0118      0.0118      0.0118s
* hd     *      0.0118      0.0118      0.0118      0.0118s
* hd     *      0.0118      0.0118      0.0118      0.0118e
* icflg  *           0           0           0           0s
* icflg  *           0           0           0           0s
* icflg  *           0           0           0           0s
* icflg  *           0           0           0           0s
* icflg  *           0           0           0           0s
* icflg  *           0           0           0           0s
* icflg  *           0           0           0           0e
* nff    *        -100        -100        -100        -100s
* nff    *        -100        -100        -100        -100s
* nff    *        -100        -100        -100        -100s
* nff    *        -100        -100        -100        -100s
* nff    *        -100        -100        -100        -100s
* nff    *        -100        -100        -100        -100s
* nff    *        -100        -100        -100        -100e
* 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         0.0e
* vl     *         5.0         5.0         5.0         5.0s
* vl     *         5.0         5.0         5.0         5.0s
* vl     *         5.0         5.0         5.0         5.0s
* vl     *         5.0         5.0         5.0         5.0s
* vl     *         5.0         5.0         5.0         5.0s
* vl     *         5.0         5.0         5.0         5.0s
* vl     *         5.0         5.0         5.0         5.0e
* vv     *         5.0         5.0         5.0         5.0s
* vv     *         5.0         5.0         5.0         5.0s
* vv     *         5.0         5.0         5.0         5.0s
* vv     *         5.0         5.0         5.0         5.0s
* vv     *         5.0         5.0         5.0         5.0s
* vv     *         5.0         5.0         5.0         5.0s
* vv     *         5.0         5.0         5.0         5.0e
* tl     *       300.0       301.0       562.0       562.0s
* tl     *       562.0       562.0       562.0       562.0s
* tl     *       562.0       562.0       562.0       562.0s
* tl     *       562.0       562.0       562.0       562.0s
* tl     *       562.0       562.0       562.0       562.0s
* tl     *       562.0       562.0       562.0       562.0s
* tl     *       562.0       562.0       562.0e
* tv     *       559.6       559.6       562.0       562.0s
* tv     *       562.0       562.0       562.0       562.0s
* tv     *       562.0       562.0       562.0       562.0s
* tv     *       562.0       562.0       562.0       562.0s
* tv     *       562.0       562.0       562.0       562.0s
* tv     *       562.0       562.0       562.0       562.0s
* tv     *       562.0       562.0       562.0e
* p      *       1.5E7       1.5E7       1.5E7       1.5E7s
* p      *       1.5E7       1.5E7       1.5E7       1.5E7s
```

```
* p      *         1.5E7        1.5E7        1.5E7        1.5E7s
* p      *         1.5E7        1.5E7        1.5E7        1.5E7s
* p      *         1.5E7        1.5E7        1.5E7        1.5E7s
* p      *         1.5E7        1.5E7        1.5E7        1.5E7s
* p      *         1.5E7        1.5E7        1.5E7e
* 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          0.0e
*  qppp  * f 0.0e
* mat    *            50 e
* tw     *      300.01       301.47       301.04       301.47       302.51s
* tw     *      301.48       304.41       301.48       306.74       301.48s
* tw     *      309.46       301.49       312.55       301.49       315.96s
* tw     *       301.5       319.66       301.51        323.6       301.51s
* tw     *      327.73       301.52        332.0       301.53       336.35s
* tw     *      301.54       340.73       301.55       345.08       301.55s
* tw     *      349.36       301.56       353.49       301.57       357.44s
* tw     *      301.58       361.15       301.59       364.58       301.59s
* tw     *      367.68        301.6       370.41        301.6       372.75s
* tw     *      301.61       374.67       301.61       376.14       301.61s
* tw     *      377.18       301.61       377.18       301.61e
* idrod  *           1e
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2            2            2            2s
* nhcel  *           2            2e
*  rdx   *        64.0e
* radrd  *         0.0    1.044E-3     2.088E-3     3.132E-3     4.176E-3s
* radrd  *     5.22E-3     5.325E-3     5.725E-3     6.125E-3e
* matrd  *           1            1            1            1 s
* matrd  *           1            3            2            2 e
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0            0            0            0s
*  nfax  *           0e
*  rftn  *      317.37       317.12       316.36        315.1s
*  rftn  *      313.35        311.1       308.87        308.3s
*  rftn  *      307.77       334.23       333.73       332.22s
*  rftn  *      329.73       326.25       321.82       317.51s
*  rftn  *      316.43       315.41        352.0       351.23s
*  rftn  *      348.92       345.09       339.77       333.01s
*  rftn  *      326.59       324.99        323.5       369.26s
*  rftn  *      368.22       365.09       359.91       352.74s
*  rftn  *      343.67       335.23       333.15        331.2s
*  rftn  *      385.83       384.51       380.57       374.07s
*  rftn  *      365.08       353.74       343.41       340.88s
*  rftn  *      338.52       401.47       399.88       395.16s
```

```
*   rftn *       387.37       376.64       363.14       351.07s
*   rftn *       348.14        345.4       415.95       414.12s
*   rftn *       408.66       399.67        387.3       371.78s
*   rftn *       358.16       354.87       351.79       429.06s
*   rftn *       427.01       420.88       410.79       396.95s
*   rftn *       379.62       364.63       361.03       357.67s
*   rftn *       440.66       438.41       431.69       420.65s
*   rftn *       405.53       386.63       370.49       366.64s
*   rftn *       363.04       450.61       448.19       440.99s
*   rftn *       429.17       412.98       392.78       375.74s
*   rftn *       371.69       367.91       458.79       456.25s
*   rftn *       448.68       436.25       419.26       398.08s
*   rftn *       380.38       376.19       372.28       465.14s
*   rftn *       462.51        454.7       441.87       424.34s
*   rftn *       402.52       384.41       380.15       376.17s
*   rftn *       469.56        466.9       458.97       445.97s
*   rftn *        428.2       406.07       387.84       383.56s
*   rftn *       379.56       472.03       469.37       461.48s
*   rftn *       448.52       430.81       408.75       390.65s
*   rftn *       386.41       382.45       472.56       469.96s
*   rftn *       462.23       449.54       432.18       410.56s
*   rftn *       392.85       388.72       384.85       471.18s
*   rftn *       468.68       461.25       449.04       432.34s
*   rftn *        411.5       394.46       390.48       386.76s
*   rftn *       467.96       465.61        458.6       447.09s
*   rftn *       431.31       411.59       395.46        391.7s
*   rftn *       388.18       462.99       460.82       454.35s
*   rftn *       443.72       429.13       410.85       395.86s
*   rftn *       392.37       389.11       456.38       454.42s
*   rftn *        448.6       439.01       425.83       409.28s
*   rftn *       395.65       392.48       389.52       448.24s
*   rftn *       446.53       441.44       433.04       421.47s
*   rftn *        406.9       394.83       392.02        389.4s
*   rftn *       438.74       437.31       433.01       425.91s
*   rftn *        416.1       403.72        393.4       390.99s
*   rftn *       388.74       428.06        426.9       423.45s
*   rftn *       417.73       409.81       399.79       391.35s
*   rftn *       389.38       387.54       416.37       415.51s
*   rftn *       412.93       408.64        402.7       395.14s
*   rftn *       388.72       387.22       385.82       403.89s
*   rftn *       403.32       401.62       398.79       394.86s
*   rftn *       389.84       385.53       384.52       383.57s
*   rftn *       391.44       391.16       390.29       388.85s
*   rftn *       386.84       384.27       382.03       381.51s
*   rftn *       381.02e
* rdpwr *          1.0          1.0          1.0          1.0          1.0s
* rdpwr *          1.0          0.0          0.0          0.0e
* cpowr *     0.015625e
* radpw *      0.18568      0.35797      0.53312      0.70045      0.85753s
* radpw *        1.002       1.1319       1.2451       1.3401       1.4155s
* radpw *       1.4701       1.5033       1.5144       1.5033       1.4702s
* radpw *       1.4155       1.3401       1.2451       1.1319        1.002s
* radpw *      0.85753      0.70046      0.53312      0.35798      0.18568e
* fpuo2 *          0.0e
*   ftd *         0.95e
*  gmix * f        0.0e
```

```
* pgapt *        7.0E6e
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4     1.9647E4     1.9647E4     1.9647E4s
*  burn *     1.9647E4e
*viewgrp *      0.83459      0.16541       0.7467       0.2533e
*beamlen *    5.0662E-3     0.012984     0.012984     0.024143e
*
*
******* type          num       userid           component name
fill                    5            0                    fill
*       jun1        ifty        ioff
           4           2            0
*     twtold        rfmx       concin        felv
         0.0         5.0          0.0         0.0
*        dxin       volin       alpin        vlin        tlin
         1.0    0.246875         0.0         0.0       562.0
*         pin        pain      flowin        vvin        tvin
       1.5E7          0.        37.5         0.0       562.0
*
*
******* type          num       userid           component name
fill                    6            0                    fill
*       jun1        ifty        ioff
        2281           2            0
*     twtold        rfmx       concin        felv
         0.0         5.0          0.0         0.0
*        dxin       volin       alpin        vlin        tlin
         1.0    0.246875         0.0         0.0       562.0
*         pin        pain      flowin        vvin        tvin
       1.5E7         0.0        37.5         0.0       562.0
*
*
******* type          num       userid           component name
fill                    7            0                    fill
*       jun1        ifty        ioff
        2271           2            0
*     twtold        rfmx       concin        felv
         0.0         5.0          0.0         0.0
*        dxin       volin       alpin        vlin        tlin
         1.0    0.246875         0.0         0.0       562.0
*         pin        pain      flowin        vvin        tvin
       1.5E7         0.0        37.5         0.0       562.0
*
*
******* type          num       userid           component name
fill                    8            0                    fill
*       jun1        ifty        ioff
        2261           2            0
*     twtold        rfmx       concin        felv
         0.0         5.0          0.0         0.0
*        dxin       volin       alpin        vlin        tlin
         1.0    0.246875         0.0         0.0       562.0
```

```
*          pin          pain        flowin          vvin          tvin
        1.5E7           0.0          37.5           0.0         562.0
*
*
******* type           num        userid          component name
break                    9             0                     Break
*        jun1         ibty          isat          ioff     adjpress
            5            0             3             0            0
*        dxin         volin         alpin           tin           pin
           1.0     0.246875           0.0         590.0        1.5E7
*        pain        concin         rbmx          poff          belv
           0.0          0.0           0.0           0.0           0.0
*
*
******* type           num        userid          component name
break                   10             0                     Break
*        jun1         ibty          isat          ioff     adjpress
          2201            0             3             0            0
*        dxin         volin         alpin           tin           pin
           1.0     0.246875           0.0         590.0        1.5E7
*        pain        concin         rbmx          poff          belv
           0.0          0.0           0.0           0.0           0.0
*
*
******* type           num        userid          component name
break                   11             0                     Break
*        jun1         ibty          isat          ioff     adjpress
          2211            0             3             0            0
*        dxin         volin         alpin           tin           pin
           1.0     0.246875           0.0         590.0        1.5E7
*        pain        concin         rbmx          poff          belv
           0.0          0.0           0.0           0.0           0.0
*
*
******* type           num        userid          component name
break                   12             0                     Break
*        jun1         ibty          isat          ioff     adjpress
          2221            0             3             0            0
*        dxin         volin         alpin           tin           pin
           1.0     0.246875           0.0         590.0        1.5E7
*        pain        concin         rbmx          poff          belv
           0.0          0.0           0.0           0.0           0.0
*
*
******* type           num        userid          component name
pipe                    13             0              outlet pipe
*      ncells        nodes          jun1          jun2          epsw
            5            0             7             5           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
```

```
* dx     *          1.0          1.0          1.0          1.0s
* dx     *          1.0e
* vol    *     0.246875     0.246875     0.246875     0.246875s
* vol    *     0.246875e
* fa     *     0.246875     0.246875     0.246875     0.246875s
* fa     *     0.246875     0.246875e
* kfac   *          0.0          0.0          0.0          0.0s
* kfac   *          0.0          0.0e
* grav   *          0.0          0.0          0.0          0.0s
* grav   *          0.0          0.0e
* hd     *   0.56065231   0.56065231   0.56065231   0.56065231s
* hd     *   0.56065231   0.56065231e
* icflg  *            0            0            0           0s
* icflg  *            0            0e
* nff    *         -100         -100         -100         -100s
* nff    *         -100         -100e
* alp    *          0.0          0.0          0.0          0.0s
* alp    *          0.0e
* vl     *      0.01022      0.01022      0.01022      0.01022s
* vl     *      0.01022      0.01022e
* vv     *         0.02         0.02         0.02         0.02s
* vv     *         0.02         0.02e
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0e
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0e
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7e
* pa     *          0.0          0.0          0.0          0.0s
* pa     *          0.0e
*
*
******* type        num       userid           component name
pipe                 14            0                outlet  pipe
*      ncells      nodes         jun1         jun2          epsw
             5          0         2171         2201           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     *          1.0          1.0          1.0          1.0s
* dx     *          1.0e
* vol    *     0.246875     0.246875     0.246875     0.246875s
* vol    *     0.246875e
* fa     *     0.246875     0.246875     0.246875     0.246875s
* fa     *     0.246875     0.246875e
* kfac   *          0.0          0.0          0.0          0.0s
* kfac   *          0.0          0.0e
* grav   *          0.0          0.0          0.0          0.0s
* grav   *          0.0          0.0e
* hd     *   0.56065231   0.56065231   0.56065231   0.56065231s
* hd     *   0.56065231   0.56065231e
```

```
* icflg  *            0            0            0            0s
* icflg  *            0            0e
* nff    *         -100         -100         -100         -100s
* nff    *         -100         -100e
* alp    *          0.0          0.0          0.0          0.0s
* alp    *          0.0e
* vl     *      0.01022      0.01022      0.01022      0.01022s
* vl     *      0.01022      0.01022e
* vv     *         0.02         0.02         0.02         0.02s
* vv     *         0.02         0.02e
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0e
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0e
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7e
* pa     *          0.0          0.0          0.0          0.0s
* pa     *          0.0e
*
*
******* type          num       userid           component name
pipe                   15            0               outlet pipe
*      ncells        nodes         jun1         jun2          epsw
             5            0         2181         2211           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     *          1.0          1.0          1.0          1.0s
* dx     *          1.0e
* vol    *     0.246875     0.246875     0.246875     0.246875s
* vol    *     0.246875e
* fa     *     0.246875     0.246875     0.246875     0.246875s
* fa     *     0.246875     0.246875e
* kfac   *          0.0          0.0          0.0          0.0s
* kfac   *          0.0          0.0e
* grav   *          0.0          0.0          0.0          0.0s
* grav   *          0.0          0.0e
* hd     *   0.56065231   0.56065231   0.56065231   0.56065231s
* hd     *   0.56065231   0.56065231e
* icflg  *            0            0            0            0s
* icflg  *            0            0e
* nff    *         -100         -100         -100         -100s
* nff    *         -100         -100e
* alp    *          0.0          0.0          0.0          0.0s
* alp    *          0.0e
* vl     *      0.01022      0.01022      0.01022      0.01022s
* vl     *      0.01022      0.01022e
* vv     *         0.02         0.02         0.02         0.02s
* vv     *         0.02         0.02e
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0e
```

```
* tv     *          562.0          562.0          562.0          562.0s
* tv     *          562.0e
* p      *          1.5E7          1.5E7          1.5E7          1.5E7s
* p      *          1.5E7e
* pa     *            0.0            0.0            0.0            0.0s
* pa     *            0.0e
*
*
******* type           num         userid           component name
pipe                    16              0              outlet pipe
*      ncells         nodes           jun1           jun2           epsw
             5             0           2191           2221            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     *            1.0            1.0            1.0            1.0s
* dx     *            1.0e
* vol    *       0.246875       0.246875       0.246875       0.246875s
* vol    *       0.246875e
* fa     *       0.246875       0.246875       0.246875       0.246875s
* fa     *       0.246875       0.246875e
* kfac   *            0.0            0.0            0.0            0.0s
* kfac   *            0.0            0.0e
* grav   *            0.0            0.0            0.0            0.0s
* grav   *            0.0            0.0e
* hd     *    0.56065231     0.56065231     0.56065231     0.56065231s
* hd     *    0.56065231     0.56065231e
* icflg  *             0              0              0             0s
* icflg  *             0              0e
* nff    *          -100           -100           -100           -100s
* nff    *          -100           -100e
* alp    *           0.0            0.0            0.0            0.0s
* alp    *           0.0e
* vl     *       0.01022        0.01022        0.01022        0.01022s
* vl     *       0.01022        0.01022e
* vv     *          0.02           0.02           0.02           0.02s
* vv     *          0.02           0.02e
* tl     *          562.0          562.0          562.0          562.0s
* tl     *          562.0e
* tv     *          562.0          562.0          562.0          562.0s
* tv     *          562.0e
* p      *          1.5E7          1.5E7          1.5E7          1.5E7s
* p      *          1.5E7e
* pa     *            0.0            0.0            0.0            0.0s
* pa     *            0.0e
*
*
******* type           num         userid           component name
pipe                    17              0          feed water pipe
*      ncells         nodes           jun1           jun2           epsw
             5             0           2261           2231            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     *          1.0          1.0          1.0          1.0s
* dx     *          1.0e
* vol    *     0.246875     0.246875     0.246875     0.246875s
* vol    *     0.246875e
* fa     *     0.246875     0.246875     0.246875     0.246875s
* fa     *     0.246875     0.246875e
* kfac   *          0.0          0.0          0.0          0.0s
* kfac   *          0.0          0.0e
* grav   *          0.0          0.0          0.0          0.0s
* grav   *          0.0          0.0e
* hd     *   0.56065231   0.56065231   0.56065231   0.56065231s
* hd     *   0.56065231   0.56065231e
* icflg  *            0            0            0            0s
* icflg  *            0            0e
* nff    *         -100         -100         -100         -100s
* nff    *         -100         -100e
* alp    *          0.0          0.0          0.0          0.0s
* alp    *          0.0e
* vl     *      0.01013      0.01013      0.01013      0.01013s
* vl     *      0.01013      0.01013e
* vv     *         0.02         0.02         0.02         0.02s
* vv     *         0.02         0.02e
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0e
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0e
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7e
* pa     *          0.0          0.0          0.0          0.0s
* pa     *          0.0e
*
*
****** type          num       userid          component name
pipe                  18            0         feed water pipe
*      ncells       nodes         jun1         jun2         epsw
           5            0         2271         2241          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     *          1.0          1.0          1.0          1.0s
* dx     *          1.0e
* vol    *     0.246875     0.246875     0.246875     0.246875s
* vol    *     0.246875e
```

```
* fa     *      0.246875     0.246875     0.246875     0.246875s
* fa     *      0.246875     0.246875e
* kfac   *           0.0          0.0          0.0          0.0s
* kfac   *           0.0          0.0e
* grav   *           0.0          0.0          0.0          0.0s
* grav   *           0.0          0.0e
* hd     *    0.56065231   0.56065231   0.56065231   0.56065231s
* hd     *    0.56065231   0.56065231e
* icflg  *             0            0            0           0s
* icflg  *             0            0e
* nff    *          -100         -100         -100         -100s
* nff    *          -100         -100e
* alp    *           0.0          0.0          0.0          0.0s
* alp    *           0.0e
* vl     *       0.01013      0.01013      0.01013      0.01013s
* vl     *       0.01013      0.01013e
* vv     *          0.02         0.02         0.02         0.02s
* vv     *          0.02         0.02e
* tl     *         562.0        562.0        562.0        562.0s
* tl     *         562.0e
* tv     *         562.0        562.0        562.0        562.0s
* tv     *         562.0e
* p      *         1.5E7        1.5E7        1.5E7        1.5E7s
* p      *         1.5E7e
* pa     *           0.0          0.0          0.0          0.0s
* pa     *           0.0e
*
*
******* type          num       userid          component name
pipe                   19            0         feed water pipe
*      ncells        nodes         jun1         jun2         epsw
             5            0            4            6          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     *           1.0          1.0          1.0          1.0s
* dx     *           1.0e
* vol    *      0.246875     0.246875     0.246875     0.246875s
* vol    *      0.246875e
* fa     *      0.246875     0.246875     0.246875     0.246875s
* fa     *      0.246875     0.246875e
* kfac   *           0.0          0.0          0.0          0.0s
* kfac   *           0.0          0.0e
* grav   *           0.0          0.0          0.0          0.0s
* grav   *           0.0          0.0e
* hd     *    0.56065231   0.56065231   0.56065231   0.56065231s
* hd     *    0.56065231   0.56065231e
* icflg  *             0            0            0           0s
* icflg  *             0            0e
* nff    *          -100         -100         -100         -100s
* nff    *          -100         -100e
```

```
* alp    *          0.0          0.0          0.0          0.0s
* alp    *          0.0e
* vl     *      0.01013      0.01013      0.01013      0.01013s
* vl     *      0.01013      0.01013e
* vv     *         0.02         0.02         0.02         0.02s
* vv     *         0.02         0.02e
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0e
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0e
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7e
* pa     *          0.0          0.0          0.0          0.0s
* pa     *          0.0e
*
*
******* type          num       userid           component name
pipe                   20            0         feed water pipe
*      ncells        nodes         jun1         jun2         epsw
             5            0         2281         2251          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     *          1.0          1.0          1.0          1.0s
* dx     *          1.0e
* vol    *     0.246875     0.246875     0.246875     0.246875s
* vol    *     0.246875e
* fa     *     0.246875     0.246875     0.246875     0.246875s
* fa     *     0.246875     0.246875e
* kfac   *          0.0          0.0          0.0          0.0s
* kfac   *          0.0          0.0e
* grav   *          0.0          0.0          0.0          0.0s
* grav   *          0.0          0.0e
* hd     *   0.56065231   0.56065231   0.56065231   0.56065231s
* hd     *   0.56065231   0.56065231e
* icflg  *            0            0            0           0s
* icflg  *            0           0e
* nff    *         -100         -100         -100         -100s
* nff    *         -100         -100e
* alp    *          0.0          0.0          0.0          0.0s
* alp    *          0.0e
* vl     *      0.01013      0.01013      0.01013      0.01013s
* vl     *      0.01013      0.01013e
* vv     *         0.02         0.02         0.02         0.02s
* vv     *         0.02         0.02e
* tl     *        562.0        562.0        562.0        562.0s
* tl     *        562.0e
* tv     *        562.0        562.0        562.0        562.0s
* tv     *        562.0e
* p      *        1.5E7        1.5E7        1.5E7        1.5E7s
* p      *        1.5E7e
```

```
* pa     *          0.0          0.0          0.0          0.0s
* pa     *          0.0e
*
*
******* type          num       userid          component name
vessel                21            0          Reactor Vessel
*        nasx         nrsx         ntsx         ncsr       ivssbf
            3            2            4           16            0
*        idcu         idcl         idcr         icru         icrl
            3            1            1            2            1
*        icrr        ilcsp        iucsp         iuhp        iconc
            1            0            0            0            0
*       igeom        nvent        nvvtb       nsgrid     vesstype
            0            0            0            0            0
*       shelv         epsw
          0.0          0.0
*    z     *          2.0        6.398          9.0e
*    r     *     0.158584     0.218584e
*    t     *         90.0        180.0        270.0        360.0e
*      lisrl      lisrc      lisrf      ljuns       zfrac
            1          1          2         10
            1          2          2       2111
            1          3          2       2131
            1          4          2       2151
            3          1         -2       2101
            3          1          3          7
            3          2         -2       2121
            3          2          3       2171
            3          3         -2       2141
            3          3          3       2181
            3          4         -2       2161
            3          4          3       2191
            3          5          3       2231
            3          6          3          6
            3          6          3       2241
            3          8          3       2251
* level 1
*
*   cfzlyt *          0.0          0.0          0.0          0.1s
*   cfzlyt *          0.0          0.0          0.0          0.1e
*   cfzlz  *          0.1          0.1          0.1          0.1s
*   cfzlz  *          0.1          0.1          0.1          0.1e
*   cfzlxr *          0.0          0.0          0.0          0.0s
*   cfzlxr *          0.0          0.0          0.0          0.0e
*   cfzvyt *          0.0          0.0          0.0          0.1s
*   cfzvyt *          0.0          0.0          0.0          0.1e
*   cfzvz  *          0.1          0.1          0.1          0.1s
*   cfzvz  *          0.1          0.1          0.1          0.1e
*   cfzvxr *          0.0          0.0          0.0          0.0s
*   cfzvxr *          0.0          0.0          0.0          0.0e
*    frvol *   0.19483965   0.19483965   0.19483965   0.19483965s
*    frvol *   0.22536376   0.22536376   0.22536376   0.22536376e
*   frfayt *          0.0          0.0          0.0          0.0s
*   frfayt *          1.0          1.0          1.0          1.0e
*   frfaz  *  0.097419826  0.097419826  0.097419826  0.097419826s
*   frfaz  *   0.22536376   0.22536376   0.22536376   0.22536376e
```

```
*   frfaxr *    0.9999873    0.9999873    0.9999873    0.9999873s
*   frfaxr *          0.0          0.0          0.0          0.0e
*     hdyt *          0.0          0.0          0.0          0.0s
*     hdyt *          0.0          0.0          0.0          0.0e
*     hdz  *        0.322        0.322        0.322        0.322s
*     hdz  *        0.322        0.322        0.322        0.322e
*     hdxr *          0.0          0.0          0.0          0.0s
*     hdxr *          0.0          0.0          0.0          0.0e
*    alpn  *          0.0          0.0          0.0          0.0s
*    alpn  *          0.0          0.0          0.0          0.0e
*    vvnyt *          0.0          0.0          0.0          0.0s
*    vvnyt *          0.0          0.0          0.0          0.0e
*    vvnz  *          1.0          1.0          1.0          1.0s
*    vvnz  *         −1.0         −1.0         −1.0         −1.0e
*    vvnxr *          0.0          0.0          0.0          0.0s
*    vvnxr *          0.0          0.0          0.0          0.0e
*    vlnyt *          0.0          0.0          0.0          0.0s
*    vlnyt *          0.0          0.0          0.0          0.0e
*    vlnz  *       0.9176       0.9176       0.9176       0.9176s
*    vlnz  *       −0.624       −0.624       −0.624       −0.624e
*    vlnxr *          0.0          0.0          0.0          0.0s
*    vlnxr *          0.0          0.0          0.0          0.0e
*      tvn *        560.0        560.0        560.0        560.0s
*      tvn *        560.0        560.0        560.0        560.0e
*      tln *        560.0        560.0        560.0        560.0s
*      tln *        560.0        560.0        560.0        560.0e
*       pn *        1.5E7        1.5E7        1.5E7        1.5E7s
*       pn *        1.5E7        1.5E7        1.5E7        1.5E7e
*      pan *          0.0          0.0          0.0          0.0s
*      pan *          0.0          0.0          0.0          0.0e
* level 2
*
*   cfzlyt *          0.0          0.0          0.0          0.1s
*   cfzlyt *          0.0          0.0          0.0          0.1e
*   cfzlz  *          0.1          0.1          0.1          0.1s
*   cfzlz  *          0.1          0.1          0.1          0.1e
*   cfzlxr *          0.0          0.0          0.0          0.0s
*   cfzlxr *          0.0          0.0          0.0          0.0e
*   cfzvyt *          0.0          0.0          0.0          0.1s
*   cfzvyt *          0.0          0.0          0.0          0.1e
*   cfzvz  *          0.1          0.1          0.1          0.1s
*   cfzvz  *          0.1          0.1          0.1          0.1e
*   cfzvxr *          0.0          0.0          0.0          0.0s
*   cfzvxr *          0.0          0.0          0.0          0.0e
*    frvol *  0.19483965   0.19483965   0.19483965   0.19483965s
*    frvol *  0.22536376   0.22536376   0.22536376   0.22536376e
*   frfayt *          0.0          0.0          0.0          0.0s
*   frfayt *          1.0          1.0          1.0          1.0e
*   frfaz  * 0.097419826  0.097419826  0.097419826  0.097419826s
*   frfaz  *  0.22536376   0.22536376   0.22536376   0.22536376e
*   frfaxr *          0.0          0.0          0.0          0.0s
*   frfaxr *          0.0          0.0          0.0          0.0e
*     hdyt *          0.0          0.0          0.0          0.0s
*     hdyt *          0.0          0.0          0.0          0.0e
*     hdz  *        0.322        0.322        0.322        0.322s
*     hdz  *        0.322        0.322        0.322        0.322e
```

```
*     hdxr *          0.0          0.0          0.0          0.0s
*     hdxr *          0.0          0.0          0.0          0.0e
*    alpn  *          0.0          0.0          0.0          0.0s
*    alpn  *          0.0          0.0          0.0          0.0e
*    vvnyt *          0.0          0.0          0.0          0.0s
*    vvnyt *          0.0          0.0          0.0          0.0e
*    vvnz  *          2.0          2.0          2.0          2.0s
*    vvnz  *         -2.0         -2.0         -2.0         -2.0e
*    vvnxr *          0.0          0.0          0.0          0.0s
*    vvnxr *          0.0          0.0          0.0          0.0e
*    vlnyt *          0.0          0.0          0.0          0.0s
*    vlnyt *          0.0          0.0          0.0          0.0e
*    vlnz  *       0.9211       0.9211       0.9211       0.9211s
*    vlnz  *      -0.6241      -0.6241      -0.6241      -0.6241e
*    vlnxr *          0.0          0.0          0.0          0.0s
*    vlnxr *          0.0          0.0          0.0          0.0e
*      tvn *        560.0        560.0        560.0        560.0s
*      tvn *        560.0        560.0        560.0        560.0e
*      tln *        560.0        560.0        560.0        560.0s
*      tln *        560.0        560.0        560.0        560.0e
*       pn *        1.5E7        1.5E7        1.5E7        1.5E7s
*       pn *        1.5E7        1.5E7        1.5E7        1.5E7e
*      pan *          0.0          0.0          0.0          0.0s
*      pan *          0.0          0.0          0.0          0.0e
* level 3
*
*   cfzlyt *          0.0          0.0          0.0          0.1s
*   cfzlyt *          0.0          0.0          0.0          0.1e
*   cfzlz  *          0.1          0.1          0.1          0.1s
*   cfzlz  *          0.1          0.1          0.1          0.1e
*   cfzlxr *          0.0          0.0          0.0          0.0s
*   cfzlxr *          0.0          0.0          0.0          0.0e
*   cfzvyt *          0.0          0.0          0.0          0.1s
*   cfzvyt *          0.0          0.0          0.0          0.1e
*   cfzvz  *          0.1          0.1          0.1          0.1s
*   cfzvz  *          0.1          0.1          0.1          0.1e
*   cfzvxr *          0.0          0.0          0.0          0.0s
*   cfzvxr *          0.0          0.0          0.0          0.0e
*    frvol *   0.19483965   0.19483965   0.19483965   0.19483965s
*    frvol *   0.22536376   0.22536376   0.22536376   0.22536376e
*   frfayt *          1.0          1.0          1.0          1.0s
*   frfayt *   0.99987189   0.99987189   0.99987189   0.99987189e
*   frfaz  *          0.0          0.0          0.0          0.0s
*   frfaz  *          0.0          0.0          0.0          0.0e
*   frfaxr *          0.0          0.0          0.0          0.0s
*   frfaxr *          0.0          0.0          0.0          0.0e
*     hdyt *          0.0          0.0          0.0          0.0s
*     hdyt *          0.0          0.0          0.0          0.0e
*     hdz  *        0.322        0.322        0.322        0.322s
*     hdz  *        0.322        0.322        0.322        0.322e
*     hdxr *          0.0          0.0          0.0          0.0s
*     hdxr *          0.0          0.0          0.0          0.0e
*    alpn  *          0.0          0.0          0.0          0.0s
*    alpn  *          0.0          0.0          0.0          0.0e
*    vvnyt *          0.0          0.0          0.0          0.0s
*    vvnyt *          0.0          0.0          0.0          0.0e
```

```
*    vvnz  *          0.0          0.0          0.0          0.0s
*    vvnz  *          0.0          0.0          0.0          0.0e
*    vvnxr *          0.0          0.0          0.0          0.0s
*    vvnxr *          0.0          0.0          0.0          0.0e
*    vlnyt *          0.0          0.0          0.0          0.0s
*    vlnyt *          0.0          0.0          0.0          0.0e
*    vlnz  *          0.0          0.0          0.0          0.0s
*    vlnz  *          0.0          0.0          0.0          0.0e
*    vlnxr *          0.0          0.0          0.0          0.0s
*    vlnxr *          0.0          0.0          0.0          0.0e
*      tvn *        560.0        560.0        560.0        560.0s
*      tvn *        560.0        560.0        560.0        560.0e
*      tln *        560.0        560.0        560.0        560.0s
*      tln *        560.0        560.0        560.0        560.0e
*       pn *        1.5E7        1.5E7        1.5E7        1.5E7s
*       pn *        1.5E7        1.5E7        1.5E7        1.5E7e
*      pan *          0.0          0.0          0.0          0.0s
*      pan *          0.0          0.0          0.0          0.0e
*
*
****************************************
*     Starting Power Components        *
****************************************
*
******* type          num       userid          component name
power                  22            0                power com
*     numpwr     chanpow
            4           1
* htnum *          1          4          3          2 e
*     irpwty       ndgx       ndhx       nrts      nhist
            5          0          0          0          0
*     izpwtr     izpwsv     nzpwtb     nzpwsv     nzpwrf
            0          1          2          0          0
*     ipwrad     ipwdep    promheat   decaheat   wtbypass
            0          0        0.4        0.4       0.75
*      nzpwz      nzpwi     nfbpwt      nrpwr      nrpwi
           26         -1          0          1          0
*      react      tneut     rpwoff     rrpwmx     rpwscl
          0.0  4.5085E-4        0.0     1.0E20        1.0
*     rpowri      zpwin     zpwoff     rzpwmx
          1.0E7        0.0        0.0     1.0E20
*     extsou       pldr      pdrat      fucrac
          0.0        0.0     1.2898        0.0
* zpwzt *          0.0     0.1472     0.2944     0.4416     0.5888s
* zpwzt *        0.736     0.8832     1.0304     1.1776     1.3248s
* zpwzt *        1.472     1.6192     1.7664     1.9136     2.0608s
* zpwzt *        2.208     2.3552     2.5024     2.6496     2.7968s
* zpwzt *        2.944     3.0912     3.2384     3.3856     3.5328s
* zpwzt *        3.68e
* zpwtb1*         0.0s
* zpwtb1*          0.0  0.14231484  0.28173256  0.41541501  0.54064082s
* zpwtb1*   0.65486073  0.75574957  0.84125353    0.909632  0.95949297s
* zpwtb1*   0.98982144         1.0  0.98982144  0.95949297    0.909632s
* zpwtb1*   0.84125353  0.75574957  0.65486073  0.54064082  0.41541501s
* zpwtb1*   0.28173256  0.14231484 1.22515E-16 -0.14231484 -0.28173256s
* zpwtb1* -0.41541501s
```

```
* zpwtb2*        1.0E6s
* zpwtb2*        1.0         1.0         1.0         1.0         1.0s
* zpwtb2*        1.0         1.0         1.0         1.0         1.0s
* zpwtb2*        1.0         1.0         1.0         1.0         1.0s
* zpwtb2*        1.0         1.0         1.0         1.0         1.0s
* zpwtb2*        1.0         1.0         1.0         1.0         1.0s
* zpwtb2*        1.0e
****************************************
*     Finished Power Components        *
****************************************
*
*
*
end
*
*****************
* Timestep Data *
*****************
*         dtmin          dtmax           tend          rtwfp
        1.0E-10            1.0         2000.0            1.0
*         edint          gfint         dmpint         sedint
         5000.0            1.0           10.0           10.0
*
*       endflag
           -1.0
```

C.2 TRACE Restart Input File

```
free format
*
*
*************
* main data *
*************
*
*      numtcr          ieos         inopt          nmat          id2o
            1             0             1             1             0
       Single Channel
*
*
*
*****************
* namelist data *
*****************
*
 &inopts
  dtstrt=1.0E-4,
  icflow=2,
  ikfac=1,
  ipowr=1,
  itdmr=1,
  nosets=1,
  nsolver=1,
  usesjc=3,
  npower=1
```

```
 &end
*
***************
* Model Flags *
***************
*
*        dstep         timet
            -1           0.0
*       stdyst        transi         ncomp          njun          ipak
             1             0            22            24             1
*         epso          epss
         1.0E-4        1.0E-4
*       oitmax        sitmax        isolut        ncontr         nccfl
            10            10             0             0             0
*         ntsv          ntcb          ntcf          ntrp          ntcp
             7             7             0             0             0
*
*
*  Component input order (IORDER)
*-- type ---- num ------------- name -------- +    jun1   jun2
    jun3
* CHAN      *       1 s * 8x8                  +      10   2101
* CHAN      *       2 s * 8x8                  +    2151   2161
* CHAN      *       3 s * 8x8                  +    2131   2141
* CHAN      *       4 s * 8x8                  +    2111   2121
* FILL      *       5 s * fill                 +       4
* FILL      *       6 s * fill                 +    2281
* FILL      *       7 s * fill                 +    2271
* FILL      *       8 s * fill                 +    2261
* BREAK     *       9 s * Break                +       5
* BREAK     *      10 s * Break                +    2201
* BREAK     *      11 s * Break                +    2211
* BREAK     *      12 s * Break                +    2221
* PIPE      *      13 s * outlet pipe          +       7      5
* PIPE      *      14 s * outlet pipe          +    2171   2201
* PIPE      *      15 s * outlet pipe          +    2181   2211
* PIPE      *      16 s * outlet pipe          +    2191   2221
* PIPE      *      17 s * feed water pipe      +    2261   2231
* PIPE      *      18 s * feed water pipe      +    2271   2241
* PIPE      *      19 s * feed water pipe      +       4      6
* PIPE      *      20 s * feed water pipe      +    2281   2251
* VESSEL    *      21 s * Reactor Vessel       +
* POWER     *      22 e * power com            +
*
**********************
* material properties *
**********************
*
*   matb*           50 e
*  ptbln*            1 e
*  User Defined Material : 50
*
*
*  prptb         temp          rho           cp         cond         emis
*  prptb*       200.0       7820.0        530.0     1.0E-20          0.0 e
*
```

```
*
*
*************************************************
* Starting Signal Variable Section of Model     *
*************************************************
           0             0             0            0             0
*************************************************
* Finished Signal Variable Section of Model     *
*************************************************
*
*
*
*
*****************************************
* Starting Control System Section of Model  *
*****************************************
*
****** Control Blocks ******
           0             0             0            0             0
*****************************************
* Finished Control System Section of Model  *
*****************************************
*
*
*
end
*
*****************
* Timestep Data *
*****************
*        dtmin          dtmax          tend          rtwfp
       1.0E-10          1.0          2000.0          1.0
*        edint          gfint         dmpint         sedint
         1.0              1.0           1.0            1.0
*
*       endflag
          -1.0
```

C.3 PARCS Restart Input File

```
!
CASEID VessChanSS
CNTL                                    ! control card
      CORE_TYPE     BWR                 ! core type, PWR or BWR
      CORE_POWER    100.0               !  initial relative power,
      BANK_POS      100.
      PPM           0.0                 ! boron ppm
      TH_FDBK       T                   ! True or False
      EXT_TH        T   ../../../maptab TRAC 1 1 -1.0e+4
      TRANSIENT     F
!     PRINT OPTIONS
!                   input  iteration     planar                adj
!                    edit      table      power       pin      reac
      print_opt        F          F          T         F         F
!                    fdbk       flux      planar
```

```
PARAM
! Basic Iteration Control Parameters
      n_iters    2      1500                    ! min, max
! Convergence Criteria
      conv_ss    1.0e-6  1.e-5  5.e-4  0.001 ! keff,globfis,locfis,
          tfuel
! Wielandt Shift Control
     wielandt   0.04   0.1   1.0  ! shift,intishit,keff goal
! Nonlinear Update Control
   nodal_kern hybrid      ! kernel method FDM, HYBRID, ANM
      nlupd_ss   3 3 1   ! nonlin updte: nupdcy, ninitout, nthpnod
      eps_anm    0.005   ! anm stabilization criteria
      eps_erf    0.005   ! cv for inners exit
!=============================================================================

XSEC  !table 7
      func_type  13 435
      dnp_ngrp   6
      EFIL_XS_R   '../../../PBTT_xsec_rodded'
      EFIL_XS_UR  '../../../PBTT_xsec_unrodded'
      kin_comp   1  1 -435
! pbtt specs
      dnp_beta   0.000167 0.001134 0.001022 0.002152 0.000837
          0.000214
      dnp_lambda 0.012813 0.031536 0.124703 0.328273 1.405280
          3.844728

!=============================================================================

! comp_num    2     !FUEL
! base_macro 0.2777778     0.010         0.005         0.005         0.02
!            0.8333333     0.1           0.125         0.125
!=============================================================================

GEOM
 geo_dim  2  2  27 1 1            ! nasyx,nasyy,nz
 Rad_Conf
  2 2
  2 2
 grid_x      2*13.461             ! in cm
 neutmesh_x 2*1                   !
 grid_y      2*13.461
 neutmesh_y 2*1
 grid_z      30.4  25*14.72  41.4
 Boun_Cond   0  0  0  0  1  1     !ibcw,ibce,ibcn,ibcs,ibcb,ibct
!
 planar_reg     1
  435 435
  435 435
 planar_reg     2
 365 365
 365 365
!
 PR_Assign  1 25*2 1
!
```

```
 CR_axinfo    20.60798822  3.65056 !fully inserted position and step
    size
 bank_conf
  1  1
  1  1
TH
    FA_POWPIT  2.5  15.275          !assembly power(Mw) and pitch(cm)
      CDC_DED     1  1
      gamma_frac 0.0  0.0           !direct heating fraction
TRAN
     Time_Step  1000.0  0.0010  100.0  0.10
     Scram      T 114.0  0.4  2.2
     nlupd_tr   5  1  5  10
     conv_tr    0.001
     eps_xsec   0.001
.
```