ANL-AFCI-175

Status Report on High Fidelity Reactor Simulation

Nuclear Engineering Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne, see www.anl.gov.

Availability of This Report

This report is available, at no cost, at http://www.osti.gov/bridge. It is also available on paper to the U.S. Department of Energy and its contractors, for a processing fee, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone (865) 576-8401
fax (865) 576-5728
reports@adonis.osti.gov

ANL-AFCI-175

Status Report on High Fidelity Reactor Simulation

by
G. Palmiotti, M. Smith, C. Rabiti, E. Lewis, W. Yang, M. Leclere
Nuclear Engineering Division, Argonne National Laboratory

A. Siegel, P. Fischer, D. Kaushik, J. Ragusa, J. Lottes, B. Smith
Mathematics and Computer Science Division, Argonne National Laboratory

September 15, 2006

work sponsored by

U. S. Department of Energy,
Office of Nuclear Energy, Science and Technology

UChicago ▶ Argonne LLC

Office of Science
U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC

ANL-AFCI-175

Status Report on High-Fidelity Reactor Simulation

G. Palmiotti, M. Smith, C. Rabiti, E. Lewis, W. Yang, M. Leclere
Nuclear Engineering Division

A. Siegel, P. Fischer, D. Kaushik, J. Ragusa, J. Lottes, B. Smith
Mathematics and Computer Science Division

Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439, USA

September 2006

Status Report on High-Fidelity Reactor Simulation

G. Palmiotti, M. Smith, C. Rabiti, E. Lewis, W. Yang, M. Leclere
Nuclear Engineering Division

A. Siegel, P. Fischer, D. Kaushik, J. Ragusa, J. Lottes, B. Smith
Mathematics and Computer Science Division

Abstract

This report presents the effort under way at Argonne National Laboratory toward a comprehensive, integrated computational tool intended mainly for the high-fidelity simulation of sodium-cooled fast reactors. The main activities carried out involved neutronics, thermal hydraulics, coupling strategies, software architecture, and high-performance computing. A new neutronics code, UNIC, is being developed. The first phase involves the application of a spherical harmonics method to a general, unstructured three-dimensional mesh. The method also has been interfaced with a method of characteristics. The spherical harmonics equations were implemented in a stand-alone code that was then used to solve several benchmark problems. For thermal hydraulics, a computational fluid dynamics code called Nek5000, developed in the Mathematics and Computer Science Division for coupled hydrodynamics and heat transfer, has been applied to a single-pin, periodic cell in the wire-wrap geometry typical of advanced burner reactors. Numerical strategies for multiphysics coupling have been considered and higher-accuracy efficient methods proposed to finely simulate coupled neutronic/thermal-hydraulic reactor transients. Initial steps have been taken in order to couple UNIC and Nek5000, and simplified problems have been defined and solved for testing. Furthermore, we have begun developing a lightweight computational framework, based in part on carefully selected open source tools, to nonobtrusively and efficiently integrate the individual physics modules into a unified simulation tool

Table of Contents

I Introduction

As specified in the previous deliverable ANL-AFCI-168, advanced simulation of a reactor plant calls for an integrated high-fidelity system of software tools that describe the overall nuclear plant behavior, taking into account coupling of the systems and physical phenomena during reactor operations ranging from neutronics to fuel behavior and from thermal-hydraulics to structural mechanics.

At Argonne National Laboratory, an effort has been initiated in this direction. Specifically, a new neutronics code, UNIC, is being written based on a single-step approach that will reduce approximations related to collapsing and homogenization of cross sections. In addition, the computational hydrodynamics code Nek5000 is being adapted to the needs of reactor simulation, in particular for fast sodium-cooled reactors. Coupling strategies are being devised to provide a better link between neutronics and thermal-hydraulics codes, and steps are under way to couple UNIC and Nek5000. Also under development are high-performance computing tools and a software architecture needed to carry out the huge calculations typical of this type of high-fidelity simulation.

In the following sections, we document these activities, discuss our work with UNIC and Nek5000, and present sample problems as test cases for studying the coupling of the two codes. We conclude with an evaluation of software technologies for the future integrated system.

II Development of the Neutronics Code UNIC

Current neutronics analysis requires two or more homogenization and group energy collapsing steps. The first step involves a local calculation to obtain the spectral self-shielding for the individual pin. The second step requires an assembly-level calculation to obtain the spectral self-shielding for each assembly and to produce homogenous parameters for each unique assembly in the reactor. These assembly-level cross sections are used in a global geometry calculation to obtain the flux solution for a targeted reactor system.

One of the major enhancements proposed for neutronics analysis is the elimination of the multistep calculational procedure for treating the energy variable. The objective is to exploit advances in both numerical and algorithmic efficiency, together with the significant increase in computing power offered by systems with several thousands of processors, in order to solve the neutron transport equation in a detailed three-dimensional geometry with thousands of energy groups. Such a deterministic approach would directly compete with Monte Carlo methods and would offer the clear advantage of being systematic and of providing an adjoint solution for calculating sensitivity coefficients. This approach has been adopted for the new code, called **UNIC** (**u**ltimate **n**eutronics **i**nvestigation **c**ode), being developed at Argonne.

UNIC is planned to have very flexible geometrical options, as in the case of Monte Carlo stochastic-based codes, by using a general geometry description and a very large number of energy groups (of the order of 10,000) to eliminate spatial approximations, eliminate the lattice cross-section generation step, and account accurately for energy and space resonance self-shielding effects. The intent is to enable analysis of advanced nuclear reactor designs, including fast reactor systems.

Two methods are under development for the UNIC code. The first method, PnFE, is based on the even-parity (second-order) form of the Boltzmann transport equation using the following discretization of the space-angle-energy phase space as a base method:

- multigroup energy discretization,
- three-dimensional finite elements in space, including curvilinear elements, and
- spherical harmonics expansions in angle.

The second is a method of characteristics approach based on the first-order integral transport equation. Coupling of the solution methods is ensured by continuity on boundaries of decomposed domains.

At present, the PnFE code uses a conjugate gradient method with automatic preconditioning to solve the within-group linear system of equations. The PETSc (Portable, Extensible Toolkit for Scientific computation) library developed at Argonne's Mathematics and Computer Science Division has been adapted for massively parallel solution of the multigroup problem.

After the first development phase, the UNIC code will provide a tool that combines the advantages of various methods such as spherical harmonics, discrete ordinates, and the method of characteristics. Whereas current reactor analysis tools use a form of the integral transport method (collision probability or method of characteristics) to handle the double heterogeneity, one can envision a characteristics formulation being used locally to handle the pin heterogeneity while a spherical harmonics formulation is used for the remainder of the domain.

The key is to use flexible coding structures and develop a strategy for coupling the different methodologies (discrete ordinates, characteristics, spherical harmonics, etc.). To this end, we are using the variational nodal method, with which Argonne has considerable experience. This method splits the problem domain into large spatial "nodes," or subdomains, and then uses spherical harmonic interface approximations to couple the nodes. In our recent work, we generalized the spherical harmonic interface conditions, thereby allowing different methodologies to be used in each subdomain.

In Section II.1 we derive the multigroup transport equation and cast it in the common within-group form. In Section II.2 we give the derivation of the second-order form of the transport equation, followed by the discretization of the spherical harmonics finite element method. In Section II.3 we present the characteristics formulation, and in Section II.4 we discuss the coupling of the two methodologies with the spherical harmonic conditions. In Section II.5 we provide information about mesh generation. In Section II.6 we present preliminary results obtained with the spherical harmonics-based methodology.

II.1 The Within-Group Neutron Transport Equation

The formulation starts with the time-independent form of the Boltzmann transport equation taken from Lewis and Miller [1]:

$$\hat{\Omega}\cdot\vec{\nabla}\psi(\vec{r},\hat{\Omega},E)+\Sigma_t(\vec{r},E)\psi(\vec{r},\hat{\Omega},E)= \int\int\Sigma_s(\vec{r},\hat{\Omega}'\rightarrow\hat{\Omega},E'\rightarrow E)\psi(\vec{r},\hat{\Omega}',E')d\Omega' dE'+S(\vec{r},\hat{\Omega},E) \quad (1)$$

Where $\psi(\vec{r},\hat{\Omega},E)$ represents the neutron angular flux and is a function of three space variables (x, y, z in $\vec{r}$), two angular variables (θ and ϕ in $\hat{\Omega}$), and one energy variable (E). The total cross section, $\Sigma_t(\vec{r},E)$, represents the sum of all possible neutron reaction probabilities with energy E at the point $\vec{r}$. Similarly, the scattering kernel $\Sigma_s(\vec{r},\hat{\Omega}'\rightarrow\hat{\Omega},E'\rightarrow E)d\Omega dE$ represents the probability that a particle at $\vec{r}$ with energy E' traveling in the direction $\hat{\Omega}'$ is scattered into energy dE about E with direction $d\Omega$ about $\hat{\Omega}$. $S(r,\hat{\Omega},E)$ is a generic neutron source that includes fission sources as well as fixed and external sources.

For the initial derivation, isotropic scattering is used to simplify the discussion (implementation of anisotropic scattering is discussed elsewhere). The substitution of an

isotropic scattering kernel in place of the anisotropic scattering kernel simplifies Eq. (1) to

$$\hat{\Omega}\cdot\vec{\nabla}\psi(\vec{r},\hat{\Omega},E)+\Sigma_t(\vec{r},E)\psi(\vec{r},\hat{\Omega},E)=\int\Sigma_s(\vec{r},E'\to E)\phi(\vec{r},E')dE'+S(\vec{r},\hat{\Omega},E)\,, \qquad (2)$$

where the scalar flux relationship $\phi(\vec{r},E)=\int\psi(\vec{r},\hat{\Omega},E)d\Omega$ has been employed.

The first step in a deterministic formulation is to apply a multigroup approach [1] to the energy dependence of the neutron flux. This approach divides the energy range of interest into G intervals with an upper energy cutoff, E_0, and lower energy cutoff, E_G, as seen in Figure 1.

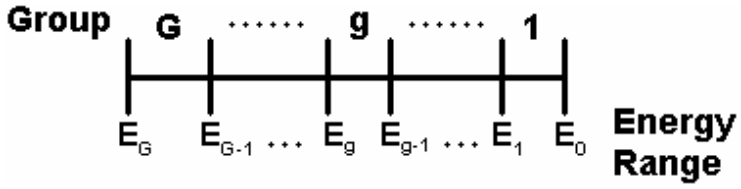

Figure 1. Splitting of the energy range into energy groups

The desire is to develop G equations based upon the group angular fluxes $\psi_g(\vec{r},\hat{\Omega})$ and the group sources $S_g(\vec{r},\hat{\Omega})$ defined by Eqs. (3) and (4).

$$\psi_g(\vec{r},\hat{\Omega})=\int_{E_g}^{E_{g-1}}\psi(\vec{r},\hat{\Omega},E)dE=\int_g\psi(\vec{r},\hat{\Omega},E)dE \qquad g=1,\ldots,G \qquad (3)$$

$$S_g(\vec{r},\hat{\Omega})=\int_g S(\vec{r},\hat{\Omega},E)dE \qquad g=1,\ldots,G \qquad (4)$$

To obtain these G equations, Eq. (2) is integrated over each energy group g and the following relations are defined for the group cross sections.

$$\Sigma_{t,g}(\vec{r})\phi_g(\vec{r})=\int_g\Sigma_{t,g}(\vec{r},E)\phi(\vec{r},E)dE \qquad (5)$$

$$\Sigma_{s,g'\to g}(\vec{r})\phi_g(\vec{r})=\int_g\int_{g'}\Sigma_s(\vec{r},E'\to E)\phi(\vec{r},E')dE'dE \qquad (6)$$

Using these relationships the multigroup Boltzmann transport equation is given as

$$\hat{\Omega}\cdot\vec{\nabla}\psi_g(\vec{r},\hat{\Omega})+\Sigma_{t,g}(\vec{r})\psi_g(\vec{r},\hat{\Omega})=\sum_{g'=1}^{G}\Sigma_{s,g'\to g}(\vec{r})\phi_{g'}(\vec{r})+S_g(\vec{r},\hat{\Omega})\,. \qquad g=1,\ldots,G \qquad (7)$$

To simplify this expression all but the within group scattering ($g=g'$) can be lumped into the group source thus arriving at the following within group form of the transport equation

$$\hat{\Omega}\cdot\vec{\nabla}\psi_g(\vec{r},\hat{\Omega})+\Sigma_{t,g}(\vec{r})\psi_g(\vec{r},\hat{\Omega})=\Sigma_{s,g\to g}(\vec{r})\phi_g(\vec{r})+S_g(\vec{r},\hat{\Omega})\,. \qquad g=1,\ldots,G \qquad (8)$$

II.2 Second-Order Even-Parity Spherical Harmonics Method

II.2.A Even-Parity Transport Equation

The first step in the spherical harmonics formulation is to transform the within-group equation into an even-parity form. This is accomplished by splitting the angular flux into even- and odd-parity components given by

$$\psi_g(\vec{r},\hat{\Omega}) = \psi_g^+(\vec{r},\hat{\Omega}) + \psi_g^-(\vec{r},\hat{\Omega}), \tag{9}$$

where + denotes even parity and – denotes odd parity. The even and odd parity components of the flux have the following properties, where the function $\phi_g(\vec{r})$ represents the group scalar flux.

Even	Odd
$\psi_g^+(\vec{r},\hat{\Omega}) = \psi_g^+(\vec{r},-\hat{\Omega})$	$\psi_g^-(\vec{r},\hat{\Omega}) = -\psi_g^-(\vec{r},-\hat{\Omega})$
$\int \psi_g^+(\vec{r},\hat{\Omega})d\Omega = \phi_g(\vec{r})$	$\int \psi_g^-(\vec{r},\hat{\Omega})d\Omega = 0$

(10)

Inserting Eq. (9) into Eq. (8) yields the equation

$$\hat{\Omega}\cdot\vec{\nabla}\left[\psi_g^+(\vec{r},\hat{\Omega}) + \psi_g^-(\vec{r},\hat{\Omega})\right] + \Sigma_{t,g}(\vec{r})\left[\psi_g^+(\vec{r},\hat{\Omega}) + \psi_g^-(\vec{r},\hat{\Omega})\right] = \int\left[\Sigma_{s,g\to g}(\vec{r})\left[\psi_g^+(\vec{r},\hat{\Omega}') + \psi_g^-(\vec{r},\hat{\Omega}')\right]d\Omega'\right] + S_g(\vec{r},\hat{\Omega}). \tag{11}$$

Equation (11) can be evaluated at $-\hat{\Omega}$ to get

$$-\hat{\Omega}\cdot\vec{\nabla}\left[\psi_g^+(\vec{r},-\hat{\Omega}) + \psi_g^-(\vec{r},-\hat{\Omega})\right] + \Sigma_{t,g}(\vec{r})\left[\psi_g^+(\vec{r},-\hat{\Omega}) + \psi_g^-(\vec{r},-\hat{\Omega})\right] = \int\left[\Sigma_{s,g\to g}(\vec{r})\left[\psi_g^+(\vec{r},-\hat{\Omega}') + \psi_g^-(\vec{r},-\hat{\Omega}')\right]d\Omega'\right] + S_g(\vec{r},-\hat{\Omega}) \tag{12}$$

Upon adding Eq. (11) to Eq. (12) and using the even and odd parity flux definitions in Eq. (10), the even parity form of the transport equation, given by Eq. (13), is obtained. In a similar fashion, subtraction of Eq. (12) from Eq. (11) will lead to the odd parity form of the transport equation in Eq. (14).

Even Parity

$$2\cdot\hat{\Omega}\cdot\vec{\nabla}\psi_g^-(\vec{r},\hat{\Omega}) + 2\cdot\Sigma_{t,g}(\vec{r})\psi_g^+(\vec{r},\hat{\Omega}) = 2\cdot\Sigma_{s,g\to g}(\vec{r})\phi_g(\vec{r}) + S_g(\vec{r},\hat{\Omega}) + S_g(\vec{r},-\hat{\Omega}) \tag{13}$$

Odd Parity

$$2\cdot\hat{\Omega}\cdot\vec{\nabla}\psi_g^+(\vec{r},\hat{\Omega}) + 2\cdot\Sigma_{t,g}(\vec{r})\psi_g^-(\vec{r},\hat{\Omega}) = S_g(\vec{r},\hat{\Omega}) - S_g(\vec{r},-\hat{\Omega}). \tag{14}$$

To simplify the derivation, we use an isotropic source in addition to an isotropic scattering kernel. This approach simplifies Eqs. (13) and (14) to Eqs. (15) and (16), respectively.

Even Parity

$$\hat{\Omega}\cdot\vec{\nabla}\psi_g^-(\vec{r},\hat{\Omega}) + \Sigma_{t,g}(\vec{r})\psi_g^+(\vec{r},\hat{\Omega}) = \Sigma_{s,g\to g}(\vec{r})\phi_g(\vec{r}) + S_g(\vec{r}) \tag{15}$$

Odd Parity

$$\hat{\Omega}\cdot\vec{\nabla}\psi_g^+(\vec{r},\hat{\Omega}) + \Sigma_{t,g}(\vec{r})\psi_g^-(\vec{r},\hat{\Omega}) = 0. \tag{16}$$

Equation (16) can now be solved for $\psi_g^-(\vec{r},\hat{\Omega})$ and substituted into Eq. (15) to obtain Eq. (17), the second-order form of the even parity transport equation.

$$\hat{\Omega}\cdot\vec{\nabla}\left[\frac{-1}{\Sigma_{t,g}(\vec{r})}\hat{\Omega}\cdot\vec{\nabla}\psi_g^+(\vec{r},\hat{\Omega})\right]+\Sigma_{t,g}(\vec{r})\psi_g^+(\vec{r},\hat{\Omega})=\Sigma_{s,g\to g}(\vec{r})\phi_g(\vec{r})+S_g(\vec{r}) \tag{17}$$

This differential equation is accompanied by boundary conditions that constrain the angular flux along the boundary of the problem domain which are discussed later.

II.2.B Weak Form of the Even-Parity Equation

Temporarily dropping the group notation for simplicity, we weight Eq. (17) with the even parity functions $\lambda^+(\vec{r},\hat{\Omega})$ and integrate over space and angle to obtain the weak form

$$\int dV\int d\Omega\lambda^+(\vec{r},\hat{\Omega})\left[\hat{\Omega}\cdot\vec{\nabla}\left[\frac{-1}{\Sigma_t(\vec{r})}\hat{\Omega}\cdot\vec{\nabla}\psi^+(\vec{r},\hat{\Omega})\right]+\Sigma_t(\vec{r})\psi^+(\vec{r},\hat{\Omega})-\Sigma_s(\vec{r})\phi(\vec{r})\right]= \tag{18}$$
$$\int dV\int d\Omega\lambda^+(\vec{r},\hat{\Omega})S(\vec{r})$$

where the integration is over the volume of the domain. Applying the divergence theorem, we obtain

$$\int\int\left[\hat{\Omega}\cdot\vec{\nabla}\lambda^+(\vec{r},\hat{\Omega})\frac{1}{\Sigma_t(\vec{r})}\hat{\Omega}\cdot\vec{\nabla}\psi^+(\vec{r},\hat{\Omega})+\lambda^+(\vec{r},\hat{\Omega})\Sigma_t(\vec{r})\psi^+(\vec{r},\hat{\Omega})\right]d\Omega dV$$
$$=\int\int\left[\lambda^+(\vec{r},\hat{\Omega})\Sigma_s(\vec{r})\phi_g(\vec{r})+\lambda^+(\vec{r},\hat{\Omega})S(\vec{r})\right]d\Omega dV \qquad , \tag{19}$$
$$-\oint\int\hat{\Omega}\cdot\hat{n}\,\lambda^+(\vec{r},\hat{\Omega})\psi^-(\vec{r},\hat{\Omega})d\Omega d\Gamma$$

where $\hat{n}$ is the outward normal from the domain surface, Γ. The incoming angular flux, $\psi_\Gamma(\vec{r},\hat{\Omega})$, is known on the domain surface and can be expressed for each group as follows.

$$\psi_\Gamma(\vec{r},\hat{\Omega})=\psi_\Gamma^+(\vec{r},\hat{\Omega})+\psi_\Gamma^-(\vec{r},\hat{\Omega}) \qquad \vec{r}\in\Gamma,\ \hat{\Omega}\cdot\hat{n}<0 \tag{20}$$

Modified natural boundary conditions can be obtained by forming weighted residuals

$$\oint d\Gamma\int\limits_{\hat{\Omega}\cdot\hat{n}<0} d\Omega\,\lambda^+(\vec{r},\hat{\Omega})\left(\psi_\Gamma^+(\vec{r},\hat{\Omega})+\psi_\Gamma^-(\vec{r},\hat{\Omega})-\psi_\Gamma(\vec{r},\hat{\Omega})\right)=0 \tag{21}$$

Using angular parity arguments, we take the boundary term from Eq. (19) and write

$$\oint d\Gamma\int d\Omega\,\hat{\Omega}\cdot\hat{n}\,\lambda^+(\vec{r},\hat{\Omega})\psi_\Gamma^-(\vec{r},\hat{\Omega})=$$
$$\oint d\Gamma\int d\Omega\left|\hat{\Omega}\cdot\hat{n}\right|\lambda^+(\vec{r},\hat{\Omega})\psi_\Gamma^+(\vec{r},\hat{\Omega})+2\oint d\Gamma\int\limits_{\hat{\Omega}\cdot\hat{n}<0} d\Omega\,\hat{\Omega}\cdot\hat{n}\,\lambda^+(\vec{r},\hat{\Omega})\psi_\Gamma(\vec{r},\hat{\Omega}) \tag{22}$$

Substituting Eq. (22) into Eq. (19), we obtain the primal form of the second-order equation with modified natural boundary conditions:

$$\int\int\left[\hat{\Omega}\cdot\vec{\nabla}\lambda^+\Sigma_t^{-1}\hat{\Omega}\cdot\vec{\nabla}\psi^+ +\lambda^+\Sigma_t\psi^+-\lambda^+\Sigma_s\phi\right]d\Omega dV$$
$$=\int\int\lambda^+Sd\Omega dV-\oint d\Gamma\int d\Omega\left|\hat{\Omega}\cdot\hat{n}\right|\lambda^+\psi^+-2\oint d\Gamma\int\limits_{\hat{\Omega}\cdot\hat{n}<0} d\Omega\,\hat{\Omega}\cdot\hat{n}\,\lambda^+\psi_\Gamma \tag{23}$$

Note that the space, angle, and group information in Eq. (23) was dropped for simplification purposes. The boundary even-parity flux was also assumed to be solved for using the even-parity approximation internal to the domain.

Given that we can solve Eq. (23) for $\psi^+(\vec{r},\hat{\Omega})$ using S and the incident angular flux ψ_Γ, then, in the weak sense, we can construct the outgoing angular flux, ψ_Γ, using

$$\oint d\Gamma \int_{\hat{\Omega}\cdot\hat{n}>0} d\Omega\, \hat{\Omega}\cdot\hat{n}\, \lambda^+\left(\psi_\Gamma^+ + \psi_\Gamma^- - \psi_\Gamma\right) = 0\,. \tag{24}$$

In a diffusion approximation where $\psi^+(\vec{r},\hat{\Omega}) \to \phi(\vec{r})$ and $\lambda^+(\vec{r},\hat{\Omega}) \to \lambda(\vec{r})$, and $\vec{J}_\Gamma(\vec{r}) = \int d\Omega\hat{\Omega}\psi_\Gamma^-(\vec{r},\hat{\Omega})$, Eq. (21) reduces to

$$\oint d\Gamma \lambda(\vec{r})\left[\tfrac{1}{4}\phi_\Gamma(\vec{r}) - \tfrac{1}{2}\hat{n}\cdot\vec{J}_\Gamma(\vec{r}) - \int_{\hat{\Omega}\cdot\hat{n}<0} d\Omega\, \hat{\Omega}\cdot\hat{n}\, \psi_\Gamma(\vec{r},\hat{\Omega})\right] = 0\,, \tag{25}$$

where $\tfrac{1}{4}\phi_\Gamma(\vec{r}) - \tfrac{1}{2}\hat{n}\cdot\vec{J}_\Gamma(\vec{r})$ is the classical form of the incoming partial current J^- assuming Rumyantsev interface conditions are imposed. Similarly, Eq. (24) reduces to

$$\oint d\Gamma \lambda(\vec{r})\left[\tfrac{1}{4}\phi_\Gamma(\vec{r}) + \tfrac{1}{2}\hat{n}\cdot\vec{J}_\Gamma(\vec{r}) - \int_{\hat{\Omega}\cdot\hat{n}>0} d\Omega\, \hat{\Omega}\cdot\hat{n}\, \psi_\Gamma(\vec{r},\hat{\Omega})\right] = 0\,, \tag{26}$$

where $\tfrac{1}{4}\phi_\Gamma(\vec{r}) + \tfrac{1}{2}\hat{n}\cdot\vec{J}_\Gamma(\vec{r})$ is the classical definition of the outgoing partial current J^+ where Rumyantsev interface conditions are again assumed. Later, the splitting of the problem domain into multiple subdomains is discussed, and the relevance of Eqs. (25) and (26) will become self-evident.

II.2.C Discretization of the Even-Parity Transport Equation

The spherical harmonic functions are defined as

$$Y_\ell^m(\hat{\Omega}) = \sqrt{C_\ell^m}\cdot P_\ell^m(\mu)\cdot \mathrm{e}^{im\phi}, \tag{27}$$

where $P_\ell^m(\mu)$ represents the associated Legendre function ($\mu = \cos(\theta)$) and the normalization constant is given by

$$C_\ell^m = \frac{(2\ell+1)(\ell-m)!}{(\ell+m)!}. \tag{28}$$

The spherical harmonic functions can be separated into sine and cosine series.

$$Y_\ell^m(\hat{\Omega}) = Y_{\ell,m}^e(\hat{\Omega}) + i\, Y_{\ell,m}^o(\hat{\Omega}) \tag{29}$$

$$Y_{\ell,m}^e(\hat{\Omega}) = \sqrt{C_\ell^m}\cdot P_\ell^m(\mu)\cdot\cos(m\phi) \tag{30}$$

$$Y_{\ell,m}^o(\hat{\Omega}) = \sqrt{C_\ell^m}\cdot P_\ell^m(\mu)\cdot\sin(m\phi) \tag{31}$$

Equations (30) and (31) are unique in that the functions $Y_{\ell,m}^e(\hat{\Omega})$ are even parity with respect to ϕ and the functions $Y_{\ell,m}^o(\hat{\Omega})$ are odd parity with respect to ϕ.

The governing equation for the finite-element domain is given earlier as Eq. (23), and the even parity flux $\psi_g^+(\vec{r},\hat{\Omega})$ and weight function $\lambda^+(\vec{r},\hat{\Omega})$ can be approximated with a set of even parity angular functions to obtain

$$\psi_g^+(\vec{r},\hat{\Omega}) = Y_+^T(\hat{\Omega})\psi^+(\vec{r}) \tag{32}$$

$$\lambda^+(\vec{r},\hat{\Omega}) = Y_+(\hat{\Omega})\lambda^+(\vec{r}) \tag{33}$$

where the set of even-parity functions $Y_+^T(\hat{\Omega})$, are given as

$$Y_+^T(\hat{\Omega}) = \left[Y_{0,m}^e(\hat{\Omega}) \quad Y_{2,m}^e(\hat{\Omega}) \quad Y_{2,m}^o(\hat{\Omega}) \quad \cdots \quad Y_{N-1,m}^e(\hat{\Omega}) \quad Y_{N-1,m}^o(\hat{\Omega})\right]. \quad |m| = 0,1,2,\ldots,l \tag{34}$$

Note that the even-parity expansion has the series property $l = 0,2,\ldots,N-1$ where N is odd, thereby selecting the even-parity set of functions with respect to μ from Eqs. (30) and (31).

We next inspect the spatial approximation, designated as the finite element method. With this method, the problem domain is subdivided by using a finite element mesh as seen in Figure 2. Each finite element in the mesh is defined through the use of spatial vertex points (typically called nodes in the finite element literature).

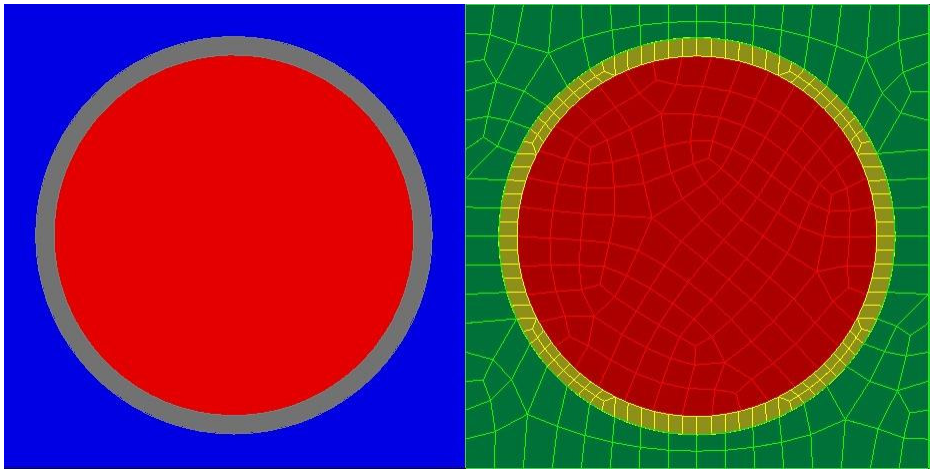

Figure 2. Finite-element mesh representation of a pin-cell geometry.

Mathematically, this subdivision is applied to Eq. (19), reproduced here as Eq. (35):

$$\begin{aligned} \mathrm{T}\left[\psi_g^+,\psi_g^-\right] = \int\int\left[\hat{\Omega}\cdot\vec{\nabla}\lambda^+\frac{1}{\Sigma_{t,g}}\hat{\Omega}\cdot\vec{\nabla}\psi_g^+ + \lambda^+\Sigma_{t,g}\psi_g^+\right]d\Omega dV \\ -\int\int\left[\lambda^+\Sigma_{s,g\to g}\phi_g + \lambda^+ S_g\right]d\Omega dV + \oint\int\hat{\Omega}\cdot\hat{n}\,\lambda^+\psi_g^- d\Omega d\Gamma \end{aligned}, \tag{35}$$

yielding the elemental equation

$$\mathrm{T}_e\left[\psi^+_{g,e},\psi^-_{g,e}\right]=\int\int\left[\hat{\Omega}\cdot\vec{\nabla}\lambda^+\frac{1}{\Sigma^e_{t,g}}\hat{\Omega}\cdot\vec{\nabla}\psi^+_{g,e}+\lambda^+\Sigma^e_{t,g}\psi^+_{g,e}\right]d\Omega dV_e \\ -\int\int\left[\lambda^+\Sigma^e_{s,g\to g}\phi_{g,e}+\lambda^+S_{g,e}\right]d\Omega dV_e+\int\int\hat{\Omega}\cdot\hat{n}\;\lambda^+\psi^-_{g,e}d\Omega d\Gamma_\gamma \quad . \tag{36}$$

and the relation between 35 and 36 of

$$\mathrm{T}\left[\psi^+_g,\psi^-_g\right]=\sum_e\mathrm{T}_e\left[\psi^+_{g,e},\psi^-_{g,e}\right]. \tag{37}$$

Note that in Eq. (36), the spatial integration is carried out over each element rather than the entire problem domain and that the surface integral has been split into a sum of integrals over each element surface Γ_γ.

The combined space-angle approximation for the even-parity flux and weight functions becomes

$$\psi^+_{g,e}(\vec{r},\hat{\Omega})=Y^T_+(\hat{\Omega})\otimes L^T_e(\vec{r})\;\tilde{\psi}^+_{g,e} \tag{38}$$

$$\lambda^+_e(\vec{r},\hat{\Omega})=Y_+(\hat{\Omega})\otimes L_e(\vec{r}) \tag{39}$$

where $\otimes$ represents a tensor product of the spatial and angular vector of trial functions. The same spatial approximation is used for the source and scalar flux, leading to

$$S_{g,e}(\vec{r})=L^T_e(\vec{r})\;\tilde{S}_{g,e}\quad, \tag{40}$$

and

$$\phi_{g,e}(\vec{r})=L^T_e(\vec{r})\;\tilde{\phi}_{g,e} \\ =\int d\Omega\;\psi^+_{g,e}(\vec{r},\hat{\Omega})=\int d\Omega\;Y^T_+(\hat{\Omega})\otimes L^T_e(\vec{r})\;\tilde{\psi}^+_{g,e}=J^T_+\otimes L^T_e(\vec{r})\;\tilde{\psi}^+_{g,e}\quad . \tag{41}$$

Inserting Eqs. (38)–(40) into Eq. (36) leads to the following matrix equation

$$\mathrm{T}_e\left[\tilde{\psi}^+_{g,e},\psi^-_{g,e}\right]=\left[H_{K,L}\otimes\frac{1}{\Sigma^e_{t,g}}P^e_{K,L}+I_+\otimes\Sigma^e_{t,g}F^e-J_+J^T_+\otimes\Sigma^e_{s,g\to g}F^e\right]\tilde{\psi}^+_{g,e} \\ -J_+\otimes F^e\tilde{S}_{g,e}+\int\int\hat{\Omega}\cdot\hat{n}\;\lambda^+\psi^-_{g,e}d\Omega d\Gamma_\gamma \quad . \tag{42}$$

The individual angular matrices are given as

$$H_{K,L}=\int\hat{\Omega}_K\hat{\Omega}_LY_+(\hat{\Omega})Y^T_+(\hat{\Omega})d\Omega\,, \tag{43}$$

$$I_+=\int Y_+(\hat{\Omega})Y^T_+(\hat{\Omega})d\Omega\,, \tag{44}$$

and

$$J_+=\int Y_+(\hat{\Omega})d\Omega\,, \tag{45}$$

while the spatial matrices are given as

$$P^e_{K,L}=\int\vec{\nabla}_KL_e(\vec{r})\vec{\nabla}_LL^T_e(\vec{r})dV_e\,, \tag{46}$$

and

$$F^e=\int L_e(\vec{r})L_e(\vec{r})dV_e\,. \tag{47}$$

Equation (42) can be simplified by defining the matrices and vectors

$$A_{g,e} = H_{K,L} \otimes \frac{1}{\Sigma^e_{t,g}} P^e_{K,L} + I_+ \otimes \Sigma^e_{t,g} F^e - J_+ J_+^T \otimes \Sigma^e_{s,g\to g} F^e \,. \tag{48}$$

$$\tilde{s}_{g,e} = J_+ \otimes F^e \tilde{S}_{g,e} \,. \tag{49}$$

to obtain

$$\mathrm{T}_e\left[\psi^+_{g,e}, \psi^-_{g,e}\right] = A_{g,e}\tilde{\psi}^+_{g,e} - \tilde{s}_{g,e} + \int\int \hat{\Omega}\cdot\hat{n}\ \lambda^+_e(\vec{r},\hat{\Omega})\ \psi^-_{g,e}(\vec{r},\hat{\Omega}) d\Omega d\Gamma_\gamma \,. \tag{50}$$

II.3 Method of Characteristics

II.3.A Integral Form of the Transport Equation

Similar to the spherical harmonics formulation, the within-group transport equation can be used as the starting point of the characteristics formulation and is reproduced in Eq. (51).

$$\hat{\Omega}\cdot\vec{\nabla}\psi_g(\vec{r},\hat{\Omega}) + \Sigma_{t,g}(\vec{r})\psi_g(\vec{r},\hat{\Omega}) = \Sigma_{s,g\to g}(\vec{r})\phi_g(\vec{r}) + S_g(\vec{r},\hat{\Omega}) \,. \qquad g = 1,\ldots,G \tag{51}$$

For simplicity, the external source is assumed to be isotropic, and the total source for each group is merged into the emission density $q_g(\vec{r})$:

$$\begin{aligned} &\hat{\Omega}\cdot\vec{\nabla}\psi_g(\vec{r},\hat{\Omega}) + \Sigma_{t,g}(\vec{r})\psi_g(\vec{r},\hat{\Omega}) = q_g(\vec{r}) \\ &q_g(\vec{r}) = \Sigma_{s,g\to g}(\vec{r})\phi_g(\vec{r}) + S_g(\vec{r}) \end{aligned} \,. \tag{52}$$

To obtain the integral form of the transport equation, we recognize that the streaming operator $(\hat{\Omega}\cdot\vec{\nabla})$ is the directional derivative along the direction of neutron travel, allowing Eq. (52) to be written as

$$\frac{d}{du}\psi_g(\vec{r}+u\hat{\Omega},\hat{\Omega}) + \Sigma_{t,g}(\vec{r}+u\hat{\Omega})\psi_g(\vec{r}+u\hat{\Omega},\hat{\Omega}) = q_g(\vec{r}+u\hat{\Omega}) \,, \tag{53}$$

where u is the direction along the line of neutron travel as shown in Figure 3. We can reverse the direction such that we consider neutrons traveling to the point of interest, thereby obtaining

$$-\frac{d}{dR}\psi_g(\vec{r}-R\hat{\Omega},\hat{\Omega}) + \Sigma_{t,g}(\vec{r}-R\hat{\Omega})\psi_g(\vec{r}-R\hat{\Omega},\hat{\Omega}) = q_g(\vec{r}-R\hat{\Omega}) \tag{54}$$

where *u* and *R* are equal in magnitude but the sign is different.

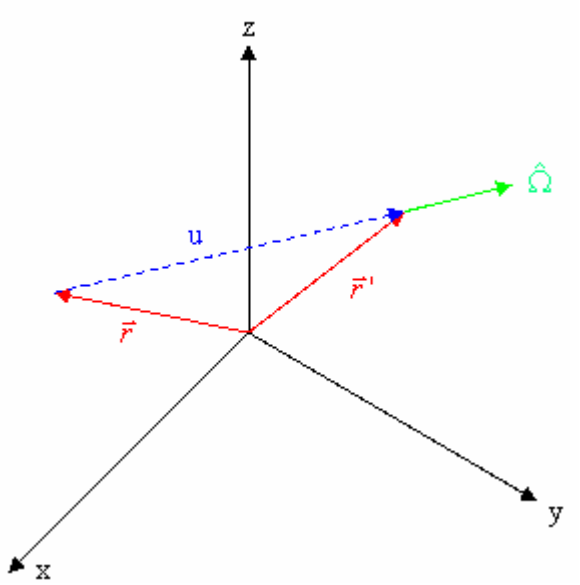

Figure 3. Diagram of the neutron travel along the direction $\hat{\Omega}$.

The derivative with respect to R can be removed by using the integrating factor

$$\exp\left[-\int_0^R \Sigma_{t,g}(\vec{r}-R'\hat{\Omega})dR'\right]. \tag{55}$$

yielding

$$\begin{aligned}\psi_g(\vec{r},\hat{\Omega}) = &\int_0^R q_g(\vec{r}-R'\hat{\Omega})\exp\left[-\int_0^{R'}\Sigma_{t,g}(\vec{r}-R''\hat{\Omega})dR''\right]dR' \\ &+\psi_g(\vec{r}-R\hat{\Omega},\hat{\Omega})\exp\left[-\int_0^R \Sigma_{t,g}(\vec{r}-R''\hat{\Omega})dR''\right]\end{aligned}. \tag{56}$$

Equation (56) gives the solution of the flux at a given point in space as a function of the source contribution over the preceding distance R and the uncollided flux at distance R along the direction $\hat{\Omega}$. Using the finite element domain decomposition, we find that the cross sections within each finite element are constant. With a spatial finite-element decomposition, the angular discretization of Eq. (56) involves drawing rays that track through the domain from boundary to boundary, as seen in Figure 4. The discrete angular directions are chosen such that the unit sphere is "sufficiently" discretized. The spatial distribution of the rays is chosen by defining a spatial quadrature on the boundary surface of the domain.

In standard characteristics an equally spaced Simpson's rule or a back projection is used. The back projection approach, in which the surfaces of each element are projected to a normal plane for a given angular direction, is generally more accurate because it allows one to spatially place the rays such that at least one ray from the boundary will pass through every finite element in the domain. In most implementations, including the one

shown here, the angular flux along each of these rays is assumed to be constant in the direction perpendicular to the ray, and thus the area, A_i, is assigned to each ray leading to a piecewise constant approximation in the normal direction. Under these assumptions, the solution of the transport equation is exact along the direction of the ray and takes the form shown in Eq. (56).

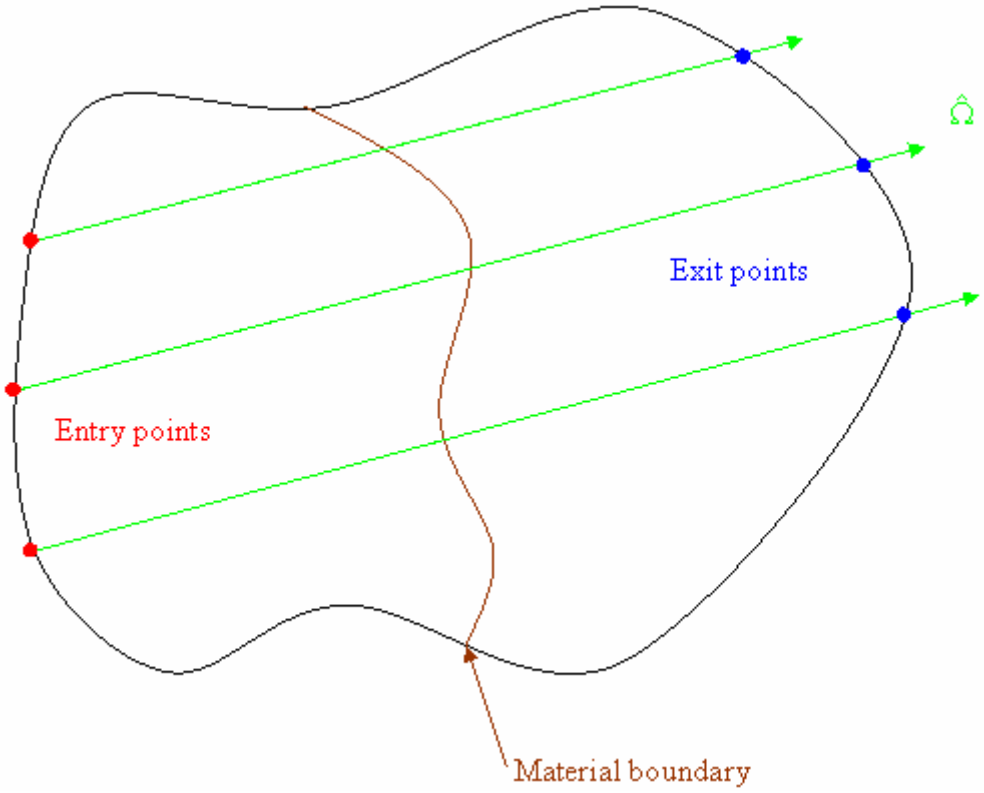

Figure 4. Example angular ray propagation through the domain.

Equation (56) can be reduced to Eq. (57) in terms of the flux at the entry point, $\psi_{g,i}^{IN}(\hat{\Omega}_n)$, and the exit point, $\psi_{g,i}^{OUT}(\hat{\Omega}_n)$, along a fixed trajectory, on the boundary of an element *e* as shown in Figure 5. This is commonly termed the propagation equation, referring to the way in which the incoming boundary flux is propagated through the domain to the exiting boundary surface for a given angular direction.

$$\psi_{g,i}^{OUT}(\hat{\Omega}_n) = \int_0^{R_{i,e}} q_g(\vec{r} - R'\hat{\Omega}) \exp\left[-\Sigma_{t,g}^e R'\right] dR' + \psi_{g,i}^{IN}(\hat{\Omega}_n) \exp\left[-\Sigma_{t,g}^e R_{i,e}\right] \tag{57}$$

If the source is assumed to be flat within each finite element (a typical approach in the method of characteristics), Eq. (57) reduces to

$$\psi_{g,i}^{OUT}(\hat{\Omega}_n) = \frac{q_g^e}{\Sigma_{t,g}^e}\left[1 - \exp\left[-\Sigma_{t,g}^e R_{i,e}\right]\right] + \psi_{g,i}^{IN}(\hat{\Omega}_n) \exp\left[-\Sigma_{t,g}^e R_{i,e}\right]. \tag{58}$$

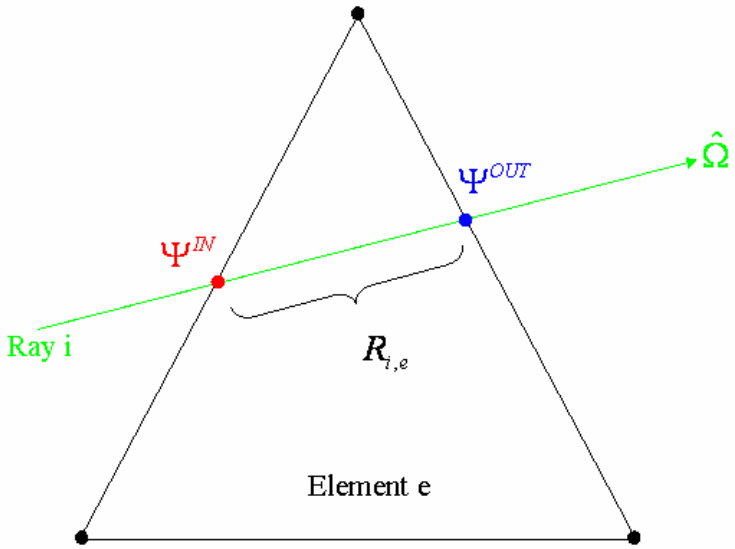

Figure 5. Ray propagation through a triangular finite element.

For each finite element one can construct a balance over the volume for each direction to obtain

$$\bar{\psi}_g^e(\hat{\Omega}_n) = \frac{q_g^e}{\Sigma_{t,g}^e} - \frac{1}{V_e \Sigma_{t,g}^e} \sum_{i \in e} A_i \left[\psi_{g,i}^{OUT}(\hat{\Omega}_n) - \psi_{g,i}^{IN}(\hat{\Omega}_n) \right]. \tag{59}$$

where V_e is the volume of the element, A_i is the area assigned to each ray, and $\bar{\psi}_g^e(\hat{\Omega}_n)$ is the element-averaged angular flux. We note that the sum encompasses all those rays with direction $\hat{\Omega}_n$ that pass through the element *e*.

Using Eq. (59), we obtain the angular flux moments within each element such that the flat source moments are reconstructed. Given that isotropic sources were used in this derivation, the scalar flux equation given by Eq. (60) is obtained.

$$\bar{\phi}_g^e = \sum_n \bar{\psi}_g^e(\hat{\Omega}_n) w_n = \frac{q_g^e}{\Sigma_{t,g}^e} - \sum_n \frac{w_n}{V_e \Sigma_{t,g}^e} \sum_{i \in e} A_i \left[\psi_{g,i}^{OUT}(\hat{\Omega}_n) - \psi_{g,i}^{IN}(\hat{\Omega}_n) \right]. \tag{60}$$

where w_n are the weights of the angular quadrature. When this product is obtained, the new source within each element can be constructed, and the source iteration procedure is straightforward.

The solution of the system of equations is quite different from other neutron transport methods in that the inner iteration does not directly involve the solution of the flux on an element-by-element basis. Instead, the incident flux along the boundary of the domain is propagated through the entire domain along individual rays in computational sweeps that are independent of each other (the solution is independent of the ordering of the rays).

The contribution to each element is accumulated from each ray as each ray is processed. Only after all of the rays in all directions have been processed can the next outer iteration be performed. This approach has significant parallel advantages, although we note that the construction of the new iteration source still requires extensive communication. The primary expense of this method is the ray tracing, which must be performed up front. Moreover, given that the projection of the boundary of curvilinear elements is very expensive, one must be careful to pick a scheme that will pick starting ray positions such that each element is crossed by at least one ray in every direction. A form of this methodology is currently being tested for implementation in the UNIC framework.

II.4 Neutron Transport Equation Multisolution Coupling Strategy

From the second-order spherical harmonics derivation, Eq. (50), the odd-parity flux along the boundary was not defined. Similarly, the angular flux along the boundary of the problem domain in the method of characteristics was also not defined. In this section we first assume a single problem domain that has been split into multiple pieces, termed subdomains, each having a closed surface Γ_D. These subdomains are coupled together by using a generic interface condition. In this work, two general interface conditions are considered, and thus coupling with the second-order even-parity method is investigated. In the first, the adjacent domain is assumed to be approximated with a spherical harmonic method, and thus a spherical harmonic interface condition must be used. In the second, the adjacent subdomain is assumed to have a method of characteristics approach, and thus a different interface condition must be used.

II.4.A Adjoining Spherical Harmonic Subdomains

For the situation where the adjoining subdomains are both treated with the spherical harmonics approximation, we begin by approximating the angular interface flux with

$$\psi^{+}_{g,\Gamma}(\vec{r},\hat{\Omega}) = Y^{T}_{+,\gamma}(\hat{\Omega}) \otimes h^{T}_{\gamma}(\vec{r})\, \tilde{\psi}^{+}_{g,\Gamma}\,. \qquad \vec{r} \in \Gamma_{D,\gamma} \qquad (61)$$

and

$$\psi^{-}_{g,\Gamma}(\vec{r},\hat{\Omega}) = Y^{T}_{-,\gamma}(\hat{\Omega}) \otimes h^{T}_{\gamma}(\vec{r})\, \tilde{\psi}^{-}_{g,\Gamma}\,. \qquad \vec{r} \in \Gamma_{D,\gamma} \qquad (62)$$

Here, $h_{\gamma}(\vec{r})$ is a vector of spatial trial functions defined only on the interface $\Gamma_{D,\gamma}$, which in their lowest form are 0^{th} order for each finite element. $Y^{T}_{+,\gamma}(\hat{\Omega})$ and $Y^{T}_{-,\gamma}(\hat{\Omega})$ are vectors of even- and odd-parity spherical harmonics aligned such that the polar axis is perpendicular to Γ_{γ}. The set of odd-parity functions is further truncated such that they meet Rumyantsev conditions along the subdomain boundary. Together these functions can be represented by using

$$Y_{+,\gamma}(\hat{\Omega}) = \boldsymbol{\mathcal{L}}_{+,\gamma} Y_{+}(\hat{\Omega})\,, \qquad (63)$$

$$Y_{-,\gamma}(\hat{\Omega}) = \Lambda_{-,\gamma} \boldsymbol{\mathcal{L}}_{-,\gamma} Y_{-}(\hat{\Omega})\,, \qquad (64)$$

where $\boldsymbol{\mathcal{L}}_{\pm,\gamma}$ is a rotation matrix to align the set of spherical harmonics with the outward normal and $\Lambda_{-,\gamma}$ is a truncation matrix to form the set of valid Rumyantsev functions.

To use these expressions, we expand Eq. (50), reproduced here as Eq. (65), where the incoming angular flux has been expanded in terms of the even- and odd-parity flux components and the boundary even-parity flux was assumed to be solved for as done before.

$$\begin{aligned}\mathrm{T}_e\left[\psi^+_{g,e},\psi^+_{g,\Gamma},\psi^-_{g,\Gamma}\right] &= A_{g,e}\tilde{\psi}^+_{g,e}-\tilde{s}_{g,e}\\ &+2\int\limits_{\hat{\Omega}\cdot\hat{n}<0}\hat{\Omega}\cdot\hat{n}\;\lambda^+_e(\vec{r},\hat{\Omega})\psi^+_{g,\Gamma}(\vec{r},\hat{\Omega})d\Omega d\Gamma_\gamma\,,\\ &+2\int\limits_{\hat{\Omega}\cdot\hat{n}<0}\hat{\Omega}\cdot\hat{n}\;\lambda^+_e(\vec{r},\hat{\Omega})\psi^-_{g,\Gamma}(\vec{r},\hat{\Omega})d\Omega d\Gamma_\gamma\end{aligned} \tag{65}$$

Substituting in the expressions from Eqs. (61) and (62), we obtain the following two expressions for the undiscretized boundary expressions in Eq. (65).

$$\begin{aligned}&\int\limits_{\hat{\Omega}\cdot\hat{n}<0}\hat{\Omega}\cdot\hat{n}\;Y_+(\hat{\Omega})\otimes L_e(\vec{r})Y^T_+(\hat{\Omega})\boldsymbol{\mathcal{L}}^T_{+,\gamma}\otimes h^T_\gamma(\vec{r})\;\tilde{\psi}^+_{g,\Gamma}d\Omega d\Gamma_\gamma = L_\gamma\otimes D_\gamma\;\tilde{\psi}^+_{g,\Gamma}\\ &\int\limits_{\hat{\Omega}\cdot\hat{n}<0}\hat{\Omega}\cdot\hat{n}\;Y_+(\hat{\Omega})\otimes L_e(\vec{r})Y^T_-(\hat{\Omega})\boldsymbol{\mathcal{L}}^T_{-,\gamma}\Lambda^T_{-,\gamma}\otimes h^T_\gamma(\vec{r})\;\tilde{\psi}^-_{g,\Gamma}d\Omega d\Gamma_\gamma = -K_\gamma\otimes D_\gamma\tilde{\psi}^-_{g,\Gamma}\end{aligned} \tag{66}$$

Defining both of these variables separately is a general mathematical violation because it effectively states that both the incoming and outgoing angular flux are explicitly known. As a result, we must define them as a single unknown.

$$j^-_{g,\Gamma} = L_\gamma\otimes I_\gamma\;\tilde{\psi}^+_{g,\Gamma}-K_\gamma\otimes I_\gamma\tilde{\psi}^-_{g,\Gamma} \tag{67}$$

For the remainder of the continuity conditions we project the angular finite-element flux solution to the boundary to obtain

$$j^+_{g,\Gamma} = L_\gamma\otimes I_\gamma\;\tilde{\psi}^+_{g,\Gamma}+K_\gamma\otimes I_\gamma\tilde{\psi}^-_{g,\Gamma}\,. \tag{68}$$

Across the interface between adjoining subdomains, the outgoing "current" for one subdomain defined by Eq. (68) becomes the incoming "current" defined by Eq. (67) for the adjoining subdomain. The reverse is also true, thereby leading to an iterative solution process.

The even-parity component of Eq. (68) is obtained by directly projecting the even-parity flux defined earlier in Eq. (38). We then have

$$\psi^+_{g,e}(\vec{r},\hat{\Omega}) = Y^T_+(\hat{\Omega})\otimes L^T_e(\vec{r})\;\tilde{\psi}^+_{g,e}\;. \tag{69}$$

Projecting to the same spatial trial functions along the interface for the even parity variables, we obtain the integral expression

$$\begin{aligned}&\int\limits_{\hat{\Omega}\cdot\hat{n}>0}\hat{\Omega}\cdot\hat{n}\;\boldsymbol{\mathcal{L}}_{+,\gamma}Y_+(\hat{\Omega})\otimes h_\gamma(\vec{r})\psi^+_{g,e}(\vec{r},\hat{\Omega}) =\\ &\int\limits_{\hat{\Omega}\cdot\hat{n}>0}\hat{\Omega}\cdot\hat{n}\;\boldsymbol{\mathcal{L}}_{+,\gamma}Y_+(\hat{\Omega})\otimes h_\gamma(\vec{r})Y^T_+(\hat{\Omega})\otimes L^T_e(\vec{r})\;\tilde{\psi}^+_{g,e}d\Omega d\Gamma_\gamma = L^T_\gamma\otimes D^T_\gamma\;\tilde{\psi}^+_{g,e}\end{aligned}\;, \tag{70}$$

For the odd-parity component of Eq. (68), the odd-parity transport equation of Eq. (16) must be used:

$$\psi^-_{g,e}(\vec{r},\hat{\Omega}) = -\frac{1}{\Sigma^e_{t,g}}\hat{\Omega}\cdot\vec{\nabla}\psi^+_{g,e}(\vec{r},\hat{\Omega})\,. \tag{71}$$

Equation (71) is valid everywhere in the domain, including along the boundary separating subdomains. Thus, there are several ways to employ this equation to obtain the targeted

solution. One straightforward approach is to simply solve for the od-parity flux everywhere in the domain. This approach uses the same spatial representation as that used for the even-parity flux (continuous finite element), and the projection to the boundary is also be the same. However, this approach results in the following elemental expression that, after element assembly, requires the inversion of the left-hand side:

$$I_- \otimes F^e \ \psi_{g,e}^- = -V_K \otimes \frac{1}{\Sigma_{t,g}^e} U_K^e \ \psi_{g,e}^+ . \tag{72}$$

This inversion is similar to that required to solve the even-parity flux, which is quite expensive and thus inadvisable.

Another approach is to implement a discontinuous spatial representation for the odd-parity flux, which is obtained only in those elements along the boundary of the domain (i.e., where it is needed). In this approach the spatial representation is flexible, and thus either the finite element approximation or a set of polynomial trial functions (not necessarily orthonormal) can be used, expressed as follows.

$$g_e^T(\vec{r}) = \begin{cases} \begin{bmatrix} 1 & x & y & z & xy & yz & xz & \ldots \end{bmatrix} & \textit{polynomial} \\ L_e^T(\vec{r}) & \textit{finite element} \end{cases} . \tag{73}$$

For the polynomial representation, a spatial expansion order is chosen such that it is sufficient for the surface projection. The total odd-parity approximation within each element is given by

$$\psi_{g,e}^-(\vec{r},\hat{\Omega}) = Y_-^T(\hat{\Omega}) \otimes g_e^T(\vec{r}) \ \tilde{\psi}_{g,e}^- . \tag{74}$$

Implementing Eqs. (74) and (38) into Eq. (71) results in the following local matrix relation

$$I_- \otimes O_e \ \tilde{\psi}_{g,e}^- = -V_K \otimes \frac{1}{\Sigma_{t,g}^e} N_K^e \ \tilde{\psi}_{g,e}^+ . \tag{75}$$

The new matrices are defined as

$$I_- = \int Y_-(\hat{\Omega}) Y_-^T(\hat{\Omega}) d\Omega , \tag{76}$$

$$V_K = \int \hat{\Omega}_K Y_-(\hat{\Omega}) Y_+^T(\hat{\Omega}) d\Omega , \tag{77}$$

$$f_e = \int g_e(\vec{r}) g_e^T(\vec{r}) dV_e , \tag{78}$$

$$u_K^e = \int g_e(\vec{r}) \vec{\nabla}_K L_e^T(\vec{r}) dV_e . \tag{79}$$

The matrix defined by Eq. (78) is relatively easy to invert and leads to the matrix relation for the odd-parity internal distribution of

$$\tilde{\psi}_{g,e}^- = -V_K \otimes \frac{1}{\Sigma_{t,g}^e} f_e^{-1} u_K^e \ \tilde{\psi}_{g,e}^+ . \tag{80}$$

The relationship between the odd-parity boundary flux and internal element odd-parity flux is given by

$$\psi_{g,\Gamma}^-(\vec{r},\hat{\Omega}) = \psi_{g,e}^-(\vec{r},\hat{\Omega}) . \qquad \vec{r} \in \Gamma_\gamma \tag{81}$$

Weighting Eq. (81) with the boundary odd-parity flux approximation yields the identity

$$\int\int \Lambda_{-,\gamma} \mathcal{L}_{-,\gamma} Y_-(\hat{\Omega}) \otimes h_\gamma^T(\vec{r}) \left[\psi_{g,\Gamma}^-(\vec{r},\hat{\Omega}) - \psi_{g,e}^-(\vec{r},\hat{\Omega}) \right] d\Omega d\Gamma_\gamma = 0 , \qquad \vec{r} \in \Gamma_\gamma \tag{82}$$

which simplifies to

$$\Lambda_{-,\gamma}\Lambda_{-,\gamma}^{T} \otimes I_{\gamma}\ \tilde{\psi}_{g,\Gamma}^{-} = \Lambda_{-,\gamma}\boldsymbol{\mathcal{L}}_{-,\gamma} \otimes d_{\gamma}^{T}\ \tilde{\psi}_{g,e}^{-}, \tag{83}$$

where the new matrix is defined as

$$d_{\gamma}^{T} = \int h_{\gamma}(\vec{r}) g_{e}^{T}(\vec{r})\, d\Gamma_{\gamma}\,. \tag{84}$$

Solving Eq. (83) for the odd parity coefficients, we have

$$\tilde{\psi}_{g,\Gamma}^{-} = \left(\Lambda_{-,\gamma}\Lambda_{-,\gamma}^{T}\right)^{-1} \Lambda_{-,\gamma}\boldsymbol{\mathcal{L}}_{-,\gamma} \otimes d_{\gamma}^{T}\ \tilde{\psi}_{g,e}^{-}, \tag{85}$$

which along with Eq. (80) can be substituted into Eq. (68) to obtain

$$j_{g,\Gamma}^{+} = \left[L_{\gamma}^{T} \otimes D_{\gamma}^{T} - K_{\gamma}\left(\Lambda_{-,\gamma}\Lambda_{-,\gamma}^{T}\right)^{-1} \Lambda_{-,\gamma}\boldsymbol{\mathcal{L}}_{-,\gamma} V_{K} \otimes d_{\gamma}^{T} \frac{1}{\Sigma_{t,g}^{e}} f_{e}^{-1} u_{K}^{e}\right] \tilde{\psi}_{g,e}^{+}, \tag{86}$$

for each element.

Another approach for using Eq. (71) is to directly apply it on the surface of the element for which the projection desired is shown in Eq. (87).

$$\psi_{g,\Gamma}^{-}(\vec{r},\hat{\Omega}) = -\frac{1}{\Sigma_{t,g}^{e}} \hat{\Omega}\cdot\vec{\nabla}\psi_{g,e}^{+}(\vec{r},\hat{\Omega})\,. \qquad \vec{r} \in \Gamma_{\gamma} \tag{87}$$

In this approach, the even parity flux approximation must be evaluated along the targeted boundary and then the derivative is applied. Weighting Eq. (87) with the boundary odd-parity flux approximation yields the following.

$$\int\int \Lambda_{-,\gamma}\boldsymbol{\mathcal{L}}_{-,\gamma} Y_{-}(\hat{\Omega}) \otimes h_{\gamma}^{T}(\vec{r}) \left[\psi_{g,\Gamma}^{-}(\vec{r},\hat{\Omega}) + \frac{1}{\Sigma_{t,g}^{e}} \hat{\Omega}\cdot\vec{\nabla}\psi_{g,e}^{+}(\vec{r},\hat{\Omega})\right] d\Omega d\Gamma_{\gamma} = 0\,. \quad \vec{r} \in \Gamma_{\gamma} \tag{88}$$

Equation (88) is reduced to the matrix relation

$$\Lambda_{-,\gamma}\Lambda_{-,\gamma}^{T} \otimes I_{\gamma}\ \tilde{\psi}_{g,\Gamma}^{-} = -\Lambda_{-,\gamma}\boldsymbol{\mathcal{L}}_{-,\gamma} V_{K} \otimes \frac{1}{\Sigma_{t,g}^{e}} T_{\gamma,K}^{e}\ \tilde{\psi}_{g,e}^{+}, \qquad \vec{r} \in \Gamma_{\gamma} \tag{89}$$

with the matrix definition

$$T_{\gamma,K}^{e} = \int h_{\gamma}(\vec{r}) \vec{\nabla}_{K} L_{e}^{T}(\vec{r} \in \Gamma_{\gamma}) d\Gamma_{\gamma}\,. \tag{90}$$

Equation (89) can be solved for the odd-parity coefficients, yielding

$$\tilde{\psi}_{g,\Gamma}^{-} = \left(\Lambda_{-,\gamma}\Lambda_{-,\gamma}^{T}\right)^{-1} \Lambda_{-,\gamma}\boldsymbol{\mathcal{L}}_{-,\gamma} V_{K} \otimes \frac{1}{\Sigma_{t,g}^{e}} T_{\gamma,K}^{e}\ \tilde{\psi}_{g,e}^{+}, \qquad \vec{r} \in \Gamma_{\gamma} \tag{91}$$

and can be substituted into Eq. (68) to obtain

$$j_{g,\Gamma}^{+} = \left[L_{\gamma}^{T} \otimes D_{\gamma}^{T} - K_{\gamma}\left(\Lambda_{-,\gamma}\Lambda_{-,\gamma}^{T}\right)^{-1} \Lambda_{-,\gamma}\boldsymbol{\mathcal{L}}_{-,\gamma} V_{K} \otimes \frac{1}{\Sigma_{t,g}^{e}} T_{\gamma,K}^{e}\right] \tilde{\psi}_{g,e}^{+}\,. \tag{92}$$

It is unclear at this point whether the use of a discontinuous finite-element flux representation is best, hence the inclusion of the polynomial representation in Eq. (73). Additionally, Eqs. (86) and (92) result in roughly the same amount of work for the polynomial approximation because the spatial integrations are independent of the cross sections (i.e., they can be performed once and stored for each element). Overall, the above representations merely pose possible choices, and only numerical studies will show which approach is the best. In conclusion, the above equations can be used to obtain

partial "current" representations for an iterative procedure for connecting to a subdomain by using a spherical harmonics flux approximation.

II.4.B Coupling with a Different Methodology

We next consider the case where an adjoining subdomain is treated with a method of characteristics (MOC) approach. We note that the approach for a discrete ordinates approach would be remarkably similar to MOC because both use a discrete set of angular directions to approximate the angular variable. As a result, we term these methods discrete angular (DA) methods. In a DA method, the angular representation along the subdomain boundary is a discrete set of angular directions, and the angular flux representation is given by

$$\psi(\vec{r}\in\Gamma_\gamma,\hat{\Omega}_n)=p_\gamma^T(\vec{r}\in\Gamma_\gamma)\psi_{\gamma,\Gamma}(\hat{\Omega}_n)=p_\gamma^T(\vec{r}\in\Gamma_\gamma)\tilde{\psi}_{\gamma,\Gamma,n}\,, \tag{93}$$

where $p_\gamma(\vec{r})$ is a vector of the spatial trial functions. For a discrete ordinates subdomain these spatial functions can be finite element trial functions similar to the preceding spherical harmonics method. For a MOC subdomain the spatial functions will be piecewise flat on each element surface along the subdomain boundary. The spatial relation for the incoming and outgoing flux between the DA domain and the spherical harmonics domain is a simple translation of the spatial approximation used in Eq. (93) and Eqs. (78) and (79) given by Eq. (94), where the limiting case is generally the method with the lowest-order spatial interface approximation.

$$R_\gamma=\int h_\gamma(\vec{r})\,p^T(\vec{r}\in\Gamma_\gamma)d\Gamma_\gamma\,, \tag{94}$$

The angular relation is more complex because the partial current variables used earlier are not truly a representation of the incoming and outgoing angular flux. In general, the angular approximation requires the numerical projection of the interface angular spherical harmonics solution to the discrete angular directions incoming to the DA subdomain. The solution for the angular flux as given by the spherical harmonics subdomain is shown in Eq. (95), where the spatial representation has been removed for simplicity:

$$\psi_{g,\Gamma}(\hat{\Omega})=Y_{+,\gamma}^T(\hat{\Omega})\tilde{\psi}_{g,\Gamma}^+ + Y_{-,\gamma}^T(\hat{\Omega})\tilde{\psi}_{g,\Gamma}^-\,. \tag{95}$$

Equation (95) can be expressed in terms of the partial current variables defined earlier in Eqs. (67) and (68), yielding

$$\begin{aligned}\psi_{g,\Gamma}(\hat{\Omega})&=\tfrac{1}{2}Y_{+,\gamma}^T(\hat{\Omega})L_\gamma^{-1}\left(j_{g,\Gamma}^+ + j_{g,\Gamma}^-\right)+\tfrac{1}{2}Y_{-,\gamma}^T(\hat{\Omega})K_\gamma^{-1}\left(j_{g,\Gamma}^+ - j_{g,\Gamma}^-\right)\\ &=\tfrac{1}{2}\left(Y_{+,\gamma}^T(\hat{\Omega})L_\gamma^{-1}+Y_{-,\gamma}^T(\hat{\Omega})K_\gamma^{-1}\right)j_{g,\Gamma}^+ + \tfrac{1}{2}\left(Y_{+,\gamma}^T(\hat{\Omega})L_\gamma^{-1}-Y_{-,\gamma}^T(\hat{\Omega})K_\gamma^{-1}\right)j_{g,\Gamma}^-\end{aligned}\,. \tag{96}$$

For a discrete set of angular directions $\hat{\Omega}_n$ that are entering the DA subdomain ($\hat{\Omega}_n\cdot\hat{n}<0$), Eq. (96) reduces to

$$\begin{aligned}\psi_{g,\Gamma}(\hat{\Omega}_n)&=\psi_{g,\Gamma,n}\\ &=\tfrac{1}{2}\left(Y_{+,\gamma}^T(\hat{\Omega}_n)L_\gamma^{-1}+Y_{-,\gamma}^T(\hat{\Omega}_n)K_\gamma^{-1}\right)j_{g,\Gamma}^+ + \tfrac{1}{2}\left(Y_{+,\gamma}^T(\hat{\Omega}_n)L_\gamma^{-1}-Y_{-,\gamma}^T(\hat{\Omega}_n)K_\gamma^{-1}\right)j_{g,\Gamma}^-\end{aligned}\,. \tag{97}$$

Note that in Eq. (97), the incoming angular flux solution into the DA subdomain includes the entire angular flux solution along the boundary for the spherical harmonics domain. The resulting iterative solution technique will suffer from latent coupling because it will not pose a true response matrix routine as that seen in the preceding spherical harmonics

method. In short, the exiting flux solution from the previous iteration for the DA domain directly contributes to the new flux solution coming into the DA domain.

The update of the spherical harmonics solution is more complicated because the entire DA angular solution must be used to construct the new incoming flux. To obtain the components of the spherical harmonics approximation, we use a least squares approach:

$$\tilde{\psi}^{+}_{g,\Gamma} = \left(M_{+}^{T} M_{+}\right)^{-1} \psi_{g,\Gamma,n} \qquad M_{+} = \begin{bmatrix} Y_{+,1}\left(\hat{\Omega}_n\right) & \cdots & Y_{+,N}\left(\hat{\Omega}_n\right) \\ \vdots & \ddots & \vdots \\ Y_{+,1}\left(\hat{\Omega}_N\right) & \cdots & Y_{+,N}\left(\hat{\Omega}_N\right) \end{bmatrix}. \tag{98}$$

$$\tilde{\psi}^{-}_{g,\Gamma} = \left(M_{-}^{T} M_{-}\right)^{-1} \psi_{g,\Gamma,n} \qquad M_{-} = \begin{bmatrix} Y_{-,1}\left(\hat{\Omega}_n\right) & \cdots & Y_{-,N}\left(\hat{\Omega}_n\right) \\ \vdots & \ddots & \vdots \\ Y_{-,1}\left(\hat{\Omega}_N\right) & \cdots & Y_{-,N}\left(\hat{\Omega}_N\right) \end{bmatrix}. \tag{99}$$

Although the above process can be carried out by using only the exiting DA directions, it would not yield a spherical harmonics solution consistent with that used in the spherical harmonics equations. Given that the DA flux solution is converted to an equivalent spherical harmonics vector, the above can be inserted into Eq. (67), as follows.

$$j^{-}_{g,\Gamma} = L_{\gamma} \otimes I_{\gamma}\, \tilde{\psi}^{+}_{g,\Gamma} - K_{\gamma} \otimes I_{\gamma} \tilde{\psi}^{-}_{g,\Gamma} \tag{100}$$

Combining the space angle approximations, we obtain the final equation for the angular flux solution entering the DA subdomain

$$\begin{aligned} \psi_{g,\Gamma,n} = {} & \tfrac{1}{2}\left(Y^{T}_{+,\gamma}(\hat{\Omega}_n) L_{\gamma}^{-1} + Y^{T}_{-,\gamma}(\hat{\Omega}_n) K_{\gamma}^{-1}\right) \otimes R_{\gamma}^{T}\; j^{+}_{g,\Gamma} \\ & + \tfrac{1}{2}\left(Y^{T}_{+,\gamma}(\hat{\Omega}_n) L_{\gamma}^{-1} - Y^{T}_{-,\gamma}(\hat{\Omega}_n) K_{\gamma}^{-1}\right) \otimes R_{\gamma}^{T}\; j^{-}_{g,\Gamma} \end{aligned} \qquad \hat{\Omega}_n \cdot \hat{n} < 0 \tag{101}$$

and the angular flux solution entering the spherical harmonic subdomain:

$$j^{-}_{g,\Gamma} = \left[L_{\gamma}\left(M_{+}^{T} M_{+}\right)^{-1} - K_{\gamma}\left(M_{-}^{T} M_{-}\right)^{-1}\right] \otimes R_{\gamma}^{T}\; \psi_{g,\Gamma,n}\,. \tag{102}$$

II.5 Mesh Generator

As part of the UNIC effort, several mesh generators will be evaluated and incorporated into UNIC for effortless mesh generation. The aim is to simplify the exchange of data for neutron transport calculations and provide detailed data for the high-resolution display of results in a postprocessing module. Currently, we are evaluating the coupling between the CUBIT code (mesh generator) and the finite element code PnFE (finite-element, second-order form transport solver of the UNIC code). Initial results have provided a better understanding of the capability of the mesh generator and, at the same time, have indicated potential issues to be tackled in the future:

- The mesh generator is capable of exporting quadratic-order hexahedron that is needed for accurate treatment of curved surfaces.

- The mesh generated by CUBIT currently must be strictly controlled by the user such that the mesh is coarse. For transport problems, the number of elements used to represent the structure can be coarser than those used for heat transfer and fluid flow calculations.
- A process will be developed so that the mesh is controlled by user input, saving computational time and providing an easy way to control the geometry and mesh creation process.

An effort also was undertaken to implement a translator between the MCNP geometry input file and the ACIS CAD file format. This translator allows importing directly the MCNP geometries in the CUBIT code for the meshing process. MCNP is known to have a practical geometry input that allows complex structures to be defined efficiently. Once cells are defined, they can be grouped in "universes" and then replicated or used in a lattice structure by using a simple card.

Generally, deterministic codes inputs are much harder to define because they need mesh information, which must come from some special software with which the geometry is, in most cases, defined by hand; moreover, the meshing process is not yet automatic. Therefore, our objective was to create an interface from MCNP to a geometry format, so that MCNP geometry can be easily converted into a finite-element input for UNIC.

Specifically, the interface must take as an input a MCNP input file, read it and extract the geometry information, create the corresponding solid model, and export it in a CAD file format. The format chosen is ACIS (*.sat, *.acis and *.sab files), which is widespread and portable. Moreover, the program must be able to export (if the user wishes) information about the geometry that will make the meshing process easier.

The program is divided into two parts :

- The **parser**, which reads the MCNP input file and stores the geometry information into relevant data structures. The parser was programmed in Fortran 90/95.
- The **converter, which** reads the data structures, creates the geometry, and exports it. The converter was programmed in C++ and is based on the CGM library by Sandia National Laboratories.

The first task was to create the parser. The Fortran 90 program is divided into three subprograms reflecting the hierarchy of the input.

- The first module *(MCNP_GEOM_PARSER)* loads the input file and reads it line by line.
- The second module *MCNP_GEOM_LINE_PARSER* contains three main subroutines. Each one reads a card depending on its type (cell, surface or data).

- The third module *TREE_PARSER*, as mentioned above, reads the combinatorial geometry information.

The main program is a short Fortran routine that takes into account the command-line parameters, launches the parser and then the converter, and then cleans the files.

- The converter is the program that actually creates and exports the geometry. It is based on the CGM library developed by Sandia National Laboratories. The library contains the abstract entities for volumes, surfaces, curves, vertices, and so forth and provides easy-to-use interfaces to create and modify the geometry.

Figures 6a and 6b show the block geometry for a pin subset of an Advanced Burner Test Reactor assembly and the associated 2D meshing for CUBIT obtained starting from a geometrical description in MCNP.

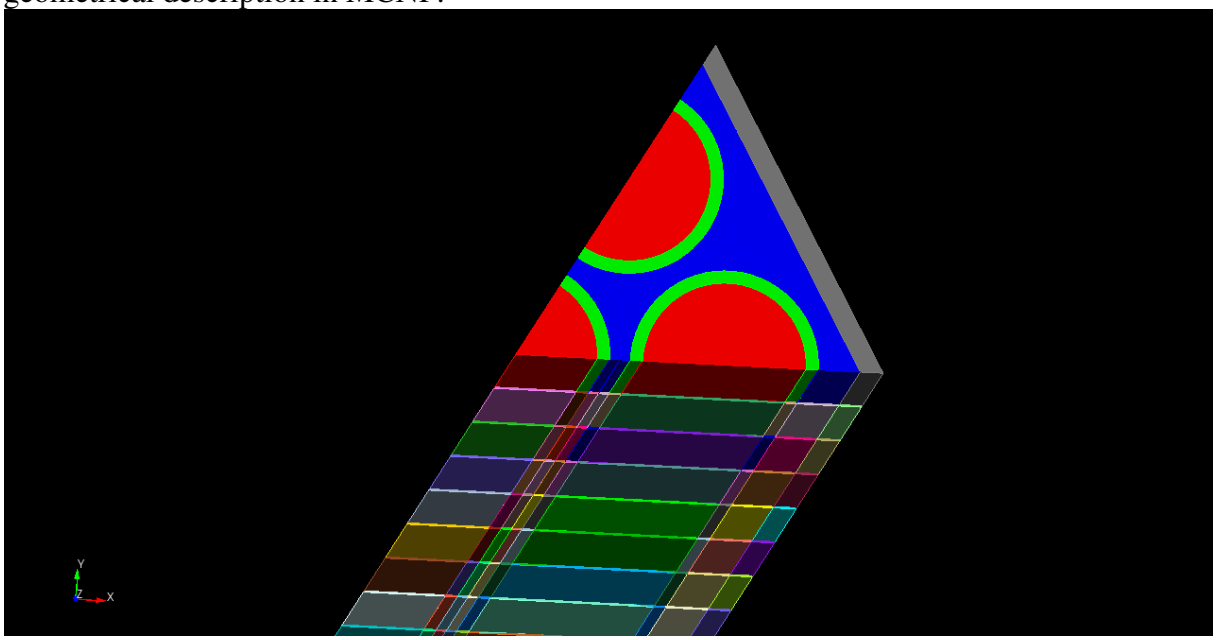

Figure 6a. 3D geometry in CUBIT with blocks.

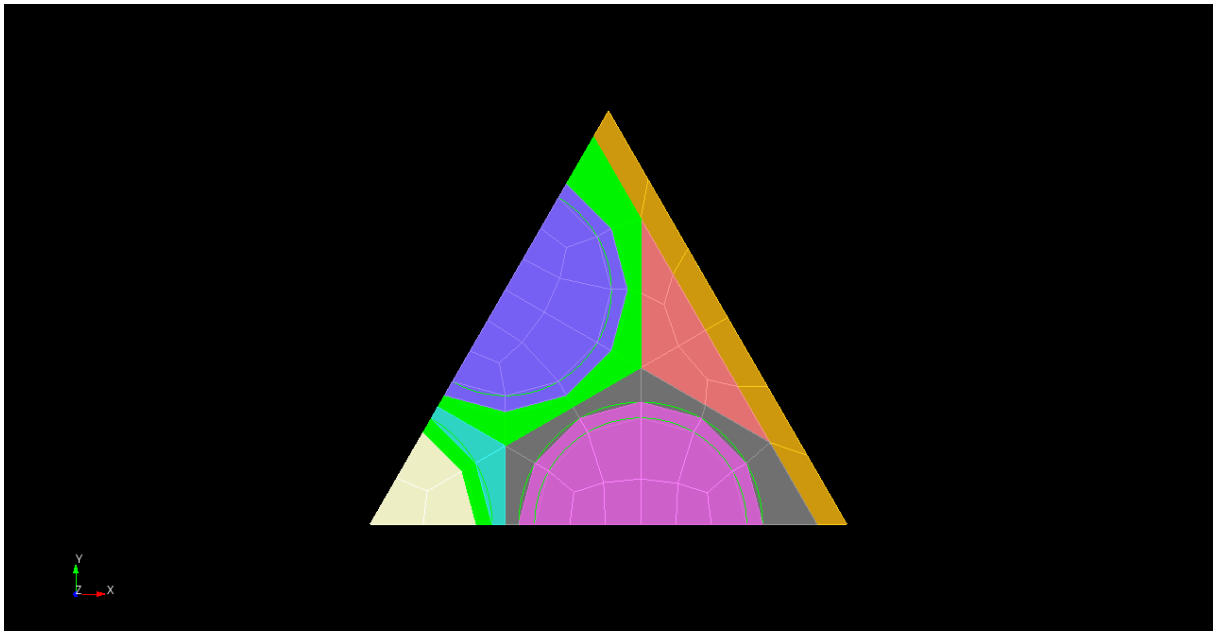

Figure 6b. 2D CUBIT meshing associated to Figure 6a.

II.6 Results

Of the preceding derivation work, only the spherical harmonics method has been implemented to date. The first benchmark problem considered was a simple two-group benchmark calculation [2]. The geometry consists of three regions as shown in Figure 7, the cross sections for which are given in Table 1.

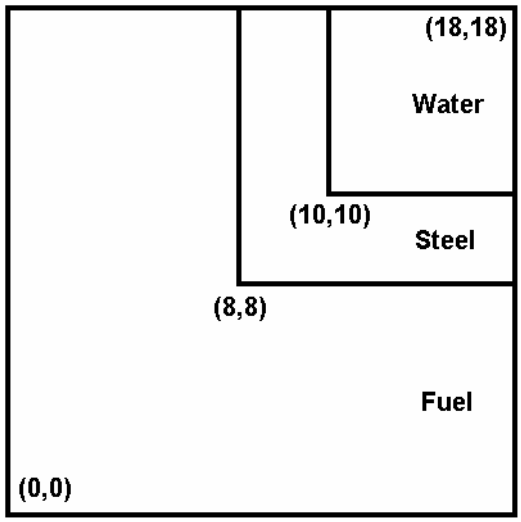

Figure 7. Two-group benchmark geometry.

Table 1. Two-Group Benchmark Cross Sections

	Fuel Composition		Steel Composition		Water Composition	
Cross Section	**1**	**2**	**1**	**2**	**1**	**2**
Σ_a	0.01	0.07	0.003	0.11	0.001	0.03
$\nu\Sigma_f$	0.006	0.1	0.0	0.0	0.0	0.0
Σ_t	0.22	0.8	0.53	0.77	0.701	2.0
$\Sigma_{1\rightarrow2}$		0.017		0.001		0.05
Σ_s	0.176	0.73	0.526	0.83	0.65	1.80
χ	1.0	0.0				

Two configurations of the boundary conditions were considered, and the reference solution was obtained by using the VARIANT code [3–5]. For the first configuration, reflected boundary conditions were applied on all sides of the domain, and the reference eigenvalue solution of 0.90152 was obtained with VARIANT (P_{17} with sixth-order flux/source and quadratic interface). For the second configuration, reflected boundary conditions were applied to the left and bottom sides of Figure 7, while vacuum boundary conditions were applied to the top and right. The reference VARIANT eigenvalue solution was 0.64562. Tables 2 and 3 give the eigenvalue solutions of the two boundary configurations obtained for two different mesh refinements (coarse and fine) in the new code, along with lower-order angular solutions obtained with the VARIANT code. As can be seen from Tables 2 and 3, the solutions obtained by using a coarse mesh are insufficient and converge to an incorrect solution. The fine mesh produces a better solution, consistent with that of the VARIANT code, but additional mesh refinement is necessary to match the reference solution as evident by the remaining error in the P_{17} solutions for the fine mesh.

Table 2. Configuration 1 Results for the Two-Group Benchmark Problem

	VARIANT		Coarse		Fine	
P_N	Eigenvalue	Error	Eigenvalue	Error	Eigenvalue	Error
1	0.89250	-1.00	0.88735	-1.57	0.89250	-1.00
3	0.90100	-0.06	0.89477	-0.75	0.90088	-0.07
5	0.90138	-0.02	0.89497	-0.73	0.90121	-0.03
7	0.90145	-0.01	0.89499	-0.72	0.90125	-0.03
9	0.90148	0.00	0.89499	-0.72	0.90127	-0.03
17	0.90152				0.90128	-0.03

Table 3. Configuration 2 Results for the Two-Group Benchmark Problem

	VARIANT		Coarse		Fine	
P_N	Eigenvalue	Error	Eigenvalue	Error	Eigenvalue	Error
1	0.63236	-2.05	0.62664	-2.94	0.62905	-2.57
3	0.64444	-0.18	0.64117	-0.69	0.64468	-0.15
5	0.64516	-0.07	0.64165	-0.61	0.64539	-0.04
7	0.64538	-0.04	0.64175	-0.60	0.64559	-0.01
9	0.64548	-0.02	0.64178	-0.59	0.64567	0.01
17	0.64562				0.64574	0.01

The next benchmark problem is taken from previous work where the necessary space-angle approximation is known to be very high [6–8]. A single pin-cell lattice calculation is solved where again the eigenvalue solution is sought for the geometry [8]. The reference solution was obtained by using a Monte Carlo code. For the new spherical harmonics code, three mesh approximations were implemented: coarse, medium, and fine. Figure 8 shows the eigenvalue results obtained by using the new code. As can be seen, the P_{33} angular approximation that was implemented is still not angularly converged. More important, additional spatial refinement is necessary to obtain an accurate solution. Given the relative difficulty of this problem, the solutions obtained by using the existing coding (poor preconditioner) are good, and therefore no additional effort was undertaken to improve the solution.

Figure 9 plots the scalar flux solution for P_1, P_{11}, and P_{21} angular approximations where each octant is the flux solution for each group. The final octant is the fission rate for the cell, where obviously no fission occurs in the water region. The color shading has been normalized by group for all three angular approximations such that the relative differences in the flux gradients can be observed in each group between the different angular approximations. As can be seen, there is virtually no gradient in the diffusion approximation; but as the angular approximation is refined, the scalar flux develops a very steep flux gradient near the material boundary. For problems of this type, the spherical harmonics method (and the discrete ordinates method) are not well suited, hence the focus on the method of characteristics.

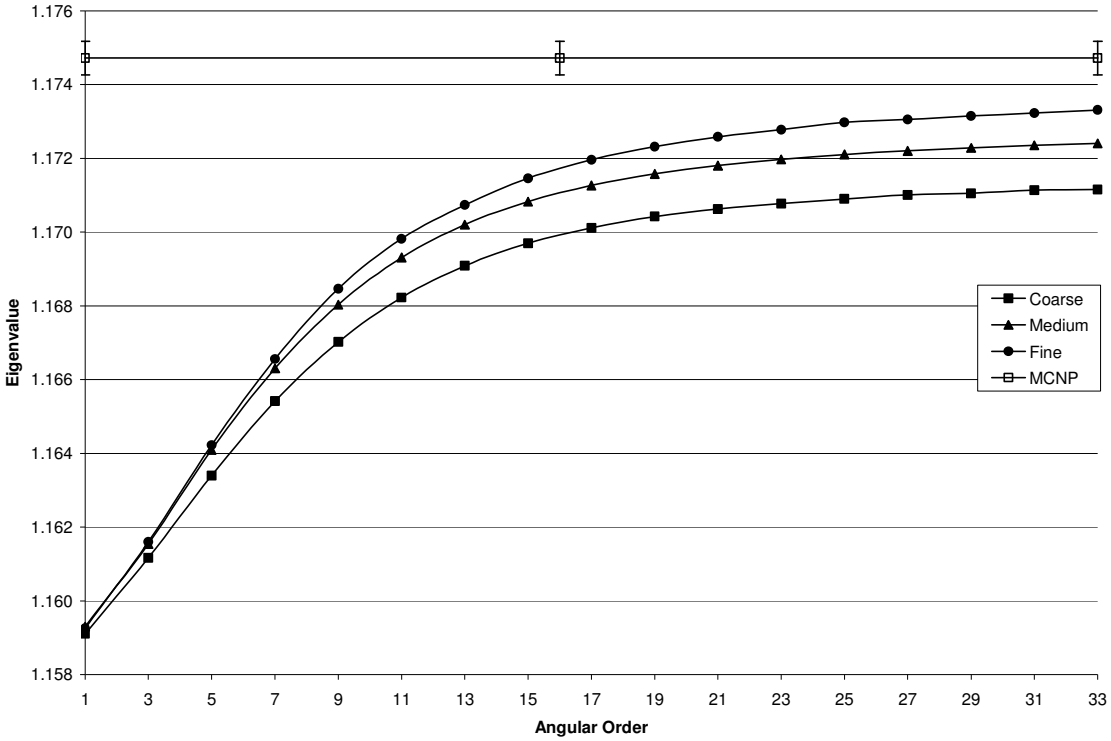

Figure 8. Eigenvalue solution for the seven-group pin-cell problem.

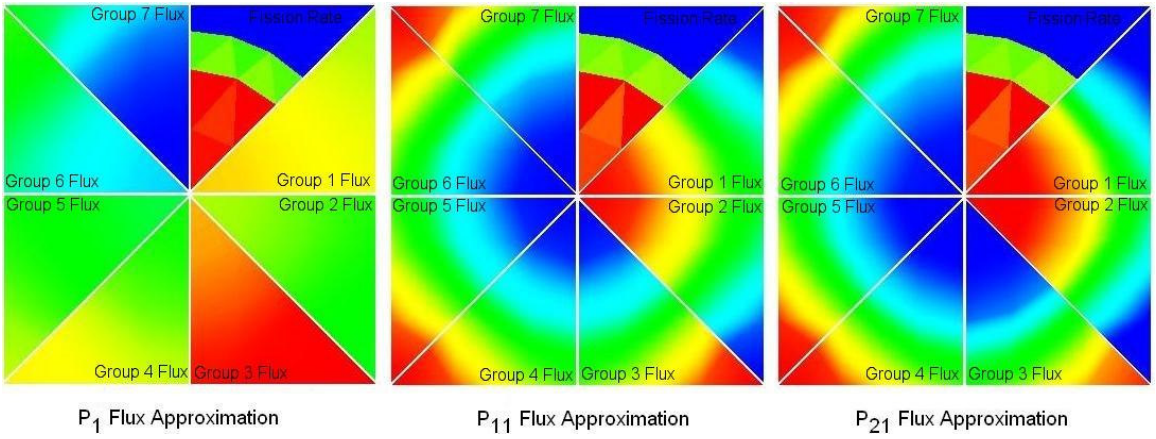

Figure 9. Flux solution plot for the seven-group pin-cell problem

The third benchmark problem solved with the new code is the first of four Takeda benchmarks [9]. The details of the three-dimensional geometry can be found in Reference 9. Two problems are described, but only one was solved for this work. The reference , given as 0.9624 ± 0.00071, was obtained by using a Monte Carlo code. Table 4 gives the solutions obtained by using the VARIANT code and the new UNIC module called PNFE. As can be seen, several meshes were attempted, with differing results. Overall, the medium-fine mesh, shown in Figure 10, produces solutions very close to that of the reference solution. Further refinements in angle are necessary but could not be obtained with the existing code because of limitations in computer memory. Both codes are observed to converge toward the reference solution, with the VARIANT solution obtaining slightly better results. We note that at P_{15}, the VARIANT solution is 0.96241.

Table 4. Eigenvalue Solutions for the Takeda #1 Benchmark

	PNFE						**VARIANT**	
	Coarse	Error (%)	**Medium**	Error (%)	**Fine**	Error (%)		Error (%)
1	0.92719	-3.6586%	0.93232	-3.1254%	0.93233	-3.1245%	0.932850	-3.0705%
3	0.97477	1.2855%	0.96119	-0.1260%	0.96119	-0.1256%	0.961223	-0.1223%
5	0.98032	1.8623%	0.96214	-0.0270%	0.96215	-0.0261%	0.962171	-0.0238%
7	0.98189	2.0254%	0.96231	-0.0097%			0.962338	-0.0064%
9	0.98251	2.0893%	0.96235	-0.0048%			0.962391	-0.0009%
11	0.98280	2.1198%					0.962400	0.0000%

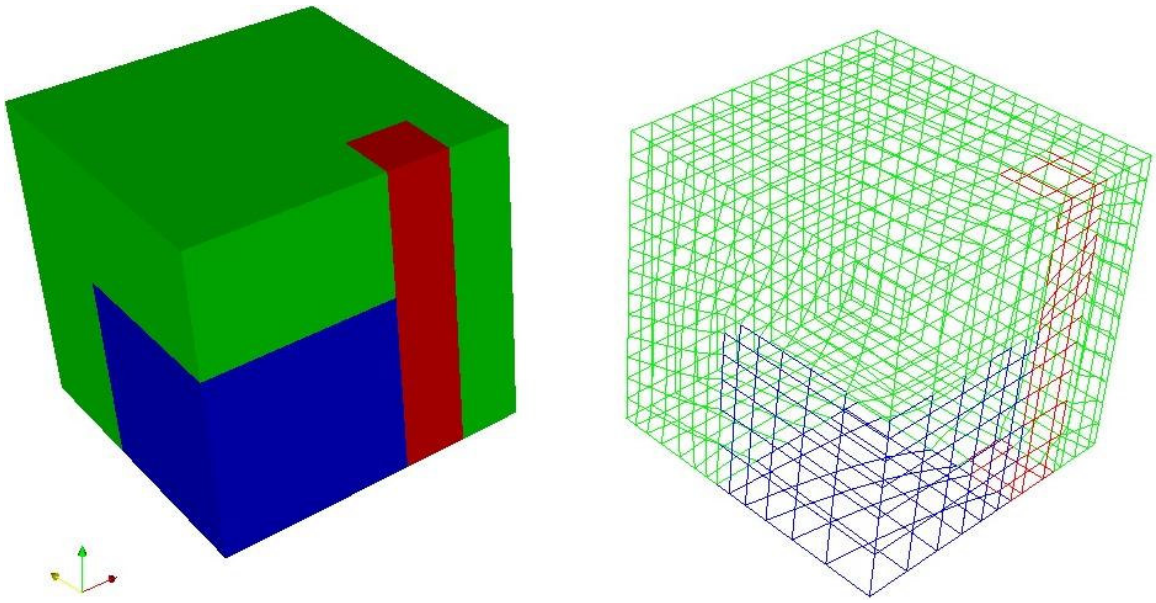

Figure 10. Takeda #1 benchmark mesh.

The fourth benchmark solved with the new code is the Takeda #4 Benchmark [9]. In this problem only a single mesh was attempted in the PNFE code; see Figure 11. As was the case with the Takeda #1 benchmark, only one configuration (unrodded) was performed; the results are given in Table 5. The reference Monte Carlo solution was 1.09514 ± 0.0004. As one can see from Table 5, more angular refinement is necessary in the PNFE code, but additional coding modifications are required before this can be achieved. We note that similar changes in the VARIANT code allow it to run on a single-processor machine. In both cases, additional space-angle refinement is needed in the PNFE and VARIANT codes to achieve the reference solution.

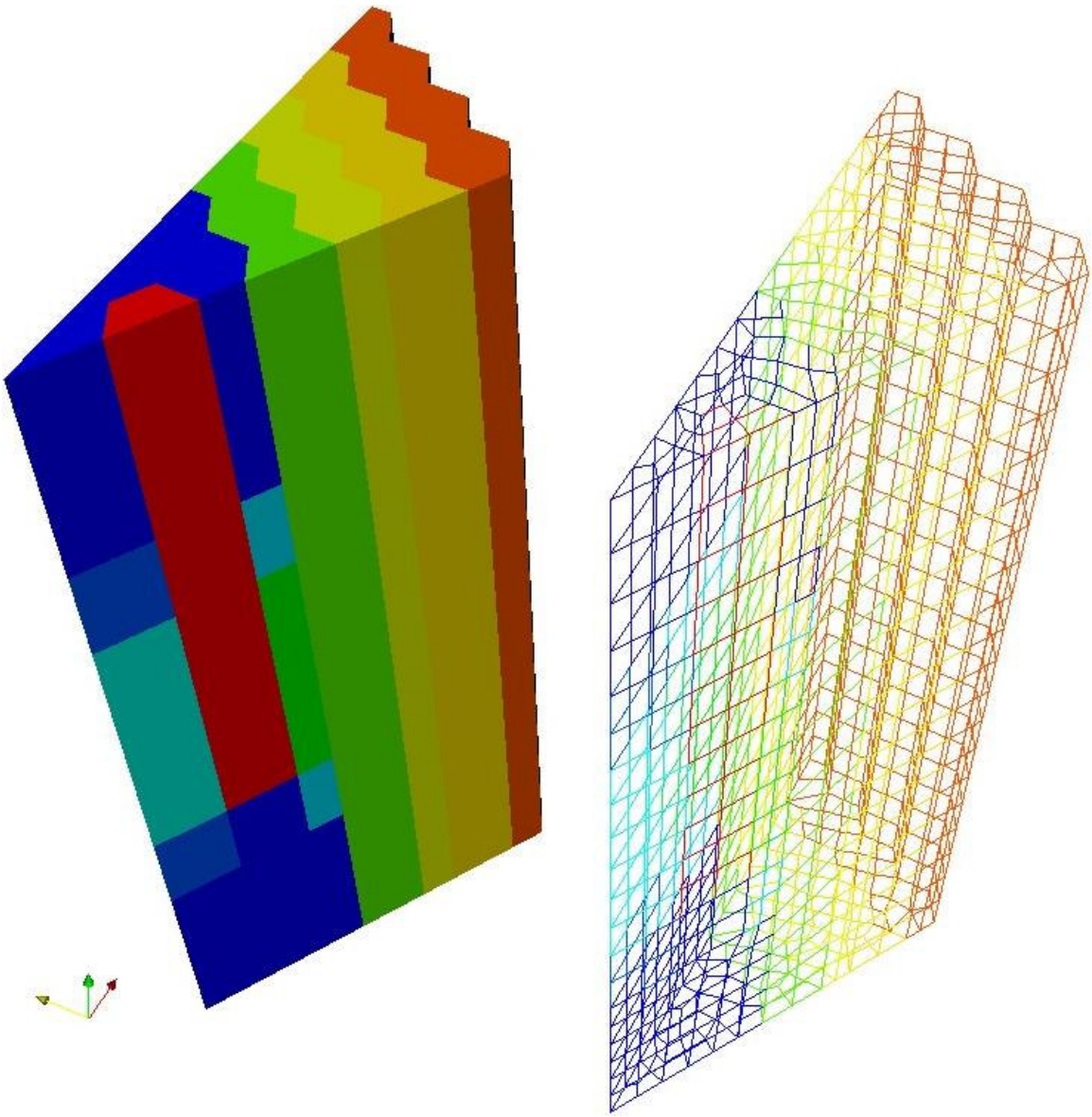

Figure 11. Takeda #4 benchmark mesh.

Table 5. Eigenvalue Solutions for the Takeda #4 Benchmark

	PNFE		VARIANT	
1	1.073385	-1.9874%	1.075429	-1.8007%
3	1.094553	-0.0545%	1.094409	-0.0676%
5	1.096111	0.0878%	1.095767	0.0564%
7			1.096142	0.0905%
9			1.096283	0.1034%
11			1.096363	0.1108%

III Thermal-Hydraulics with Nek5000

Nek5000 is a computational fluid dynamics code for coupled hydrodynamics and heat transfer developed in the Mathematics and Computer Science Division at Argonne. The code is based on the spectral element method (SEM), which is a high-order weighted residual technique that combines the rapid convergence of spectral methods with the geometric flexibility of the finite element method (FEM). Nek5000 employs second- or third-order semi-implicit timestepping, in which the nonlinear term is treated explicitly and the linear viscous and pressure coupling is solved by using state-of-the-art parallel multigrid algorithms. Nek5000 was recognized with the 1999 Gordon Bell high-performance computing award for performance on 4096 processors of the Intel ASCI Red machine and has recently been running in production mode on 32K processors of IBM's Blue Gene Watson platform.

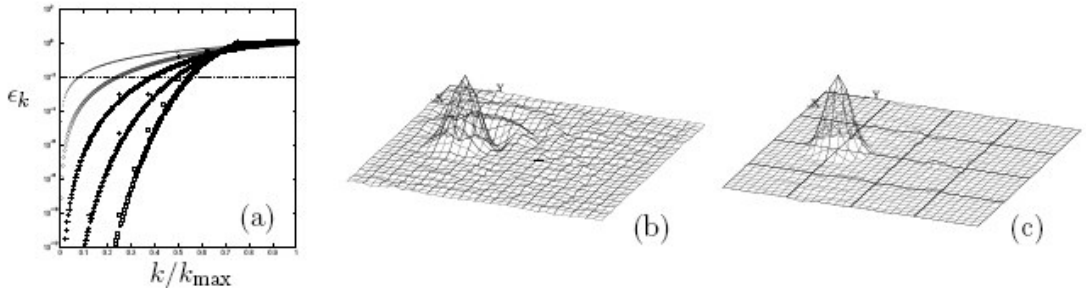

Figure 12. (a) Relative error in the 1D spectrum of $u_x = \lambda u$ versus fraction of resolvable modes for n=512, $E := n/N$, and N=1, 2, 4, 8, and 16; (b-c) spectral element solution for the convected cone problem after a single plane-rotation on a 32x32 grid: (b) (E,N) =(16x16,2), (c) (4x4,8).

Velocities and temperatures in the SEM are represented as Nth-order tensor product polynomials within each of E curvilinear brick elements, for a total of $n \sim EN^3$ gridpoints. A singular feature of the SEM is that it is designed for much higher approximation orders than are commonly used with the FEM. With the SEM, orders N = 4–16 are typical (and feasible, because of the use of matrix-free operator evaluation), implying a few hundred to a few thousand points per element. These high orders lead to excellent transport (minimal numerical diffusion and dispersion) for a significantly larger fraction of the resolved modes than is possible with the FEM. This point is illustrated in Figure 12a, which shows the error, ε_k, for eigenvalues associated with the model convection problem $u_t + u_x = 0$ on $[0, 2\pi]$ versus the fraction of resolvable modes, k/k_{max}. Here, $k_{max} = n/2$, according to the Nyquist criterion, $n = EN$ is the number of degrees of freedom for this one-dimensional problem, and $\varepsilon_k := |\kappa - k| / k$. The approximate eigenvalue is computed as $\kappa := (\phi_k', D\,\phi_k)_N / (\phi_k', \phi_k')_N$, where $\phi_k(x) := cos(kx)$, D is the spectral element derivative operator associated with E uniformly sized elements of order N, and $(.,.)_N$ is the inner product defined by using quadrature on the $N+1$ Gauss-Lobatto Legendre nodal points within each domain. (These nodal points also correspond to the Lagrange interpolation nodes for the SEM basis functions.) Figure 12a shows the errors for n=512,

N=1, 2, 4, 8, 16, and $E := n/N$. Taking 1 percent as an acceptable error threshold (indicated by the dashed line in Figure 12a), we see that only10 percent of the modes are well resolved with linear elements (N=1), whereas approximately half of the modes are well resolved for $N = 8$ and higher. Thus, the SEM provides roughly a fivefold reduction in the required number of gridpoints *per space dimension* to properly propagate waves at typical engineering tolerances. Note that because the abscissa is scaled by k_{max} the curves in Figure 12a exhibit little material change with increased resolution; as n increases, one resolves more waves, but the relative fraction remains unchanged. By the same token, one cannot circumvent the Nyquist sampling criterion by simply increasing N. In fact, as N increases, one can resolve at most $(2/\pi)\ k_{max}$ waves because of the spacing of stable (Gauss-type) nodal point distributions. The use of moderate values of N is motivated by the fact that one resolves nearly this number of waves with N ~8–16.

The benefits of a minimally dispersive/dissipative spatial discretization are illustrated by the two-dimensional convection problem of Figures 12b and 12c, which show the solution after an initially pointed cone is subjected to plane rotation on a pair of n = 32×32 grids. The quadratic case, with E = 16×16 elements, exhibits significant numerical dispersion after a single rotation. This dispersion is dramatically reduced as the order is increased to $N = 8$ ($E = 4 \times 4$). The improvement is striking in light of the fact the solution is *not smooth* and as such is not generally considered amenable to high-order discretizations. The improved result of Figure 12c derives from the ability of the high to accurately propagate modes that *are* resolved (i.e., corresponding to $k < (2/\pi)\ k_{max}$ in Figure 12a). The accurate transport properties of the SEM makes it ideal for propagation of small-scale structures and for long time integrations that are typical of large-eddy and direct numerical simulations. In addition to linear convection problems such as demonstrated in Figure 12, Nek5000 has been extensively verified on analytic, transitional, and turbulent Navier-Stokes benchmarks.

III.1 Preliminary Results: Wire-Wrapped Fuel Pin Analysis

The design of the advanced burner reactor calls for liquid sodium coolant. The fuel pins in the core are separated by wire wrap spacers that are wound helically along the pin axes. A single assembly will comprise 217 pins in a hexagonal array, with about five turns of the wire along the length of each pin. In addition to spacing the pins, the wire wrap is designed to promote mixing of the coolant between the channels that are formed by the pin array. Enhanced mixing can lead to reduced peak pin temperatures. The ability to predict (and optimize) mixing behavior can lead to reduced hot channel factors and thereby allow the plant to operate at higher power levels. In addition to the mixing, it is of interest in the design of the overall core to predict the distribution of the coolant flow, including the amount passing through the near-wall subchannels.

The wire-wrapped fuel pin analysis is challenging for several reasons. The Reynolds number based on the hydraulic radius, $Re=U\ D_h/\nu$, is in the range of 40000–65000, which means the flow is turbulent. The turbulence is further promoted by the wire wrap. Because it creates multiple contact points and lines, the wire wrap makes the geometry (and mesh) relatively complex. Moreover, with ~400 subchannels and L/D_h ~360, the

geometry of just a *single subassembly* is already large. There are hundreds of assemblies in the reactor.

We have taken a first step to understanding the thermal-hydraulics of the ABR core by analyzing a single periodic cell in the wire-wrap geometry. This analysis is based on the hypotheses (which can be rigorously checked *a posteriori*) that the flow in the middle of the assembly does not sense the presence of the walls, that the flow is fully developed (i.e., statistically independent) in the axial direction as one moves from one wire-wrap period to the next, and that the turbulent correlation length in the axial direction is less than the wire pitch, H_w.

Our initial simulation geometry can be seen in Figure 13 (bottom). The wire pitch was chosen to conform to an earlier experimental study with H_w/D_h=31.9. (The design calls for H_w/D_h=64.) The computational mesh comprised E=29520 elements of order N=7 (8.7 M points). Simulation of one flow-through time required 15 hours on 2048 nodes of Argonne's Blue Gene/L.

Figure 13 shows the axial velocity distribution along the periodic-cell interface for Re=4200 (top) and Re=12600 (center). A marked increase in range of scales is evident with the increase in Reynolds number. Low-speed streaks that are characteristic of wall-bounded turbulent flow are visible in the velocity isosurface distribution shown for Re=12600 at the bottom of Figure 13. This saturated nonlinear state was reached in a single flow-through time, which indicates that the flow in the core will be fully developed within the first wire pitch. This hypothesis will be checked in a future run with inflow/outflow boundary conditions.

The present simulations will be used to provide hydrodynamic building blocks for thermal transport. The simulations will be coupled with Lagrangian particle tracking that will allow quantification of interchannel mixing (modulo wall effects). The simulations also provide accurate flow fields that can be directly coupled with larger (i.e., multichannel) thermal hydraulics computations or used to provide accurate mean and fluctuating flow fields that can server as surrogate flow fields. The availability of both approaches will allow us to develop an accurate low-dimensional model for thermal transport in wire-wrap fuel pin assemblies. Future simulations involving both internal channels and wall channels will allow quantification of wall effects. At a minimum, one can perform a computation with two interior channels and two wall channels in ~30 hours with 2048 processors on the BG/L.

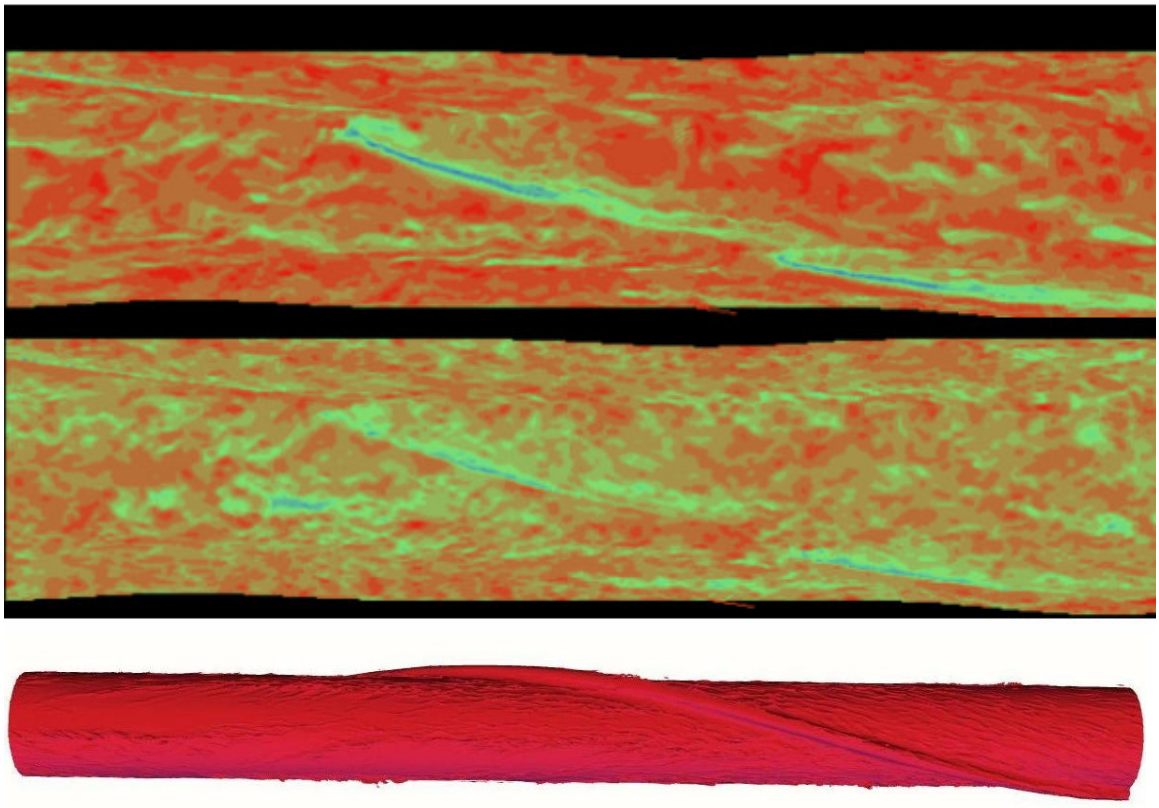

Figure 13. Nek5000 simulation of turbulent flow field in an array of wire-wrapped fuel pins: (top) axial velocity distribution at Re=4200, (center) Re=12600, (bottom) isosurface of axial velocity at 62.5% of Umean.

IV Numerical Strategies for Multiphysics Coupling

For decades, high-fidelity modeling of nuclear cores has been divided into two domains of physics: neutronics, which solves for the neutron population distribution, and thermal-hydraulics, which solves for the coolant velocity, density, and temperature fields and the fuel temperature distribution. Yet, these isolated physical models in reality describe physical processes that are intertwined and rely heavily on the solution field of one another. In the past decade or so, various monodisciplinary codes have been coupled together in a naive "black-box" fashion, where the output of one code serves as the input of another code, thereby producing solutions that we denoted hereafter as *nonlinearly inconsistent*. Such schemes, which are still the main coupling paradigm today for solving nonlinear nuclear reactor physics equations, are based on a linearization that is never resolved. The schematic of the "black-box" model is shown in Figure 14, where each physic component is solved by an independent monophysics code and the data from one code is exchanged through message-passing paradigms such as PVM or MPI.

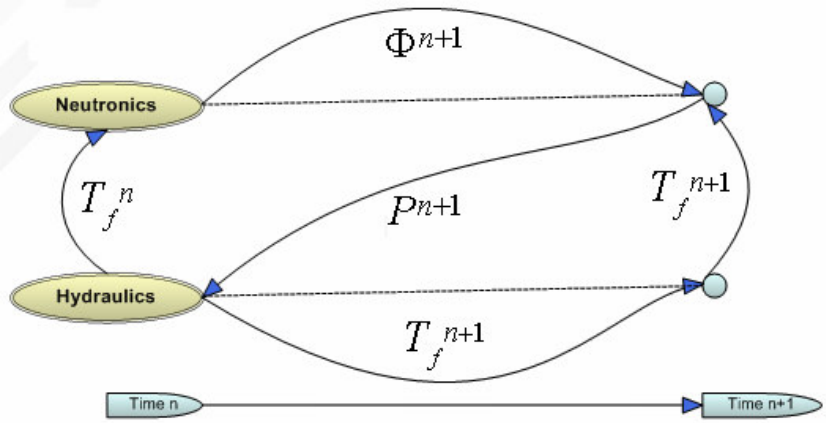

Figure 14. Schematic of a "black-box" model. -

In reactor analysis, numerous N-TH (neutronics-thermal hydraulics) codes have been coupled in such a fashion; see, for example, the OECD/NEA - US/NRC PWR Main Steam-Line Break Benchmark and the OECD/NEA - US/NRC Boiling Water Reactor Turbine Trip Benchmark, which use the conventional coupling techniques described above. This coupling technique is also mathematically described as operator splitting.

The fundamental inefficiency and hence the essential drawback of this conventional or naïve strategy is that the treatment of the nonlinear terms is not rigorous. This inconsistent treatment of nonlinear terms, wherein one or more physics solution field is lagged in time, usually results in a loss of accuracy order. Typically, in order to mitigate this loss of accuracy, smaller time steps are required to achieve a reasonable convergence, thereby taxing the CPU and increasing the overall computer calculation time.

Here, we use a Point Reactor Kinetic Model with Feedback applied to both light water reactors and sodium-cooled fast reactors. The objective is fourfold:

1. Demonstrate the weaknesses of naïve coupling paradigm
2. Present nonlinearly consistent schemes based on fixed-point iterations and extrapolation techniques
3. Investigate higher-order implicit Runge-Kutta schemes well-suited for stiff nonlinear problems
4. Investigate the potential of higher-order implicit Runge-Kutta schemes combined with automatic step size control

A nonlinear problem of ordinary differential equations can be formulated as follows:

$$\dot{u} = Lu + N(u) + b(t) = f(t,u) \tag{103}$$

where u is the vector of unknowns, $\dot{u}$ represents time derivation, L is a matrix representing a linear operator, and $N(u)$ is a nonlinear vector function. L and $N(u)$ may result, say, from spatial discretization of partial differential equations. Our test-case problems are zero-dimensional but still retain the stiffness and nonlinearities encountered in multidimensional coupled neutronics/thermal-hydraulics.

Let us consider an implicit time integration of Eq. (103) using a *theta*-scheme.

$$\frac{u^{n+1}-u^n}{t^{n+1}-t^n} = \theta(L^{n+1}u^{n+1}+N(u^{n+1})+b(t^{n+1}))+(1-\theta)\ (L^nu^n+N(u^n)+b(t^n)) \tag{104}$$

In a conventional coupling paradigm, the nonlinear term $N(u^{n+1})$ would not be known but simply approximated by $N(u^n)$, that is, lagged behind, which is a crude first-order approximation. Hence, if both physic components were solved by using a second-order scheme, in our example a Crank-Nicholson scheme (θ=1/2), then the resulting overall accuracy order of the solution would only be 1, because of the approximation $N(u^{n+1}) \approx N(u^n)$.

IV.1 Improved Coupling Strategies

Rather than lagging behind the nonlinear terms, a simple idea consists of extrapolating or predicting the value u^{n+1}. The idea is to replace the unknown quantity $N(u^{n+1})$ by a predicted quantity $N(u^{n+1,P})$, where the crudest prediction $N(u^{n+1,P})=N(u^n)$ yields back the conventional coupling first-order result. For instance, a second-order prediction is given by

$$u^{n+1,p} = 2u^n - u^{n-1} + O(h^2) \tag{105}$$

Equation (105) is an explicit second order prediction approximation used for nonlinear terms only. When combined with, say, the implicit Crank Nicolson scheme for the linear terms, we obtain a semi-implicit/semi-explicit of second order. Even though this idea seems appealing, it does not assure unconditional stability of the overall scheme because of the explicit treatment of some terms. Additionally, high-order accurate prediction formulas are more difficult to obtain for methods were nonfixed time-step sizes.

Fixed-point iteration (or Newton's method) over the time step can be used to resolve the nonlinearities, hence restoring the convergence order of the time integrator, but at the cost of a certain number of iterations per time step. Fixed-point iteration can be accelerated by using Steffenssen's technique (a variant of Aitken's Δ^2 process).

The θ time discretization scheme was implemented for the zero-dimensional Point Reactor Kinetic Model with Feedback. Figure 15 shows a calculation wherein at t=250 ms, a control rod is ejected. The ramp ejection duration is 250 ms with amplitude of 1.2$. The reference computation for the transient was performed by using a time-step size of 0.4 ms. Three other computations were performed with a time-step size of 10 ms:

1) Conventional coupling paradigm as shown in Figure 14
2) Fixed-point iterations (i.e., conventional scheme iterated)
3) Explicit higher-order treatment of nonlinear terms (improved prediction)

Figure 16 shows that the conventional coupling scheme over predicts the power level by more than 10%, whereas the other schemes are off by at most only 1%. Obviously, the improved prediction scheme was much cheaper than the fixed-point method because there are no iterations over all the physics within each time-step calculation.

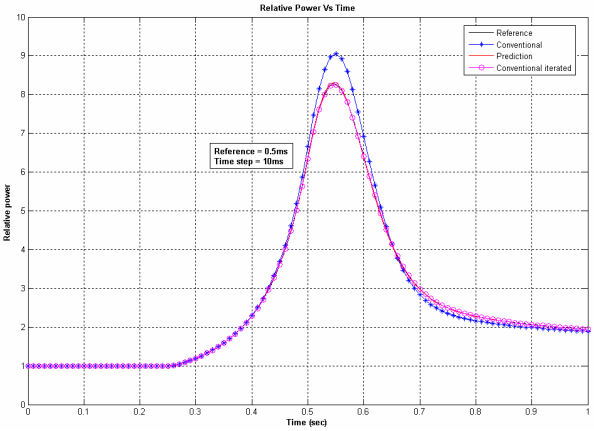

Figure 15: Comparison between schemes.

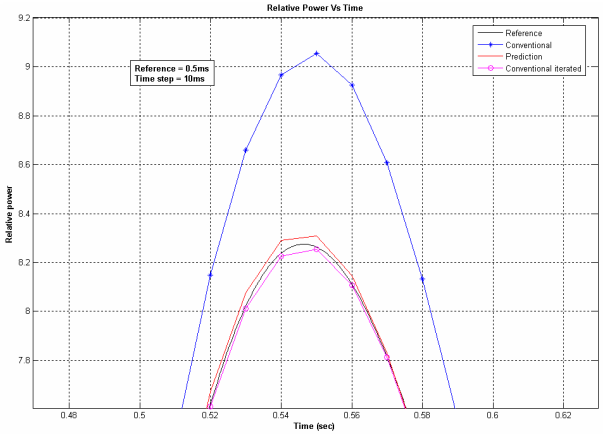

Figure 16: Enlarged at power peak.

Based on the calculations, the number of fixed-point iterations per time step was tabulated for different values of time steps for each of the schemes mentioned above. The results are plotted in Figure 17. The figure shows that using either solution prediction or acceleration definitely improves the number of fixed-point iterations. However, whereas using only prediction does not yield a considerable reduction in number of iterations, the synergistic effects of using both acceleration and prediction provides a reduction of more than 30% in the number of iterations per time step. On a large time scale, the total reduction in CPU time can then be threefold because, on an average, only one-third of the iterations are needed to fully converge the nonlinearities between the different physics. Figure 17 also shows that the usage of acceleration for finer time steps yields no perceptible improvement because fewer iterations are needed to converge nonlinearities, while acceleration for coarser time steps results in a considerably faster convergence, still retaining the order of accuracy.

The predictive methods also will restore the lost order of accuracy for the coupled transient scenario even without outer fixed-point iteration. In order to analyze this, a ramp transient, identical to the one plotted in Figure 15, was simulated and the orders of accuracy of the various different schemes were measured. Figure 18 presents the accuracy order obtained for the three aforementioned schemes: (1) conventional coupling, (2) fixed-point iterations, and (3) explicit high-order treatment of nonlinear terms (improved prediction). The time-step sizes varied from 0.8 ms to 100 ms.

The conventional coupling scheme yields only first-order accuracy, whereas fixed-point iterations and improved prediction both yield second-order accuracy: these schemes are therefore nonlinearly consistent and yield the expected higher accuracy order. Clearly, the improvement in the number of iterations by Steffensen's acceleration technique does not change the order accuracy. But the solution prediction method does improve the conventional first-order solution to yield a second-order-accurate solution with a simple extrapolation based on the previous history of the solution.

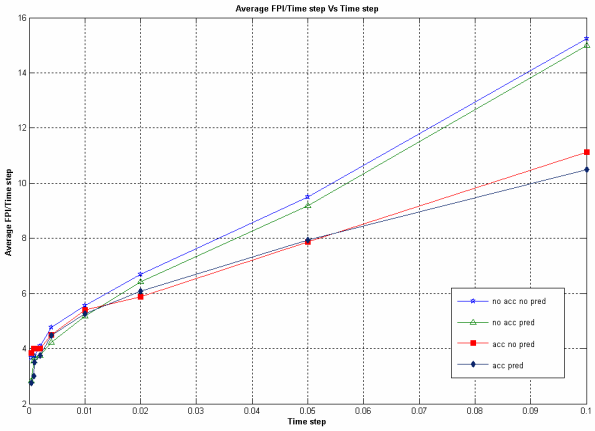

Figure 17. Efficiency of acceleration.

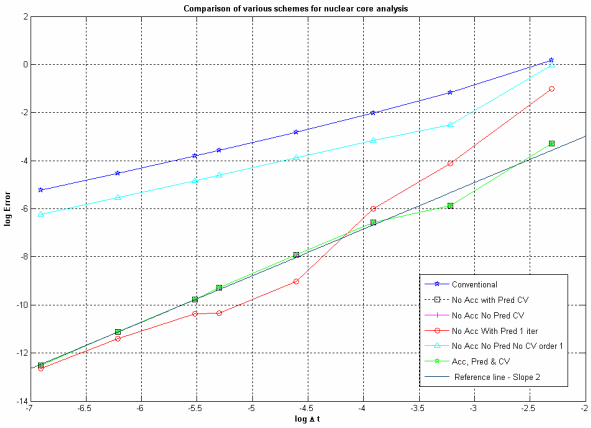

Figure 18. Order of accuracy for schemes.

IV.2 Higher-Order Implicit Runge-Kutta Methods

Even though the results in the previous section are encouraging, issues related to the (1) derivation of extrapolation formulae for higher orders, (2) the stability properties of semi-explicit / semi-implicit time integration, and (3) the overhead of fixed-point iterations led us to investigate higher-order implicit Runge-Kutta methods. Time implicitation is mandatory and imposed by the stiff nature of the physical phenomena occurring during a reactor transient. Implicit Runge-Kutta methods offer the possibility of much higher-accuracy orders, efficient treatment of the nonlinear terms, and automatic time-adaptation.

A generic *s*-stage implicit Runge-Kutta method is given by

$$g_i = u^n + h\sum_{j=1}^{s} a_{ij} f\left(t^n + c_j h, g_j\right) \text{ for } i = 1,...,s$$

$$u^{n+1} = u^n + h\sum_{j=1}^{s} b_j f\left(t^n + c_j h, g_j\right)$$

which can be easily represented by using a Butcher tableau.

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array}$$

Implicit Runge-Kutta methods are obtained when the matrix $A = \left(a_{ij}\right)$ is not strictly lower triangular. In this case, the nonlinear function f implicitly depends on some yet unknown values of g_j. These nonlinear equations must be solved at each time step by using Newton's method (or an inexact Newton's method). Replacing the Jacobians $\frac{\partial f}{\partial t}\left(t^n + c_i h; g_i\right)$ by the approximation $J = \frac{\partial f}{\partial t}\left(t^n; u^n\right)$, we get the simplified Newton iterations:

$$\left(I - hA \otimes J\right)\left(Z^{k+1} - Z^k\right) = -Z^k + h\left(A \otimes I\right) F\left(Z^k\right)$$

where

$Z^k = \left(z_1^k,...,z_s^k\right)^{\mathrm{T}}$ with $z_i = g_i - u^n$ and $F\left(Z^k\right) = \left(f\left(t^n + c_1 h; z_1^k + u^n\right) z_1^k,...,f\left(t^n + c_s h; z_s^k + u^n\right)\right)^{\mathrm{T}}$

The question regarding the computation of the Jacobian needs to be addressed. Three solutions are possible: (1) the Jacobian is built, which may require the assembly of a very large matrix for multidimensional applications, (2) the Jacobian is evaluated numerically by using finite differences, which requires n^2 function evaluations per Jacobian evaluation but can be a viable approach if a new Jacobian is not needed at each time step (inexact Newton), or (3) the Jacobian matrix need not be formed if Newton's method uses a Krylov subspace method for its linear solver, in which case Jacobian-free techniques, which require only $n+1$ function evaluations, can be used. In this work, solution 2 was used to evaluate the Jacobian matrix.

A particular class of implicit Runge-Kutta methods, whose matrix *A* is only lower triangular with identical coefficients on the main diagonal (methods known as singly diagonal implicit Runge-Kutta, or SDIRK), is also of interest because of the reduced number of coefficients and the simplified algebra associated with it. When an SDIRK method is applied to a nonlinear system for which only one Newton's iteration is performed at each step (i.e., a linearized Newton's method), the class of Rosenbrock-Runge-Kutta methods is thus devised.

An attractive feature of the Runge-Kutta method is that, for a given set of coefficients a_{ij} and c_i, there exist several sets of coefficients b_i yielding embedded numerical approximations of different orders.

$$u^{n+1} = u^n + h\sum_{j=1}^{s} b_j f\left(t^n + c_j h, g_j\right) + O\left(h^p\right)$$

$$\hat{u}^{n+1} = u^n + h\sum_{j=1}^{s} \hat{b}_j f\left(t^n + c_j h, g_j\right) + O\left(h^{\hat{p}}\right)$$

The new optimal step size is then given by

$$h_{opt} = h\left(\frac{tol}{\left\| u^{n+1} - \hat{u}^{n+1} \right\|}\right)^{1/(q+1)}$$

with $q = \min\left(p, \hat{p}\right)$. This procedure also allows for step rejection when the error is greater than the user-prescribed tolerance *tol*.

The three-stage Radau IIA method of order 5 and the Kaps-Rentrop GRK4A (of the Rosenbrock method family) were tested for rod ejection transient accidents with data typical of an LWR and an SFR. Numerical behavior of the LWR- and SFR related results were similar. We report here the SFR transients, whose time integration is more challenging because of a smaller neutron mean generation time. We note that the results reported here have no real physical meaning because of the many approximations (e.g., no time delay taken into account for thermo-mechanical feedbacks), but they demonstrate the numerical capabilities of the proposed methodologies.

The following rapid transient was modeled: at t=0 s, a \$0.4 reactivity was extracted (step reactivity), and at t=0.2 s, a SCRAM was simulated by step-inserting \$0.5 of reactivity.

Figure 19 provides the global physical trend of the transient simulated: the maximum power level, attained within around 1.5 ms, is about 1.68 times the initial power; fuel Doppler and expansion effects overturn the power increase after that time. The sodium temperature rises up to 448 C (no boiling), fuel temperature rises to around 575 C; the SCRAM initiates a temperature decrease after 0.2 s. The last plot in Figure 19 shows the total and partial reactivity evolutions during the transient.

Figure 20 describes the log of the time-step length used using the Radau IIA method with three user-specified tolerances: *tol* =1E-3, 1E-4, 1E-5. A Matlab automated step control built-in function (ODE23s) was used as a reference with *tol* =1E-6. ODE23s required about 350 steps, whereas the Radau IIA methods, with crudest tolerances, required 35–55 steps only.

Figure 21 shows how precisely the rapid portion of the transient was captured by Radau IIA and how accurate was the SCRAM automatically captured by the step control algorithm of Radau IIA, even with relatively crude tolerances.

Table 6 compares Radau IIA and Rosenbrock GRK4A for the same user-specified tolerances. More steps are generally required for the GRK4A Rosenbrock method (which is only an order 4 method).

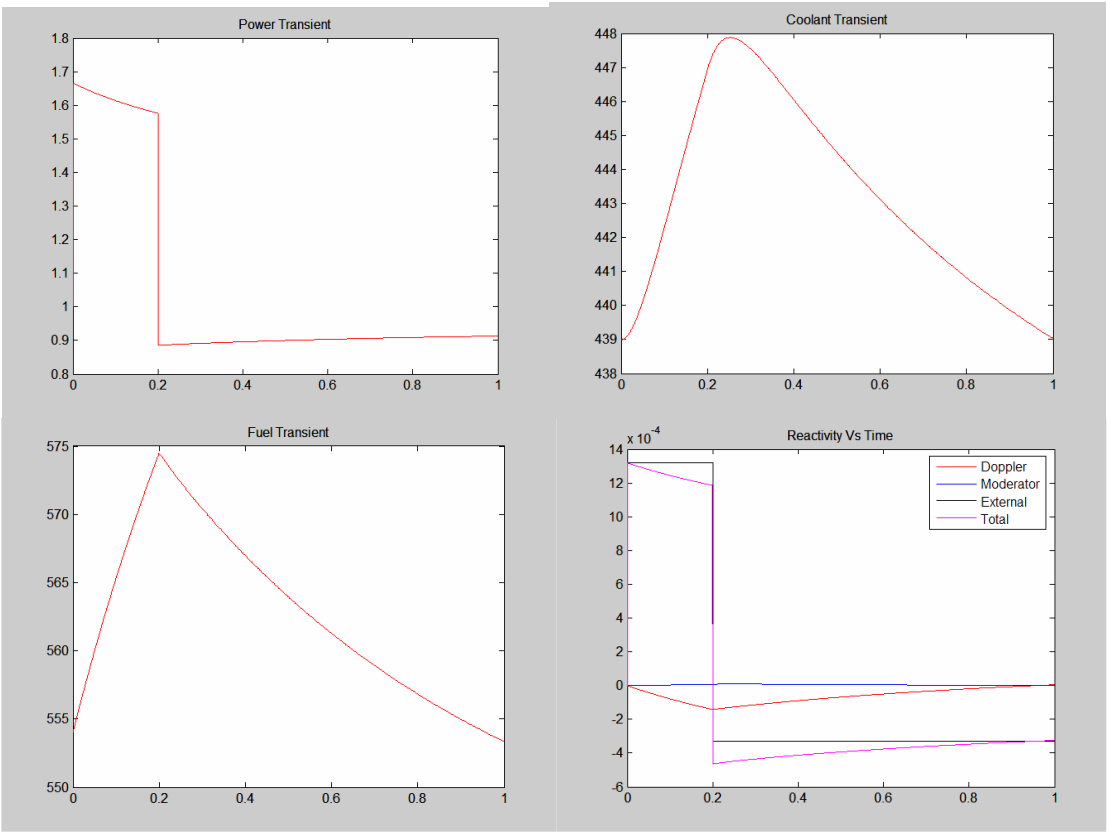

Figure 19: SFR: $0.4 rod ejection at t=0 followed by a $0.5 rod insertion at t=0.2: power level, coolant and fuel temperature, reactivity

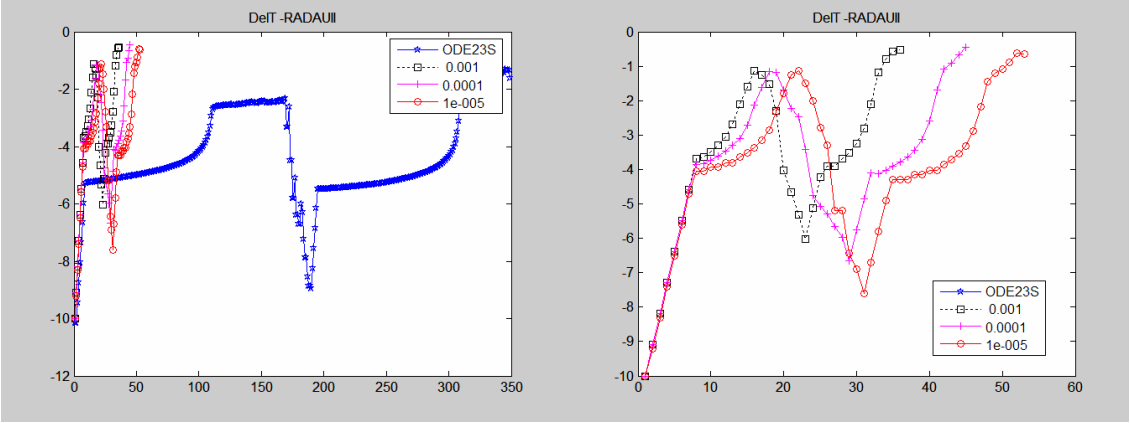

Figure 20: Log (Delta t) for step size control in Radau IIA, for 3 user-prescribed tolerances: *tol* =1E-3, 1E-4, 1E-5. Matlab reference (ODE23s is also shown).

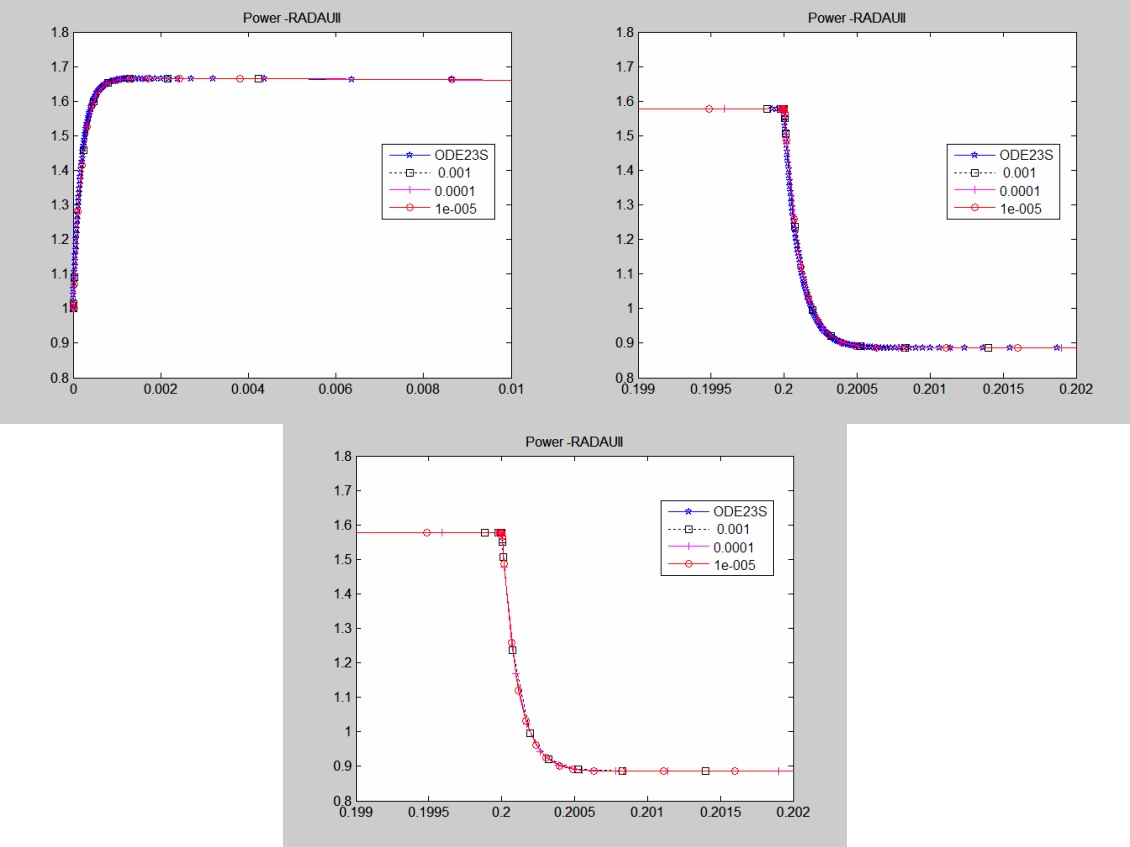

Figure 21: Power level for t∈[0,0.01] (top left) and around t=0.2 (time of the SCRAM, top right and bottom figures) for Radau IIA, for 3 different user prescribed tolerances: *tol* =1E-3, 1E-4, 1E-5. Matlab reference (ODE23s is also shown).

Table 6. Time-step comparison of Radau IIA and Rosenbrock GRK4A for different tolerances

Algorithm	**1.00E-03**	**1.00E-04**	**1.00E-05**
Radau IIA			
steps	53	64	77
successful steps	36	45	53
Rosenbrock GRK4A			
steps	82	89	109
successful steps	61	67	86

V Steps for Coupling UNIC to Nek5000

In the previous section we discussed numerical strategies for coupling multiphysics codes, in particular neutronics and thermal-hydraulics codes. In this section we describe the steps taken to couple UNIC and Nek5000 in order to have an integrated simulation of the rector core. Sample problems, addressing steady state solutions, are being also defined, that, even if limited in size, provide significant test cases for studying the coupling among the two codes.

NEUTRONXS, a general representation scheme for broad-group cross sections, has been developed for the coupling calculations for a wide range of applications. In the NEUTRONXS approach a user-definable cross-section table is used to store the microscopic cross-section information. Figure 22 shows an example having tabulated data with respect to burnup (Bu), fuel temperature (TF), and moderator temperature (TM). The ISOTXS tag is a standardized cross-section information file storage format for neutron transport methods and is used to indicate the type of microscopic data maintained in NEUTRONXS. At each tabulation point (Bu, TF, TM), the cross section can optionally have additional embedded polynomial dependencies of the cross-section data, such as enrichment. All of this cross-section data is stored in a random access file for rapid access. The most important advantage of the NEUTRONXS approach is its ability to selectively use either polynomial or tabulated cross-section data. Future development of this package will involve linking to the parallel netCDF library for storage of and access to the cross-section data in a parallel environment.

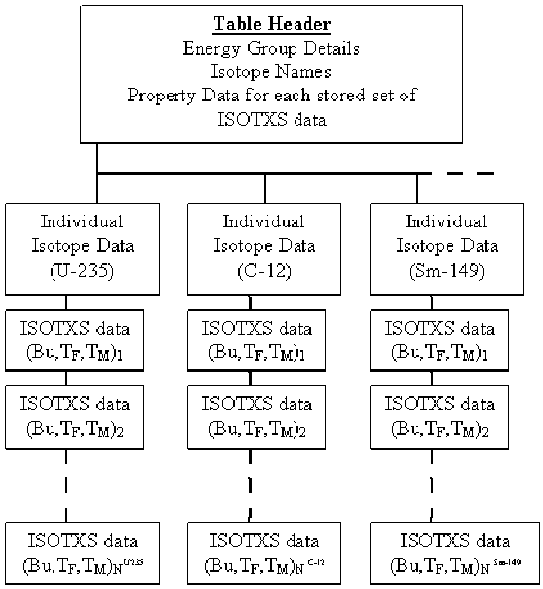

Figure 22. NEUTRONXS table setup.

In UNIC, we developed a general object-oriented interface to interpolate isotopes from the NEUTRONXS table and form a homogenized composition. Since the material density description is necessary for neutron transport problems, it was quite straightforward to include a list of correlated property information for each isotope (temperature, burnup, etc.), which is necessary for retrieving cross-section data from NEUTRONXS because

each material is mapped into individual geometrical zones in the finite element mesh. The NEUTRONXS approach is the primary means by which thermal-hydraulic feedback and fuel depletion effects currently are managed in the UNIC code. Our objective is to fully interpolate the temperature and coolant density solution data from the Nek5000 code to the UNIC code by means of these data structures.

With regard to the thermal-hydraulics calculations, we have identified several steps to prepare Nek5000 for coupling to UNIC. The first step is to prepare multiple interfaces to Nek5000 that allow one to set up a problem (read the mesh, define the geometry, prepare the preconditioners, etc.), exchange solution data, repeatedly integrate for a prescribed length of time or to a prescribed condition (for steady state), and output results. The second step is to update the communication routines in Nek5000 so that they do not rely on mpi_comm_world. This change is required in order to decouple communications between Nek5000 and any other MPI-based application that is running at the same time. (This process is near completion; there remains one additional routine to modify.) The third step is to develop prototype (Matlab) and production routines that provide the material properties to the code as a function of temperature. The final step is to develop and validate a Matlab code that computes single-pin axial and radial temperature distributions based on enthalpy fluxes that provides a coarse validation (sanity check) for the detailed thermal hydraulics computations.

V.1 Specifications of Sample Coupled Problems

In this section we define simplified problems for testing the functionality of the coupled physics calculations. The intention here is to check the validity of the coupling of the two codes, rather than obtaining physical solutions. Based on the information obtained from the ABR design group, we decided to compute two problems for the steady state solution of a coupled neutronics and thermal-hydraulics sodium-cooled fast reactor: a simplified problem with the first two rows of pins, and a more complex problem with one entire subassembly. In both cases the problem is limited to one-sixth of the geometry in view of the existing symmetry. For simplicity, variations in cross sections due to the temperature are considered only in the fuel zones and variation of the density only in the coolant zones.

V.1.A First Problem

As shown in Figures 23–25, the first problem has only one-sixth of the central pin and two halves of a pin belonging to the second row. A reduced (to minimize impact on K_{eff}) thickness wall is added at the boundary of the geometry. Axially the configuration has: 80 cm of active core zone, on the bottom the lower reflector (20 cm), which implies that HT9 replaces fuel inside the pin clad and on the top, the upper plenum (20 cm) with sodium replacing the fuel inside the pin clad. Boundary conditions are reflective on all the horizontal directions and void vertically. The densities used for the different regions are shown in Table 7. Cross sections have been generated for the 850 K nominal fuel temperature in a five-group energy structure; if memory allows, 9-group or 33-group

cross sections could be quickly generated. Two extra sets of cross sections for 750 K and 950 K have been also generated for interpolation during the coupled calculation. The total power for normalizing the flux for this problem is 22,400 watts. The initial condition for the thermal-hydraulic calculation is an inlet coolant temperature of 355 °C; thermal conductivities for metal fuel (U-Pu-Zr), HT9 and sodium, as sodium density temperature dependence, have been provided by the ABR design group. Dimensions for the wire wrap radius and pitch also have been provided by the design group. Knowing that the outlet coolant temperature should be around 510 °C , averaging along the axial profile should give a good initial guess for both temperature distribution and flow rates. This should speed up the convergence for the steady-state calculation. The thermal hydraulic calculation should provide the temperature distribution and density coolant variation along the z-axis averaged every five centimeters and horizontally averaged in each pin region and coolant around it in the basic hexagon associated to a pin (see Figures 23 and 25).

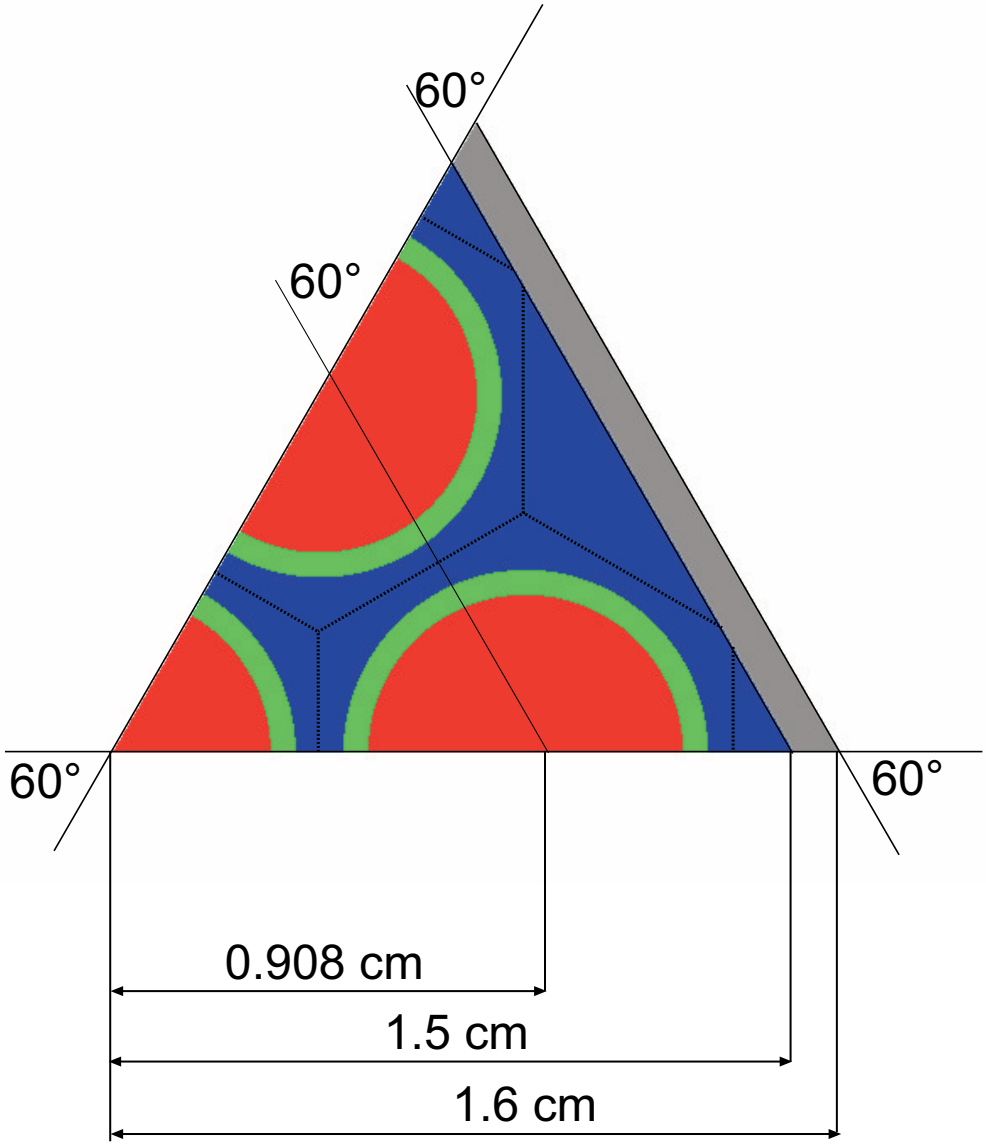

Figure 23. Problem 1 radial geometry.

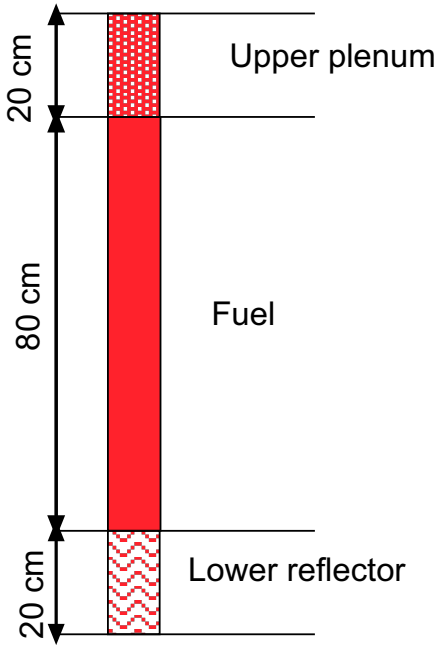

Figure 24. Problem 1 axial geometry.

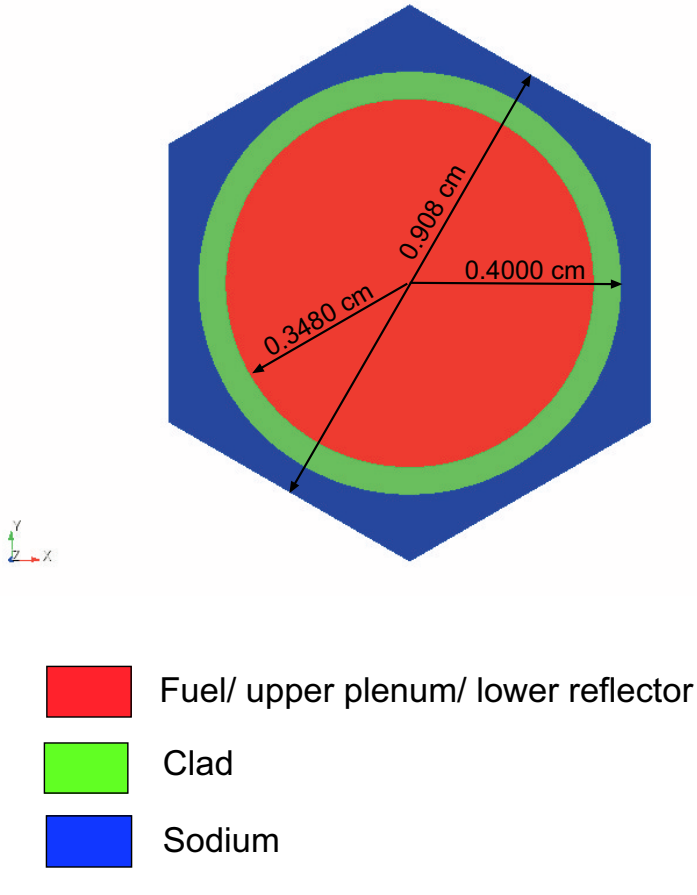

Figure 25. Pin radial geometry.

Table 7. Nuclide densities

Nuclide	Nuclide Name on Cross Section File	Fuel	Structure (Clad, Hex can wall)	Coolant
		U-Pu-Zr	HT9	Sodium
U235	U235IC	5.6711E-05		
U-238	U238IC	2.7941E-02		
Pu238	PU38IC	5.5444E-07		
Pu239	PU39IC	5.1794E-03		
Pu240	PU40IC	3.1944E-04		
Pu241	PU41IC	1.9163E-05		
Pu242	PU42IC	1.0905E-06		
Zr	ZIRCIC	9.7253E-03		
Na	NA23IC			2.2272E-02
Fe	FE IC		7.0426E-02	
Ni	NI IC		4.1315E-04	
Cr	CR IC		1.1247E-02	
Mn55	MN55IC		4.7156E-04	
Mo	MO IC		2.8819E-04	

V.1.B Second Problem

See Figure 26 for the radial geometry; the axial geometry stays the same as for problem 1. The specifications also are as for the first problem, but with the geometry extended to the full assembly with one-sixth symmetry, which corresponds to a total of 36 1/6 pins. Note that this time the wall thickness of the hex can is the actual one used by the ABR design group. Total power to be used for normalizing the flux is 694,400 watts**.**

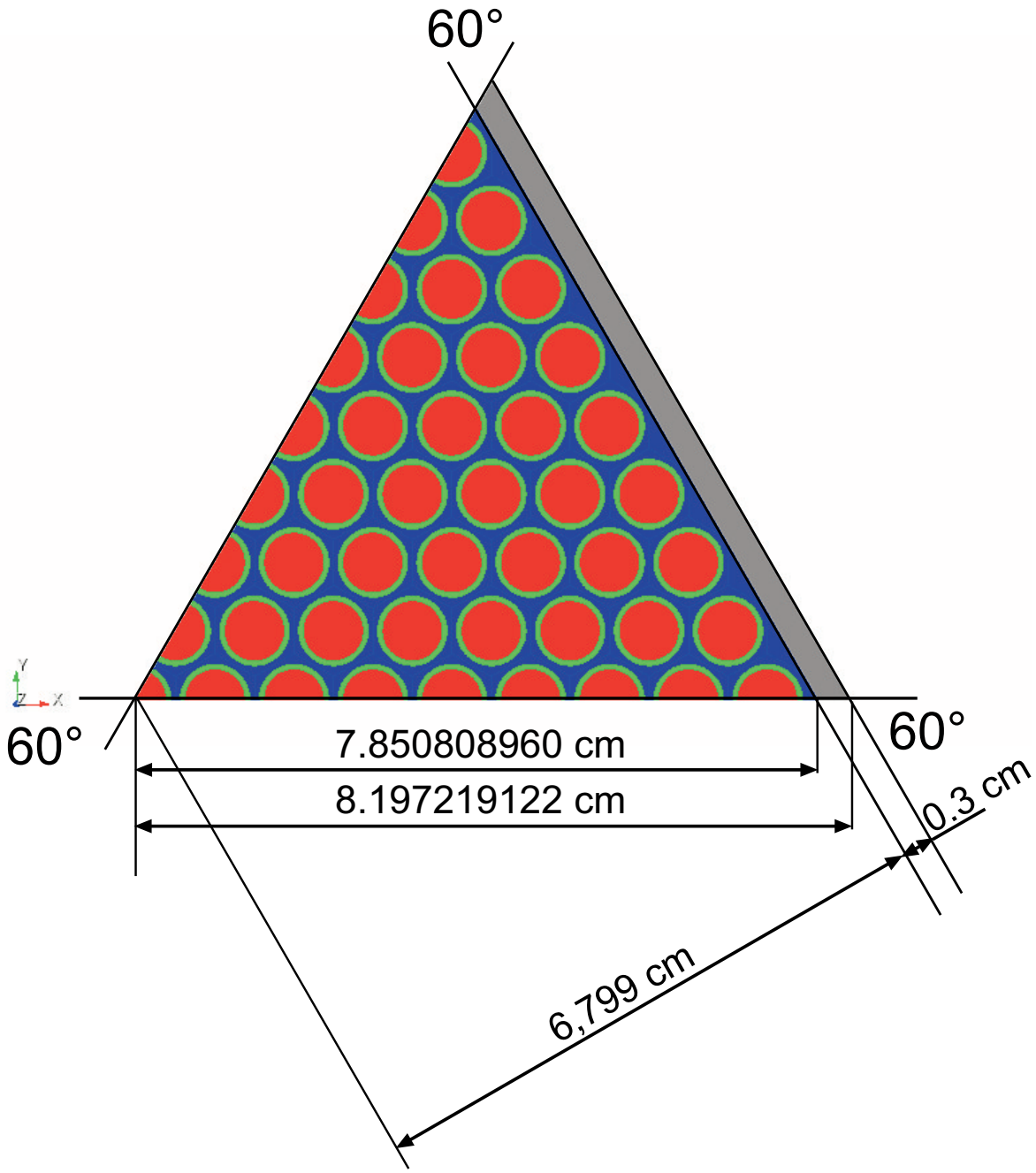

Figure 26. Problem 2 radial geometry.

V.1.C Results

Calculations for the neutronics code have been run on a serial machine, but because of memory and time limitations only the first problem was carried out. Table 8 shows the eigenvalues up to the P_{11} angular approximation, and Figure 27 shows the power distribution at different axial heights.

From Table 8, we can infer that the convergence on the angular variable will occur around P_{17} or P_{19}. This number will be bigger when a larger number of groups are used. The differences between linear and quadratic meshing increase when more refined angular approximations are used. It is believed that quadratic mesh approximation should be adequate if a sufficient number of meshes are used in describing the fuel region.

Table 8. UNIC computed eigenvalues for sample problem 1.

P_N	Linear Mesh	Quadratic Mesh
1	1.36479	1.36519
3	1.40524	1.40660
5	1.41874	1.42059
7	1.42539	–
9	1.42936	–
11	1.43180	–

.

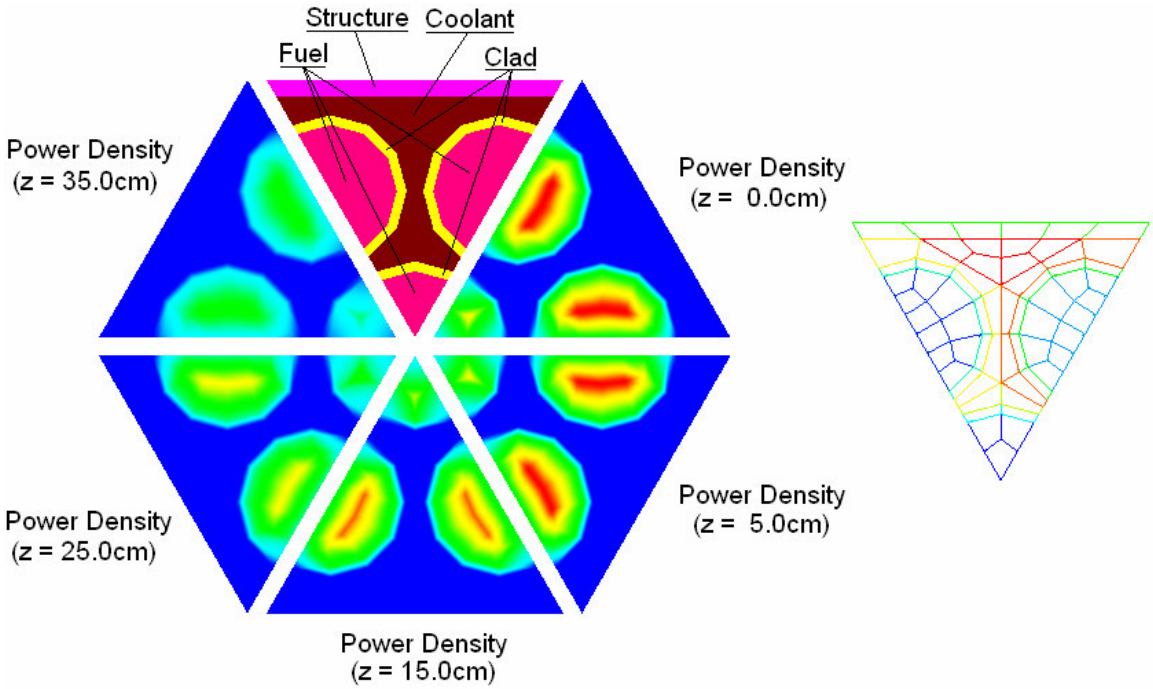

Figure 27. UNIC power distribution at different heights (z=0.0 is midplane) for sample problem 1.

Following the UNIC neutronics calculations, the power distribution was interpolated over the mesh structure defined for the Nek5000 code in order to calculate the temperature distribution of sample problem 1. Figure 28 shows the Nek5000 temperature distribution

in the pin, clad, and liquid sodium at several cross-sections for the three-pin model problem.

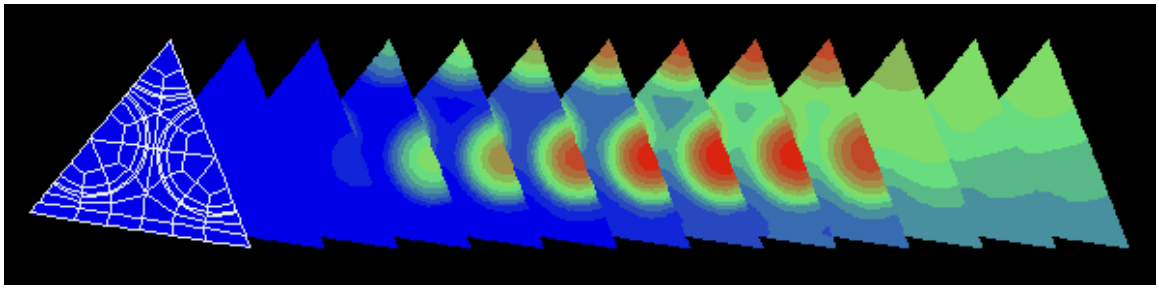

Figure 28. Temperature distribution for a Nek5000 thermal-hydraulics computation based on UNIC-generated power data. The spectral element mesh distribution is visible in the leading slice.

VI Software Design

Software *architecture, process,* and *framework* are heavily overloaded terms that lead to considerable confusion without a careful discussion of their meaning in a given context. For the current project, we define these terms as follows.

By software *architecture* we generally refer not to specific physics implementations (i.e., the solvers themselves) but rather to a set of rules and formalities that describe how the subcomponents of a larger integrated project communicate with one another—how data is shared, how one implementation can be swapped for another, how modules are unit tested, where platform-specific code can be placed, and so forth. The concept of architecture presumes a larger, integrated whole composed of many subpieces (components, modules, etc.), and the architectural specification defines how these components interact independently of how the details of how they are implemented.

By software *process* we generally refer to a set of formalities that enable large group projects to move ahead efficiently. Bug tracking, automated testing, meeting schedules, code review, repository management, and distribution are common examples. Overall, this set of processes distinguishes small cottage-industry coding efforts from large, integrated software projects and is indispensable for producing code on the scale of the design effort here. What are the roles of the various developers? How are decisions ultimately made? What are the coding standards? How is documentation produced? While research environments necessarily eschew too rigid a set of processes, some significant degree of formality is necessary to keep the process moving forward. We have chosen specific tools in each of the relevant areas that we are putting into place as the project ramps up.

Framework generically refers to any set of services that are provided to a set of implementations to augment their capabilities. This is the most general of the three terms (see Figure 29) and can take many forms depending on the needs of the application: parallel solvers for systems of equations; meshing capabilities; loose coupling for multiphysics applications; visualization, and so forth. Usually, framework implies some degree of end-to-end services, so, typically, a framework that emphasizes any one of the above services may provide at least simple versions of the other services; for example, a solver framework such as PETSc might provide some simple plotting tools.

As an important part of our initial study for the fast reactor simulator, we have evaluated existing technologies in these areas and have made some decisions about which tools and approaches to adopt.

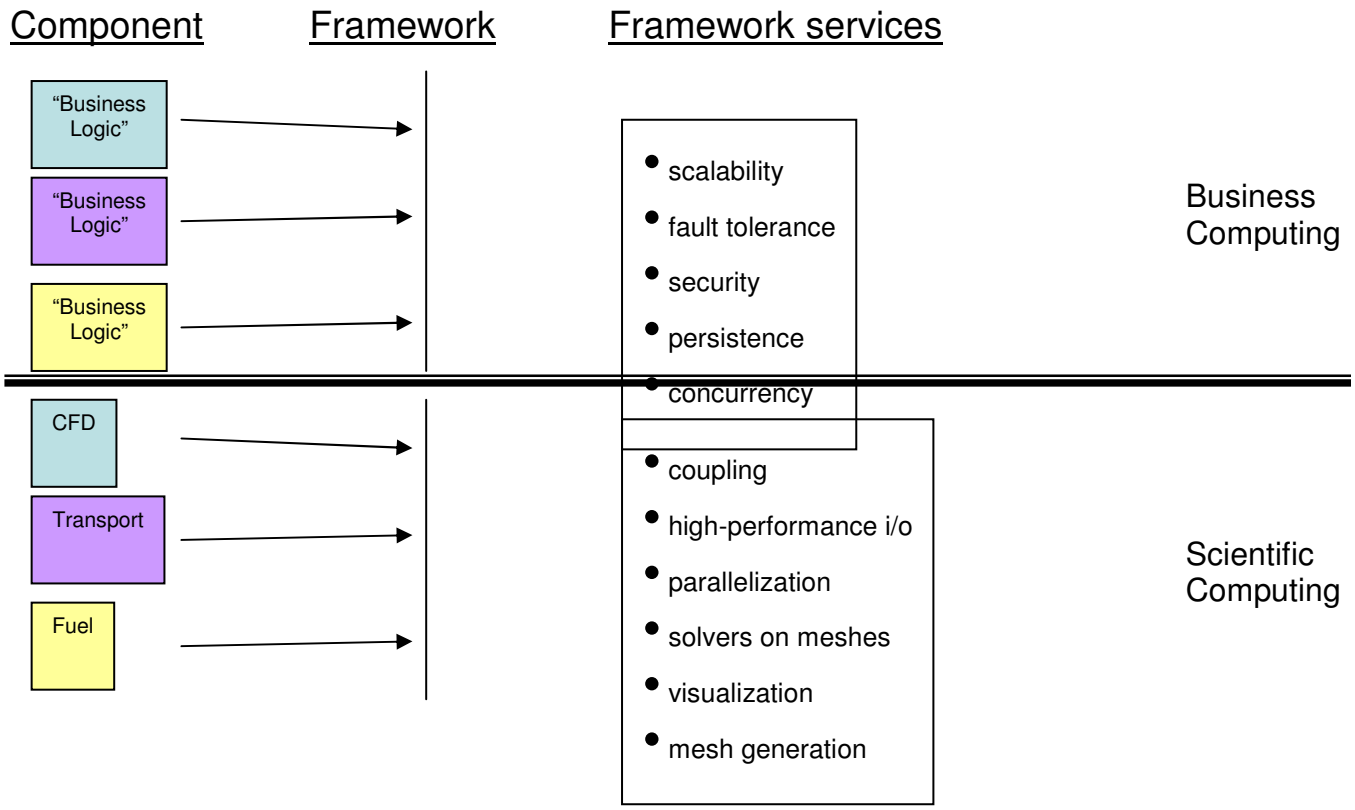

Figure 29. Generic depiction of the role of a software framework in the development of multicomponent software. Note that for scientific computing the decoupling between framework services and component implementation is less clean and is dictated by the degree of decoupling in the numerics.

Our ultimate project goal is the development of a fully integrated, state-of-the-art high-fidelity fast reactor core simulation tool. This necessarily implies (1) a strong emphasis on leveraging existing state-of-the-art computational hardware and (2) a group software process and software architecture that will enable large code development with a heterogeneous development team (likely over 30 contributors and 1 million lines).

Point (1) is true because our state-of-the-art simulation tool aims to reduce the use of traditional empirical correlations to the greatest extent possible and move toward the solution of more fundamental governing equations. Hence, extremely fine spatial and temporal resolutions are needed that go beyond what can be achieved even on the currently proposed petascale architectures (e.g., BG/P and Cray Cascade). Point (2) recognizes that, when simulation codes move beyond several hundred thousand lines, the architectural complexity begins to compete with the individual module algorithmic complexity as the main barrier to progress. Twenty years of research in software engineering and software process has illuminated this point clearly for industrial applications. Relatively little work in this area has been done for high-performance scientific computing, however.

Additionally, we require that the integrated simulation tool be designed both to produce results with current state-of-the-art models and to be flexible enough to easily incorporate improved models as they become available. This requirement is particularly important when considering licensing and certification. Semi-empirical models that are well validated for certain regimes (*computational paths)* will be used while verification and validation research is carried out for next-generation models. When new modules are validated, they can replace or live alongside existing modules with the same coupling capabilities to peer modules.

Our initial finding is that many large multiphysics software projects have failed because they have overemphasized and decoupled the framework design too much from the ultimate simulation goals of the project. That is, one-size-fits-all frameworks, which are overly sophisticated and based on overly general and complex abstractions, have interfered with desirable traits such as performance, usability, simplicity, and memory footprint and have placed the elegance of the architecture ahead of the physics and engineering requirements. Architectural features must demonstrate that they serve both the short- and long-term simulation goals of the project. At the same time, one must move out of a familiar comfort zone to avoid going to the other extreme—"spaghetti code," which is fragile, unreadable, and difficult to unit test, with no clear integration strategy and no path to growth (i.e., all implementation and no architecture).

At the highest level the main physics modules for the integrated reactor core simulation are thermal hydraulics, neutronics, structural mechanics (for fuel pins, hex cans, etc.), and fuel behavior. As reported, our initial studies have focused on neutronics and thermal-hydraulics modules. These form a set of nonlinearly coupled PDEs where considerable care has to be taken in the numerical aspects of coupling for specified accuracy requirements (see Section III). A *software coupling* strategy must accompany and be consistent with the numerical coupling strategy. The software coupling can be viewed as defining a clear set of public *interfaces* by which each module communicates its data to other modules. That is, a formal specification is required for each module in terms of its accessible method and its input and output parameters; the only "window" into each module is via this public interface. This level of encapsulation ensures that each module can be developed and tested independently and that changes to one module do not affect another.

This approach is far more complex than it initially appears for several reasons: (1) data lives on a computational mesh, and each module needs flexibility to define its own mesh (e.g., neutronics and thermal hydraulics require resolution in very different spatial locations); (2) performance is often a first-order concern, and copying or restructuring mesh data can be prohibitively costly; (3) sufficiently general interfaces are often difficult to define for numerical components; (4) operator split methods are often inadequate for given accuracy requirements and tighter coupling (e.g., Jacobi-free Newton Krylov) is required; and (5) coupling must be *scalable* to O[100K] processors, a scaling that is not an obvious extension of the serial case.

A *coupling toolkit* should provide high-level abstractions to define the relationships among specified components and to implement the data exchange between them, an effort that includes understanding or defining the computational mesh for each module, providing tools to extrapolate or interpolate data between meshes, and providing multiple parallelization strategies for carrying this out in a scalable way (for example, each module could share all processors in a communicator, be split across communicators, etc.).

We evaluated three tools as possible candidates for adoption in the fast reactor simulation project: Salome, the Model Coupling Toolkit (MCT), and MpCCI. Salome (see Figure 30) is an open-source GUI-based tool that includes both pre- and postprocessing tools as well as a high-level coupling interface. It is developed by the French Nuclear Agency (CEA) in collaboration with industry, targeting in particular nuclear applications (but in theory applicable to other areas as well). MCT is an open source project that arose out of the coupled climate modeling project and evolved into a more general tool. MpCCI is a commercial project. Both MCT and MpCCI are designed specifically for parallel applications, while Salome has no deep built-in concept of parallelization (though one could imagine crudely coupling parallel components and doing much of the coupling by hand).

In summary, our evaluation indicates that none of the above tools, as they stand, meet the requirements of our current project. Commercial tools are in general a bad choice given that we target leadership-class computers with semi-exotic architectures (e.g., BG/P), and commercial applications have shown little interest in quickly adapting and optimizing for these architectures; restricting ourselves with such a choice is too risky. MCT is not sufficiently general in its handling of complex mesh types and is not directly suited for nuclear applications. Salome is the most promising tool, with sophisticated software component design capabilities, but we found the following design choices made it imperfect for our needs: (1) a reliance on CORBA to provide very coarse-grained module coupling; (2) no built-in parallelization capabilities; (3) no built-in data interpolation capabilities; and (4) too heavyweight for our purposes, locking in choice of pre- and postprocessing tools and persistence model and requiring an intricate and delicate build process with over a dozen fairly complicated set of library dependencies. We also found the user interface portion of Salome cumbersome and limited.

We have therefore chosen to develop our own lightweight coupler and loosely integrate it with existing solvers and pre- and postprocessing tools to form a complete software infrastructure for our solver modules. We have begun design of this coupler in close coordination with meshing experts and borrowing heavily from ideas put forth by the CEA and the TSTT SciDAC project. We intend to flexibly support several methods of software coupling on highly scalable architectures such as BG/P. Furthermore, as mentioned early in this report, we have chosen the PETSc toolkit as our parallel solver engine, CUBIT as a preprocessing tool, and tentatively VISIT as a postprocessing tool. We intend to partially fund work in each of these areas to evolve the tools to meet the specific needs of the current project.

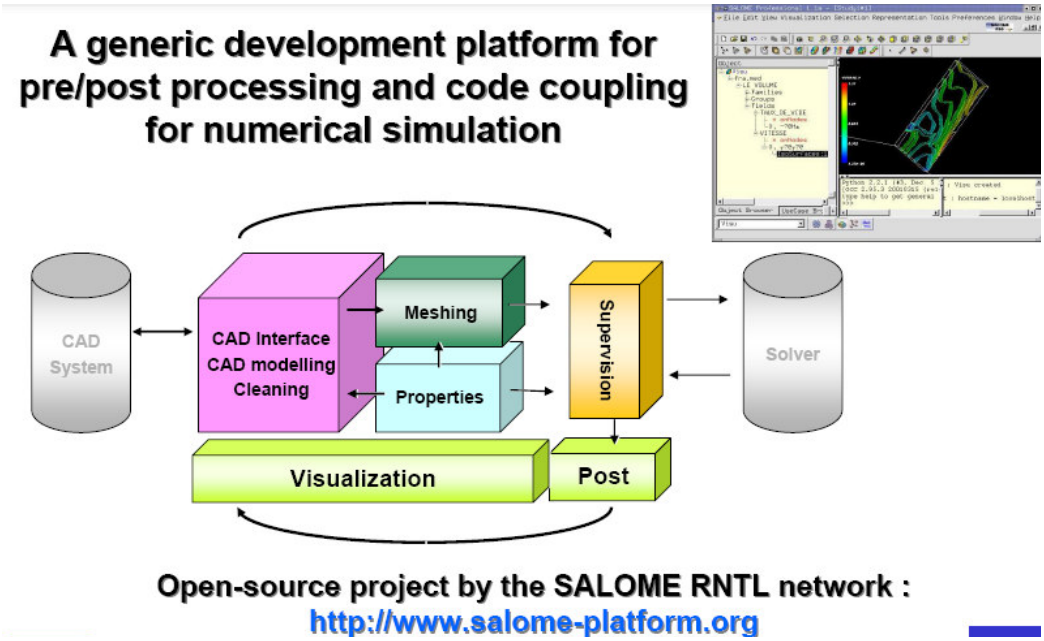

Figure 30. Architecture of the Salome coupling framework.

VII Conclusions

This report has presented the effort under way at Argonne to develop a comprehensive, integrated computational tool intended mainly for the high-fidelity simulation of sodium-cooled fast reactors. Our principal activities have focused on neutronics, thermal hydraulics, coupling strategies, and software architecture.

Neutronics. For the neutronics UNIC code, we investigated a spherical harmonic finite element method. The details of both the spatial and angular approximation were considered, along with the two boundary conditions typically applied for neutron transport problems. The interfacing of this method with a method of characteristics domainwas also evaluated. We implemented the spherical harmonics equations in a stand-alone code and used the code to solve several benchmark problems. Overall, the new spherical harmonics code performs well, although more development is necessary. Specifically, refinements of the code structure and better implementation of the PETSc package should alleviate the current problems with the space-angle limitations. After these refinements are made, the code should be optimized for a parallel environment. This optimization will allow for the modeling of more complex benchmark problems in addition to producing more accurate solutions. After this work is completed, the interfacing method proposed in Section II can be further investigated.

Thermal hydraulics. Nek5000, a computational fluid dynamics code for coupled hydrodynamics and heat transfer developed in the MCS Division, was applied to a single-pin, periodic cell, in the wire-wrap geometry typical of ABR. The results of the simulations with this code will be used to provide hydrodynamic building blocks for thermal transport. The simulations will be coupled with Lagrangian particle tracking that will allow quantification of interchannel mixing (modulo wall effects). The simulations will also provide accurate flow fields that can be directly coupled with larger (i.e., multichannel) thermal hydraulics computations or used to provide accurate mean and fluctuating flow fields that can serve as surrogate flow fields.

Coupling. Conventional strategies for coupling multiphysics codes are at best first-order accurate because of the poor treatment of the implicit nonlinear terms. Higher-accuracy methods will be required to finely simulate coupled neutronic/thermal-hydraulic reactor transients. Implicit Runge-Kutta methods offer such possibilities at a potentially reasonable cost thanks to their automatic step-size control features. Two implicit Runge-Kutta methods were tested here: a full implicit Radau method and a singly diagonal implicit Rosenbrock method. Both methods performed well and automatically captured extremely sharp transients. The Radau method, whose implementation was more complex, delivered a better performance in terms of function evaluations because of its higher-accuracy order. For both methods, the Jacobian matrix was calculated numerically. Jacobian-free Newton Krylov techniques will have to be analyzedmodified in order to reduce the cost associated with the computation of the Jacobian matrix. Another possibility is to devise a selective update for the Jacobian matrix so that it would not be re-evaluated at each time step. We have taken steps to couple UNIC and Nek5000 in order to have an integrated simulation of the reactor core. Simplified problems,

addressing steady-state solutions, have been defined that, while limited in size, provide significant test cases for studying the coupling of the two codes. We have already gained insights about mesh approximation from the first sample problem.

Software. In the field of software design we have evaluated existing technologies in related areas and have decided which tools and approaches to adopt. Unfortunately, none of the existing tools, as they stand, meet the requirements of our current project. Therefore, we have chosen to develop our own lightweight coupler and loosely integrate it with existing solvers and pre- and postprocessing tools to form a complete software infrastructure for our solver modules. We have begun design on this coupler in close coordination with meshing experts and borrowing heavily from ideas put forth by other projects.

During the next fiscal year we will build the integrated high-fidelity software tool. Inclusion of other physical phenomena, such as structural mechanics and nuclear fuel behavior, will be essential for a better simulation of the reactor core.

References

1. E.E. Lewis and W.F. Miller Jr., Computational Methods of Neutron Transport, John Wiley & Sons, New York (1984).
2. M. A. Smith, G. Palmiotti, E. E. Lewis, and N. Tsoulfanidis, "An Integral Form of the Variational Nodal Method," Nucl. Sci. Eng. 146, 141 (2004).
3. G. Palmiotti, E. E. Lewis & C. B. Carrico, "VARIANT: VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation," Argonne National Laboratory Report ANL-95/40 (1995).
4. E. E. Lewis, G. Palmiotti, and T. Taiwo, "Space-Angle Approximations in the Variational Nodal Method," in Proc. Int. Conf. Mathematics and Computations, Reactor Analysis and Environmental Analysis in Nuclear Applications, Vol. 1, 827. September 27-30, Madrid, Spain (1999).
5. C. B. Carrico, E.E. Lewis and G. Palmiotti, "Three Dimensional Variational Nodal Transport Methods for Cartesian, Triangular and Hexagonal Criticality Calculations," Nucl. Sci. Eng. 111, 168 (1992).
6. M. A. Smith, N. Tsoulfanidis, E. E. Lewis, G. Palmiotti, and T. A. Taiwo, "A Finite Subelement Generalization of the Variational Nodal Method," Nucl. Sci. Eng. 144, 36 (2003).
7. M. A. Smith, E. E. Lewis, Byung-Chan Na, "Recent Benchmarking Experience of the OECD/NEA Expert Group on Three-dimensional Radiation Transport," Supercomputing in Nuclear Applications, Paris France, September 2003 (2003).
8. M. A. Smith, N. Tsoulfanidis, E. E. Lewis, and G. Palmiotti, "Comparison of Angular Approximations for PWR Cell Calculations," Trans. Am. Nucl. Soc. 84, 90 (2001).
9. T. Takeda and H. Ikeda, "3-D Neutron Transport Benchmarks," NEACRP-1-300 OECD/NEA, Organization of Economic Cooperation and Development/Nuclear Energy Agency (March 1991).

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC