ANL/NE-12/46

Development Status of PROTEUS-MOC

Nuclear Engineering Division

ANL/NE-12/46

Development Status of PROTEUS-MOC

A. Marin-Lafleche, M. A. Smith, E. E. Lewis, C. H. Lee
Nuclear Engineering Division
Argonne National Laboratory

September 28, 2012

SUMMARY

The PROTEUS-MOC solver is intended to resolve the outstanding computational challenges that the PROTEUS-SN is limited by and bridge the gap between PROTEUS-SN and existing homogenized structured grid solvers. The primary push behind all of this work is to carry out coupled thermal-mechanical, fluid dynamics, and neutronics physics problems. The PROTEUS-MOC development has taken advantage of the combined experience with PROTEUS-SN and DeCART with the goal of producing a tool which combines the best features of both tools.

Taking the DeCART tool as the model, it merges a subgroup cross section interpolation technique with a 2D method of characteristics approach and a 1D SN formulation. DeCART has been specifically demonstrated to be accurate on PWR and VHTR problems, but it has displayed several issues with regard to accuracy on thin axial mesh sizes which are attributable to the 2D1D scheme it is built upon. There are also notable problems with performance and scalability all of which we hope to resolve in PROTEUS-MOC.

PROTEUS-MOC is at its core, based upon the success achieved with MOCFE which demonstrated excellent parallelization with respect to angle and good parallelization with respect to space. More recent efforts have indicated parallelization with respect to energy is also feasible. The work completed yielded a novel methodology where the axial basis functions are treated separately from the radial ones, but the geometry is extruded. This eliminates the need to carry out and store 3D ray tracing data and simultaneously allows the coarse axial node sizes typical with DeCART. To rapidly build a prototype, a standard power iteration algorithm was constructed using an unpreconditioned GMRES solution algorithm for solving the flux vector in each fission source iteration.

The new PROTEUS-MOC solver was deployed on two test problems: Takeda and a heterogeneous cylindrical heavy water system. The Takeda benchmark problem is a standard homogeneous test problem which was previously found to be too difficult for the 3D MOCFE code to solve. Using the PROTEUS-MOC we were able to show space-angle convergence although full convergence requires further improvements in performance. The second benchmark problem was constructed more for future work than this year, but the results generated are promising indicating that it will be very useful in highlighting the weaknesses and strengths of the PROTEUS-MOC tool.

While we are far from a production tool, the results included in this report give us confidence that we are on the right path to constructing an interceding code between the plethora of structured grid homogenized assembly solvers and the supercomputer consuming PROTEUS-SN code. With time, the creation of a suitable preconditioner along with typical optimization should reduce the performance shortcomings that are observable today.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The PROTEUS-MOC solver is intended to resolve the outstanding computational challenges that the PROTEUS-SN is limited by [1,2,3,4,5] and bridge the gap between PROTEUS-SN and existing homogenized structured grid solvers [6,7]. The primary push behind building the PROTEUS-SN solver is to couple the complex thermal-mechanical and fluid dynamics physics with neutronics for various off-nominal operational scenarios. The development of PROTEUS-SN is an ongoing research effort and has to date been closely reliant upon the availability of a supercomputing machine. Given the cutting edge nature of its development, it has yet to prove its ability to routinely perform the desired heterogeneous full core calculations of NEAMS, although it has demonstrated immediate application to classic assembly homogenized methodologies [8,9]. Ignoring some of the unresolved cross section issues (a topic of research), the problem size is the primary problem that plagues the PROTEUS-SN code today.

A full core heterogeneous neutronics calculation targeted by PROTEUS-SN is expected to require upwards of 100 trillion degrees of freedom (per time step) which is beyond the capability of existing supercomputing machines. In contrast DeCART [10] has demonstrated success by routinely providing good solutions to lower resolution heterogeneous problems in a coupled environment. Further, there are existing homogeneous structured grid solvers such as DIF3D [6] which routinely perform similar calculations in mere seconds. At issue of course is the requirements put on solution fidelity and proof of reliability on a wide range of reactor designs with regard to both of these tools. In the case of DIF3D (and other similar tools), they have been demonstrated to work on steady state, normal mode reactor operations, such that routine analysis and core depletion predictions (their primary purpose) are accurate and reliable on a wide range of reactors: LMFBR, HWR, PWR, BWR, VHTR, etc. Of course when one wants to study the dynamics behavior where nominal temperature and coolant flow rates are not assumed, then one immediately runs into difficulties as a homogenized neutronics model does not explicitly include the geometry. While one typically assumes the cross section homogenization (equivalence) procedure accounts for these effects, it introduces significant uncertainty (and bias) in the resulting dynamics calculation.

The DeCART tool addresses the dynamics issue by merging a subgroup cross section interpolation technique [11] with a 2D method of characteristics approach (heterogeneous planer geometry) and a 1D SN formulation (hence 2D1D). DeCART has been specifically demonstrated to be accurate on PWR and VHTR problems, but it has displayed several issues with regard to accuracy on thin axial mesh sizes arising from the 2D1D scheme it uses (Richardson iteration between the two systems) and performance with regard to problem size. DeCART is also limited with regard to the type of geometry it can handle given that it has an underlying structured grid (hexagonal prism or Cartesian grid) which is of secondary concern. The typical DeCART calculation defines a +1 trillion DOF problem, but in contrast to PROTEUS-SN, the solution algorithm does not focus on getting the fine level system fully converged. Instead, DeCART projects a loosely converged fine mesh solution onto a coarse grained coarse-mesh finite-difference (CMFD) grid from which it obtains an eigenvalue/eigenvector solution. Note that the CMFD grid size is typically set to be a single pin-cell. This is by no means a bad decision, but just a rational one given the computational expense associated with solving a +1 trillion DOF problem on limited computational

resources. The resulting ~1 billion DOF problem is much more manageable to solve on a small workstation or parallel computer. Several of the features of DeCART are desirable, but the lack of a flexible geometry and its performance issues continue to hamper its routine usage. As one example, DeCART was intended for thermal reactor systems and while it does handle basic fluid-dynamics changes, it can only be done on a coarse grained level (10-20 cm axial node size) which restricts it to a few large production reactor designs of interest.

The focus of the PROTEUS-MOC work is to combine the knowledge gained from development efforts on PROTEUS-SN and DeCART along with our previous work on MOC [1] to create a more robust tool without the performance problems of either PROTEUS-SN or DeCART. The first step in that process is to take advantage of the DeCART geometry concept of building an extruded geometry, but use the unstructured mesh treatment from PROTEUS-SN to define the planer geometry. This allows complex geometries such as the Advanced Test Reactor [12] shown in Figure 1 to be modeled with PROTEUS-MOC.

Figure 1. Two-Dimensional ATR Geometry and Mesh.

With regard to the transport equation, the focus of this fiscal year was to develop and demonstrate a methodology that does not suffer the algorithmic problems of DeCART. This was accomplished by building PROTEUS-MOC on a fully three-dimensional grid where a discontinuous finite element method is used axially and the MOC is used in the plane. This approach allows relatively fine axial meshing (2-8 cm meshes) which should resolve the algorithmic problems of DeCART and allow a more detailed thermal-hydraulic coupling.

The following section gives a brief overview of the innovative methodology used in PROTEUS-MOC. The results from two separate benchmark problems are provided to demonstrate the methodologies validity. The first benchmark is a simple homogeneous example which is small enough to assess the space-angle convergence and computational performance of the methodology. The second benchmark is a heterogeneous problem derived

from a CANDU reactor and was built to show the ability of PROTEUS-MOC to handle arbitrary geometry problems like the ATR.

2 Methodology

Similar to PROTEUS-SN, PROTEUS-MOC was developed to solve the steady-state neutron transport equation cast into the common k eigenvalue mode [13].

$$\vec{\Omega}\cdot\vec{\nabla}\Psi(E,\vec{\Omega},\vec{r})+\Sigma^t(E,\vec{r})\Psi(E,\vec{\Omega},\vec{r})=\iint dE'd\vec{\Omega}'\Sigma^s(E'\to E,\vec{\Omega}'\to\vec{\Omega},\vec{r})\Psi(E',\vec{\Omega}',\vec{r}) \\ +\frac{1}{k}\int dE'\nu(E'\to E,\vec{r})\Sigma^f(E',\vec{\Omega},\vec{r})\Psi(E',\vec{\Omega},\vec{r}) \quad (1)$$

where $\Psi(E,\vec{\Omega},\vec{r})$ is the flux vector, $\Sigma^t(E,\vec{r})$ is the total cross-section, $\Sigma^s(E'\to E,\vec{\Omega}'\to\vec{\Omega},\vec{r})$ is the differential scattering cross-section, $\Sigma^f(E',\vec{\Omega},\vec{r})$ the fission cross-section and $\nu(E'\to E,\vec{r})$ is the neutron yield per fission. The fundamental mode or maximum eigenvalue k_{eff} and its associated eigenvector are the typical goals of solving equation 1. Given a solver which provides that solution, one can easily transform the solver to work on a fixed source steady state problem or various formulations of the time dependent transport equation. For demonstration purposes, PROTEUS-MOC uses standard power iteration [13] to obtain a solution to equation 1 noting that large thermal reactor systems have high dominance ratios and thus require acceleration of some form.

2.1 Discretization of the neutron transport equation

The neutron angular flux, $\Psi(E,\vec{\Omega},\vec{r})$, is a function of three variables in the steady state regime. These are the energy of the neutron (E), the angular direction of travel of the neutron ($\vec{\Omega}$), and the position of the neutron ($\vec{r}$). Solutions for the neutron flux are obtained by solving the Boltzmann equation which can be written as follows in the steady-state regime:

$$\vec{\Omega}\cdot\vec{\nabla}\Psi(E,\vec{\Omega},\vec{r})+\Sigma^t(E,\vec{r})\Psi(E,\vec{\Omega},\vec{r})-W(E,\vec{\Omega},\vec{r})=S(E,\vec{\Omega},\vec{r}) \quad (2)$$

where $W(E,\vec{\Omega},\vec{r})$ is the scattering source term and $S(E,\vec{\Omega},\vec{r})$ is the fission source term which is fixed for a given power method (outer) iteration. In order to solve equation with a deterministic code, the three variables have to be discretized.

The energy variable typically ranges from 10^{-5} eV to 10 MeV and is discretized using the multi-group methodology (energy group) noting that the number of energy groups is problem dependent. Integral flux, source and cross-sections quantities are defined over each energy group transforming the Boltzmann equation into a set of G (=number of groups) equations:

$$\vec{\Omega}\cdot\vec{\nabla}\Psi_g(\vec{\Omega},\vec{r})+\Sigma_g^t(\vec{r})\Psi_g(\vec{\Omega},\vec{r})-W_g(\vec{\Omega},\vec{r})=S_g(\vec{\Omega},\vec{r}) \quad (3)$$

Note that these equations are coupled together via scattering in $W_g(\vec{\Omega},\vec{r})$ and fission in $S(E,\vec{\Omega},\vec{r})$.

The angular variable is treated using a discrete ordinates method [13]. In this method, the angular flux solution is sought only for a finite numbers of angular directions (ordinates) where the weights are typically defined to exactly integrate the spherical harmonics that

appear in the scattering kernel. This approximation turns each group equation above into N (=total number of directions) angular equations:

$$\vec{\Omega}_n \cdot \vec{\nabla}\Psi_{g,n}(\vec{r}) + \Sigma_g^t(\vec{r})\Psi_{g,n}(\vec{r}) - W_{g,n}(\vec{r}) = S_{g,n}(\vec{r}) \tag{4}$$

Finally, the spatial dependence is first discretized into a mesh of finite elements where the cross section data is assumed to be spatially independent (constant) within each element. In this methodology, an extruded geometry definition is assumed with respect to the treatment of the axial variable (z) and the treatment of the planar variables (x-y). Consequently, the valid finite element types only include triangular prism and hexahedrons whose axial faces are rectangular (i.e. no tetrahedrons). In the extruded geometry, we utilize a single radial mesh (two-dimensional planer mesh) and index each element by two indices: (e) gives the element id in the radial mesh and (k) gives the axial mesh or plane number. Within each element (e,k) the flux solution is approximated by a flat term and a linear term in z.

$$\Psi_{g,n}(\vec{r}) = \Psi^0_{g,n,e,k} + \Psi^1_{g,n,e,k} \times z \tag{5}$$

The solution vector will consist of the flat and linear coefficients for all elements (E) on all planes (K) in all directions (N) and all energy groups (G). In order to be able to treat generic boundary conditions and compute leakage, the outgoing flux on the radial surfaces exiting each plane must also be included as part of the solution vector. The outgoing flux on the upper or lower boundary surfaces can be obtained from evaluating the flux on the surface in the corresponding plane. Finally, we note that when parallelization in space is applied, the broken trajectory flux (characteristic lines are broken during the spatial decomposition) also becomes part of the solution vector as outlined in our previous work. Assuming a radial mesh containing E elements can be decomposed into a structured grid with $E=T^2$ total elements, the size of the solution vector can be estimated to be roughly $2KT^2NG + 2KTNG$ where we assume an axial mesh with K planes, an angular cubature with N directions, and G energy groups. For 10^6 elements, 20 planes, 200 angles, and 100 energy groups this defines a problem with ~10^{12} DOFs noting that PROTEUS-SN typically defines 100-200 axial meshes.

2.2 Derivation of the 2D1D methodology

Starting with the within-group discrete ordinate equation, the linear axial dependence is obtained by applying the Discontinuous Finite Element Methodology (DFEM) to the axial variable. Assuming that the domain is decomposed into K axial planes, we define the following functions:

$$b_{k,1}(z) = \begin{cases} 1, z \in [z_{k-1}, z_k] \\ 0, \text{otherwise} \end{cases}$$
$$b_{k,2}(z) = \begin{cases} \dfrac{2}{(z_k - z_{k-1})}\left(z - \dfrac{z_{k-1}+z_k}{2}\right), z \in [z_{k-1}, z_k] \\ 0, \text{otherwise} \end{cases} \tag{6}$$

The DFEM consists of (1) multiplying equation 4 by a trial function $b_{k,i}$ and integrating over the axial domain, (2) using integration by part on the leakage term, (3) replacing quantities on

the zone surfaces with their upstream values, $\Psi_{g,n}^{inc}(x,y,z)$, (4) approximating the unknowns, $\Psi_{g,n}(\vec{r})$, $W_{g,n}(\vec{r})$ and $S_{g,n}(\vec{r})$ as expansions in basis functions $b_{k,i}$, finally obtaining:

$$\begin{aligned}
&\xi_n\left(\sum_{i'=1}^{2}\tilde{\psi}_{g,n,k,i'}(x,y)b_{k,i'}(z_k)b_{k,i}(z_k)-\sum_{i'=1}^{2}\tilde{\psi}_{g,n,k,i'}(x,y)\int_{z_{k-1}}^{z_k}b_{k,i'}(z)\frac{db_{k,i}(z)}{dz}dz\right)\\
&+\left(\mu_n\frac{\partial\tilde{\psi}_{g,n,k,i'}(x,y)}{\partial x}+\eta_n\frac{\partial\tilde{\psi}_{g,n,k,i'}(x,y)}{\partial y}+\Sigma_{g,k}^{t}(x,y)\tilde{\psi}_{g,n,k,i'}(x,y)-\tilde{w}_{g,n,k,i}(x,y)\right)\\
&\times\int_{z_{k-1}}^{z_k}\left(b_{k,i}(z)\right)^2dz\\
&=\tilde{s}_{g,n,k,i}(x,y)\int_{z_{k-1}}^{z_k}\left(b_{k,i}(z)\right)^2dz+\Psi_{g,n}^{inc}(x,y,z_{k-1})b_{k,i}(z_{k-1})\ ,\ \xi_n>0\ ,\ i=1,2\\
&\xi_n\left(-\sum_{i'=1}^{2}\tilde{\psi}_{g,n,k,i'}(x,y)b_{k,i'}(z_{k-1})b_{k,i}(z_{k-1})-\sum_{i'=1}^{2}\tilde{\psi}_{g,n,k,i'}(x,y)\int_{z_{k-1}}^{z_k}b_{k,i'}(z)\frac{db_{k,i}(z)}{dz}dz\right)\\
&+\left(\mu_n\frac{\partial\tilde{\psi}_{g,n,k,i'}(x,y)}{\partial x}+\eta_n\frac{\partial\tilde{\psi}_{g,n,k,i'}(x,y)}{\partial y}+\Sigma_{g,k}^{t}(x,y)\tilde{\psi}_{g,n,k,i'}(x,y)-\tilde{w}_{g,n,k,i}(x,y)\right)\\
&\times\int_{z_{k-1}}^{z_k}\left(b_{k,i}(z)\right)^2dz\\
&=\tilde{s}_{g,n,k,i}(x,y)\int_{z_{k-1}}^{z_k}\left(b_{k,i}(z)\right)^2dz-\Psi_{g,n}^{inc}(x,y,z_k)b_{k,i}(z_k)\ ,\ \xi_n<0\ ,\ i=1,2
\end{aligned}\tag{7}$$

In order to simplify the equation notation, a matrix-vector notation is used:

$$\overline{\overline{F}}_k\equiv\begin{pmatrix}\int_{z_{k-1}}^{z_k}\partial z b_{k,1}(z)b_{k,1}(z) & \int_{z_{k-1}}^{z_k}\partial z b_{k,1}(z)b_{k,2}(z)\\ \int_{z_{k-1}}^{z_k}\partial z b_{k,2}(z)b_{k,1}(z) & \int_{z_{k-1}}^{z_k}\partial z b_{k,2}(z)b_{k,2}(z)\end{pmatrix}=\begin{pmatrix}z_k-z_{k-1} & 0\\ 0 & \frac{z_k-z_{k-1}}{3}\end{pmatrix}\tag{8}$$

$$\overline{\overline{U}}_k\equiv\begin{pmatrix}\int_{z_{k-1}}^{z_k}\partial z\frac{\partial b_{k,1}(z)}{\partial z}b_{k,1}(z) & \int_{z_{k-1}}^{z_k}\partial z\frac{\partial b_{k,1}(z)}{\partial z}b_{k,2}(z)\\ \int_{z_{k-1}}^{z_k}\partial z\frac{\partial b_{k,2}(z)}{\partial z}b_{k,1}(z) & \int_{z_{k-1}}^{z_k}\partial z\frac{\partial b_{k,2}(z)}{\partial z}b_{k,2}(z)\end{pmatrix}=\begin{pmatrix}0 & 0\\ 2 & 0\end{pmatrix}\tag{9}$$

$$\begin{aligned}
\overline{\overline{W}}_k^{-}&\equiv\begin{pmatrix}b_{k,1}(z_{k-1})b_{k,1}(z_{k-1}) & b_{k,1}(z_{k-1})b_{k,2}(z_{k-1})\\ b_{k,2}(z_{k-1})b_{k,1}(z_{k-1}) & b_{k,2}(z_{k-1})b_{k,2}(z_{k-1})\end{pmatrix}=\begin{pmatrix}1 & -1\\ -1 & 1\end{pmatrix}\\
\overline{\overline{W}}_k^{+}&\equiv\begin{pmatrix}b_{k,1}(z_k)b_{k,1}(z_k) & b_{k,1}(z_k)b_{k,2}(z_k)\\ b_{k,2}(z_k)b_{k,1}(z_k) & b_{k,2}(z_k)b_{k,2}(z_k)\end{pmatrix}=\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}
\end{aligned}\tag{10}$$

$$\overline{\overline{X}}_k^- \equiv \begin{pmatrix} b_{k,1}(z_{k-1}) \\ b_{k,2}(z_{k-1}) \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$
$$\overline{\overline{X}}_k^+ \equiv \begin{pmatrix} b_{k,1}(z_k) \\ b_{k,2}(z_k) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (11)$$

$$\overline{\psi}_{g,n,k}(x,y) \equiv \begin{pmatrix} \tilde{\psi}_{g,n,k,1}(x,y) \\ \tilde{\psi}_{g,n,k,2}(x,y) \end{pmatrix} \quad (12)$$

$$\overline{s}_{g,n,k}(x,y) \equiv \begin{pmatrix} \tilde{s}_{g,n,k,1}(x,y) \\ \tilde{s}_{g,n,k,2}(x,y) \end{pmatrix} \quad (13)$$

$$\overline{w}_{g,n,k}(x,y) \equiv \begin{pmatrix} \tilde{w}_{g,n,k,1}(x,y) \\ \tilde{w}_{g,n,k,2}(x,y) \end{pmatrix} \quad (14)$$

This yields the simpler form:

$$\begin{aligned} &\overline{\overline{F}}_k \left(\mu_n \frac{\partial \overline{\psi}_{g,n,k}(x,y)}{\partial x} + \eta_n \frac{\partial \overline{\psi}_{g,n,k}(x,y)}{\partial y} \right) + \left(\xi_n \left(\overline{\overline{W}}_k^+ - \overline{\overline{U}}_k \right) + \Sigma_{g,k}^t(x,y) \overline{\overline{F}}_k \right) \overline{\psi}_{g,n,k}(x,y) \\ &\quad - \overline{\overline{F}}_k \overline{w}_{g,n,k}(x,y) \\ &\qquad = \overline{\overline{F}}_k \overline{s}_{g,n,k}(x,y) + \Psi_{g,n}^{inc}(x,y,z_{k-1}) \overline{X}_k^- \quad , \quad \xi_n > 0 \\ &\overline{\overline{F}}_k \left(\mu_n \frac{\partial \overline{\psi}_{g,n,k}(x,y)}{\partial x} + \eta_n \frac{\partial \overline{\psi}_{g,n,k}(x,y)}{\partial y} \right) + \left(\xi_n \left(-\overline{\overline{W}}_k^- - \overline{\overline{U}}_k \right) + \Sigma_{g,k}^t(x,y) \overline{\overline{F}}_k \right) \overline{\psi}_{g,n,k}(x,y) \\ &\quad - \overline{\overline{F}}_k \overline{w}_{g,n,k}(x,y) \\ &\qquad = \overline{\overline{F}}_k \overline{s}_{g,n,k}(x,y) - \Psi_{g,n}^{inc}(x,y,z_k) \overline{X}_k^+ \quad , \quad \xi_n < 0 \end{aligned} \quad (15)$$

Introducing more simplifying notation:

$$\overline{\overline{\Sigma}}_{g,n,k}(x,y) \equiv \begin{cases} \xi_n \overline{\overline{F}}_k^{-1} \left(\overline{\overline{W}}_k^+ - \overline{\overline{U}}_k \right) + \Sigma_{g,k}^t(x,y) \overline{\overline{I}} \quad , \quad \xi_n > 0 \\ \xi_n \overline{\overline{F}}_k^{-1} \left(-\overline{\overline{W}}_k^- - \overline{\overline{U}}_k \right) + \Sigma_{g,k}^t(x,y) \overline{\overline{I}} \quad , \quad \xi_n < 0 \end{cases}$$
$$\overline{Q}_{g,n,k}(x,y) \equiv \begin{cases} \overline{s}_{g,n,k}(x,y) + \overline{w}_{g,n,k}(x,y) + \Psi_{g,n}^{inc}(x,y,z_{k-1}) \overline{\overline{F}}^{-1} \overline{X}_k^- \quad , \quad \xi_n > 0 \\ \overline{s}_{g,n,k}(x,y) + \overline{w}_{g,n,k}(x,y) - \Psi_{g,n}^{inc}(x,y,z_k) \overline{\overline{F}}^{-1} \overline{X}_k^+ \quad , \quad \xi_n < 0 \end{cases} \quad (16)$$

we can write:

$$\vec{\Omega}_n \cdot \vec{\nabla} \overline{\psi}_{g,n,k}(x,y) + \overline{\overline{\Sigma}}_{g,n,k}(x,y) \overline{\psi}_{g,n,k}(x,y) = \overline{Q}_{g,n,k}(x,y) \quad (17)$$

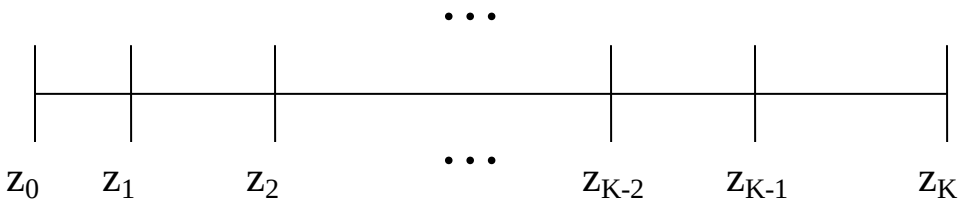

Figure 2. Axial DFEM Geometry

As stated above, the radial dependence in (x,y) is approximated by the average flux in each element, which takes the form:

$$\overline{\psi}_{g,n,k}(x,y) \cong \overline{\psi}_{g,n,k,e} = \frac{1}{V_e}\iint_{V_e} \partial x \partial y \overline{\psi}_{g,n,k}(x,y), (x,y) \in V_e \tag{18}$$

The method of characteristics (MOC) is used to evaluate the integral term.

Defining $\vec{\Omega}'_n = \left(\frac{\mu_n}{\sqrt{1-\xi_n^2}} \quad \frac{\eta_n}{\sqrt{1-\xi_n^2}} \quad 0 \right)$, we remark that

$$\vec{\Omega}_n \cdot \vec{\nabla} \overline{\psi}_{g,n,k}(x,y) = \sqrt{1-\xi_n^2}\left(\vec{\Omega}'_n \cdot \vec{\nabla} \overline{\psi}_{g,n,k}(x,y)\right) \tag{19}$$

The first step of the derivation of MOC formulation starts with the definition of a new coordinate system which allows the $\vec{\Omega}'_n \cdot \vec{\nabla}$ operator to be written as a first-order mono-dimensional derivative term. To do this, we project the incident portion ($\vec{\Omega}'_n \cdot \vec{\nabla} < 0$ where $\hat{n}$ is the outward normal from the domain surface) of the 2D radial problem domain boundary **A** for a given direction $\vec{\Omega}'_n = \left(\frac{\mu_n}{\sqrt{1-\xi_n^2}} \quad \frac{\eta_n}{\sqrt{1-\xi_n^2}} \quad 0 \right)$ to a line $A_\perp$ that is both exterior to the problem domain and perpendicular to the direction $\vec{\Omega}'_n$ as shown in Figure 3. We can define $\vec{s}_\perp$ to be a one-dimensional coordinate on the line $A_\perp$ which intersects the problem domain boundary at points $I_\perp$ and $O_\perp$. Between these points we can define the position l to be the distance measured from the incident point $I_\perp$ to some position $\vec{r} \in V$ within the volume. If the problem domain is convex, then the position $(I_\perp, l, \vec{\Omega}'_n)$ defines an alternative coordinate system to that used in equation 17. In this new coordinate system, the $\vec{\Omega}'_n \cdot \vec{\nabla}$ operator becomes a partial derivative with respect to l and we can rewrite the within-group discrete ordinate equation as

$$\sqrt{1-\xi_n^2}\frac{d\overline{\psi}_{g,n,k}(I_\perp, l)}{dl} + \overline{\overline{\Sigma}}_{g,n,k}(I_\perp, l)\overline{\psi}_{g,n,k}(I_\perp, l) = \overline{Q}_{g,n,k}(I_\perp, l) \tag{20}$$

where $\overline{\psi}_{g,n,k}(I_\perp, l)$ is used to indicate that the flux vector is in the alternative coordinate system. As can be seen, this equation is dependent upon straight line paths (termed trajectories) that penetrate the 2D radial problem domain in the direction $\vec{\Omega}'_n$ which are referred to as the mathematical characteristics of the neutron transport equation.

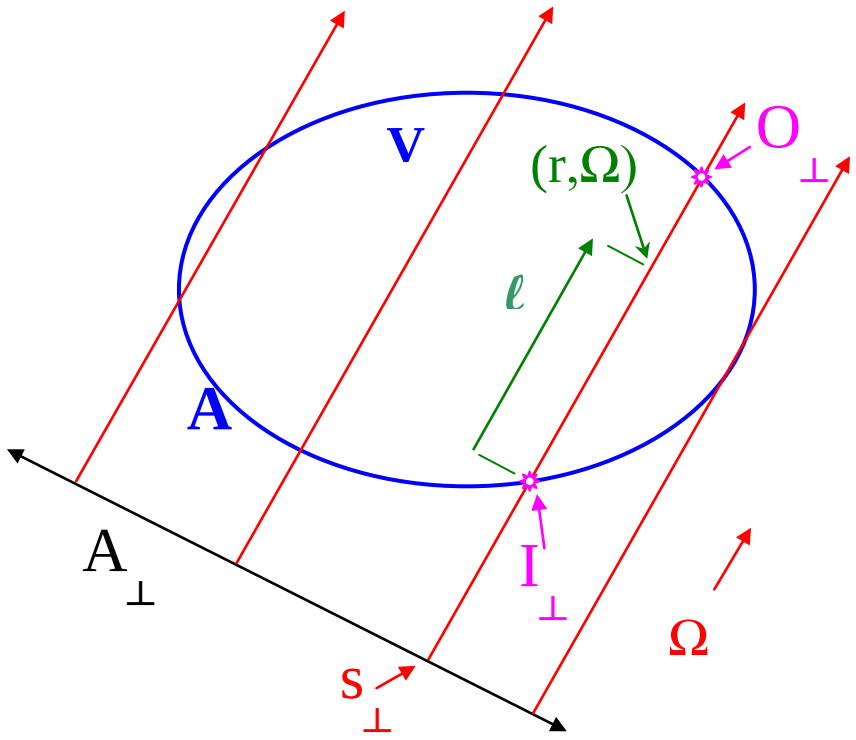

Figure 3. Coordinate System for the Characteristic Lines

Within a single element (e,k) the cross-section and the source are assumed constant and the strategy is to integrate the differential equation over the segment of trajectory intersecting the element. We obtain the integral formula:

$$\int_{V_e} \partial l\, \overline{\psi}_{g,n,k}(I_\perp, l) = \overline{\overline{\Sigma}}^{-1}_{g,n,k,e}\left(\sqrt{1-\xi_n^2}\left(\overline{\psi}_{g,n,k}(I_\perp, l_e^{in}) - \overline{\psi}_{g,n,k}(I_\perp, l_e^{out})\right) + \left(l_e^{out} - l_e^{in}\right)\overline{Q}_{g,n,k,e}\right) \tag{21}$$

Since there are typically multiple trajectories crossing a given element and each one of these is assigned a weight (area on the plane $A_\perp$), we can define the average angular flux per element in equation 18 using the Rectangle integration rule (Riemann like) :

$$\overline{\psi}_{g,n,k,e} \approx \frac{1}{V_e}\sum_{t\cap V_e \neq 0} A_t \int_{V_e} \partial l\, \overline{\psi}_{g,n,k}(I_\perp, l) \tag{22}$$

The last task is to obtain values for $\overline{\psi}_{g,n,k}(I_\perp, l_e^{in})$ and $\overline{\psi}_{g,n,k}(I_\perp, l_e^{out})$ in equation 21 which is done by performing a transport "sweep." Equation 20 can be solved analytically, noting that the exponential term is a matrix:

$$\begin{aligned}\overline{\psi}_{g,n,k}(I_\perp, l_e^{out}) = \exp\left(-\frac{l_e^{out} - l_e^{in}}{\sqrt{1-\zeta_n^2}}\overline{\overline{\Sigma}}_{g,n,k,e}\right)\overline{\psi}_{g,n,k,t}\left(l_e^{in}\right) \\ + \overline{\overline{\Sigma}}^{-1}_{g,n,k,e}\left(\overline{\overline{I}} - \exp\left(-\frac{l_e^{out} - l_e^{in}}{\sqrt{1-\zeta_n^2}}\overline{\overline{\Sigma}}_{g,n,k,e}\right)\right)\overline{Q}_{g,n,k,e}\end{aligned} \tag{23}$$

Starting from the surface of the domain for which the direction is incoming, we know the value for $\overline{\psi}_{g,n,k,t}\left(l_e^{in}\right)$ from the boundary conditions. Then $\overline{\psi}_{g,n,k,t}\left(l_e^{out}\right)$ is computed from the equation 23 for the first element. Then, if *e'* is the next element on the trajectory, we see that $\overline{\psi}_{g,n,k,t}\left(l_{e'}^{in}\right)=\overline{\psi}_{g,n,k,t}\left(l_e^{out}\right)$, and we can compute $\overline{\psi}_{g,n,k,t}\left(l_{e'}^{out}\right)$ using the same equation as above but with different a cross section, trajectory length, and source. The same approach is used along the trajectory until we reach the outgoing boundary of the domain. On the outgoing boundary a surface average flux is computed in a similar fashion as the element average flux:

$$\overline{\psi}_{g,n,k,s}^{out} \approx \frac{1}{S_s}\sum_{t\cap S_s\neq 0} A_t \overline{\psi}_{g,n,k}(I_\perp, l_e^{out}) \tag{24}$$

Defining the solution vector and the source vector, we have:

$$\overline{\Psi}_{g,n,k} \equiv \begin{pmatrix} \overline{\psi}_{g,n,k,1} \\ \vdots \\ \overline{\psi}_{g,n,k,E} \\ \overline{\psi}_{g,n,k,1}^{out} \\ \vdots \\ \overline{\psi}_{g,n,k,S}^{out} \end{pmatrix}, \overline{\Psi}_{g,n} \equiv \begin{pmatrix} \overline{\Psi}_{g,n,1} \\ \vdots \\ \overline{\Psi}_{g,n,K} \end{pmatrix}, \overline{\Psi}_{g} \equiv \begin{pmatrix} \overline{\Psi}_{g,1} \\ \vdots \\ \overline{\Psi}_{g,N} \end{pmatrix}, \overline{\Psi} \equiv \begin{pmatrix} \overline{\Psi}_{1} \\ \vdots \\ \overline{\Psi}_{G} \end{pmatrix} \tag{25}$$

$$\overline{Q}_{g,n,k} \equiv \begin{pmatrix} \overline{Q}_{g,n,k,1} \\ \vdots \\ \overline{Q}_{g,n,k,E} \\ \overline{\psi}_{g,n,k,1}^{in} \\ \vdots \\ \overline{\psi}_{g,n,k,S}^{in} \end{pmatrix}, \overline{Q}_{g,n} \equiv \begin{pmatrix} \overline{Q}_{g,n,1} \\ \vdots \\ \overline{Q}_{g,n,K} \end{pmatrix}, \overline{Q}_{g} \equiv \begin{pmatrix} \overline{Q}_{g,1} \\ \vdots \\ \overline{Q}_{g,N} \end{pmatrix}, \overline{Q} \equiv \begin{pmatrix} \overline{Q}_{1} \\ \vdots \\ \overline{Q}_{G} \end{pmatrix} \tag{26}$$

We can view the transport sweep operation above as the application of a matrix $\overline{\overline{\Lambda}}_{g,n,k}$ on the source vector which defines the flux in each element and on the outgoing boundary surface.

$$\overline{\Psi}_{g,n,k} = \overline{\overline{\Lambda}}_{g,n,k}\overline{Q}_{g,n,k} \tag{27}$$

Until now we have hidden the fact that $\overline{Q}$ is a function of $\overline{\Psi}$ which only complicates the process of obtaining a solution for $\overline{\Psi}$. We also need to explicitly account for the boundary conditions and the scattering source. Boundary conditions link surface outgoing angular flux values for a given energy group to yield surface incoming flux values. We can define the boundary conditions operation as $\overline{\overline{B}}\overline{\Psi}_g$ where $\overline{\overline{B}}$ is a sparse matrix which only operates on the surface outgoing flux part in $\overline{\Psi}_g$. It treats the flat and linear terms independently, except for the top and bottom boundary conditions where it operates on the element-wise quantity and combines the flat and linear term together to translate the outgoing solution on the surface to the incoming constraint on the same surface. The scattering source is obtained as $\overline{\overline{W}}_g\overline{\Psi}_g$ where $\overline{\overline{W}}_g$ only operates on the element-wise quantity of the solution vector, treating the flat and the linear terms seperately. Finally the reader will note that the element-wise source term

also includes an incoming leakage term from the adjacent plane. The computation of this leakage term is included in the transport sweep procedure and is therefore hidden within the definition of $\overline{\overline{\Lambda}}_{g,n}$. Thus, for directions going upward the transport sweep goes from the bottom most plane to the topmost plane and proceeds conversely for downward traveling directions. Taking all of this into account, equation 27 can be written as

$$\left(\bar{\bar{I}} - \overline{\overline{\Lambda}}\,\overline{\overline{W}}\,\overline{\overline{\mathrm{B}}}\right)\overline{\Psi} = \overline{\overline{\Lambda}}\,\overline{\Theta}\,, \tag{28}$$

where $\overline{\Theta}$ is similar to $\overline{Q}$, but only contains the fixed source term of the outer iteration, and $\overline{\overline{\Lambda}}\,\overline{\Theta}$ has the same dimension as $\overline{\Psi}$. One can now see that it is not straight-forward to evaluate $\overline{\Psi}$ (unless there is no scattering) since the matrix $\overline{\overline{A}} = \bar{\bar{I}} - \overline{\overline{\Lambda}}\,\overline{\overline{W}}\,\overline{\overline{\mathrm{B}}}$ must be inverted. It is obviously impractical to invert the matrix directly and thus an iterative method is used. In the current implementation, the method chosen is an unpreconditionned GMRES [15] which is of the family of the Krylov subspace methods.

2.3 Remarks on the current implementation

The current implementation only allows linear expansions in the axial dimension. However, it can be seen that by choosing a different set of basis/trial functions, an implementation with higher order polynomial can be implemented. The main difficulty arises in computing the coefficients of the exponential matrix that appears in equations 8 to 11. These were analytically computed for flat, linear, and quadratic approximations and thus both flat and quadratic approximations could be implemented in a rather short amount of time. Higher order approximations should lead to quicker convergence with respect to the axial mesh size; however the computational effort increases considerably (1,4,9 multiplications per segment per trajectory for flat, linear, and quadratic respectively), and therefore a careful study will be needed to determine the advantages and disadvantages of using higher order approximations.

In the derivation above, the basis functions have the same form in each plane. This is a restriction only on the current implementation and not one of the underlying methodology as the basis functions can be different in each plane. More specifically, one could choose a linear approximation in regions with small flux gradients and a higher order approximation in regions with greater flux gradients.

At present, no preconditioner was used to accelerate the flux solution step which is typically a bad idea. The creation of an efficient preconditioner is not trivial and was not targeted for work this year as the main desire was to replicate the desired input features and address the convergence problems of DeCART. Also note that the additional desire to build a “arbitrary sized CMFD like grid” which the user actually desires the solution on (as opposed to the trillion DOF system) requires considerable research.

Overall, the current implementation is far from being a production level-code and is only meant to verify the feasibility of the methodology. This is consistent with the fact that the methodology itself is novel and thus all of the work shown here forms a simple baseline for follow on work to compare against. Several parts of the code have been identified where memory storage and computational performance can be optimized. Additionally, the flux solution cannot be plotted at this time, but will be added in future versions. Finally, only

parallelization in angle has been verified to work where parallelization in plane, group, and space has been implemented, but not tested.

3 Takeda-1 benchmark

3.1 Description of the benchmark

In 1988, a set of 3D neutron transport benchmark problems was proposed from Osaka University to NEARCP [15]. These have become known as the Takeda benchmarks. The purpose was to compare results obtained from different numerical methods. They are also regularly used to check the validity of newly developed methods.

In order to test the accuracy of PROTEUS-MOC, the first of the Takeda benchmarks was chosen to test eigenvalue convergence. This benchmark models 1/8 of a thermal core with reflective boundary conditions on 3 of the surfaces to obtain a full core problem. As can be seen in Figure 4, it consists of 3 regions (core (blue), control rod (red), and reflector (green)). The model size is a cube of 25 cm edge length. This core is actually a simplified model of the Kyoto University Critical Assembly (KUCA). Two group cross-section data are provided and reproduced in the table. The reference eigenvalue was obtained with a Monte-Carlo code and is given as 0.97800 (± 0.00060).

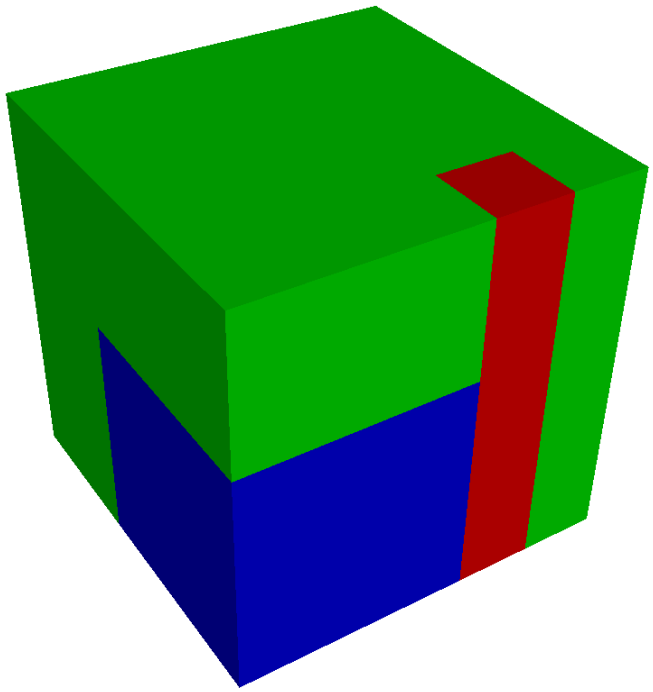

Figure 4. Geometry of the Takeda-1 Benchmark

Table 1. Energy groups and fission spectrum

Group	Energy range (eV)		Fission Spectrum
	Upper	Lower	
1	1.0000e+7	6.8256e-1	1.0000
2	6.8256e-1	1.0000e-5	0.0000

Table 2 Cross Section data (unit: cm^{-1})

	Core		
Group	Transport	Absorption	Production
1	2.23775e-01	8.52709E-03	9.09319E-03
2	1.03864E+00	1.58196E-01	2.90183E-01
Scattering Matrix			
Group	1→	2→	
1	1.92423E-01	0.0	
2	2.28253E-02	8.80439E-01	
	Reflector		
Group	Transport	Absorption	Production
1	2.50367E-01	4.16392E-04	0.0
2	1.64482E+00	2.02999E-02	0.0
Scattering Matrix			
Group	1→	2→	
1	1.93446E-01	0.0	
2	5.65042E-02	1.62452E+00	
	Empty (Void)		
Group	Transport	Absorption	Production
1	1.28407E-02	4.65132E-05	0.0
2	1.20676E-02	1.32890E-03	0.0
Scattering Matrix			
Group	1→	2→	
1	1.27700E-02	0.0	
2	2.40997E-05	1.07387E-02	

3.2 Convergence study of PROTEUS-MOC

In order to study the convergence and the computational effort involved in PROTEUS-MOC, various combinations of the 2D mesh resolutions, axial mesh resolutions and angular cubature were tested. The 2D mesh size varies from 625 to 40000 elements, the number of axial planes ranges from 5 to 40, and the number of angles from 8 (L1T1) to 128 (L7T7). We ran all of these calculations on a Linux machine using upwards of 32 parallel processes (cores). Unfortunately, some of the largest problems required more memory than the machine could accommodate, so we could not generate a complete table of results.

Figures 5 through 9 demonstrate eigenvalue convergence with respect to each independent degree of freedom. Starting with Figure 5 and Table 3, a 2500 element mesh (radial) with 10

axial planes was combined with the Legendre Tchebychev cubature where the Legendre (axial order) can be varied independently from the Tchebychev (radial) order. Figure 5 shows the convergence trend for simultaneous changes in Legendre and Tchebychev where sufficient convergence is observable at a cubature setting of L3T5 (48 angles). This result is consistent with the MOCFE and SN2ND codes [4] on the same benchmark. Using different radial meshes and different numbers of axial planes yield similar plots indicating that angular convergence is not strongly coupled to space in this problem.

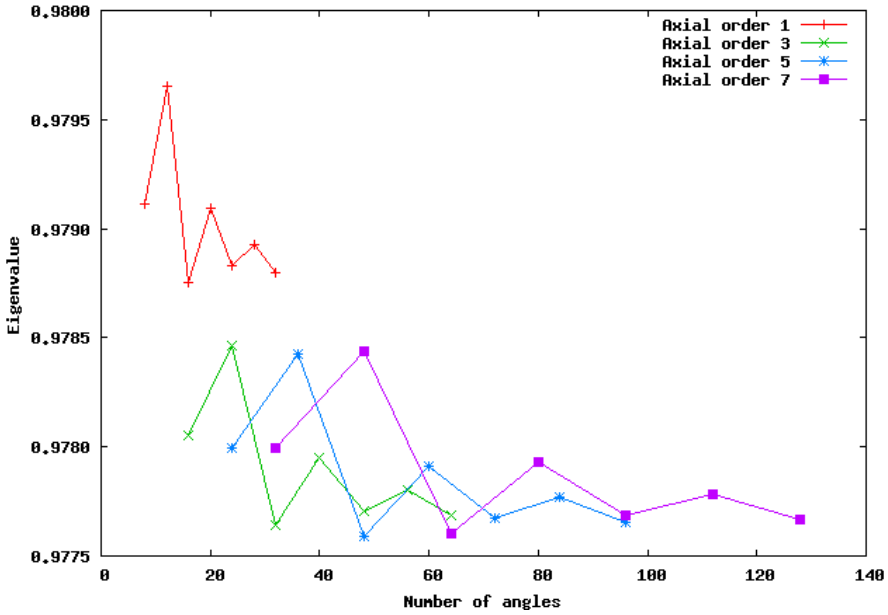

Figure 5. Eigenvalue Convergence with Respect to the Number of Angles

Table 3. Eigenvalue Convergence with Respect to the Number of Angles

Number of radial elements	Number of axial elements	Cubature	Number of angles	Eigenvalue	Difference with reference (pcm)	Difference with previous (pcm)
2500	10	L1T1	8	0.97911	111	
2500	10	L1T3	16	0.97875	75	-36
2500	10	L3T3	32	0.97764	-36	-111
2500	10	L3T5	48	0.97771	-29	7
2500	10	L5T5	72	0.97767	-33	-3
2500	10	L5T7	96	0.97766	-34	-2
2500	10	L7T7	128	0.97767	-33	1

In contrast, the convergence with respect to axial meshing is shown in Figures 6 and 7 with some of the data detailed in Table 4. In Figure 6, the cubature (L1T1) and the number of elements (625 elements) are kept low in order to be able to use a large number of axial meshes. As can be seen, convergence is rather slow and a large number of axial planes is needed to achieve convergence in this configuration. Figure 7 demonstrates that the

asymptotic behavior of the finest mesh is consistent with other angular cubatures and a finer mesh.

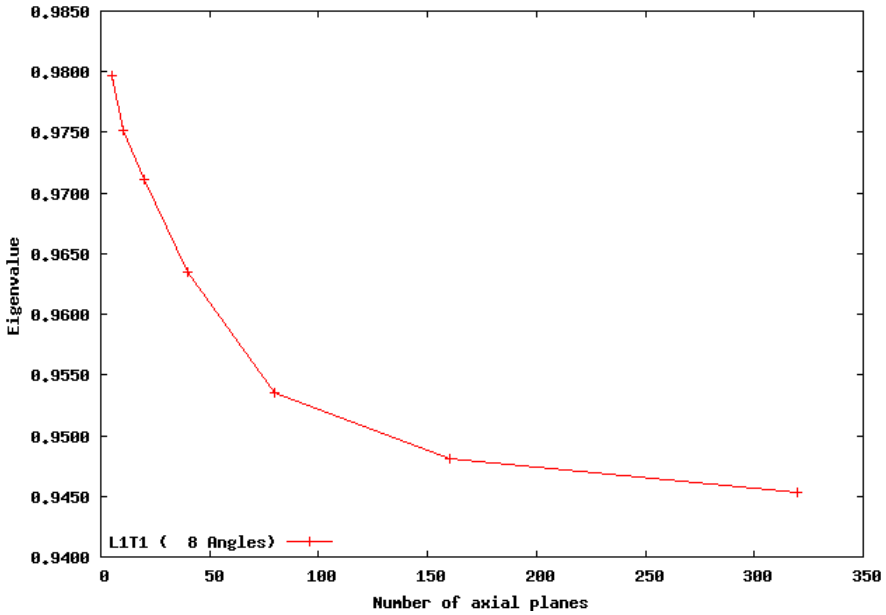

Figure 6. Coarse Mesh Eigenvalue Convergence with Respect to Number of Axial Planes

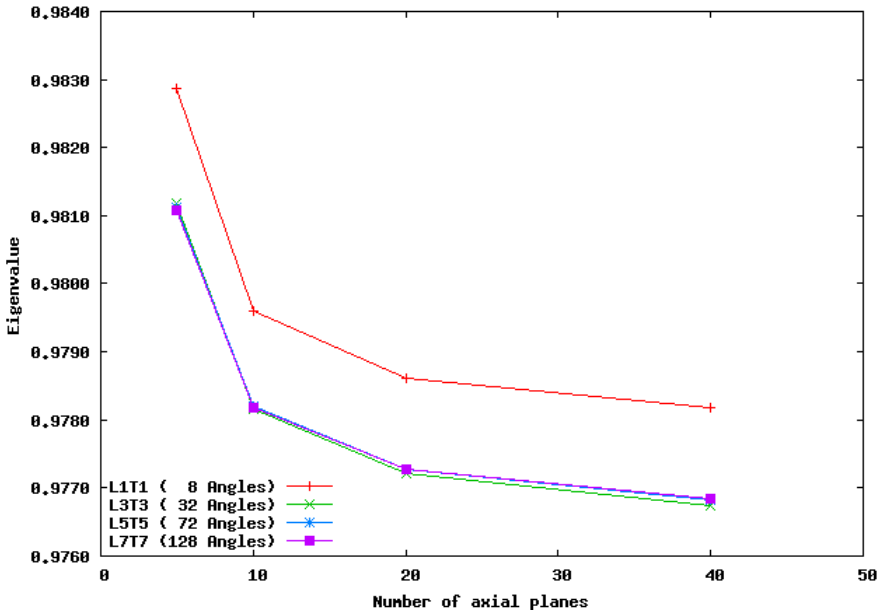

Figure 7. Fine Mesh Eigenvalue Convergence with Respect to Number of Axial Planes.

Table 4. Eigenvalue Convergence with Respect to the Number of Axial Planes

Number of radial elements	Number of axial elements	Cubature	Number of angles	Eigenvalue	Difference with reference (pcm)	Difference with previous (pcm)
40000	5	L3T3	32	0.98119	319	
40000	10	L3T3	32	0.97821	21	-298
40000	20	L3T3	32	0.97737	-63	-84
40000	40	L3T3	32	0.97714	-86	-23

For completeness, the last aspect to consider is radial mesh refinement. Figure 8 shows convergence with respect to radial mesh refinement for a fixed 10 axial planes while Figure 9 shows convergence for a fixed angular cubature. Table 5 shows that we are nearing asymptotic convergence when we reach 40,000 elements in the radial plane. The choice of angular cubature does not seem to impact the rate of convergence, whereas with a finer axial mesh we need finer radial meshing as well.

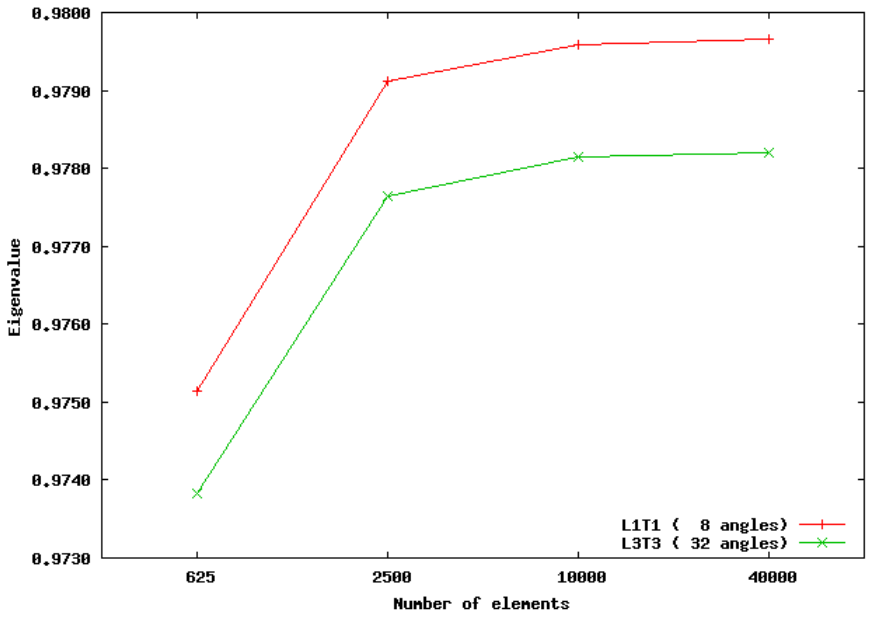

Figure 8. Eigenvalue Convergence with Respect to the Radial Meshing with 10 Axial Planes

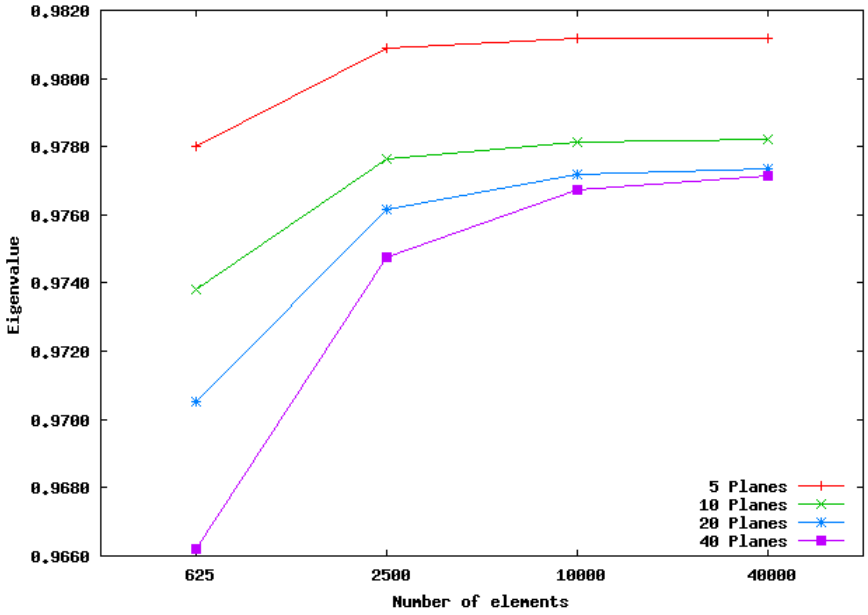

Figure 9. Eigenvalue Convergence with Respect to Radial Meshing with a L3T3 Cubature.

Table 5. Eigenvalue convergence with Respect to the Radial Meshing

Number of radial elements	Number of axial elements	Cubature	Number of angles	Eigenvalue	Difference with reference (pcm)	Difference with previous (pcm)
625	40	L3T3	32	0.966214854	-1179	
2500	40	L3T3	32	0.974773444	-323	856
10000	40	L3T3	32	0.976733383	-127	196
40000	40	L3T3	32	0.977136662	-86	40

3.3 Computational performance

PROTEUS-MOC was used on a desktop Linux machine which has 32 Intel CPUs at 2.27GHz and 132GB of shared memory. The case with a radial mesh of 10,000 elements, a L5T3 cubature (48 angles) and 40 axial planes (38.4 million DOF) ran in about 1:15 hours using 24 processors (two angles per process). While this time measure needs to be improved to compete with other codes, it was still tractable to run sufficient cases for the preceding study. The main issue is that when the problem size increases, the memory limit is quickly reached and the machine is then forced to use virtual memory (disk), which alters the performance in a very negative way.

An interesting point to study is the timing behavior with respect to change in parameters. Figures 10 through 12 show the timing performance with respect to the number of axial planes, number of angular directions, and number of elements in the radial mesh, respectively. In each of these plots, perfect parallel scaling was assumed to compute an estimation of the time for a one process job and it was then normalized with respect to the maximum value of the plot. Ignoring communication costs, the main components of computational effort can be broken into the time required performing the transport sweep and doing the GMRES

orthogonalization work. Without a preconditioner, the GMRES work is at a minimum a quadratic relationship with regard to an increase in the number of DOF, but if there is a change in the spectral radius of the underlying matrix-vector system (i.e. mesh refinement), it can be much worse. Conversely, if the matrix is nearly an identity matrix it can approach strictly linear if not less than linear. With regard to the transport sweep, the total computational effort goes linearly with the number of directions, linearly with respect to the number of planes, and linearly with respect to the square root of the number of elements in the radial mesh.

In Figure 10, one can see that the computational effort increases almost linearly with respect to the number of axial planes which means the GMRES contribution to the total time is likely not important. More importantly, since the geometry size is fixed, the spectral radius is not observed to increase as the axial mesh size is decreased from 5 cm to 0.625 cm clearly indicating that the convergence problems observed with DeCART are not present in PROTEUS-MOC. Note that the net number of GMRES iterations confirms this conclusion. In Figure 11, the timing has more of a quadratic increase with respect to the number of angles which we believe is due to the angle communication work combined with a change in the number of intersections that arise as the angular cubature is changed. In future work, a study will be performed where the Tchebychev cubature is fixed and only the Legendre setting is varied to verify this. In Figure 12, there is a slightly higher than linear trend observed with respect to radial meshing which is not consistent with the expected trajectory intersection results. This is likely due to the fact that the total number of intersections is not increasing linearly with respect to the number of radial elements.

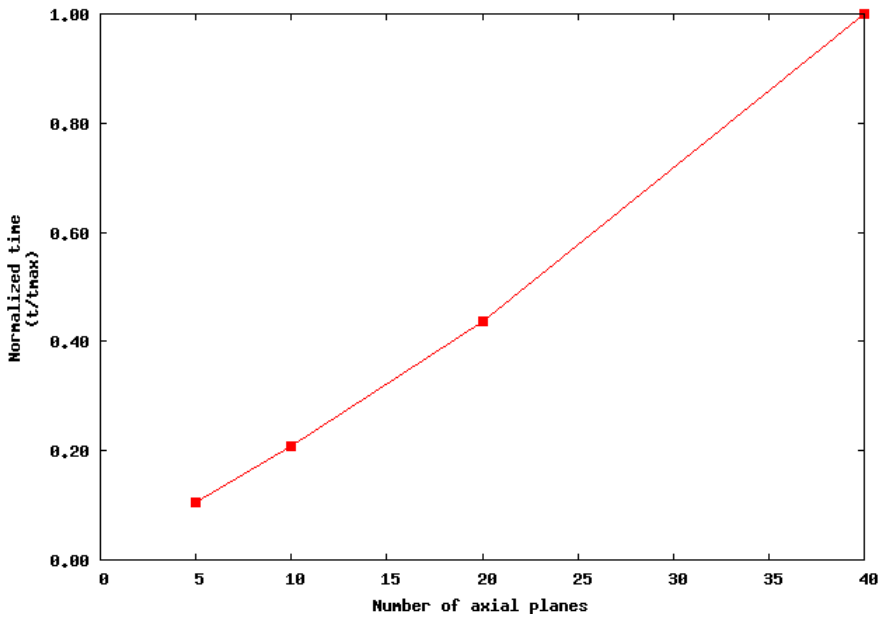

Figure 10. Dependence of Computational Effort on the Number of Axial Planes

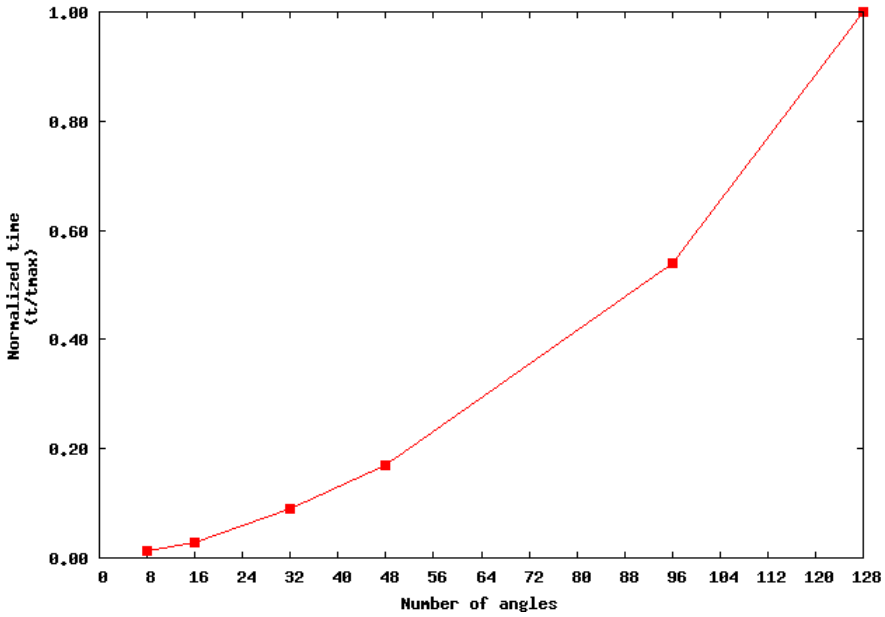

Figure 11. Dependence of Computational Effort on the Number of Angles

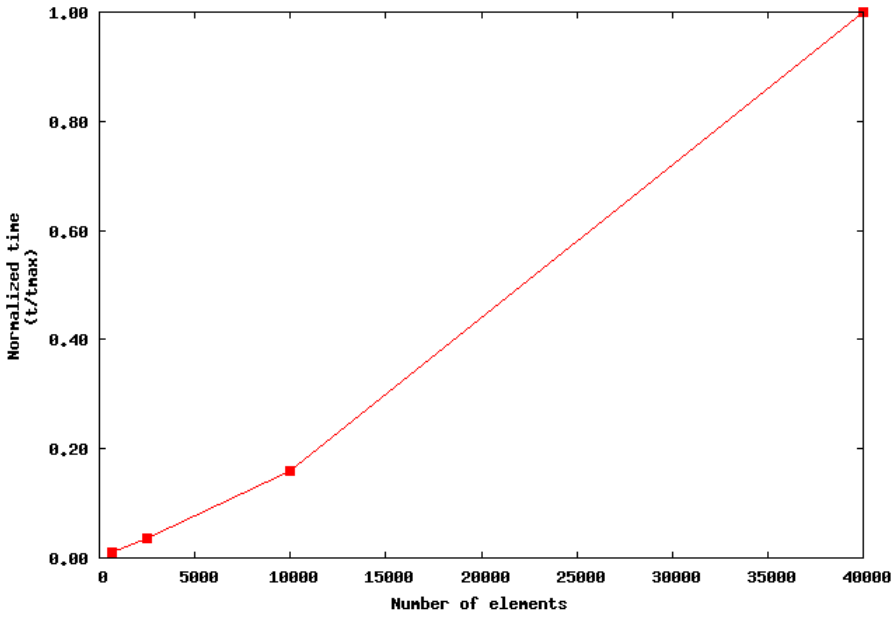

Figure 12. Dependence of Computational Effort on Radial Meshing

Overall, the computational results and convergence of PROTEUS-MOC on the preceding Takeda benchmark are as expected. Noting that the unpreconditioned GMRES system in the current implementation is effectively just the global scattering system, we did not observe any real problems noting that a larger, less leaky system might require more computational effort. Most important, PROTEUS-MOC was able to handle rather small mesh sizes without a commensurate increase in the number of GMRES iterations.

4 NonCartesian Demonstration Problem

In order to show the unique capability of PROTEUS-MOC to handle arbitrary radial geometry models, a mock-up core with a circular pattern of pins was created. Similar to the Takeda benchmark, the goal of this benchmark is to test the behavior of PROTEUS-MOC on more complicated problems. Multigroup heavy water moderated cross section data was generated and used in this problem. Comparative solutions were obtained with the PROTEUS-SN code in order to compare the results. Unfortunately a complete space-angle convergence study could not be completed at this time and we only give preliminary results on this problem with the understanding that it will become one of the key verification problems in future work.

4.1 Description of the problem

Figure 13 shows several views of the reactor problem which is 64 cm in diameter and contains 7 concentric rings of fuel pins with 2.0 cm diameters. The distance between each ring of pins is 4 cm and 6 pins are added per ring where all pins on each ring are distributed evenly. In the fifth ring, a single fuel pin is replaced by a control rod (light green). The active core was defined to be 60 cm tall with 10 cm axial reflectors, thus the total height of the system is 80 cm. A 13 energy group set of cross section data was generated using DRAGON [17] and the problem was defined to contain just three compositions: moderator/reflector (blue/grey), fuel (red) and control rod (light green). The initial planar mesh built with CUBIT contains 5,270 triangular elements. Four other finer meshes were also built containing 7376, 12418, 31177 and 127599 elements respectively in order to conduct a mesh refinement study. The control rod (in green) is inserted half way in the rodded case and the tip starts at 40cm. A top and bottom reflector regions (in grey) were added. On the right hand side of Figure 13, a view of the geometry is given at 45cm height where the control rod is inserted and at 35 cm below the control rod insertion point. The problem has strong upscattering and was setup to be near critical with a dominance ratio of ~0.6.

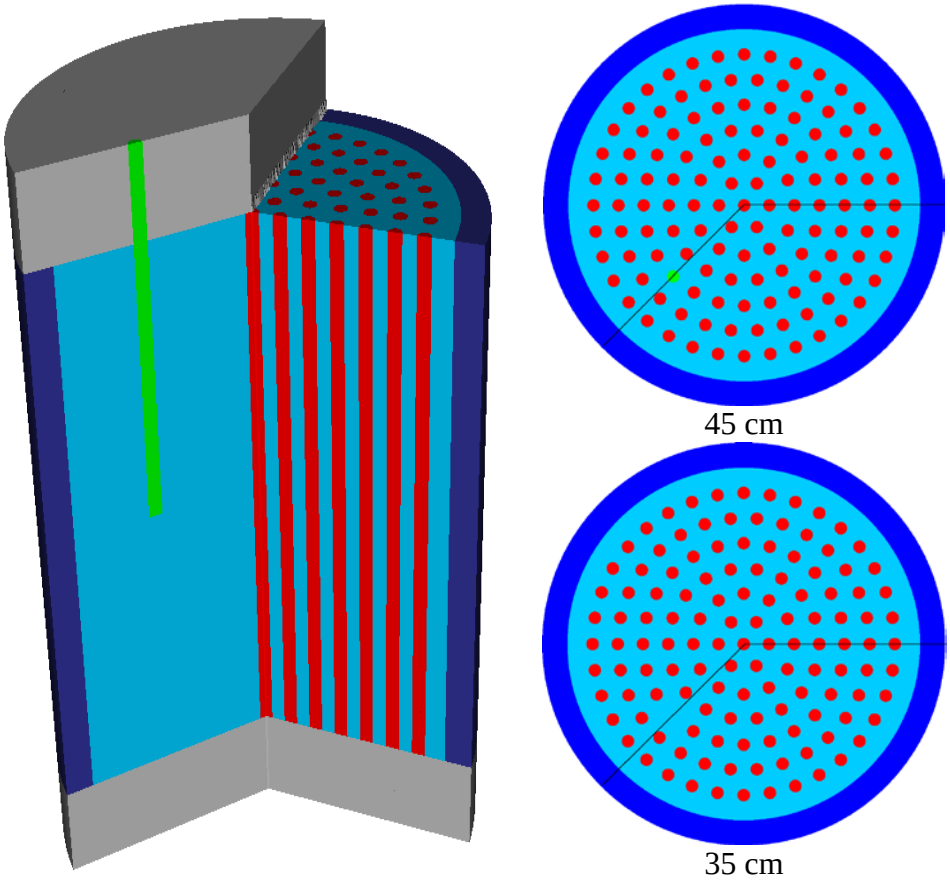

Figure 13. Rodded Geometry of the Heterogeneous Test Problem

4.2 PROTEUS-SN Preliminary Results

A rodded (as shown) and unrodded configuration (replace control rod with moderator) were constructed and BlueGene/P [17] was used to obtain a reference calculation with PROTEUS-SN code. Note that the more interesting aspect to consider is radial and axial mesh refinement relative to Takeda 1. We used a L5T7 cubature for all of the PROTEUS-SN calculations which was shown to be reasonably converged in a preliminary study.

Table 6 and Table 7 show eigenvalue results for the unrodded case. While convergence is reached rapidly with axial mesh refinement, as it is clear in Table 6, convergence with respect to radial mesh refinement is not present at 127599 elements. This is likely due to the fact that boundary layer meshing is required in SN2ND and was not resolved for this study. Note that equally spaced mesh refinement schemes were used.

Table 6. Unrodded Eigenvalue Convergence of PROTEUS-SN with Axial Mesh Refinement

Number of vertices	Number of elements in radial plane	Number of elements axially	Eigenvalue	Difference with previous (pcm)
425085	12418	8	0.91692	
825165	12418	16	0.91697	5
1625325	12418	32	0.91694	-3

Table 7. Unrodded Eigenvalue Convergence of PROTEUS-SN with Radial Mesh Refinement

Number of vertices	Number of elements in radial plane	Number of elements axially	Eigenvalue	Difference with previous (pcm)
352011	5270	16	0.91576	
492393	7376	16	0.91648	72
825165	12418	16	0.91697	49
2065998	31177	16	0.91743	46
8438166	127599	16	0.91774	31

Comparing against the most refined result from SN2ND (0.91774), Table 8 shows results obtained for the rodded case. Again, convergence with respect to the eigenvalue is not observable with the obvious need for additional radial mesh refinement. Figures 14 and 15 display the dramatic effect that the control rod has on the neutron flux and the power. Clearly the control rod is absorbing all the neutrons in neighborhood thus breaking the symmetry both axially and radially. From Figure 14, one can clearly see that there are strong gradients within the fuel pins, which explains the need for more refinement. In future work, a different meshing scheme will be used which allows the boundary layer to be inserted trivially. In Figure 15 one can see that the power level in the fuel pin surrounding the control rod is also impacted.

Table 8. Rodded Eigenvalue Convergence of PROTEUS-SN with Radial Mesh Refinement

Number of vertices	Number of elements in radial plane	Number of elements axially	Eigenvalue	Difference with previous (pcm)
1625325	12418	32	0.90727	
2065998	31177	16	0.90789	62
8438166	127599	16	0.90858	69

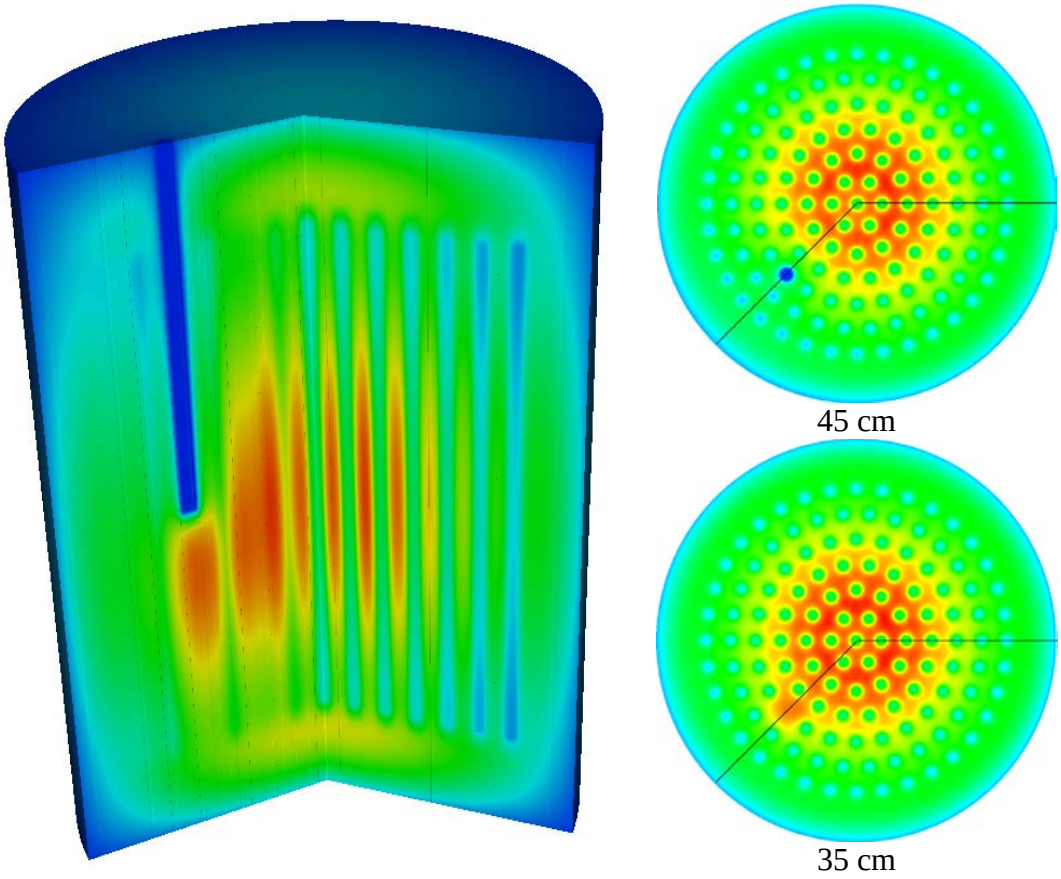

Figure 14. PROTEUS-SN Thermal (0.5eV-1.045eV) Neutron Flux Plots for the Rodded Case

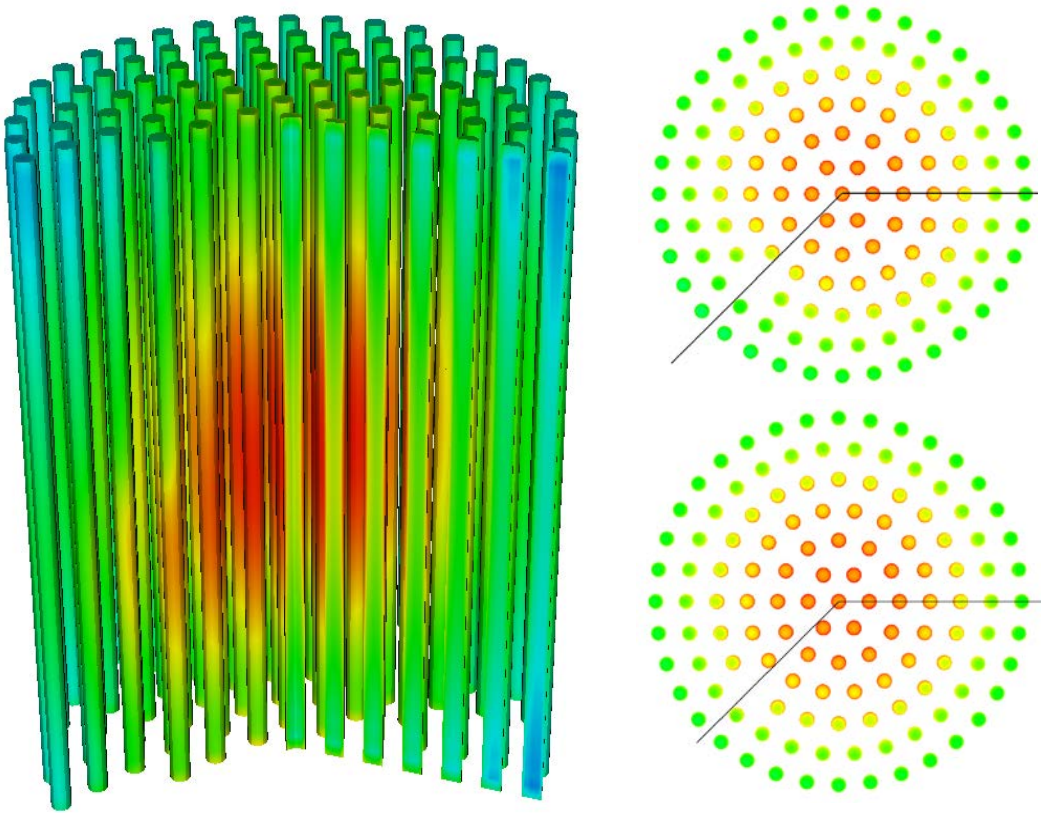

Figure 15 Power in the fuel pin in the rodded case, top view on the right are taken at 45cm (top) and 35cm(bottom)

4.3 Results obtained with PROTEUS-MOC

From the PROTEUS-SN mesh refinement requirements, clearly the problem size is beyond the abilities of the current PROTEUS-MOC. Given that the unrodded configuration was expected to converge more quickly, a majority of the calculations were focused on it. Table 9 shows the results for a radial mesh convergence study when the cubature and the number of axial planes are kept constant.

Table 9. Unrodded Eigenvalue Convergence of PROTEUS-MOC for Radial Meshing

Cubature	Radial Mesh Elements	Axial Elements	Eigenvalue	Error to Reference (pcm)	Error to Previous (pcm)
L3T3	5270	8	0.91087	687	
L3T3	7376	8	0.91318	456	-231
L3T3	12418	8	0.91471	303	-153

From Table 9, one can assume that the eigenvalue will continue increasing with more radial mesh refinement, but clearly convergence is not present. Table 10 shows the eigenvalue convergence with respect to angular cubature where the coarsest radial mesh with 16 axial planes was used. As was the case for radial mesh refinement, convergence with respect to angle is not present, however, it is interesting to note the oscillatory behavior which was similarly observed in the Takeda benchmark. Finally, Table 11 shows eigenvalue convergence behavior for axial meshing where again full convergence is not reached. Similar to the Takeda benchmark, the eigenvalue is observed to increase with axial mesh refinement which combined with angular cubature are the dominate reasons why the eigenvalue is considerably lower than PROTEUS-SN. This problem will prove very useful when comparing the accuracy of using linear axial basis functions relative to quadratic basis functions. Note that no significant efforts were made to obtain results for the rodded configuration given the results of the unrodded case.

Table 10. Unrodded Eigenvalue Convergence of PROTEUS-MOC for Angular Cubature

Cubature	Number of elements in radial plane	Number of elements axially	Eigenvalue	Difference with reference (pcm)	Difference with previous (pcm)
L1T1	5270	16	0.90703	1071	
L1T3	5270	16	0.90747	1027	-44
L3T3	5270	16	0.91040	734	-293
L3T5	5270	16	0.91016	758	23
L5T5	5270	16	0.91064	710	-47

Table 11. Unrodded Eigenvalue Convergence of PROTEUS-MOC for Axial Meshing

Cubature	Number of elements in radial plane	Number of elements axially	Eigenvalue	Difference with reference (pcm)	Difference with previous (pcm)
L3T3	7376	8	0.91318	456	
L3T3	7376	16	0.91295	479	23
L3T3	7376	32	0.91203	571	91

5 Conclusions

The PROTEUS-MOC effort is an active area of research meant to reproduce the exceptional qualities of DeCART and PROTEUS-SN, but replace the shortcomings in both. The work completed this year focused on a new formulation which does not suffer from the axial mesh size problems of DeCART. In that regard, a novel methodology was produced which treats the axial basis functions separately from the radial approximation thus allowing an extruded geometry concept to be built. This eliminates the need to carry out and store 3D ray tracing data and simultaneously preserves the coarse axial node sizes typical with DeCART (high order axial basis functions). To rapidly build a prototype, a standard power

iteration algorithm was constructed using an unpreconditioned GMRES solution algorithm for solving the flux vector in each fission source iteration.

The new solver was deployed on two test problems: Takeda and a heterogeneous cylindrical heavy water system. The Takeda benchmark problem is a standard homogeneous test problem which was previously found to be too difficult for the 3D MOCFE to solve. Using the new version we are reasonably confident that space-angle mesh convergence is present and can be fully achieved with improvements in performance. The heterogeneous cylindrical heavy water test problem was constructed more for future work than this year. There are multiple goals of this test problem the first of which is to be small enough to solve on a reasonable parallel machine yet display the physics properties of a strong upscattering moderator and a relatively high dominance ratio (~0.6). The second goal is to be very simple axially (unrodded configuration) such that the accuracy and performance characteristics of the axial basis functions (linear, quadratic, cubic, etc.) can be assessed. Finally, in the rodded configuration, the steep gradients around the test problem will prove difficult for even the most rigorous neutronics methodologies of which the MOC is typically considered to be the most optimal. Preliminary results from this benchmark are promising although clearly extensive radial mesh refinement is required before accurate results will be achieved.

While we are far from a production tool, the preceding results are promising indicating that PROTEUS-MOC is well on its way to a usable tool for NEAMS work. The preceding benchmark problems clearly demonstrate convergence of the algorithm and elimination of the small axial mesh size issue along with the capability of treating complex geometries. With time, the creation of a suitable preconditioner along with typical optimization should reduce the performance shortcomings that are observable today. One of the most important aspects of this work is developing the solution methodology which allows the user to target the domain size of interest with respect to accuracy rather than focus on getting a fully converged fine grained solution.

6 References

1. M. A. Smith, A. Marin-Lafleche, W. S. Yang, D. Kaushik, A. Siegel, “Method of Characteristics Development Targeting the High Performance Blue Gene/P Computer at Argonne National Laboratory,” International Conference on Advances in Mathematics, Computational Methods, and Reactor Physics (M&C), Rio de Janeiro, Brazil, in May 8-12, 2011.
2. M.A. Smith, A. Mohamed, A. Marin-Lafleche, E. E. Lewis, K. Derstine, C.H. Lee, A. Wollaber and W. S. Yang. FY2010 Status Report on Advanced Neutronics Modeling and Validation, ANL-GenIV-149 (September 2010).
3. M. A. Smith, D. Kaushik, A. Wollaber, W. S. Yang, B. Smith, C. Rabiti, G. Palmiotti, “Recent Research Progress on UNIC at Argonne National Laboratory,” International Conference on Mathematics, Computational Methods & Reactor Physics, Saratoga Springs, New York, May 3-7 (2009).
4. A. Siegel, T. Tautges, A. Caceres, D. Kaushik, and P. Fischer, “Software Design of SHARP”, Proceedings of the Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications, Monterey, CA, April 15-19, American Nuclear Society, Lagrange Park, IL (2007).
5. D. Kaushik, M. Smith, A. Wollaber, B. Smith, A. Siegel, and W. S. Yang, "Enabling High Fidelity Neutron Transport Simulations on Petascale Architectures," Proceedings of the Conference on

High Performance Computing Networking, Storage and Analysis, vol. 67, Portland, Oregon, ACM (2009).

6. K. L. Derstine, DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite-Difference Diffusion Theory Problems, ANL-82-64, Argonne National Laboratory, Argonne, IL (1984).
7. G. Palmiotti, E. E. Lewis, and C. B. Carrico, VARIANT: VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation, ANL-95/40, Argonne National Laboratory, Argonne, IL (1995).
8. A. Mohamed, W. S. Yang, M. A. Smith, and C. H. Lee, "Spectrum Interaction Effects on Calculating Foil Reaction Rates in ZPR Assemblies," International Conference on Advances in Mathematics, Computational Methods, and Reactor Physics (M&C), Rio de Janeiro, Brazil, in May 8-12 (2011).
9. M. A. Smith, A. Mohamed, C. H. Lee, A. Marin-Lafleche, and W. S. Yang, "Zero Power Fast Reactor Deterministic Modeling with Heterogeneous Geometry," American Nuclear Society, Hollywood, FL, June 26-30 (2011).
10. J. Y. Cho, et al., "Whole Core Transport Calculation Employing Hexagonal Modular Ray Tracing and CMFD Formulation," J. Nucl. Sci. Tech., 45, 745-751 (2008).
11. HELIOS Methods, Studsvik Scandpower, November 20 (2003).
12. E. R. Wolters and M. A. Smith, "Scoping Analysis of the Advanced Test Rector Using SN2ND," ANL-12/31, June (2012).
13. E.E. Lewis, W.F. Miller Jr, "Computational methods of neutron transport"
14. Y. Saad, M. H. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., Vol. 7, No. 3, July 1986
15. T. Takeda & H. Ikeda , *3D-Neutron Transport Benchmarks*, NEACRP-L-330,03/1991.
16. Marleau, G., A. Hébert, R. Roy, A User's Guide for DRAGON, Ecole Polytechnique de Montréal (December 1997).
17. http://www.alcf.anl.gov/resource-guides/intrepid-and-surveyor-guide

Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

www.anl.gov