

# **FY15 Status Report on NEAMS Neutronics Activities**

---

**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 and its pioneering science and technology programs, see [www.anl.gov](http://www.anl.gov).

### **DOCUMENT AVAILABILITY**

**Online Access:** U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via DOE's SciTech Connect (<http://www.osti.gov/scitech/>)

### **Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):**

U.S. Department of Commerce  
National Technical Information Service  
5301 Shawnee Rd  
Alexandria, VA 22312  
**[www.ntis.gov](http://www.ntis.gov)**  
Phone: (800) 553-NTIS (6847) or (703) 605-6000  
Fax: (703) 605-6900  
Email: [orders@ntis.gov](mailto:orders@ntis.gov)

### **Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):**

U.S. Department of Energy  
Office of Scientific and Technical Information  
P.O. Box 62  
Oak Ridge, TN 37831-0062  
**[www.osti.gov](http://www.osti.gov)**  
Phone: (865) 576-8401  
Fax: (865) 576-5728

### **Disclaimer**

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

## **FY15 Status Report on NEAMS Neutronics Activities**

---

prepared by  
C. H. Lee, E. R. Shemon, M. A. Smith, H. M. Connaway, and G. Aliberti  
Nuclear Engineering Division, Argonne National Laboratory

September 30, 2015



## EXECUTIVE ABSTRACT

Under the U.S. DOE NEAMS program, the high-fidelity multi-physics modeling and simulation capability SHARP for nuclear reactor design and analysis has been developed. SHARP is a suite of physics simulation software modules and computational framework components that enables users to accurately evaluate the physical processes of nuclear reactors including neutron transport, thermal fluid, and fuel and structure behaviors. Among the SHARP components, the goal of the NEAMS neutronics effort is to develop a high-fidelity deterministic neutron transport code PROTEUS and a cross section library and methodology for use primarily on sodium-cooled fast reactors (SFRs) and extendable to other reactor types such as light water reactors (LWRs) and high temperature reactors (HTRs).

The objectives of FY2015 NEAMS neutronics work are to update and improve PROTEUS in order to support a multi-physics demonstration of a reactor coupled with other physics tools, to verify and validate the code together with the cross section libraries for fast and thermal reactors, and to support PROTEUS users.

This report summarizes the current status of NEAMS activities in FY2015. The tasks this year are (1) to improve solution methods for steady-state and transient conditions, (2) to develop features and user friendliness to increase the usability and applicability of the code, (3) to improve and verify the multigroup cross section generation scheme, (4) to perform verification and validation tests of the code using SFRs and thermal reactor cores, and (5) to support early users of PROTEUS and update the user manuals.

PROTEUS was composed of three different high-fidelity transport solvers: SN2ND, MOCFE, and MOCEX. Among them, MOCEX (2D MOC coupled with the discontinuous Galenkin method axially based on the extruded geometry in the axial direction) requires less memory and computation time but is still under verification. To provide more options for users, the development of intermediate-fidelity transport solvers (MOC 2D/1D and NODAL) were initiated, which allow the code to produce the solutions that the user needs with practical time and computing resources. Those new solvers will be completed and tested in the following years.

The current transient solution scheme of PROTEUS with the adiabatic method was reviewed to find an efficient way to improve it with the improved quasi-static (IQS) method. It was suggested that the IQS method be implemented for a first order solver which can compute the full angular flux rather than only the scalar flux (and even-parity angular flux). Therefore, an actual implementation of the IQS will be conducted to the existing MOC solvers or the first-order SN which should be developed in advance.

To improve the usability of PROTEUS, a user-friendly mesh generation capability, including Ufmesh and GRID, was developed for typical Cartesian or hexagonal geometries, which generates the mesh input based on user inputs instead of using CUBIT. This allows the user to easily build a mesh file with input cards and options and thus to quickly update the mesh file by changing user inputs. As the Ufmesh is generated preserving the original geometry volumes, no additional adjustment that is sometimes required when using CUBIT-

generated meshes is necessary. This capability fills up the gap that RGG (MeshKit) based upon CUBIT does not support.

The ANL cross section library (ACSL) has been further verified using the selected VERA PWR benchmark problems, showing good agreement with MCNP Monte Carlo solutions. Previously, the application programming interface (API) was implemented into the SN2ND and MOCEX solvers of PROTEUS. This year, it was successfully connected to the MOCFE solver as well. To speed up the on-the-fly cross section generation, an additional resonance self-shielding option based on the Dancoff approach was tested, which allows the code to reduce the number of the FSPs to solve by more than an order of magnitude.

A 2D transport capability based upon MOC was implemented into the MC<sup>2</sup>-3 code in order to better account for the 2D spatial variance of the resonance self-shielding. Preliminary verification tests using ZPR-6/7 fuel drawers indicated that the 2D transport calculation of MC<sup>2</sup>-3 can better account for the local heterogeneity effect whereas the previously used 1D models have accuracy limitations.

As a continued effort of code verification and validation, three reactors were simulated and analyzed with PROTEUS. First, the 3D ABTR cores with different heterogeneity level configurations were modelled using MC<sup>2</sup>-3/PROTEUS: homogeneous assembly model, partially homogeneous (duct heterogeneous) model, and partially homogeneous model with the fuel heterogeneity effect. Note that a fully heterogeneous assembly model is not necessary because the heterogeneity effect is not significant in a fast reactor system. It was found that most of the PROTEUS-SN eigenvalues with 116 groups were within 100 pcm of the MCNP solutions. The total heterogeneity effect and control rod worths were in very good agreement with MCNP. The initial thermal expansion tests showed good agreement in the reactivity change between PROTEUS and MCNP as well.

The 3D ASTRID cores were simulated using MC<sup>2</sup>-3/PROTEUS with three different configurations similarly to the ABTR simulation: homogeneous assembly model, partially homogeneous assembly model (an explicit representation of wrapper tube and inter-assembly sodium gap), and partially homogeneous model with a fully heterogeneous assembly. The core  $k_{\text{eff}}$  for the homogeneous assembly model showed good agreement between PROTEUS and MCNP. Unlike the ABTR simulation, however, those for the partially homogeneous assembly model showed a noticeable discrepancy between the two codes, which could be attributed to the inconsistency in generating the multigroup cross sections using MC<sup>2</sup>-3 and should be investigated further in the future.

The simulation of the 3D TREAT cores was initiated using PROTEUS and MCNP with the homogeneous assembly model and the heterogeneous assembly model. The eigenvalue solutions were in reasonable agreement between the two codes. For detailed investigation, 3D single assembly models with two different heterogeneous configurations were built: one including a zircaloy can smeared over the whole fuel assembly and the other including zircaloy cladding on the fuel and aluminum cladding on the reflector. The eigenvalue

solutions from the two codes were in good agreement within 90 pcm. Further analysis is ongoing and the final report on TREAT will be written in a separate document in October.

As an effort of the PROTEUS user support, the PROTEUS manual was revised to include new options and changes made to the code (Revision 2.0). Technical supports were also made to the ORNL team who has been developing the depletion module for PROTEUS using the ORIGEN API. Additionally, we took the initiative to write up the MOAB API specifications for interfacing MOAB with PROTEUS to minimize current and potential problems arising from the multi-physics coupling between three physics tools via SIGMA.

## TABLE OF CONTENTS

Executive Abstract .....	i
Table of Contents .....	iv
List of Figures .....	v
List of Tables.....	vi
1. Introduction.....	1
2. Improved and New Capabilities for PROTEUS .....	4
2.1 Review of the Improved Quasi-Static Method for Kinetics.....	6
2.1.1 Derivation of the IQS Equations .....	7
2.1.2 Implementation of the IQS Method.....	13
2.1.3 Transformation of the Shape Equation into Even-Parity Form.....	15
2.1.4 Summary.....	18
2.2 User-friendly Finite Element Mesh Generation Capability .....	18
2.2.1 Ufmesh and GRID .....	18
2.2.2 Verification Tests .....	21
3. Cross Section Library and API .....	24
3.1 Verification of Argonne Cross Section Library and API.....	24
3.2 Alternative Resonance Self-Shielding Method.....	27
3.3 Cross Section Generation Using Monte Carlo Codes .....	30
4. Multidimensional Capability of MC <sup>2</sup> -3 .....	32
4.1 Development of 2D MOC Solver .....	32
4.2 Integration of 2D MOC Solver into MC <sup>2</sup> -3 Code.....	34
4.3 Verification Tests of 2D Transport Capability.....	36
4.4 Summary .....	42
5. Verification and Validation of PROTEUS.....	43
5.1 ABTR.....	43
5.1.1 Multigroup Cross Section Generation .....	43
5.1.2 Verification Tests .....	45
5.1.3 Summary.....	50
5.2 ASTRID .....	50
5.2.1 Fully Assembly Homogenized Models of ASTRID.....	51
5.2.2 Partially Homogenized Geometry .....	52
5.2.3 Summary.....	57
5.3 TREAT.....	58
5.3.1 Homogenized Fuel Assembly in Infinite Lattice.....	58
5.3.2 Full Core Model.....	60
6. Conclusions .....	64
References .....	67

## LIST OF FIGURES

Figure 1. Inputs, Outputs, and Components of PROTEUS .....	5
Figure 2. Meshes Generated from RGG-MeshKit (left) and User-friendly Mesh Generation (right).....	19
Figure 3. Example Cartesian Sector and Face Meshing Control UFmesh File .....	20
Figure 4. Example Hexagonal Assembly UFmesh File.....	20
Figure 5. Cartesian and Hexagonal Meshes Resulting From the User-friendly Mesh Generation Capability of PROTEUS .....	21
Figure 6. Eigenvalue Convergence with Mesh (element/vertex) and Angular Refinement for a Pin Cell.....	22
Figure 7. 2D Core Mesh Using UFmesh and GRID .....	23
Figure 8. Comparison of Temperature Coefficients of HTR Fuel Pin and Assembly.....	25
Figure 9. Selected Fuel Assemblies from the VERA PWR Benchmarks .....	25
Figure 10. Comparison of $k_{\infty}$ between DeCART and PROTEUS (Using the 4-group Cross Section Library) for Homogeneous Compositions .....	26
Figure 11. Difference in $k_{\infty}$ between DeCART and PROTEUS (Using the New 4-group Cross Section Library) for Pin-cell Problems .....	27
Figure 12. The 70 Group Structure Used for the Dancoff Approach.....	29
Figure 13. Percent Differences of Absorption Cross Sections between Dancoff Approach and MCNP for 3.8 wt% MOX Pin Cell .....	29
Figure 14. ISOTXS Generation Using Serpent Multigroup Cross Section .....	30
Figure 15. Composition Assignment and Fast and Thermal Fluxes of the C5 Problem Resulting from DIF3D with the 23-group Multigroup Cross Sections Generated from Serpent.....	31
Figure 16. Illustrations of 2D Unit Cell Models of MC <sup>2</sup> -3.....	33
Figure 17. 2D HFG Transport Calculation Scheme.....	35
Figure 18. Spectrum Comparison of 1D CPM and 2D MOC UFG Solutions for 1D Slab Model Problem of ZPR-6 Assembly 7 Fuel Drawer (Sliced).....	37
Figure 19. Effect of $P_1$ and $P_0$ Scattering on Spectrum for 1D Slab Model Problem of ZPR-6 Assembly 7 Fuel Drawer (Sliced).....	37
Figure 20. Configuration of ZPR-6/7 Fuel Drawer.....	39
Figure 21. Comparisons of Fe-56 Total Cross Section for Different Unit Cell Models.....	41
Figure 22. ABTR Core Loading.....	43
Figure 23. (a)-(c) Spatial models used to represent fuel assembly, (d) 1D equivalent model to the heterogeneous model in (c). Other assembly types are not shown (different number of ducts/pins) .....	45
Figure 24. Scalar flux (Clockwise from top left: Group 5, 10, 14, 19 out of 33), homogeneous assemblies, 33 groups, L7T7, P1 scattering, control rods in (slice at 140 cm).....	48
Figure 25. ABTR absorption rate for partially homogeneous assemblies with heterogeneity effects, L7T7, P1 scattering, 33 groups, control rods in (slice at 140 cm) .....	48
Figure 26. Absorption rate in non-fuel, vertical slice of the core midplane, partially homogeneous w/ pin heterogeneity effects, L7T7, P1 scattering, 33 groups, control rods in .....	49
Figure 28. Flux spectra comparison in a central zone of inner core .....	52
Figure 29. PROTEUS Models for ASTRID .....	53

Figure 30. PROTEUS Meshes for ASTRID ..... 54  
 Figure 31. 33 energy-groups flux spectra for fuel pins in positions 1, 2, 3 and 7..... 56  
 Figure 32. Cross-section view of TREAT Fuel Assembly in MCNP ..... 58  
 Figure 33. Lengthwise View of TREAT Fuel Assembly in MCNP, (a) Simplified Geometry  
 and (b) True Geometry ..... 58  
 Figure 34. PROTEUS Meshes Used for Infinite Lattice of Homogenized Assemblies, (a)  
 10x10x29 and (b) 20x20x29 ..... 60  
 Figure 35. PROTEUS Model for TREAT, (a) Radial Slice and (b) Axial Slice ..... 61  
 Figure 36. MCNP TREAT Core, (a) X-Y View and (b) X-Z View ..... 61  
 Figure 37. Meshes Used in Current PROTEUS Simulations..... 62  
 Figure 38. Example CUBIT-generated TREAT fuel assembly mesh for use with PROTEUS63

### LIST OF TABLES

Table 1. Eigenvalue Comparison between PROTEUS and MCNP Using 4G Multigroup Cross  
 Sections ..... 22  
 Table 2. Eigenvalue Comparison between PROTEUS and DeCART Using 4G Cross Section  
 Library..... 23  
 Table 3. Eigenvalue Comparison between MCNP and DeCART (204 Group Library) for the  
 VERA PWR Benchmark Problems..... 26  
 Table 4. Eigenvalue Comparison between MCNP and Dancoff Approach Using the 70-group  
 Library for UO2 and MOX Pin Cells..... 30  
 Table 5. Eigenvalue Comparison for the C5 Benchmark Problem between MCNP, Serpent,  
 and DIF3D (23-group MG cross sections)..... 31  
 Table 6.  $k_{\text{inf}}$  Comparison of 1D CPM and 2D MOC Solvers of MC<sup>2</sup>-3 for 1D Slab Model  
 Problem of ZPR-6 Assembly 7 Fuel Drawer (Sliced). ..... 36  
 Table 7. 2D Transport Effect in  $k_{\text{inf}}$  for 2D Unit Cell Calculation of ZPR-6 Assembly 7 Fuel  
 Drawer..... 38  
 Table 8. Eigenvalue Comparisons of Different Unit Cell Models of ZPR-6/7 Fuel Drawers. 40  
 Table 9. Summary of Calculation Types and Spatial Representations in Codes ..... 45  
 Table 10. PROTEUS-SN eigenvalue results, control rods out (L7T7 cubature, P1 scattering).  
 ..... 46  
 Table 11. PROTEUS-SN eigenvalue results, control rods in (L7T7 cubature, P1 scattering). 46  
 Table 12. Duct and Pin heterogeneity Effect Calculated from MCNP ..... 47  
 Table 13. Heterogeneity Effect Calculated from PROTEUS-SN (116 groups, L7T7, P1  
 scattering)..... 47  
 Table 14. Assembly Heterogeneity Effect Calculated from MC<sup>2</sup>-3 ..... 47  
 Table 15. Comparison of  $k_{\text{effective}}$  for a Fully Homogenized Assembly Model of ASTRID.  
 ..... 51  
 Table 16.  $k_{\text{eff}}$  values for Different ASTRID Models Using MCNP5..... 53  
 Table 17. Comparison of calculated  $k_{\text{eff}}$  values for the 2D ASTRID model with DIF3D and  
 PROTEUS ..... 55  
 Table 18. Power fractions for fuel pins in positions 1 to 7. .... 55  
 Table 19. Comparison of calculated  $k_{\text{eff}}$  values for the 3D ASTRID model with PROTEUS. 57

Table 20. MCNP Results for Infinite Lattice of Simplified TREAT Fuel Assemblies .....	59
Table 21. MCNP and PROTEUS Simulations of Infinite Lattice of Homogenized, Simplified TREAT Fuel Assemblies .....	60
Table 22. PROTEUS, MCNP, and DIF3D Results for TREAT Core .....	62



## 1. Introduction

Under the U.S. DOE NEAMS program, the high-fidelity multi-physics modeling and simulation capability SHARP [1] for nuclear reactor design and analysis has been developed. SHARP is a suite of physics simulation software modules and computational framework components that enables users to accurately evaluate the physical processes of nuclear reactors including neutron transport, thermal fluid, and fuel and structure behaviors. Among the SHARP components, the goal of the NEAMS neutronics effort [2,3] is to develop a high-fidelity deterministic neutron transport code PROTEUS [4,5] and a cross section library and methodology for use primarily on sodium-cooled fast reactors (SFRs) and extendable to other reactor types such as light water reactors (LWRs) and high temperature reactors (HTRs).

The objectives of FY2015 NEAMS neutronics work are to update and improve PROTEUS in order to support a multi-physics demonstration of a reactor coupled with other physics tools, to verify and validate the code together with the cross section libraries for fast and thermal reactors, and to support PROTEUS users.

To meet those objectives, the work scope of FY2015 includes improving solution methods for steady-state and transient conditions, enhancing or developing features and user friendliness to increase the usability and applicability of the code, improving and verifying the multigroup cross section generation scheme, and performing verification and validation tests of the code using SFRs and thermal reactor cores such as ABTR [6], ASTRID [7], and TREAT [8].

Due to its capability to handle unstructured finite element geometries, PROTEUS has minimal geometry modeling restrictions. PROTEUS uses largely two different transport solution methods: discrete ordinate (SN) and method of characteristics (MOC). The SN is currently based upon the even-parity, second-order formulation (SN2ND) while the MOC takes two different approaches: full 3D MOC (MOCFE) and 2D MOC coupled with the discontinuous Galenkin method axially based on the extruded geometry in the axial direction (MOCEX) [9]. Compared to MOCFE, MOCEX is advantageous in computation resource and time for 3D problems. To provide more options for users, the development of intermediate-fidelity transport solvers (MOC 2D/1D and NODAL) were initiated, which allow the code to produce the solutions that the user needs with practical time and computing resources. In future, the new solution options may make it possible to solve multi-resolution problems since solvers with different levels of fidelity are available within a code.

As an effort of improving the PROTEUS solution capabilities, the transient solution scheme that was developed with the simple adiabatic method [9] was reviewed this year to find an efficient way to implement the improved quasi-static (IQS) method. A tentative conclusion suggests to update a transport solver at the same time rather than to add the IQS option to the existing solution method which should take more time than expected. Thus, an actual implementation of the determined approach will take place in the following year.

To improve the usability of PROTEUS, we developed a user-friendly mesh generation capability which generates the mesh input based on user inputs instead of using CUBIT [11]. This allows the user to easily build a mesh file with input cards and options and thus to quickly update the mesh file by changing a few user inputs. Even though MeshKit [12] is a convenient tool which provides more flexibility in generating mesh files for various geometry components, its

mesh is based on CUBIT requiring any updates using the tool only. This sometimes makes it difficult or inconvenient to verify the mesh files in terms of geometry and boundary condition. As can be seen in many conventional codes, the user-friendly mesh generation capability removes intermediate steps between user inputs and mesh files not only to simplify the mesh generation process, but also to easily verify the mesh file. However, this new capability is limited to typical Cartesian or hexagonal geometry problems.

In previous years, the ANL cross section library (ACSL) and application programming interface (API) have been developed for the application to various reactor types including LWR, HTR, and SFR. [12] Currently, the cross section API includes two resonance self-shielding options: the subgroup method and the resonance table method. This year, efforts have been focused on verifying the cross section libraries and API. During the verification, we noticed that the total computation time was significantly increased due to solving many fixed source problems (FSPs) to estimate the heterogeneity effect for the multigroup cross sections. In order to reduce the number of the FSPs to solve, an additional resonance self-shielding option based on the Dancoff approach has been implemented into the cross section API. With the Dancoff approach [14], the number of the FSPs is reduced by more than an order of magnitude. As an alternative and verification option to the ACSL, a procedure of generating multigroup cross sections using Monte Carlo codes was developed, which would be useful to debug and verify the multigroup cross sections for complex geometry or spectrum reactor cores.

The MC<sup>2</sup>-3 code [15] does still play an important role in preparing multigroup cross sections for the high-fidelity simulation of fast reactors because cross sections for fast reactors require ultrafine-groups which is beyond the memory capacity of the API or any real whole core analysis objective today. To improve the accuracy of MC<sup>2</sup>-3, efforts have been made to enable a 2D transport capability in order to capture the 2D spatial variance in the resonance self-shielding. Verification tests were performed using the ZPR cores which naturally include large 2D effects.

In the previous year, an initial effort of PROTEUS verification with the heterogeneous cross sections was made for core components and 2D cores of ABTR. This year, we continued verification calculations for 3D cores and extended the effort to simulating ASTRID whose core specifications were taken from the benchmark exercise that was proposed by a collaborative effort between U.S. DOE-INL & ANL and France CEA-DEN [7]. Since both cores are SFRs, we used MC<sup>2</sup>-3 to generate multigroup cross sections. In addition, a thermal reactor TREAT was simulated using PROTEUS. The initial efforts made on TREAT are discussed in this report but the detailed progress and results will be presented in a separate report in October.

For the PROTEUS user support, the PROTEUS manual was revised to include new options and changes made to the code (Revision 2.0 [16]). Technical supports were also made to the ORNL team who has been developing the depletion module for PROTEUS using the ORIGEN API [17]. Additionally, we took the initiative to write up the MOAB API specifications for interfacing MOAB with PROTEUS to minimize current and potential problems arising from the multi-physics coupling between three physics tools via SIGMA [17]. The discussion is being led and progressed by the SIGMA development team.

Section 2 discusses the improvements and new development made in PROTEUS. Section 3 and 4 present continued development and verification tests performed for the cross section generation. Section 5 shows verification and validation activities for PROTEUS using ABTR, ASTRID, and TREAT cores. Conclusions are discussed in Section 6.

## 2. Improved and New Capabilities for PROTEUS

As a key component of multi-physics tool SHARP, a high-fidelity deterministic neutronics code PROTEUS-SN was developed initially for analysis and design of sodium fast reactors (SFRs). In particular, a thermal expansion behavior is one of the most important phenomena in a SFR which are very difficult to simulate using the existing computer codes. Our multi-physics and associated neutronics exercises in the NEAMS program have been mostly so far focused on simulating SFRs such as EBR-II and ABTR [6]. In recent years, efforts have been made to extend the applicability of the tools to other reactor types such as light water reactors (LWRs) and high temperature reactors (HTRs). For accurately simulating thermal reactors, multigroup cross sections should first be prepared appropriately which led us to invest lots of efforts on developing the cross section libraries, associated methodologies, and cross section API.

For PROTEUS, three high-fidelity deterministic neutron transport solvers were initially developed based on the finite element method: PN2ND (second order transport equation, spherical harmonics), SN2ND (second order transport equation, discrete ordinates), and MOCFE (first order transport equation, method of characteristics). Both PN2ND and SN2ND were designed to use 100,000+ processors for reasonably large problems with reduced homogenization approximations. MOCFE targeted fully heterogeneous fine-group problems. The variety of solvers was intended to allow the gradual replacement of the homogenization methodology by progressively applying more accurate treatments of the entire space-angle-energy phase space.

Of the three solvers, SN2ND outperformed the other solvers PN2ND and MOCFE because of its ability to seamlessly handle homogeneous or heterogeneous problems as well as problems with some spatial details. The use of off-the-shelf components like PETSc [19] helped facilitate the use of large supercomputers in solving problems with  $10^{12}$  degrees of freedom. MOCFE was found to scale, but the enormous memory usage and relative novelty limited its application. SN2ND was therefore carried forward as the primary candidate for large-scale heterogeneous geometry calculations. More recently, a reduced-memory method of characteristics methodology based upon extruded geometry modeling was developed. [9] Therefore, PROTEUS-MOC includes two options: full 3D MOC (MOCFE) and 2D MOC with the discontinuous Galerkin formulation axially (MOCEX).

Since PN2ND was found to be inefficient as a high-fidelity neutron transport method for heterogeneous geometry problems, PROTEUS-SN and PROTEUS-MOC are considered as the NEAMS neutronics tool. Currently, PROTEUS-SN is more advanced and has also been used in multi-physics coupling simulations due to its longer development time than PROTEUS-MOC.

The current solvers in PROTEUS were built to target particular behaviors of specific reactor cores (e.g., a thermal expansion of SFR and a curvilinear geometry of ATR [20]). The underlying expense associated with fully unstructured 3D transport calculations is still beyond the existing computing resources. This means that the PROTEUS code has a practical limit for the routine analysis that is desired for most reactor systems. Unlike MOCFE, the development of MOCEX relieves the computing requirements considerably, but it still requires a significant amount of computing resources compared with older methodologies. The requirement of a large computing resource prevents PROTEUS from being used on a wider set of reactor applications where users are willing to accept reduced levels of complexity and fidelity.

Thus, to improve the usability of PROTEUS, we decided to add intermediate-fidelity solvers which can solve the problems with smaller resources. One of them is the MOC 2D/1D solver [21] which uses the 2D MOC coupled with the 1D pin-based nodal transport method (NTM) and solves the 3D CMFD equation using the equivalence cross sections and parameters calculated from the 2D MOC/1D NTM. The implementation of this solver was initiated this year. The other one is the nodal solver for Cartesian and hexagonal geometries which uses the similar method to DIF3D-VARIANT [22]. The implementation of the nodal solver is the least worked on aspect, but simplest and thus it should be completed in the following year.

The finite element mesh input of PROTEUS can be generated using CUBIT or RGG (MeshKit) which calls CUBIT inside the toolkit. For complex geometry problems, there is no better way to generate meshes than using CUBIT. For typical nuclear reactor geometries, however, RGG would be a useful tool to generate meshes. For many reactor cores with regular Cartesian and hexagonal geometries, the user-friendly mesh generation capability developed for PROTEUS this year should be more convenient to provide meshes with direct user inputs and without using CUBIT. [22] This allows users to verify the geometry input with ease as well. When the preservation of volumes for each composition region is a primary concern for the CUBIT mesh, the Ufmesh is generated preserving all region volumes, which greatly simplifies the mesh verification process.

The multigroup cross sections can be generated with an independent code (e.g., MC<sup>2</sup>-3) or calculated on the fly inside PROTEUS using the cross section API which reads and processes the cross section library generated from the Genesis code [24]. Since the resonance self-shielding methods available in the cross section API requires many fixed source problems (FSPs) ( $G \times R \times I$  times where  $G$  is the number of energy groups,  $R$  is the number of resonance isotopes or categories, and  $I$  is the number of subgroups for the subgroup method or the number of iterations to converge total cross sections for the resonance table method), the Dancoff approach is added to greatly reduce the FSPs to solve ( $G \times 2$  times).

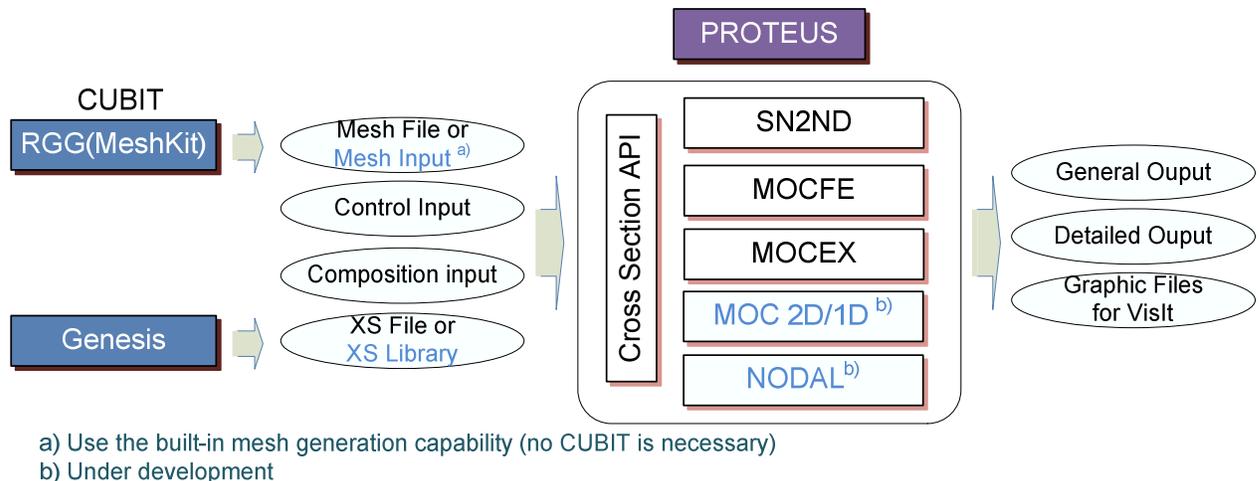


Figure 1. Inputs, Outputs, and Components of PROTEUS

To support the time-dependent multi-physics modeling capability that is one of the goals of NEAMS, a simple time-dependent modeling capability was added to PROTEUS based on the

adiabatic method in the previous year. This year, the transport method formulation has been reviewed in detail to find an efficient way of implementing the improved quasi-static method.

Improvements made to PROTEUS, as briefly noted above, are discussed in detail in this section. Those for the cross section library and API are presented in the following separate section.

### **2.1 Review of the Improved Quasi-Static Method for Kinetics**

To deterministically solve time-dependent neutron transport problems, the time variable must be treated and discretized along with the other phase space variables. The choice of a kinetics treatment is dictated by the applicability of the particular numerical approximation to the problem of interest as well as the computational resources available. Numerous methods have been developed to treat the time dependence of the angular flux. In between the two extremes of the point kinetics approximation and full-blown spatial kinetics lies the quasi-static class of approximations which comprise methods of varying accuracy: adiabatic, quasi-static (QS), improved quasi-static (IQS).

All of the quasi-static methods factorize the angular flux into a shape function and time-dependent amplitude function, i.e.  $\psi(\vec{r}, E, \hat{\Omega}, t) = n(t)\tilde{\psi}(\vec{r}, E, \hat{\Omega}, t)$ . This factorization is chosen such that the shape function  $\tilde{\psi}(\vec{r}, E, \hat{\Omega}, t)$  varies only slowly with time. This factorization choice provides a physically meaningful basis for applying approximations and naturally implies a splitting of the time-dependent transport equation into separate equations for the amplitude and shape functions. Each quasi-static method therefore yields a set of coupled equations for the shape function and amplitude function to be solved iteratively. The complexity of the shape function equation is reduced as more approximations are applied, whereas the amplitude function equations always looks like the point kinetics equations. The IQS method is the most accurate (and most complex) of the three quasi-static methods as it incurs the fewest approximations.

The approximations made in the different quasi-static methods can be quickly summarized [25] as follows:

- IQS: The time derivative  $\frac{\partial}{\partial t}\tilde{\psi}(r, E, \hat{\Omega}, t)$  in the shape function equation is approximated by a backward difference of first order.
- QS: The time derivative  $\frac{\partial}{\partial t}\tilde{\psi}(r, E, \hat{\Omega}, t)$  in the shape function equation is neglected under the justification that the time variation of the shape function is minimal compared to the time variation of the amplitude function.
- Adiabatic: The shape of the delayed neutron source is not distinguished from the shape of the prompt source (the time delay in the shape of the precursor distribution is neglected). The time derivatives  $\frac{\partial}{\partial t}\tilde{\psi}(r, E, \hat{\Omega}, t)$  and  $\frac{\partial}{\partial t}n(t)$  in the shape function equation are neglected and an eigenvalue is applied to reduce the shape function equation to a steady state eigenvalue problem.

PROTEUS-SN currently implements the adiabatic method. As noted above, the adiabatic method employs various approximations to reduce the time-dependent transport equation into the well-known point kinetics equations for the amplitude function and steady state equation for the shape function. The equations are coupled through the point kinetics parameters which are defined in terms of the angular flux solution. The steady state equation is recomputed periodically (on a longer time-scale) to update the point kinetics parameters, and the point kinetics equations are solved frequently (on a short time scale). While this method is low-cost to implement and run (it requires no modifications to the existing steady state solver and minimal effort to implement the PKE's) the adiabatic method employs certain assumptions that may not be valid for all transients of all reactors.

In order to extend the accuracy of PROTEUS-SN for a wider variety of transient events, implementation of the more accurate IQS method may be necessary. However, studies are strongly recommended to determine exactly which focus problems are not well-treated by the adiabatic method before expending effort to implement more advanced kinetics schemes such as IQS. However, in preparation for considering advanced kinetics options, we review the theory and implementation of IQS. In particular, we identify issues with IQS implementation in the even-parity method currently used by PROTEUS-SN which was chosen as a short-cut to implementing a first order continuous finite element methodology that does not suffer from condition number problems. It is important to note that most continuous finite element based methods will suffer condition number and all methods can have accuracy problems when cast into any discrete time dependent formulation (i.e. full blown kinetics like form). Thus additional research will be needed into these issues even after a particular first order formulation is chosen assuming that non-simple reactor physics problems are to be studied.

### *2.1.1 Derivation of the IQS Equations*

In this chapter, we derive the IQS equations beginning with the time dependent neutron transport equation and the time-dependent balance equation for the precursors. We apply a factorizing expression for the angular flux and perform a series of operations on these equations. The factorizing expression involves no approximation and simply separates the time dependence of the angular flux into two different functions: shape and amplitude. In the quasi-static series of approximations for kinetics, the amplitude function is assumed to contain most of the time dependence, and the shape function is assumed to vary slowly with time. These assumptions are used later to simplify the equations. The end result is a set of coupled equations which are simpler to solve than the full spatial kinetics equations but yield higher accuracy than simple point kinetics equations.

#### *Time-Dependent Neutron Transport Equation*

The time-dependent neutron transport equation describes the neutron flux  $\psi$  as a function of space  $\vec{r}$ , angle  $\hat{\Omega}$ , energy  $E$ , and time  $t$ :

$$\begin{aligned}
 & \frac{1}{v(E)} \frac{\partial \psi}{\partial t}(\vec{r}, E, \hat{\Omega}, t) + \hat{\Omega} \cdot \nabla \psi(\vec{r}, E, \hat{\Omega}, t) + \Sigma_t(\vec{r}, E, t) \psi(\vec{r}, E, \hat{\Omega}, t) \\
 & = \int_0^\infty dE' \int_{\Omega'} d\Omega' \Sigma_s(\vec{r}, E' \rightarrow E, \hat{\Omega} \cdot \hat{\Omega}', t) \psi(\vec{r}, E', \hat{\Omega}', t) \\
 & + \frac{1}{4\pi} \sum_j \chi^j(E) \int_0^\infty dE' \int_{\Omega'} d\Omega' v^j(E') (1 - \beta^j) \Sigma_f^j(\vec{r}, E', t) \psi(\vec{r}, E', \hat{\Omega}', t) \\
 & + \frac{1}{4\pi} \sum_i \chi_i(E) \lambda_i C_i(\vec{r}, t) + q(\vec{r}, E, \hat{\Omega}, t).
 \end{aligned} \tag{2.1}$$

The equation includes a source term due to the delayed radioactive neutron decay of “precursors” created during the fission process. When a fission neutron strikes a fissile isotope, it may not promptly cause a fission reaction and instead become absorbed temporarily creating a neutron precursor. Delayed neutron precursors are typically lumped into six families according to their characteristic decay constant which governs the time delay for producing neutrons after the initial fission reaction occurred. Balance equations for the density of each precursor group,  $C_i(\vec{r}, t)$ , include a production term representing the fraction of fissions in various isotopes ( $j$ ) that produce delayed precursors in that group, and a loss term representing radioactive decay (delayed neutron emission):

$$\frac{\partial C_i(\vec{r}, t)}{\partial t} = \sum_j \int_0^\infty dE' v^j(E') \beta_i^j \Sigma_f^j(\vec{r}, E', t) \int_{\Omega'} d\Omega' \psi(\vec{r}, E', \hat{\Omega}', t) - \lambda_i C_i(\vec{r}, t), \quad i=1, \dots, I \tag{2.2}$$

The notation is standard for the angular flux and interaction cross sections. We clarify the following quantities which are introduced to distinguish the time-dependent fission source contributions:

$v^j(E)$  = total number of neutrons (delayed and prompt) emitted from fission of isotope  $j$  due to an incident neutron with energy  $E$

$\beta^j = \sum_i \beta_i^j$  = fraction of fission neutrons that are delayed (emitted from any group of delayed neutron precursors) due to fission event in isotope  $j$

$1 - \beta^j$  = fraction of fission neutrons emitted promptly due to fission event in isotope  $j$

$\chi^j(E)$  = for isotope  $j$  (prompt neutron emitter), this is the fraction of (prompt) neutrons emitted at energy  $E$ ;  $\int_0^\infty dE \chi^j(E) = 1$

$\chi_i(E)$  = for delayed neutron precursor type  $i$  (delayed neutron emitter), this is the fraction of (delayed) neutrons emitted at energy  $E$ ;  $\int_0^\infty dE \chi_i(E) = 1$

$C_i(\vec{r}, t)$  = density of delayed neutron precursor group  $i$  at location  $\vec{r}$  and time  $t$

$\lambda_i$  = radioactive decay constant (in units of  $s^{-1}$ ) for delayed neutron precursor type  $i$ , i.e. probability per second that a delayed neutron is emitted from a given precursor atom

We also introduce the source-free, steady state adjoint equation for the initial state of the reactor ( $t=0$ ) which includes the total fission source (prompt and delayed):

$$\begin{aligned} & \hat{\Omega} \cdot \nabla \psi_0^*(\vec{r}, E, \hat{\Omega}) - \Sigma_{t,0}(\vec{r}, E) \psi_0^*(\vec{r}, E, \hat{\Omega}) \\ & + \int_0^\infty dE' \int_{\Omega'} d\Omega' \Sigma_{s,0}(\vec{r}, E \rightarrow E', \hat{\Omega} \cdot \hat{\Omega}') \psi_0^*(\vec{r}, E', \hat{\Omega}') \\ & + \frac{1}{4\pi} \sum_j \int_{\Omega'} d\Omega' \int_0^\infty dE' \chi_i^j(E') v^j(E) \Sigma_{f,0}^j(\vec{r}, E) \psi_0^*(\vec{r}, E', \hat{\Omega}') = 0. \end{aligned} \quad (2.3)$$

Similarly, the source-free, steady state forward equation for the initial state of the reactor ( $t=0$ ) is:

$$\begin{aligned} & \hat{\Omega} \cdot \nabla \psi_0(\vec{r}, E, \hat{\Omega}) + \Sigma_{t,0}(\vec{r}, E) \psi_0(\vec{r}, E, \hat{\Omega}) \\ & - \int_0^\infty dE' \int_{\Omega'} d\Omega' \Sigma_{s,0}(\vec{r}, E' \rightarrow E, \hat{\Omega} \cdot \hat{\Omega}') \psi_0(\vec{r}, E', \hat{\Omega}') \\ & - \frac{1}{4\pi} \sum_j \chi_i^j(E) \int_{\Omega'} d\Omega' \int_0^\infty dE' v^j(E') \Sigma_{f,0}^j(\vec{r}, E') \psi_0(\vec{r}, E', \hat{\Omega}') = 0. \end{aligned} \quad (2.4)$$

In Eq. (2.3), the average fission spectrum  $\chi_i^j(E)$  for isotope  $j$  was introduced as the weighted contributions over the prompt and delayed spectra:

$$\chi_i^j(E) \equiv \chi^j(E)(1 - \beta^j) + \sum_i \chi_i(E) \beta_i^j. \quad (2.5)$$

### Definition of Coupled Kinetics Parameters

Substituting the simplified scattering kernel and fission source expressions in Eq. (2.1) and Eq. (2.2), we rewrite Eq. (2.1):

$$\begin{aligned}
 \frac{dn(t)}{dt} = \frac{1}{\alpha} n(t) & \left\{ \begin{aligned}
 & - \int_V dV \int_0^\infty dE \phi_0^*(r, E) \left[ \int_\Omega d\Omega \hat{\Omega} \cdot \nabla \tilde{\psi}(\vec{r}, E, \hat{\Omega}, t) \right] \\
 & - \int_V dV \int_0^\infty dE \tilde{\phi}(r, E, t) \left[ \int_\Omega d\Omega \hat{\Omega} \cdot \nabla \psi_0^*(\vec{r}, E, \hat{\Omega}) \right] \\
 & - \int_V dV \int_0^\infty dE \delta\Sigma_t(\vec{r}, E, t) \tilde{\phi}(\vec{r}, E, t) \phi_0^*(r, E) \\
 & + \int_V dV \int_0^\infty dE \int_0^\infty dE' \delta\Sigma_s^0(\vec{r}, E' \rightarrow E, t) \tilde{\phi}(\vec{r}, E', t) \phi_0^*(r, E) \\
 & + \sum_j \int_V dV \int_0^\infty dE \int_0^\infty dE' \nu^j(E') \chi_i^j(E) \delta\Sigma_f^j(\vec{r}, E', t) \phi_0^*(r, E) \tilde{\phi}(\vec{r}, E', t) \\
 & - \sum_j \int_V dV \int_0^\infty dE \int_0^\infty dE' \nu^j(E') \Sigma_f^j(\vec{r}, E', t) \sum_i \chi_i(E) \beta_i^j \phi_0^*(r, E) \tilde{\phi}(\vec{r}, E', t)
 \end{aligned} \right\} \\
 & + \frac{1}{\alpha} \sum_i \lambda_i \left[ \int_V dV \int_0^\infty dE \chi_i(E) C_i(\vec{r}, t) \phi_0^*(r, E) \right] \\
 & + \frac{1}{\alpha} \left[ \int_V dV \int_0^\infty dE \phi_0^*(r, E) \int_\Omega q(\vec{r}, E, \hat{\Omega}, t) d\Omega dV dE \right]
 \end{aligned} \tag{2.6}$$

where differential cross section terms have been introduced to represent the cross section perturbations from time 0 to time t:

$$\begin{aligned}
 \delta\Sigma_t(\vec{r}, E, t) &= \Sigma_t(\vec{r}, E, t) - \Sigma_{t,0}(\vec{r}, E) \\
 \delta\Sigma_s^0(\vec{r}, E' \rightarrow E, t) &= \Sigma_s^0(\vec{r}, E' \rightarrow E, t) - \Sigma_s^0(\vec{r}, E' \rightarrow E, 0) \\
 \delta\Sigma_f^j(\vec{r}, E', t) &= \Sigma_f^j(\vec{r}, E', t) - \Sigma_{f,0}^j(\vec{r}, E')
 \end{aligned} \tag{2.7}$$

Eq. (2.2) is rewritten as,

$$\frac{dn(t)}{dt} = \frac{\rho(t) - \bar{\beta}(t)}{\Lambda(t)} n(t) + \sum_i \lambda_i c_i(t) + Q(t), \tag{2.8}$$

with the following definitions:

$$\rho(t) = \frac{1}{F(t)} \int_V dV \int_0^\infty dE \left[ \begin{aligned}
 & - \phi_0^*(r, E) \int_\Omega d\Omega \hat{\Omega} \cdot \nabla \tilde{\psi}(\vec{r}, E, \hat{\Omega}, t) \\
 & - \tilde{\phi}(r, E, t) \int_\Omega d\Omega \hat{\Omega} \cdot \nabla \psi_0^*(\vec{r}, E, \hat{\Omega}) \\
 & - \delta\Sigma_t(\vec{r}, E, t) \tilde{\phi}(\vec{r}, E, t) \phi_0^*(r, E) \\
 & + \int_0^\infty dE' \delta\Sigma_s^0(\vec{r}, E' \rightarrow E, t) \tilde{\phi}(\vec{r}, E', t) \phi_0^*(r, E) \\
 & + \sum_j \int_0^\infty dE' \nu^j(E') \chi_i^j(E) \delta\Sigma_f^j(\vec{r}, E', t) \tilde{\phi}(\vec{r}, E', t) \phi_0^*(r, E)
 \end{aligned} \right] \tag{2.9}$$

$$\bar{\beta}(t) = \sum_i \bar{\beta}_i(t) \tag{2.10}$$

$$\bar{\beta}_i(t) = \frac{1}{F(t)} \int_V dV \int_0^\infty dE \int_0^\infty dE' \sum_j \chi_i(E) v^j(E') \beta_i^j \Sigma_f^j(\bar{r}, E', t) \phi_0^*(r, E) \tilde{\phi}(\bar{r}, E', t), \quad (2.11)$$

$$\Lambda(t) = \frac{\alpha}{F(t)}, \quad (2.12)$$

$$c_i(t) = \frac{1}{\alpha} \int_V dV \int_0^\infty dE \chi_i(E) C_i(\bar{r}, t) \phi_0^*(r, E), \quad (2.13)$$

$$Q(t) = \frac{1}{\alpha} \int_V dV \int_0^\infty dE \phi_0^*(r, E) \int_\Omega q(\bar{r}, E, \hat{\Omega}, t) d\Omega dV dE, \quad (2.14)$$

$$F(t) = \int_V dV \int_0^\infty dE \int_0^\infty dE' \sum_j \chi_i^j(E) v^j(E') \Sigma_f^j(\bar{r}, E', t) \phi_0^*(r, E) \tilde{\phi}(\bar{r}, E', t). \quad (2.15)$$

We note that the definition of  $F(t)$  is arbitrary; it scales  $\rho$ ,  $\bar{\beta}$ , and  $\Lambda$  by the same factor and cancels out of the final equations. Due to the arbitrary definition of  $F(t)$ , the quantities  $\rho$ ,  $\bar{\beta}$ , and  $\Lambda$  should therefore not be attached to any specific physical meaning although we can choose  $F(t)$  such that these numbers are similar to the constants typically applied to point kinetics methodologies. Also note that  $\alpha = \Lambda(t)F(t)$  is the time-independent value determined by the normalization condition in Eq. (2.15).

#### Operations on Precursor Equations

To derive the corresponding equation for the precursor densities, we apply the operator  $\int_V \int_0^\infty \phi_0^*(r, E) \chi_i(E) \langle \bullet \rangle dV dE$  to the time-dependent precursor equation which yields:

$$\begin{aligned} & \left[ \int_V \int_0^\infty \phi_0^*(r, E) \chi_i(E) \frac{\partial}{\partial t} C_i(\bar{r}, t) dV dE \right] \\ &= n(t) \left[ \int_V \int_0^\infty \phi_0^*(r, E) \chi_i(E) \sum_j \int_0^\infty dE' v^j(E') \beta_i^j \Sigma_f^j(\bar{r}, E', t) \tilde{\phi}(\bar{r}, E', t) dV dE \right] \\ & \quad - \lambda_i \left[ \int_V \int_0^\infty \phi_0^*(r, E) \chi_i(E) C_i(\bar{r}, t) dV dE \right]. \end{aligned} \quad (2.16)$$

We multiply by the constant  $\frac{1}{\alpha}$  and move the partial derivative outside the integral, and replace the partial derivative with a normal derivative since the quantity inside is only a function of time:

$$\begin{aligned}
 & \frac{d}{dt} \left[ \int_V \int_0^\infty \phi_0^*(r, E) \chi_i(E) C_i(\bar{r}, t) dV dE \right] \cdot \frac{1}{\alpha} \\
 &= n(t) \left[ \int_V \int_0^\infty \phi_0^*(r, E) \chi_i(E) \sum_j \int_0^\infty dE' \nu^j(E') \beta_i^j \Sigma_f^j(\bar{r}, E', t) \tilde{\phi}(\bar{r}, E', t) dV dE \right] \cdot \frac{1}{\alpha} \\
 & \quad - \lambda_i \left[ \int_V \int_0^\infty \phi_0^*(r, E) \chi_i(E) C_i(\bar{r}, t) dV dE \right] \cdot \frac{1}{\alpha}
 \end{aligned} \tag{2.17}$$

The reduced form of the precursor equation is then directly obtained with all terms having previously been defined.

$$\frac{dc_i(t)}{dt} = \frac{\bar{\beta}_i(t)}{\Lambda(t)} n(t) - \lambda_i c_i(t), \quad i = 1, \dots, I \tag{2.18}$$

These equations form the system of equations for the amplitude function,  $n(t)$ . They have the same structure as the well-known point kinetics equations, but unlike the point kinetics methodology, they contain time-dependent coefficients which are directly coupled to the shape function  $\tilde{\psi}(r, E, \hat{\Omega}, t)$ . Thus the point kinetics equations cannot be solved independently from the shape function.

#### Derivation of Shape Function Equations

Thus far, we have derived compact equations for the amplitude function which are coupled to the shape function through integral parameters. The shape function equation is easily derived by substitution of the factorization expression into the transport equation. The time derivative term is expanded using the chain rule, and the entire equation is divided by  $n(t)$  to yield the following expression:

$$\begin{aligned}
 & \frac{1}{v(E)} \left[ \frac{\partial \tilde{\psi}(\bar{r}, E, \hat{\Omega}, t)}{\partial t} + \tilde{\psi}(\bar{r}, E, \hat{\Omega}, t) \frac{dn(t)}{dt} \cdot \frac{1}{n(t)} \right] \\
 & + \hat{\Omega} \cdot \nabla \tilde{\psi}(\bar{r}, E, \hat{\Omega}, t) + \Sigma_t(\bar{r}, E, t) \tilde{\psi}(\bar{r}, E, \hat{\Omega}, t) \\
 & = \int_0^\infty dE' \int d\Omega' \Sigma_s(\bar{r}, E' \rightarrow E, \hat{\Omega} \cdot \hat{\Omega}', t) \tilde{\psi}(\bar{r}, E', \hat{\Omega}', t) \\
 & \quad + \frac{1}{4\pi} \sum_j \chi^j(E) \left\{ \int dE' \nu^j(E') (1 - \beta^j) \Sigma_f^j(\bar{r}, E', t) \int d\Omega' \tilde{\psi}(\bar{r}, E', \hat{\Omega}', t) \right\} \\
 & \quad + \frac{1}{n(t)} \left[ \frac{1}{4\pi} \sum_i \chi_i(E) \lambda_i C_i(\bar{r}, t) + q(\bar{r}, E, \hat{\Omega}, t) \right].
 \end{aligned} \tag{2.19}$$

In the improved quasi-static approximation, Eq. (2.19) is solved by replacing the derivative of  $\tilde{\psi}$  with a backward difference of first order,

$$\frac{\partial \tilde{\psi}(\vec{r}, E, \hat{\Omega}, t)}{\partial t} = \frac{\tilde{\psi}(\vec{r}, E, \hat{\Omega}, t) - \tilde{\psi}(\vec{r}, E, \hat{\Omega}, t - \Delta t)}{\Delta t}, \quad (2.20)$$

where  $t - \Delta t$  is the time corresponding to the last shape function calculation. The improved quasi-static shape equation is therefore written as

$$\begin{aligned} & \hat{\Omega} \cdot \nabla \tilde{\psi}(\vec{r}, E, \hat{\Omega}, t) + \left[ \Sigma_t(\vec{r}, E, t) + \frac{1}{\nu(E)} \left( \frac{1}{\Delta t} + \frac{dn(t)}{dt} \cdot \frac{1}{n(t)} \right) \right] \tilde{\psi}(\vec{r}, E, \hat{\Omega}, t) \\ & = \int_0^\infty dE' \int d\Omega' \Sigma_s(\vec{r}, E' \rightarrow E, \hat{\Omega} \cdot \hat{\Omega}', t) \tilde{\psi}(\vec{r}, E', \hat{\Omega}', t) \\ & + \frac{1}{4\pi} \sum_j \chi^j(E) \left\{ \int dE' \nu^j(E') (1 - \beta^j) \Sigma_f^j(\vec{r}, E', t) \int d\Omega' \tilde{\psi}(\vec{r}, E', \hat{\Omega}', t) \right\} \\ & + \frac{1}{n(t)} \left[ \frac{1}{4\pi} \sum_i \chi_i(E) \lambda_i C_i(\vec{r}, t) + q(\vec{r}, E, \hat{\Omega}, t) \right] + \frac{1}{\nu(E) \Delta t} \tilde{\psi}(\vec{r}, E, \hat{\Omega}, t - \Delta t) \end{aligned} \quad (2.21)$$

We also insert Eq. (2.20) into the precursor density equation:

$$\frac{\partial C_i(\vec{r}, t)}{\partial t} = \sum_j \int_0^\infty dE' \nu^j(E') \beta_i^j \Sigma_f^j(\vec{r}, E', t) n(t) \tilde{\phi}(\vec{r}, E', t) - \lambda_i C_i(\vec{r}, t), \quad i = 1, \dots, I \quad (2.22)$$

Eq. (2.22) can be analytically solved in terms of the amplitude and shape functions for all previous time:

$$C_i(\vec{r}, t) = \int_{-\infty}^t \left[ \sum_j \int_0^\infty dE' \nu^j(E') \beta_i^j \Sigma_f^j(\vec{r}, E', t') \tilde{\phi}(\vec{r}, E', t') \right] n(t') e^{-\lambda_i(t-t')} dt', \quad i = 1, \dots, I \quad (2.23)$$

The precursor density at a given time  $t$  is dependent on the delayed fission source weighted and integrated over all preceding times. Eq. (2.23) can be inserted into Eq. (2.22) to eliminate the precursor density variable.

### 2.1.2 Implementation of the IQS Method

The preceding coupled IQS equations govern the amplitude  $n(t)$  and shape  $\tilde{\psi}(\vec{r}, E, \hat{\Omega}, t)$  functions whose product is the angular flux,  $\psi(\vec{r}, E, \hat{\Omega}, t) = n(t) \tilde{\psi}(\vec{r}, E, \hat{\Omega}, t)$ . The ‘‘point kinetics’’ parameters  $\rho$ ,  $\bar{\beta}$ , and  $\Lambda$  in the amplitude equations depend on the shape function computed at time  $t$ . The shape equation includes dependencies on the amplitude and precursor densities. Thus the IQS equations are nonlinearly coupled. The key idea behind the QS factorization choice is that the time dependence of the shape function is small compared to that of the amplitude function. The shape function can be solved on a longer macro-time step scale,  $\Delta t$ , and the simpler amplitude (PK) equations can be solved on a shorter micro-time step scale,  $\delta t$ .

The IQS algorithm can be summarized with the following steps [26] noting that the PROTEUS-SN code only computes the even-parity flux and thus we have to approximate the angular flux terms as  $\psi(r, \hat{\Omega}, E) = \frac{1}{4\pi} \phi(r, \hat{\Omega}, E)$ :

**Step 1.** Compute the initial condition (t=0) steady state adjoint and forward solutions including the delayed neutron fission source, i.e.  $\psi_0^*$  and  $\psi_0$  from Eq. (2.3) and Eq. (2.4).

**Step 2.** Evaluate the PK parameters for the interval  $[0, \Delta t]$  using the solution  $\psi_0$  from Step 1a. This assumes that the shape function does not change over this interval. Solve the PK equations over the interval  $[0, \Delta t]$  using micro-timesteps  $\delta t$  to estimate the amplitude function  $n(t)$  and its time derivative  $\frac{dn(t)}{dt}$  at  $t = \Delta t$ .

**Step 3.** Solve Eq. (2.21) for the shape equation for the end of the interval  $[0, \Delta t]$  using a conventional steady-state transport solver with external source capability, provided the total cross section and external source are normalized properly:

$$\tilde{\Sigma}_t(\vec{r}, E, t) = \Sigma_t(\vec{r}, E, t) + \frac{1}{v(E)} \left( \frac{1}{\Delta t} + \frac{dn(t)}{dt} \cdot \frac{1}{n(t)} \right) \quad (2.24)$$

$$\tilde{Q}(\vec{r}, E, \hat{\Omega}, t) = \frac{1}{n(t)} \left[ \frac{1}{4\pi} \sum_i \chi_i(E) \lambda_i C_i(\vec{r}, t) + q(\vec{r}, E, \hat{\Omega}, t) \right] + \frac{1}{v(E)\Delta t} \tilde{\psi}(\vec{r}, E, \hat{\Omega}, t - \Delta t) \quad (2.25)$$

This step will produce the shape function at time  $t = \Delta t$ ,  $\tilde{\psi}(\vec{r}, E, \hat{\Omega}, \Delta t)$ . [Ref. [26] indicates that the shape equation should be “integrated” over time and solved and it is unclear if the above procedure is equivalent].

**Step 4.** Evaluate the error on the normalization of the shape function by evaluating the inner product of the initial condition adjoint scalar flux (known from Step 1) and scalar shape function with inverse velocity,  $\frac{1}{v} \tilde{\phi}(\vec{r}, E, \Delta t)$  (known) and comparing to the assigned value:

$$\varepsilon_{\tilde{\psi}} = \frac{\left| \left\langle \phi_0^*, \frac{1}{v} \tilde{\phi} \right\rangle - \alpha \right|}{\alpha} \quad (2.26)$$

If the relative error is greater than some criteria epsilon, the shape function is then re-normalized such that the normalization condition is satisfied. [Note: In Ref. [26] the angular adjoint flux and angular shape function are used in this normalization.]

**Step 5.** The re-normalized shape function is used to re-compute the PK parameters over the first macro time step. Re-solve the PK equations using the new PK parameters, obtain a new estimate of the amplitude  $n(t)$  and its time derivative at  $\Delta t$ .

**Step 6.** Repeat Steps 3-5 until the epsilon criteria is satisfied. Once satisfied, then the solution  $n(\Delta t)$  and  $\tilde{\psi}(\vec{r}, E, \hat{\Omega}, \Delta t)$  have been estimated.

**Step 7.** The solution over the next macro-timestep can be initiated starting with Step 2 and estimating the point kinetics parameters initially using the solution  $\tilde{\psi}(\vec{r}, E, \hat{\Omega}, \Delta t)$ .

While we have not yet described the actual implementation into the PROTEUS-SN solver, the existing SN solver with external source capability can be used to solve the second order even parity form of the shape equation, yielding the angle-dependent even parity flux. However, without the corresponding odd-parity angle-dependent flux (which is not calculated in PROTEUS-SN), we can only compute the scalar flux (independent of angle). It appears to be common in classical kinetics codes to utilize an isotropic approximation for the angular flux when computing the PK coefficients and adjusted source term, but such an approximation clearly introduces some inconsistency into the IQS equations. A recent paper [27] describes some of the errors which might be expected by using a classical (vs consistent) method when the code does not directly produce the angular flux such as in PROTEUS-SN. To use a fully consistent approach (computation of angular flux), the PROTEUS-SN formulation would have to be significantly changed.

### 2.1.3 Transformation of the Shape Equation into Even-Parity Form

Here we derive the even-parity multigroup form of the IQS shape equation to determine its form in PROTEUS-SN. Beginning with the first order continuous energy shape equation, we apply the multigroup approximation to obtain

$$\begin{aligned}
 & \hat{\Omega} \cdot \nabla \tilde{\psi}_g(\vec{r}, \hat{\Omega}, t) + \left[ \Sigma_{t,g}(\vec{r}, t) + v_g^{-1} \left( \frac{1}{\Delta t} + \frac{dn(t)}{dt} \cdot \frac{1}{n(t)} \right) \right] \tilde{\psi}_g(\vec{r}, \hat{\Omega}, t) \\
 &= \sum_{g'=1}^G \int d\Omega' \Sigma_{s,g' \rightarrow g}(\vec{r}, \hat{\Omega} \cdot \hat{\Omega}', t) \tilde{\psi}_{g'}(\vec{r}, \hat{\Omega}', t) \\
 &+ \frac{1}{4\pi} \sum_j \chi_g^j (1 - \beta^j) \sum_{g'=1}^G v \Sigma_{f,g'}^j(\vec{r}, t) \tilde{\phi}_{g'}(\vec{r}, t) \\
 &+ \frac{1}{n(t)} \left[ \frac{1}{4\pi} \sum_i \chi_{i,g} \lambda_i C_i(\vec{r}, t) + q_g(\vec{r}, \hat{\Omega}, t) \right] \\
 &+ v_g^{-1} \frac{1}{\Delta t} \tilde{\psi}_g(\vec{r}, \hat{\Omega}, t - \Delta t)
 \end{aligned} \tag{2.27}$$

Note that a conventional approximation has been introduced in the definition of  $\Sigma_{t,g}(\vec{r}, t)$  which should actually be weighted with the angular shape function instead of the scalar flux. However this results in an angle dependent total cross section which is not practical. We define a “virtual” total cross section as:

$$\tilde{\Sigma}_{t,g}(\vec{r},t) = \Sigma_{t,g}(\vec{r},t) + v_g^{-1} \left( \frac{1}{\Delta t} + \frac{dn(t)}{dt} \cdot \frac{1}{n(t)} \right). \quad (2.28)$$

We also lump the group-to-group scattering sources (except the within group scattering term) into one term called the group source in order to obtain the following equation:

$$\begin{aligned} \hat{\Omega} \cdot \nabla \tilde{\psi}_g(\vec{r},\hat{\Omega},t) + \tilde{\Sigma}_{t,g}(\vec{r},t) \tilde{\psi}_g(\vec{r},\hat{\Omega},t) &= \sum_{g'=1}^G \int \Sigma_{s,g' \rightarrow g}(\vec{r},\hat{\Omega} \cdot \hat{\Omega}',t) \tilde{\psi}_{g'}(\vec{r},\hat{\Omega}',t) d\Omega' \\ &+ \frac{1}{4\pi} \sum_j \chi_g^j (1 - \beta^j) \sum_{g'=1}^G v \Sigma_{f,g'}^j(\vec{r},t) \tilde{\phi}_{g'}(\vec{r},t) \\ &+ \frac{1}{n(t)} \frac{1}{4\pi} \sum_i \chi_{i,g} \lambda_i C_i(\vec{r},t) + \frac{1}{n(t)} q_g(\vec{r},\hat{\Omega},t) \\ &+ v_g^{-1} \frac{1}{\Delta t} \tilde{\psi}_g(\vec{r},\hat{\Omega},t - \Delta t) \end{aligned} \quad (2.29)$$

We lump the group-to-group scattering sources (except the within group scattering term) into a single term called the group source in order to obtain Eq. (2.30), a familiar form for the within group transport equation:

$$\hat{\Omega} \cdot \bar{\nabla} \tilde{\psi}_g(\vec{r},\hat{\Omega},t) + \tilde{\Sigma}_{t,g}(\vec{r},t) \tilde{\psi}_g(\vec{r},\hat{\Omega},t) = \tilde{W}_g(\vec{r},\hat{\Omega}) + \tilde{S}_g(\vec{r},\hat{\Omega}). \quad (2.30)$$

The term  $\tilde{W}_g(\vec{r},\hat{\Omega})$  represents the within-group scattering source, and the term  $\tilde{S}_g(\vec{r},\hat{\Omega})$  represents the adjusted sources particular to the IQS method:

$$\begin{aligned} \tilde{W}_g(\vec{r},\hat{\Omega}) &= \int \Sigma_{s,g \rightarrow g}(\vec{r},\hat{\Omega} \cdot \hat{\Omega}',t) \tilde{\psi}_g(\vec{r},\hat{\Omega}',t) d\Omega' \\ &= \int \Sigma_{s,g \rightarrow g,m}(\vec{r},t) P_m(\hat{\Omega} \cdot \hat{\Omega}') \tilde{\psi}_g(\vec{r},\hat{\Omega}',t) d\Omega' \end{aligned} \quad (2.31)$$

$$\begin{aligned} \tilde{S}_g(\vec{r},\hat{\Omega}) &= \sum_{g' \neq g}^G \int \Sigma_{s,g' \rightarrow g}(\vec{r},\hat{\Omega} \cdot \hat{\Omega}',t) \tilde{\psi}_{g'}(\vec{r},\hat{\Omega}',t) d\Omega' \\ &+ \frac{1}{4\pi} \sum_j \chi_g^j (1 - \beta^j) \sum_{g'=1}^G v \Sigma_{f,g'}^j(\vec{r},t) \tilde{\phi}_{g'}(\vec{r},t) \\ &+ \frac{1}{n(t)} \frac{1}{4\pi} \sum_i \chi_{i,g} \lambda_i C_i(\vec{r},t) + \frac{1}{n(t)} q_g(\vec{r},\hat{\Omega},t) \\ &+ v_g^{-1} \frac{1}{\Delta t} \tilde{\psi}_g(\vec{r},\hat{\Omega},t - \Delta t) \end{aligned} \quad (2.32)$$

These definitions differ from the conventional even-parity method in their use of the shape function rather than the angular flux, and the definition of source term which contains terms specific to IQS.

In the even parity method, even- and odd-parity components of the shape function,  $\tilde{\psi}_g^+(\vec{r}, \hat{\Omega})$ , are defined as the sum and difference, respectively, of the shape function evaluated at directions  $\hat{\Omega}$  and  $-\hat{\Omega}$ :

$$\tilde{\psi}_g^+(\vec{r}, \hat{\Omega}) \equiv \frac{1}{2} \left( \tilde{\psi}_g(\vec{r}, \hat{\Omega}) \pm \tilde{\psi}_g(\vec{r}, -\hat{\Omega}) \right). \quad (2.33)$$

The shape function can be written as the sum of the even- and odd-parity components:

$$\tilde{\psi}_g(\vec{r}, \hat{\Omega}) = \tilde{\psi}_g^+(\vec{r}, \hat{\Omega}) + \tilde{\psi}_g^-(\vec{r}, \hat{\Omega}). \quad (2.34)$$

The even- and odd-parity components have the following properties, where the function  $\tilde{\phi}_g(\vec{r})$  represents the group scalar shape function.

$$\tilde{\psi}_g^+(\vec{r}, \hat{\Omega}) = \tilde{\psi}_g^+(\vec{r}, -\hat{\Omega}), \quad (2.35)$$

$$\int \tilde{\psi}_g^+(\vec{r}, \hat{\Omega}) d\Omega = \tilde{\phi}_g(\vec{r}), \quad (2.36)$$

$$\tilde{\psi}_g^-(\vec{r}, \hat{\Omega}) = -\tilde{\psi}_g^-(\vec{r}, -\hat{\Omega}), \quad (2.37)$$

$$\int \tilde{\psi}_g^-(\vec{r}, \hat{\Omega}) d\Omega = 0. \quad (2.38)$$

Substituting these into Eq. (2.30) yields the following equation.

$$\hat{\Omega} \cdot \vec{\nabla} \left[ \tilde{\psi}_g^+(\vec{r}, \hat{\Omega}) + \tilde{\psi}_g^-(\vec{r}, \hat{\Omega}) \right] + \tilde{\Sigma}_{t,g}(\vec{r}, t) \left[ \tilde{\psi}_g^+(\vec{r}, \hat{\Omega}) + \tilde{\psi}_g^-(\vec{r}, \hat{\Omega}) \right] = \tilde{W}_g(\vec{r}, \hat{\Omega}) + \tilde{S}_g(\vec{r}, \hat{\Omega}). \quad (2.39)$$

Evaluating this equation at  $-\hat{\Omega}$  and adding that equation to Eq. (2.30) yields the even-parity form of the transport equation in Eq. (2.40). Evaluating Eq. (2.39) at  $-\hat{\Omega}$  and subtracting from Eq. (2.30) yields the odd-parity form of the transport equation given by Eq. (2.41).

$$\hat{\Omega} \cdot \vec{\nabla} \tilde{\psi}_g^-(\vec{r}, \hat{\Omega}) + \tilde{\Sigma}_{t,g}(\vec{r}) \tilde{\psi}_g^-(\vec{r}, \hat{\Omega}) = \tilde{W}_g^-(\vec{r}, \hat{\Omega}) + \tilde{S}_g^-(\vec{r}, \hat{\Omega}), \quad (2.40)$$

$$\hat{\Omega} \cdot \vec{\nabla} \tilde{\psi}_g^+(\vec{r}, \hat{\Omega}) + \tilde{\Sigma}_{t,g}(\vec{r}) \tilde{\psi}_g^+(\vec{r}, \hat{\Omega}) = \tilde{W}_g^+(\vec{r}, \hat{\Omega}) + \tilde{S}_g^+(\vec{r}, \hat{\Omega}). \quad (2.41)$$

We have defined the even- and odd-parity components of the within group source by:

$$\begin{aligned} \tilde{W}_g^+(\vec{r}, \hat{\Omega}) &= \sum_{m \in \text{even}} \int_{4\pi} \Sigma_{s,g \rightarrow g,m}(\vec{r}) P_m(\hat{\Omega} \cdot \hat{\Omega}') \tilde{\psi}_g^+(\vec{r}, \hat{\Omega}') d\hat{\Omega}' \\ \tilde{W}_g^-(\vec{r}, \hat{\Omega}) &= \sum_{m \in \text{odd}} \int_{4\pi} \Sigma_{s,g \rightarrow g,m}(\vec{r}) P_m(\hat{\Omega} \cdot \hat{\Omega}') \tilde{\psi}_g^-(\vec{r}, \hat{\Omega}') d\hat{\Omega}' \end{aligned} \quad (2.42)$$

Above we have used the identities  $P_m(\hat{\Omega} \cdot \hat{\Omega}') = P_m(-\hat{\Omega} \cdot \hat{\Omega}')$  for  $m \in \text{even}$  and  $P_m(\hat{\Omega} \cdot \hat{\Omega}') = -P_m(-\hat{\Omega} \cdot \hat{\Omega}')$  for  $m \in \text{odd}$ . Furthermore, the odd parity flux is an odd function of  $\hat{\Omega}'$  and therefore integrates to zero when weighted by even order functions of  $\hat{\Omega}'$  (even order

Legendre moments and consequently spherical harmonics). This causes the cancellation of several terms. Similarly, the remaining even- and odd-parity components of the source term are defined in Eq. (2.42), where the isotropic terms drop out of the odd-parity component definition:

$$\begin{aligned}
 S_g^+(\vec{r}, \hat{\Omega}) &= \sum_{g' \neq g}^G \sum_{m=\text{even}}^M \int_{\Sigma_{s,g' \rightarrow g,m}}(\vec{r}) P_m(\hat{\Omega} \cdot \hat{\Omega}') \tilde{\psi}_{g'}^+(\vec{r}, \hat{\Omega}') d\hat{\Omega}' \\
 &+ \frac{1}{4\pi} \sum_j \chi_g^j (1 - \beta^j) \sum_{g'=1}^G v \Sigma_{f,g'}^j(\vec{r}, t) \tilde{\phi}_{g'}^+(\vec{r}, t) + \frac{1}{n(t)} \frac{1}{4\pi} \sum_i \chi_{i,g} \lambda_i C_i(\vec{r}, t) \\
 &+ \frac{1}{n(t)} q_g^+(\vec{r}) + v_g^{-1} \frac{1}{\Delta t} \tilde{\psi}_g^+(\vec{r}, \hat{\Omega}, t - \Delta t) \\
 S_g^-(\vec{r}, \hat{\Omega}) &= \sum_{g' \neq g}^G \sum_{m=\text{odd}}^M \int_{\Sigma_{s,g' \rightarrow g,m}}(\vec{r}) P_m(\hat{\Omega} \cdot \hat{\Omega}') \tilde{\psi}_{g'}^-(\vec{r}, \hat{\Omega}') d\hat{\Omega}' \\
 &+ \frac{1}{n(t)} q_g^-(\vec{r}) + v_g^{-1} \frac{1}{\Delta t} \tilde{\psi}_g^-(\vec{r}, \hat{\Omega}, t - \Delta t)
 \end{aligned} \tag{2.43}$$

The additional IQS terms in the source term components of Eq. (2.43) causes considerable complications in PROTEUS-SN. The even-parity source term can be straightforwardly computed as the even-parity angular flux is computed in PROTEUS-SN. However, the odd-parity source term requires the odd-parity angular flux to be computed (at all discrete ordinates, at all vertices, and stored for the previous time step). While mathematically we can show that this odd-parity term is consistent with the even-parity methodology, this procedure requires major changes in the PROTEUS-SN code which still leaves us with an even-parity code when we are done.

The interested reader should refer to the PROTEUS-SN methodology manual [4] to understand which steps are taken (discrete ordinates approximation and finite element approximation) past this stage.

#### 2.1.4 Summary

The improved quasi-static (IQS) method for kinetics has been reviewed. The IQS equations were derived from the time-dependent Boltzmann transport equation (with accompanying precursor balance equation) and implementation guidelines were reviewed based on recent journal articles. Finally, the discussion on issues with implementation in the even-parity method is included. It is suggested that the IQS method be implemented for a first order solver which can compute the full angular flux rather than only the scalar flux (and even-parity angular flux).

## 2.2 User-friendly Finite Element Mesh Generation Capability

### 2.2.1 UFmesh and GRID

The PROTEUS code has a built-in finite element library which provides one-, two-, and three-dimensional modeling capabilities. The finite element library is defined by its data structure setup which consists of a serial instance called NTmesh and a parallel instance called PNTmesh. [22] The latter of these contains the parallel details of a given mesh which maps the locally

visible mesh information into the global quantities on the parallel system. For PROTEUS, one needs to provide a mesh as input and while mesh generation tools such as CUBIT [11] are available, they can be considerably complicated to use for simple geometries.

Given the complexity of real world problems, experience has shown that using commercial mesh generator to create rather simple input geometries is overly complex and slow. As a consequence, significant effort has been put into place to create multiple codes that help assist in the mesh generation and manipulation. Even though MeshKit [12] is a convenient tool which provides more flexibility in generating mesh files for various geometry components, its mesh is based on CUBIT. This sometimes makes it difficult or inconvenient to verify the mesh files in terms of geometry and boundary condition. In addition, preserving the geometry volume with finite element meshes is a big concern for the CUBIT-generated meshes.

To improve the user convenience for mesh generation, a user-friendly built-in mesh generation tool was developed to generate meshes for simple Cartesian and hexagonal geometries without using CUBIT: Ufmesh and GRID. The Ufmesh is a simple way to generate two-dimensional Cartesian and hexagonal fuel assembly geometries. The GRID input allows users to generate conventional homogenized geometries for structured grid codes containing options for Cartesian, hexagonal, and regular triangular geometry options. Because it does its visualization using the finite element method, one can use the GRID input to construct a finite element mesh of the domain. Figure 2 shows how different the meshes generated from RGG (MeshKit) and Ufmesh tools look.

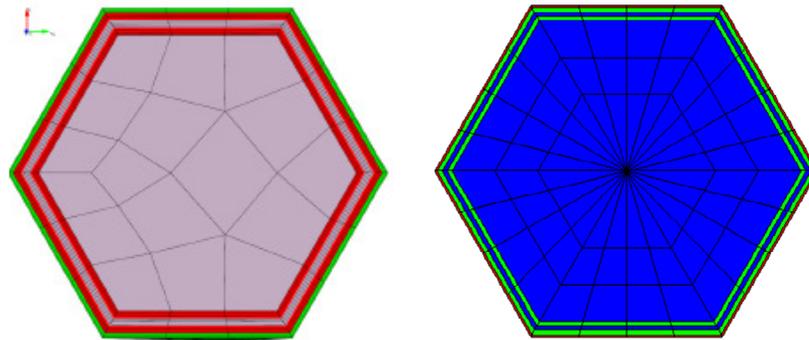


Figure 2. Meshes Generated from RGG-MeshKit (left) and User-friendly Mesh Generation (right)

The Ufmesh format was developed to assist in the creation of meshes for pin-cells and to better control the meshing behavior than what is practical in CUBIT. The Ufmesh is a keyword-based input description. The current Ufmesh capability only supports Cartesian and Hexagonal geometries. Note that Ufmesh only creates linear two-dimensional meshes of circular geometries and thus curvilinear surfaces are always represented using faceted surfaces. Thus all cylinders are linearly tessellated and the radius of each cylinder is adjusted to give the correct volume. While the input does allow the conversion to higher order basis functions, the cylindrical surfaces are

always based upon the linear tessellation such that the cylinder volume is always exact. We have significant extra work to ensure this in CUBIT and MeshKit.

Figure 3 and Figure 4 show the example inputs for Cartesian and hexagonal geometry fuel assemblies, respectively. The resulting UFmesh outputs are illustrated in Figure 5. More detailed information is discussed in Reference [22].

```

GRID_TYPE CARTESIAN 5 5
GRID_PITCH 2.0 2.0
ELEMENT_ORDER 1
ASSEMBLY_SIZES 10.0 11.0 12.0
ASSEMBLY_REGIONS DUCT1 DUCT2
ASSEMBLY_MESHES 1 2
PINCELL_RADII A1 0.1 0.5 0.75
PINCELL_REGIONS A1 FUEL1 FUEL1 CLAD1 MODERATOR
PINCELL_MESHES A1 1 2 1 2
PINCELL_RADII A2 0.1 0.5 0.6 0.7 0.75
PINCELL_REGIONS A2 FUEL1 FUEL1 INNERCLAD2 POISON2 OUTERCLAD2 MODERATOR
PINCELL_MESHES A2 1 2 1 1 1 2
PINCELL_REGIONS A3 MODERATOR
PINCELL_MESHES A3 3
GRID_MAP 5 A2 A1 A1 A1 A2
GRID_MAP 4 A1 A2 A2 A2 A1
GRID_MAP 3 A1 A2 A3 A2 A1
GRID_MAP 2 A1 A2 A2 A2 A1
GRID_MAP 1 A3 A1 A1 A1 A3
PINCELL_AZIMUTHAL 3 4 3 4 3 4 3 4
ASSEMBLY_BORDER 22 18 22 18

```

Figure 3. Example Cartesian Sector and Face Meshing Control UFmesh File

```

GRID_TYPE HEXAGONAL 5 5
GRID_PITCH 2.0
ELEMENT_ORDER 2
ASSEMBLY_SIZES 10.20 10.3
ASSEMBLY_REGIONS DUCT1
ASSEMBLY_MESHES 1
PINCELL_RADII A1 0.1 0.5 0.75
PINCELL_REGIONS A1 FUEL1 FUEL1 CLAD1 MODERATOR
PINCELL_MESHES A1 1 2 1 2
PINCELL_RADII A2 0.1 0.5 0.6 0.7 0.75
PINCELL_REGIONS A2 FUEL1 FUEL1 INNERCLAD2 POISON2 OUTERCLAD2 MODERATOR
PINCELL_MESHES A2 1 2 1 1 1 2
PINCELL_REGIONS A3 MODERATOR
PINCELL_MESHES A3 3
GRID_MAP 5 A1 A1 A1 O O
GRID_MAP 4 A1 A3 A2 A1 O
GRID_MAP 3 A1 A2 A1 A2 A1
GRID_MAP 2 O A1 A2 A3 A1
GRID_MAP 1 O O A1 A1 A1
PINCELL_AZIMUTHAL 4

```

Figure 4. Example Hexagonal Assembly UFmesh File

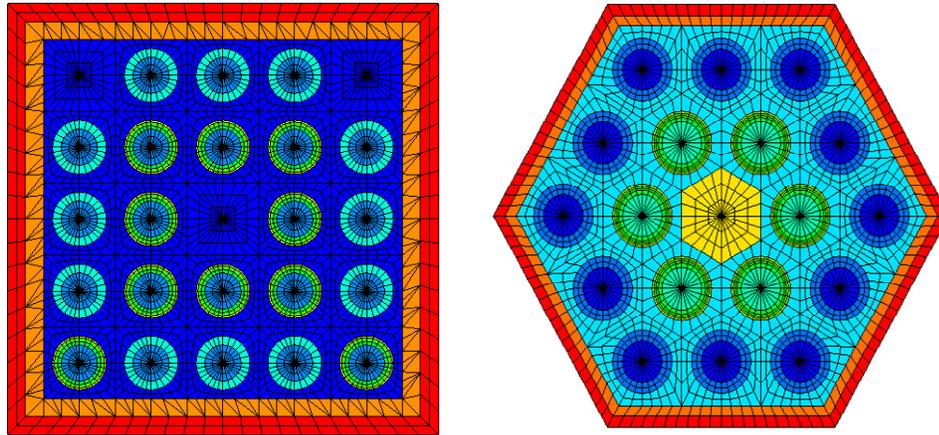


Figure 5. Cartesian and Hexagonal Meshes Resulting From the User-friendly Mesh Generation Capability of PROTEUS

### 2.2.2 Verification Tests

Verification tests of UFmesh were conducted using the C5 benchmark problem for which came from the C5G7 PWR benchmark problem [28] with the unrodded condition by changing two-region (fuel and cladding+moderator) to three-region pin (fuel, cladding, and moderator) with typical Westinghouse-type  $\text{UO}_2$  and MOX geometry and isotopic compositions.

With PROTEUS, a mesh and angular convergence study was conducted to determine the optimal mesh refinement and Legendre-Tchebychev ( $L_m T_n$ ) angular cubature for a pin cell: spatial mesh varied from 80 to 528 elements and angular cubature order from L3T3 to L7T35. PROTEUS-SN2ND (using linear and quadratic order element meshes) and PROTEUS-MOCFE (using linear order element mesh) were run for the 3.1 wt%  $\text{UO}_2$  pin cell of the C5 benchmark problem. Figure 6 shows the convergence of mesh refinement and angular Tchebychev order with angular Legendre order L5 and 383 elements (1198 vertices). The eigenvalues of PROTEUS were compared with the MCNP [29] solutions using the same 4-group multi-group cross sections.

The mesh and angle refinement study indicates that with the UFmesh, SN requires quadratic order elements to obtain accurate solutions whereas the linear order elements are sufficient for MOC. Furthermore the MOC eigenvalue is more accurate than any of the SN for the same mesh and angle although it is important to note that MOC uses many more characteristic lines than there are elements in the domain. From the mesh and angle refinement study, 352 elements and L5T17 angular cubature were selected for testing fuel assemblies and a 2D core.

The same mesh and angle refinement study was conducted using the cross section API with the 4-group cross section library, resulting in the similar conclusion made with the 4-group multigroup cross sections above. Unlike the previous comparison, DeCART [21] solutions were used as reference solutions because the 4-group cross section library is not accurate enough to compare with continuous energy MCNP and both PROTEUS and DeCART use the same resonance self-shielding methodology.

The UOX and MOX fuel assemblies and 2D core of the C5 benchmark problem were simulated with PROTEUS using the same cross section libraries, whose eigenvalue solutions were compared with multi-group MCNP or DeCART with the same library. For the 2D core, the UFmesh and GRID meshes were used for fuel assemblies and reflector region, respectively, as

shown in Figure 7. The GRID mesh spacing was made coarser as approaching to the core boundaries.

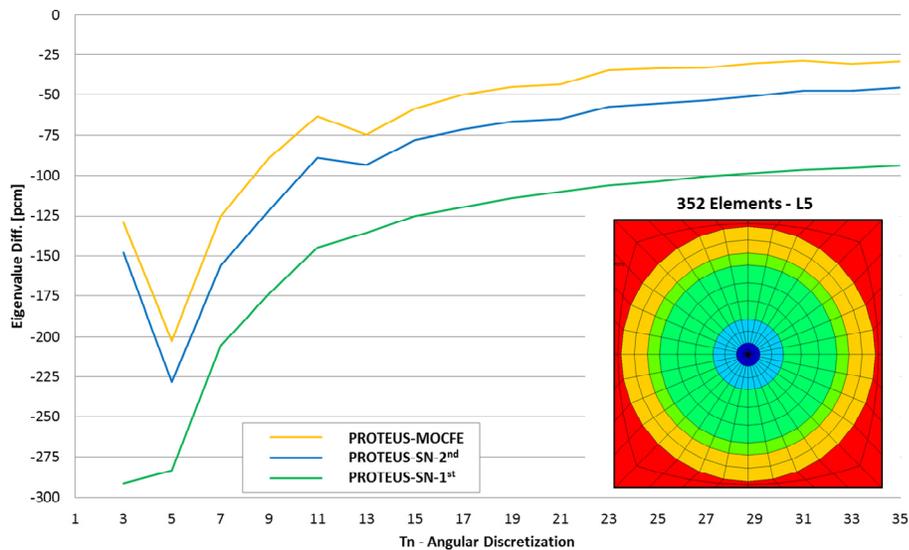


Figure 6. Eigenvalue Convergence with Mesh (element/vertex) and Angular Refinement for a Pin Cell

Table 2 digests the eigenvalue comparison between PROTEUS and MCNP when the 4-group multigroup cross sections are used for both codes. For PROTEUS, L5T15 angular cubature was used with or without boundary layer meshes. Table 2 shows the eigenvalue comparison between PROTEUS and DeCART when the 4-group cross section library is used. As seen in the tables, the MOC eigenvalues with the linear meshes were closer to the reference solutions, compared to the SN eigenvalues with the quadratic meshes in most cases except for the 2D core case. The comparison results indicate that the UFmesh and GRID meshes work well for such a typical PWR problem.

Table 1. Eigenvalue Comparison between PROTEUS and MCNP Using 4G Multigroup Cross Sections

CASE	MCNP <sup>a)</sup>	Mesh	SN <sup>b)</sup>		MOC <sup>b)</sup>
			Linear Mesh	Quadratic Mesh	Linear Mesh
UOX Pin (3.1 wt%)	1.31751	BL	-83	-41	-31
		No BL	-97	-58	-34
MOX Pin (7.4 wt%)	1.25267	BL	-161	-69	-58
		No BL	-165	-71	-61
UOX FA	1.33695	BL	-50	-19	-11
		No BL	-58	-22	-12
MOX FA	1.24818	BL	-100	-55	-43
		No BL	-105	-54	-44
2D Core	1.23961	BL	- 24	+55	-101

a) MCNP standard deviation  $\leq 7$  pcm, b)  $\Delta k$  (pcm) difference from reference

Table 2. Eigenvalue Comparison between PROTEUS and DeCART Using 4G Cross Section Library

Case	Reference	SN <sup>a)</sup>		MOC <sup>a)</sup>
		Linear Mesh	Quadratic Mesh	Linear Mesh
UOX Pin (3.1 wt%)	1.29785	-264	-86	-61
UOX FA	1.32556		-126	-98
MOX FA	1.03817		-90	-64

a)  $\Delta k$  (pcm) difference from reference

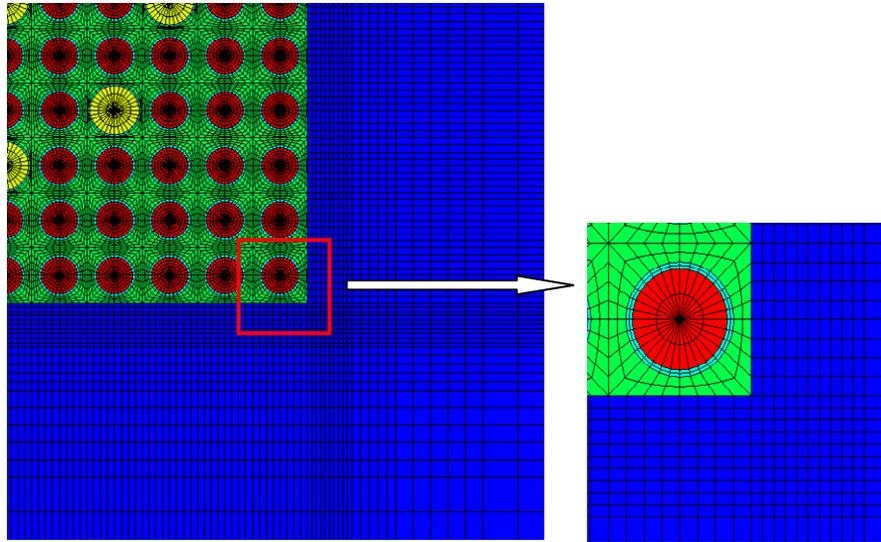


Figure 7. 2D Core Mesh Using Ufmesh and GRID

### 3. Cross Section Library and API

In the previous years, the Argonne cross section library (ACSL) and methodology were developed for application to various reactor types including LWR, HTR, and SFR. The Genesis code (formerly named GeneCS) [25] produces the ACSL using NJOY [29] and MC<sup>2</sup>-3. The base ACSL is composed of isotopic cross sections and resonance tables with 2158 energy groups which include 2123 groups with constant lethargy intervals from 20 MeV to 0.414 eV and 35 groups from 0.414 to 10<sup>-4</sup> eV with variable lethargy intervals. Genesis is able to determine the broad group structures optimized for a reactor of interest. The use of the ACSL requires solving the fixed source problems (FSPs) to determine the escape cross sections which account for the heterogeneity effect in the resonance cross section. The main reason for the use of the ultrafine group (UFG) cross section set as a base library is that the fast reactor cross section often requires a UFG spectrum calculation to maintain accuracy.

The cross section API is a set of routines which read the ACSL, assign compositions and geometry data to cross section regions, calculate the escape cross sections, and then generate multigroup cross sections. It is calling the FSP solver which should be provided by the transport solver (SN2ND, MOCFE, or MOCEX). Therefore, the preparation of the FSP solver is most of the work when connecting the cross section API to any transport code. Previously, the cross section API was connected to the SN2ND and MOCEX solvers, but this year it was linked to the MOCFE solver. Thus, all the exiting solvers are able to use the cross section API to generate multigroup cross sections on the fly although both MOCEX and MOCFE are not releasable as production versions at this time.

Previously, the cross section API included two cross section generation method options: the subgroup method [30] and the resonance table method [24]. Both methods require solving the FSPs to determine the escape cross sections which contribute to the total background cross sections. The number of the FSPs for the subgroup method is  $G \times R \times S$  where  $G$  is the number of resonance energy groups,  $R$  is the number of resonant isotopes or categories, and  $S$  is the number of subgroups. The number of categories is normally four or more, the number of resonance isotopes depends on problem compositions, and the number of subgroups is usually 4 to 7. For the resonance table method,  $S$  should be the number of iterations to converge total cross sections, which is normally 2 to 4. Basically both the subgroup method and the resonance table method require solving at least  $G \times 16$  FSPs to determine the escape cross section.

This year, as an alternative resonance self-shielding option, the Dancoff approach was developed to reduce the computation time. Since the Dancoff approach requires  $G \times 2$  FSPs only (fuel and cladding), the number of FSPs to solve is reduced by more than an order of magnitude.

#### 3.1 Verification of Argonne Cross Section Library and API

The broad-group cross section libraries contain resonance cross sections as a function of temperature and background cross sections. Previously, those libraries generated from the Genesis code have been tested for different background cross sections at 300K. Verification tests were extended for temperature coefficients. A typical fuel pin and assembly of HTR were selected for the test varying the temperature from 300 to 1500K. The  $k_{\text{eff}}$  values of the fuel pin and assembly were calculated using DeCART with the 198-group cross section library. As shown

in Figure 8, the temperature coefficients (pcm/K) agreed within 2% for all temperature conditions with the MCNP solutions except for the case of the fuel assembly with 600-900K which showed a difference of 4.6%. This appears to be a problem with the MCNP solutions, compared with the results in the fuel pin which represent very good agreement at all temperature conditions.

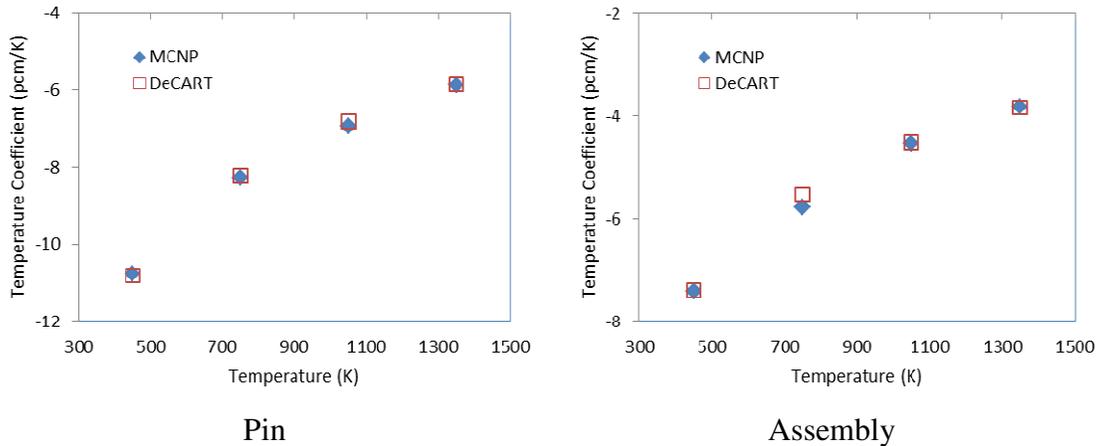


Figure 8. Comparison of Temperature Coefficients of HTR Fuel Pin and Assembly

The broad-group cross section libraries generated for LWR were tested for the selected VERA PWR benchmark problems [32] which include 2 fuel pin and 7 fuel assembly (17×17) problems with different burnable poisons and control rod absorbers. The configurations of the problems are presented in Table 3, among which the 2F and 2P fuel assembly cases are shown in Figure 9. The original configurations of the problems were modified with simplified compositions and a constant temperature of 300K. The  $k_{eff}$  values of those problems were calculated using DeCART with the 204-group cross section library and MCNP with the continuous-energy library. As seen in Table 3, the eigenvalues are in good agreement between DeCART and MCNP within 174 pcm  $\Delta k$  for fuel pins and 137 pcm  $\Delta k$  for fuel assemblies.

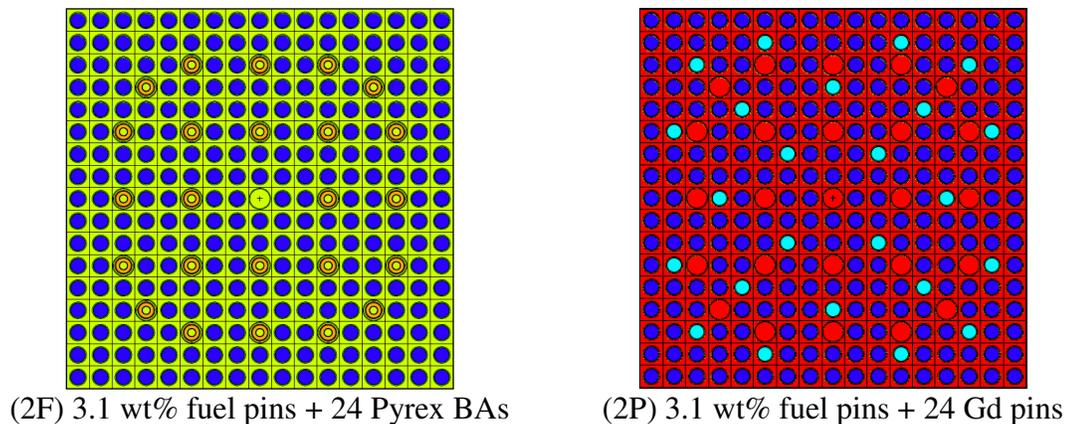


Figure 9. Selected Fuel Assemblies from the VERA PWR Benchmarks

Table 3. Eigenvalue Comparison between MCNP and DeCART (204 Group Library) for the VERA PWR Benchmark Problems

Case	Description	MCNP	DeCART 204G ( $\Delta k$ , pcm)
1A	3.1wt% fuel pin	1.21056	174
1E	3.1wt% fuel pin + IFBA	0.78878	99
2A	3.1wt% fuel pins	1.20201	71
2F	3.1wt% fuel pins + 24 Pyrex BA	1.00678	-136
2G	3.1wt% fuel pins + 24 AgInCd control rods	0.88100	-85
2K	3.1wt% & 3.6 wt% fuel pins + 24 Pyrex BA	1.04989	-113
2M	3.1wt% fuel pins + 128 IFBA pins	0.96075	137
2N	3.1wt% fuel pins + 104 IFBA pins + 20 WABA	0.89518	-109
2P	3.1wt% fuel pins + 24 Gd pins	1.19347	45

\* Standard deviation of MCNP  $\leq$  20 pcm

The cross section API with the ACSL was verified by comparing DeCART and PROTEUS solutions. Verification tests were conducted for homogeneous and pin cell problems. The four-group cross section library with the resonance table method was chosen for the tests. For homogeneous compositions with 3.1 wt% U-235, the number density of oxygen varied to provide different background cross sections of uranium isotopes. The  $k_{\infty}$  solutions of PROTEUS were compared with DeCART, resulting in only several pcm differences between the two codes as shown in Figure 10. Those differences are acceptable but larger than the expected numerical errors. We noticed that although the escape cross sections are supposed to be zero for homogeneous compositions, the fixed source problem solutions of PROTEUS may be sensitive to the default convergence criteria embedded in the code.

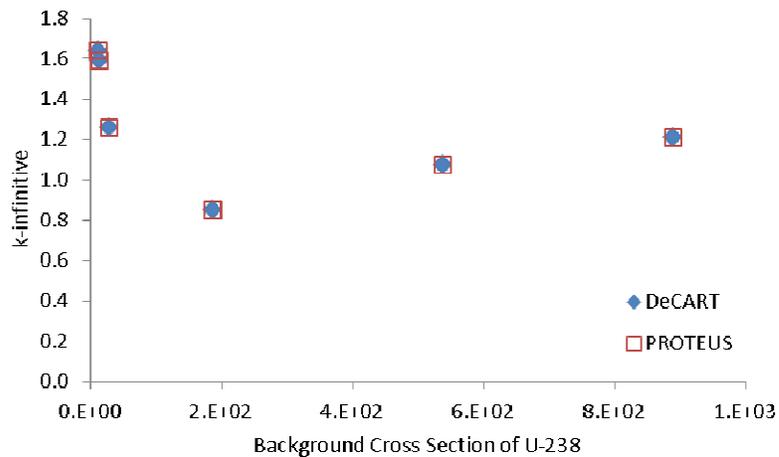


Figure 10. Comparison of  $k_{\infty}$  between DeCART and PROTEUS (Using the 4-group Cross Section Library) for Homogeneous Compositions

Additional PROTEUS calculations were conducted for pin cell problems with 3.1 or 24.2 wt% U-235 to provide different background cross sections for U-238. The spatial meshes employed for the C5 benchmark problem were used, which is composed of 308 elements (1198 vertexes) as shown in Figure 11. The Legendre-Tchebychev order was changed from L3T5 to L13T55 to see the convergence of eigenvalue. The PROTEUS eigenvalues were compared with

DeCART which were obtained with relatively little mesh refinement (solution is based upon characteristic lines and not elements). Unlike the subgroup method option with which PROTEUS-SN suffered from a large discrepancy in eigenvalue from the two codes, the converged eigenvalue of PROTEUS with the resonance table option agreed very well with DeCART within 5 pcm and 19 pcm  $\Delta k$  for 24.2 wt% ( $k_{\infty} = 1.79977$ ) and 3.1 wt% ( $k_{\infty} = 1.32893$ ) pin-cell problems, respectively.

The relatively large difference in  $k_{\infty}$  between PROTEUS-SN and DeCART observed from the subgroup method option can be attributed to the large “subgroup” flux gradient, which required using the boundary layer meshing and high-order angular cubature. With the resonance table method option, however, the flux gradient is not so much as that of the subgroup method option.

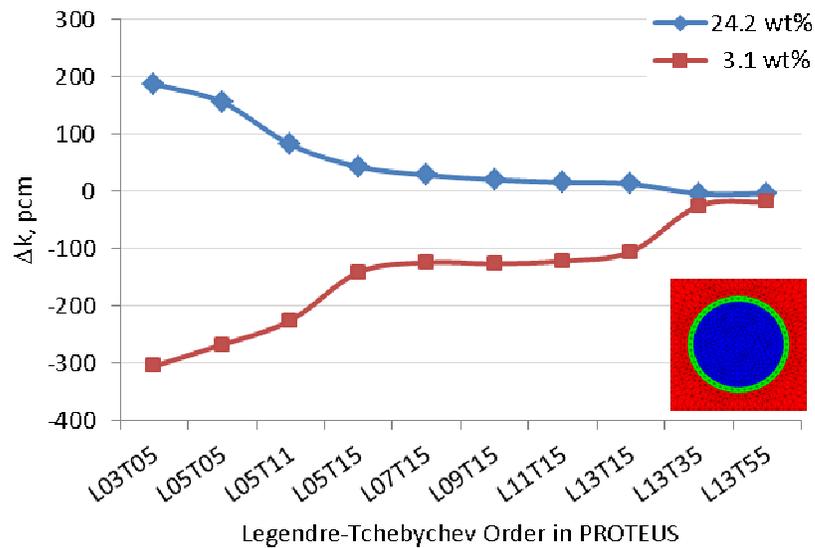


Figure 11. Difference in  $k_{\infty}$  between DeCART and PROTEUS (Using the New 4-group Cross Section Library) for Pin-cell Problems

### 3.2 Alternative Resonance Self-Shielding Method

The resonance self-shielding methods currently available in the cross section API of PROTEUS are the subgroup method and the resonance table method, both of which require solving the fixed source problems (FSPs) for each resonance isotope or category, each resonance group, and each subgroup. Solving many FSPs is a big computational burden for PROTEUS. Therefore, the Dancoff approach was added as an alternative resonance self-shielding option to reduce the computation time to be spent for the FSPs.

The Dancoff correction is to reduce the in-current of resonance neutrons into the fuel in a closely packed lattice, compared to the in-current into a single fuel rod in an infinite moderator because of the shadowing effect of adjacent rods. In other words, the Dancoff factor is the probability that a neutron emitted isotropically from the surface of the fuel region of the fuel element will have its next collision in the fuel region of any other surrounding fuel element.

Therefore, the Dancoff factor is used to adjust the first collision probabilities for lattice-cell geometry to account for the presence of other fuel rods for a single fuel lump in an infinite moderator. The Dancoff correction does not depend on the composition of a fuel while it is calculated by the geometry and cross section of the moderator. This allows the calculation process to be simple while its application is often limited to typical geometries.

For this Dancoff approach, the resonance table as a function of background and temperature for each isotope is generated from NJOY. The smooth cross sections and the intermediate resonance (IR) parameters are prepared as well. The fixed source problem with the source ( $\lambda\Sigma_p$ ) is solved to determine the Dancoff factors for fuel and cladding regions.

$$RI_t(E) = \Sigma_t^f(E) \frac{\Sigma_s^f + \Sigma_e}{\Sigma_t^f(E) + \Sigma_e} \frac{1}{E}, \quad (3.1)$$

where  $RI_t$  = total resonance integral that is calculated by solving the fixed source problem,  $\Sigma_t^f(E)$  = total cross section for fuel,  $\Sigma_s^f$  = scattering cross section for fuel,  $\Sigma_e$  = escape cross section defined by the inverse of the average chord length,  $1/\bar{\ell} = S/4V$ . Assuming a black pin (i.e.,  $\Sigma_t^f \rightarrow \infty$ ), Eq. (3.1) becomes

$$RI_t(E) = \Sigma_s^f + \Sigma_e = \Sigma_b + D\Sigma_e \rightarrow D = (RI_t(E) - \Sigma_b) / \Sigma_e, \quad (3.2)$$

where  $\Sigma_b$  = background cross sections for homogeneous mixture and  $D$  = Dancoff factor. Once the total resonance integral is solved using the FSP assuming the infinite cross sections in the fuel region, the Dancoff factor for the fuel region can be calculated using Eq. (3.2). When resonant isotopes are included in the cladding region, the similar approach can be applied to calculation the Dancoff factor for the cladding region.

Since the one-term rational approximation is known to be not accurate, the non-escape probability ( $P_{ff}$ ) can be represented by the two-term rational approximation as Eq. (3.3). Finally, the resonance cross section ( $\sigma_x^i$ ) for the reaction  $x$  of the resonant isotope  $i$  is determined by Eq. (3.4).

$$P_{ff}(E) = \sum_{n=1}^2 \frac{\beta_n \Sigma_t(E)}{\Sigma_t(E) + \alpha_n \Sigma_e}, \quad (3.3)$$

$$\sigma_x^i = \frac{\sum_{n=1}^2 \beta_n RI_{xn}^i(\sigma_{bn}^i)}{1 - \sum_{n=1}^2 \frac{\beta_n RI_{an}^i(\sigma_{bn}^i)}{\sigma_{bn}^i}}, \quad (3.4)$$

where  $RI_x$  = resonance integral of reaction  $x$ ,  $\sigma_{bn}^i = \left( \sum_k \lambda \Sigma_p^k + \alpha_n \Sigma_e \right) / N^i$ , and  $\alpha_n, \beta_n$  = n-th term coefficients.

In the Dancoff approach, only 2G times of the fixed source problems are necessary to determine the Dancoff factors for the fuel and cladding regions if resonant isotopes are included in both regions. Using the Dancoff approach which is to be considered as the third option in the

cross section API, a large reduction of the computation time is expected, compared to the other two resonance self-shielding options (the subgroup and resonance table method options).

Preliminary verification tests were conducted with the pin cells of the C5 benchmark problem: 3.1 wt% UO<sub>2</sub> and 3.8 wt% MOX. The 72 group cross section library was used for the tests, whose group structure (lethargy intervals) is shown in Figure 12. The eigenvalue solutions from the Dancoff approach were compared with the MCNP solutions, showing good agreement for 3.1 wt% UO<sub>2</sub> pin cell but relatively large overestimation for 3.8% MOX pin cell (see Table 4). As shown in Figure 13, the large underestimation of absorption cross sections of Pu-239 were observed in its resolved resonance energy range compared to the MCNP tallied cross sections, whereas the absorption cross sections of U-238 and U-235 are in good agreement with MCNP. This can be attributed to the inaccurate estimation of the resonance interferences between U-238, U-235, and Pu-239. Therefore, an accurate estimation of the resonance interference effect remains as a future work.

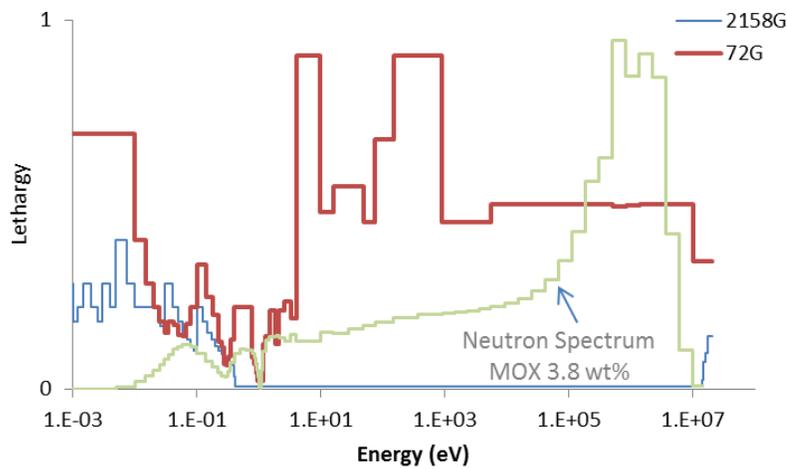


Figure 12. The 72 Group Structure Used for the Dancoff Approach

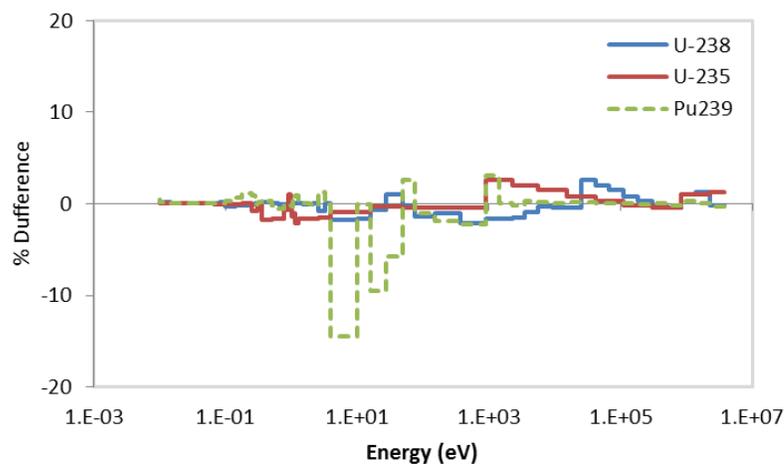


Figure 13. Percent Differences of Absorption Cross Sections between Dancoff Approach and MCNP for 3.8 wt% MOX Pin Cell

Table 4. Eigenvalue Comparison between MCNP and Dancoff Approach Using the 70-group Library for UO<sub>2</sub> and MOX Pin Cells

Fuel Type	MCNP	$\Delta k$ , pcm
3.1 wt% UO <sub>2</sub>	1.36845 ±0.00011	-53
3.8 wt% MOX	1.15161 ±0.00014	338

### 3.3 Cross Section Generation Using Monte Carlo Codes

The Serpent code [33] is a continuous energy Monte Carlo reactor physics code. The code includes a capability of generating multigroup (MG) cross sections which would be useful to perform core analysis as well as to benchmark cross sections generated by deterministic approaches. The MG cross sections generated from the latest version of Serpent have the following limitations:

- Macroscopic cross sections only: i.e., no isotopic microscopic cross sections
- Total scattering matrices only: i.e., no partial scattering matrices such as elastic, inelastic, n<sub>2n</sub>, and n<sub>3n</sub> cross sections.
- No high-order P<sub>N</sub> scattering matrices and total cross sections
- No partial principle cross sections such as (n,α), (n,p), (n,d), and (n,t) which are included in the capture cross section

Due to the limitations above, the MG cross sections generated from Serpent are so far adequate to use in diffusion calculation codes. In addition, the use of many cross sections (more than a few to several groups) may require thorough verification tests.

To use the MG cross sections generated from Serpent in our code suite, a computer code was developed, which reads the Serpent cross section output (the .m file) and produces a cross section file in the ISOTXS format. Since the contribution of the (n,2n) cross sections is significant for fast reactor analysis, those cross sections need to be separate from the total scattering cross section matrices. In addition, the high-order P<sub>N</sub> scattering matrices and total cross sections are needed for transport calculations.

To provide the data missing in the Serpent, the cross sections generated from MC<sup>2</sup>-3 were utilized when converting the Serpent cross section output to the ISOTXS format. Assuming scattering matrices do not change much, those for P<sub>N</sub> scattering and (n,2n) are obtained from MC<sup>2</sup>-3, as shown in Figure 14. In future, the use of GROUPR of NJOY will be tried instead of MC<sup>2</sup>-3 to provide accurate cross section data for the thermal energy range.

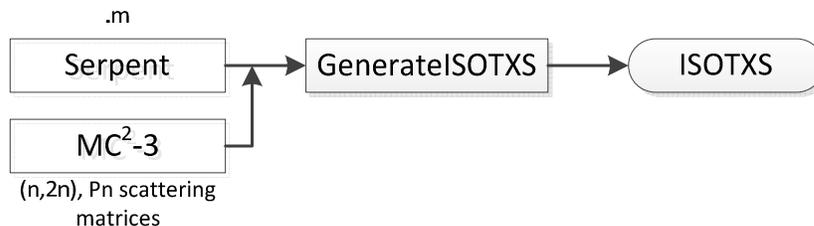


Figure 14. ISOTXS Generation Using Serpent Multigroup Cross Section

Verification tests were performed with the C5 benchmark problem, showing that the Serpent eigenvalues for pin cells and fuel assemblies were well reproduced in the DIF3D calculations with the ISOTXS files generated from Serpent and MC<sup>2</sup>-3. For MC<sup>2</sup>-3 cross sections, the homogeneous mixture model equivalent to a pin cell or a fuel assembly was used. For a 2D core calculation, the 23-group cross sections for the UO<sub>2</sub> and MOX fuel assemblies were provided from single fuel assembly models, and those for the water reflector were prepared from the two-region problem with water and a UO<sub>2</sub> assembly from which the cross sections for the water region only were tallied.

As shown in Table 5, the eigenvalue difference between MCNP and DIF3D for the 2D C5 benchmark problem (shown in Figure 15) is only -34 pcm which is smaller than the differences between MCNP and Serpent observed for fuel assemblies. Further tests with various benchmark problems will be conducted to verify the cross section generation procedure. Note that relatively large differences between MCNP and Serpent were observed in the test results because they used their own cross section library. We observed that the eigenvalue differences between the two codes were reduced to below 50 pcm when both used the same library.

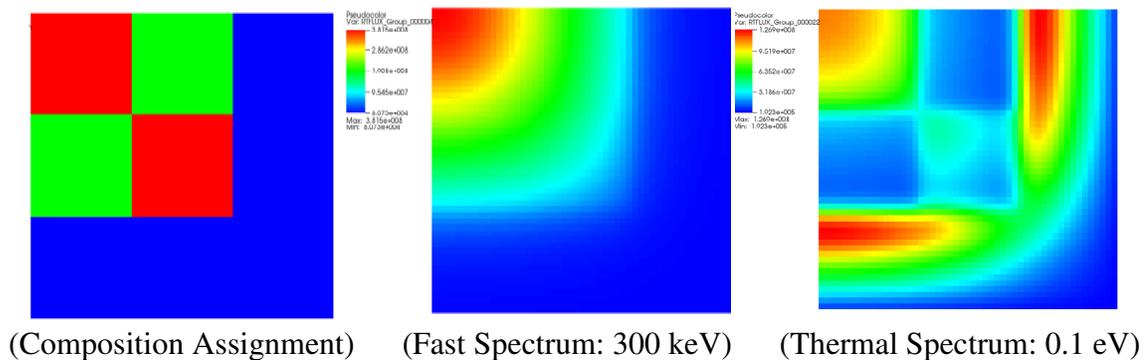


Figure 15. Composition Assignment and Fast and Thermal Fluxes of the C5 Problem Resulting from DIF3D with the 23-group Multigroup Cross Sections Generated from Serpent

Table 5. Eigenvalue Comparison for the C5 Benchmark Problem between MCNP, Serpent, and DIF3D (23-group MG cross sections)

Case	MCNP	Serpent (pcm Δk)	DIF3D (pcm Δk)
MOX FA (7.4 w/o)	1.21171	-137	
UOX FA (3.1 w/o)	1.39025	-99	
2D Core	1.19448		-34

\* Standard deviation of MCNP and Serpent ≤ 20 pcm

## 4. Multidimensional Capability of MC<sup>2</sup>-3

The unit cells of plate-type fast critical assemblies such as those seen in ZPR-3, ZPR-6 and ZPPR, are formed of steel square tubes and steel drawers containing rectangular plates of different materials such as depleted, enriched, or natural uranium, plutonium, stainless steel, sodium, etc. These unit cells show a two-dimensional nature because of the variation in plate dimensions and the drawer structure, but in the current MC<sup>2</sup>-3 with 1D transport calculation capabilities [15], they are approximated with one-dimensional (1D) slab models along the dominant direction of heterogeneity to account for the local heterogeneity effects.

In order to represent the local heterogeneity effects more accurately for these unit cells, two-dimensional (2D) transport calculation capabilities have been developed and implemented in the MC<sup>2</sup>-3 code. An efficient 2D method of characteristics (MOC) solver based on a modular ray tracing technique was implemented and a parallel computation capability was realized using OpenMP. The implemented 2D transport solver can handle the 2D geometries of the ZPR type unit cells explicitly and can perform the hyperfine-group (HFG) and ultrafine-group (UFG) slowing-down calculations with P<sub>1</sub> anisotropic scattering sources within the MC<sup>2</sup>-3 framework. For the UFG calculation, equivalence theory is used to generate self-shielded ultrafine-group cross sections for individual plates. As initial verification tests, the unit cell calculations were performed for the fuel drawer of ZPR-6 Assembly 7, and the obtained results were compared against the 1D collision probability method (CPM) solution of the current MC<sup>2</sup>-3 and the reference Monte Carlo solutions.

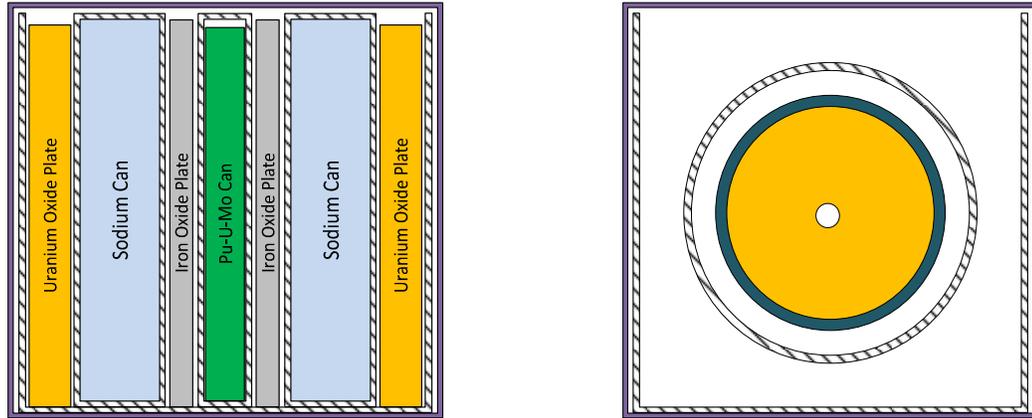
### 4.1 Development of 2D MOC Solver

#### Geometry Modeling

The primary application target of the 2D transport calculations of the MC<sup>2</sup>-3 code is the unit cell calculations for plate-type critical assemblies, illustrated in Figure 16. In the geometry treatment module, a problem domain consists of multiple cells that have rectangular shapes of arbitrary dimensions. Each cell is further divided into rectangular or annular meshes. A cell composed of annular meshes can be used for modeling a cylindrical fuel rod, and therefore, the Doppler drawer of the ZPPR-15 core shown in Figure 16(b) and the CADENZA assembly of the ZEBRA critical assembly [34] can be modeled explicitly. This type of cell composed of annular meshes could be used for lattice calculations of light water reactors (LWRs), when the slowing-down calculation capability is extended to the thermal energy region as planned. Meshes with the same composition are grouped together into a region, which is the most fundamental geometric unit of the cross section generation in the MC<sup>2</sup>-3 code, and the meshes belonging to the same region share the same UFG or HFG cross sections in a transport calculation.

#### Ray Tracing and P<sub>1</sub> Anisotropic Scattering Treatment

For a given set of UFG or HFG cross sections, a transport calculation is performed using a MOC solver with the modular ray tracing technique. The cyclic ray tracking is applied for the modular ray tracing calculation, which allows a simple treatment of reflective boundary conditions without approximation. In the ray tracing calculation, the MOC equation is solved for each ray segment that passes through a mesh over which the source is assumed flat as



(a) Plate-type (ZPR-6 Assembly 7 Drawer) (b) Fuel Rod-type (ZPPR-15 Doppler Drawer)

Figure 16. Illustrations of 2D Unit Cell Models of MC<sup>2</sup>-3

$$\varphi_{out} = \varphi_{in} e^{-\tau} + \frac{q}{\Sigma} (1 - e^{-\tau}), \quad (4.1)$$

where  $\varphi_{out}$  is the outgoing angular flux,  $\varphi_{in}$  is the incoming angular flux,  $q$  is the source,  $\Sigma$  is the macroscopic total cross section, and  $\tau$  is the optical path length of the segment. The segment-averaged value of the angular flux  $\bar{\varphi}$  is determined as

$$\bar{\varphi} = \frac{\varphi_{in} - \varphi_{out}}{\tau} + \frac{q}{\Sigma}. \quad (4.2)$$

The corresponding scalar flux and angular flux moments of the mesh are computed by summing the segment-averaged angular fluxes over the space and angles with appropriate weighting functions.

In order to reduce the computational time of 2D MOC calculations, a parallel computation capability has been implemented. Since a problem size of a unit cell calculation is small relative to that of a whole core transport calculation, a massive parallelization in the high-performance computing environment by employing the spatial domain decomposition is not applicable. Alternatively, the ray tracing angles were decomposed for the parallelized calculation and the multi-threading calculation capability for the shared memory machines was implemented in the OpenMP framework.

To account for anisotropic scattering, an anisotropic scattering source is incorporated in the 2D MOC solver. The anisotropic scattering source is evaluated from the angular flux moments that are determined from the angular flux solution of the previous inner iteration, and it is explicitly incorporated into the within-group transport sweeping. At the moment, the P<sub>1</sub> anisotropic scattering is the default option of the 2D MOC solver, and the isotropic scattering with the transport correction can be used optionally. The anisotropic scattering will be extended to a P<sub>3</sub> order later.

### Low-density Region Treatment

In principle, the 2D MOC solver that relies upon the first order transport equation can handle low-density regions, i.e. a region filled with air, without any stability issues. However, for an extremely small optical length, the segment-averaged angular flux in Eq. (4.2) is computed to be zero or almost zero due to the truncation error of floating point operations, whereas the MOC solution in Eq. (4.1) can be computed properly. Consequently, the scalar fluxes of low-density regions are obtained to be extremely small values, even though the overall transport calculation can be performed properly. It causes a problem for calculating the homogenized cross sections, which are used in the subsequent whole core analysis. For example, in the unit cell calculation of ZPR-6 Assembly 7 fuel drawer, the volume filled with air is roughly 10% of the problem domain and the homogenized drawer cross sections were significantly over-estimated.

The issue of the low-density regions was remedied by calculating the segment-averaged angular flux with a simple average of the incoming and outgoing angular fluxes as

$$\bar{\varphi} = \frac{\varphi_{in} + \varphi_{out}}{2}. \quad (4.3)$$

Due to the negligible flux attenuation in a low-density region, this approximation does not introduce any noticeable error to the eigenvalue or the global flux distribution.

## **4.2 Integration of 2D MOC Solver into MC<sup>2</sup>-3 Code**

### Ultrafine Group Transport Calculation

In the multi-dimensional transport calculation of MC<sup>2</sup>-3, the heterogeneity effect on the self-shielded UFG cross sections should be taken into account properly by using equivalence theory. In this regard, the UFG cross sections for the existing 1D CPM solver are generated by the use of isotopic escape cross sections, which are determined using the Tone's method. [15,35] The Tone's method is formulated based on the collision probability method, but the collision probability method is impractical to deploy for the 2D and 3D problems due to the complexity of computing collision probabilities for individual regions. In order to implement a 2D UFG transport calculation capability within the existing framework of the UFG cross section generation, the isotopic escape background cross section are calculated in an alternative way by solving the fixed source transport problems, which are equivalent with the Tone's Method, using the 2D MOC solver [36]. In this formulation, the following two fixed-source transport problems are solved:

$$\boldsymbol{\Omega} \cdot \nabla \psi_{1,r}^g(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma_i^g(\mathbf{r}) \psi_{1,r}^g(\mathbf{r}, \boldsymbol{\Omega}) = \sum_{k \neq r} N_k(\mathbf{r}) \sigma_{t,k}^g(\mathbf{r}), \quad (4.4)$$

$$\boldsymbol{\Omega} \cdot \nabla \psi_{2,r}^g(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma_i^g(\mathbf{r}) \psi_{2,r}^g(\mathbf{r}, \boldsymbol{\Omega}) = N_r(\mathbf{r}). \quad (4.5)$$

Using the solutions of Eq. (4.4) and Eq. (4.5), the escape cross section to be used for determining the self-shielded group g cross section of resonant isotope r in a region i can be computed as

$$\Sigma_{e,r,i}^g = N_{r,i} \frac{\int_{V_i} dV \int_{4\pi} d\Omega \psi_{1,r}^g(\mathbf{r}, \boldsymbol{\Omega})}{\int_{V_i} dV \int_{4\pi} d\Omega \psi_{2,r}^g(\mathbf{r}, \boldsymbol{\Omega})} - \sum_{k \neq r} N_k \sigma_{t,k,i}^g \quad (4.6)$$

This method for evaluating the escape cross section using the 2D MOC solver was implemented in the MC<sup>2</sup>-3 code. For each resonance group, the escape cross sections can be obtained by performing 2*N* fixed source transport calculations, where *N* is the number of resonant isotopes in the entire domain. Then, the UFG cross sections are determined by self-shielding the pointwise cross sections using the NR approximation and the calculated escape cross sections.

The slowing-down source iteration, which is to update the fission source distribution and the eigenvalue, was also implemented in the 2D transport UFG solver. If the UFG cross sections for the individual regions are prepared, the slowing down calculation for the 2D domain is performed with the source iterations. In addition, the interface routines that convert the 2D MOC solutions to the MC<sup>2</sup>-3 data structure formats were implemented so that the existing routines for processing the cross sections, i.e. homogenization and group condensation, can be readily utilized.

### Hyperfine Group Transport Calculation

The HFG calculation poses a huge computational burden not only in the transport calculation for the large number of groups but also in the explicit computation of scattering and fission sources for the HFG level. Therefore, in the current 1D transport capability of the MC<sup>2</sup>-3 code, the HFG slowing-down calculation is performed by extrapolating the fission, inelastic scattering, and (n,2n) source terms of the UFG calculation to the HFG structure, while calculating the elastic scattering source explicitly. [15] The same approach was used for 2D HFG transport calculations as illustrated in Figure 17.

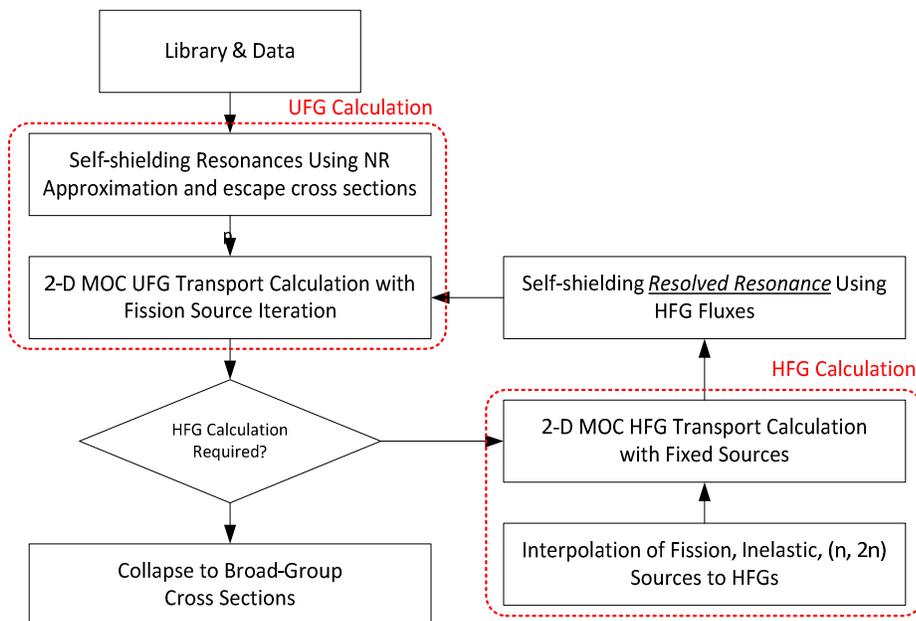


Figure 17. 2D HFG Transport Calculation Scheme

In this approach, the 2D UFG eigenvalue problem is solved using the self-shielded UFG cross sections based upon the narrow resonance (NR) approximation and the escape cross sections. By interpolating the UFG fission and non-elastic scattering sources of the converged UFG solution, the fixed source terms of HFG calculation are determined. With these fixed source terms, the 2D HFG transport calculation is performed by explicitly evaluating the elastic scattering sources. Once the HFG calculation is finished, the self-shielded cross sections in the resolved resonance energy are determined again by making use of the HFG fluxes. Because the cross section in the unresolved resonance is self-shielded on the UFG basis using the NR approximation and equivalence theory in MC<sup>2</sup>-3, the self-shielded cross sections in the unresolved resonance are not updated during the iterative calculation of HFG and UFG problems. Since the fixed source problem does not require the source iteration, the computational time can be much shorter than that of the HFG eigenvalue calculation. The iteration between ultrafine and hyperfine group calculation can be repeated until the UFG solution is converged.

### 4.3 Verification Tests of 2D Transport Capability

#### 1D Slab Model Problem of ZPR-6 Assembly 7 Fuel Drawer

For the initial verification of the newly implemented 2D transport capability, a 1D slab problem was solved using the 2D transport solver and the obtained solution was compared with the solutions of the current 1D CPM solver of MC<sup>2</sup>-3 and of the MCNP-6 code. The configuration of the 1D slab problem was derived from the ZPR-6 Assembly 7 fuel drawer [37] shown in Figure 16(a) by taking a slice along the x-direction at the vertical center of the drawer.

The eigenvalue results for the 1D slab model of ZPR-6 Assembly 7 fuel drawers are compared in Table 6. The eigenvalues obtained with the 1D CPM and 2D MOC UFG calculations are 106 pcm and 29 pcm off, respectively, from the MCNP-6 reference solution. With the HFG calculations, the eigenvalue error is reduced about 10 pcm for both of the 1D CPM and 2D MOC calculations.

Figure 18 shows the relative difference in the homogenized spectrum between the 2D MOC and 1D CPM calculations. A good agreement is observed for most of the energy region, but noticeable differences, roughly 1 %, are observed near 2.85 keV where the large resonance of Na-23 is located. The primary factor that makes the observed spectrum difference is the different anisotropic scattering treatment.

Table 6. k-inf Comparison of 1D CPM and 2D MOC Solvers of MC<sup>2</sup>-3 for 1D Slab Model Problem of ZPR-6 Assembly 7 Fuel Drawer (Sliced).

Calculation Type	MCNP-6 k-inf (std)	MC <sup>2</sup> -3 k-inf (Del_k, pcm)	
		CPM	MOC
Continuous	1.35135 (2)	-	-
UFG	-	1.35243 (106)	1.35164 (29)
HFG		1.35232 (97)	1.35155 (20)

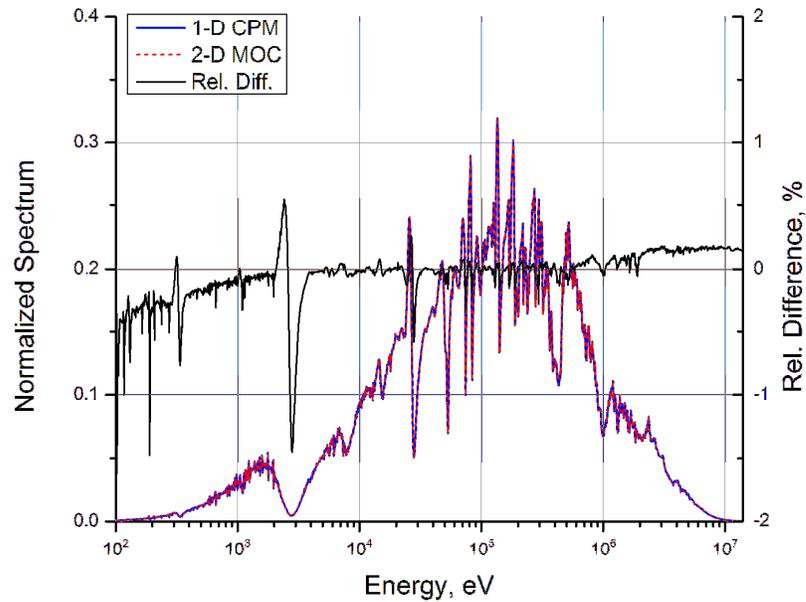


Figure 18. Spectrum Comparison of 1D CPM and 2D MOC UFG Solutions for 1D Slab Model Problem of ZPR-6 Assembly 7 Fuel Drawer (Sliced)

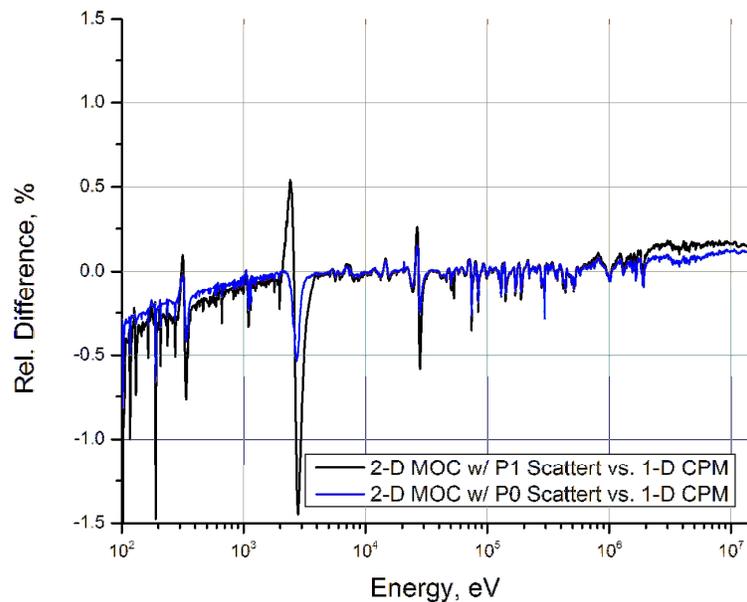


Figure 19. Effect of  $P_1$  and  $P_0$  Scattering on Spectrum for 1D Slab Model Problem of ZPR-6 Assembly 7 Fuel Drawer (Sliced)

The  $P_1$  anisotropic scattering was explicitly used in the 2D MOC calculation while the 1D CPM method uses  $P_0$  scattering. To examine the effect of the  $P_1$  anisotropic scattering, the 1D MOC calculation with  $P_0$  scattering was additionally performed. The spectrum difference of the resulting spectrum from the 1D CPM solution is given in Figure 19, which shows a significantly reduced difference at the Na-23 resonance. The eigenvalue of the 1D MOC calculation with  $P_0$  scattering is 1.35208, 73 pcm off from the MCNP-6 result, and the eigenvalue difference between 2D MOC and 1D CPM decreased from 79 pcm to 35 pcm. It is noted that the UFG calculation with the explicit  $P_1$  anisotropic scattering results in a better agreement with the reference MCNP-6 solution. Therefore, anisotropic scattering should be explicitly incorporated in unit cell calculations of plate-type critical assemblies.

2D Unit Cell Problem of ZPR-6 Assembly 7 Fuel Drawer

As the second step of the verification of the 2D capability, the 2D unit cell problem of ZPR-6 Assembly 7 fuel drawer was calculated with the 2D MOC solver of MC<sup>2</sup>-3 and MCNP-6. The 2D drawer configuration depicted in Figure 20(a) was explicitly modeled in the MC<sup>2</sup>-3 and MCNP-6 calculations. Along with the 2D calculations, the 0D mixture and 1D transport calculations were also performed with both MC<sup>2</sup>-3 and MCNP-6 to examine the 2D transport effect. The 1D unit cell model was derived from the 2D model by homogenizing each region along the y-direction. As a result, the region-wise composition of this 1D model is different with that of the 1D slice model above. The MC<sup>2</sup>-3 calculations for the 2D and 1D models were performed by the 2D MOC solver with the explicit  $P_1$  anisotropic scattering.

The obtained results for 2D, 1D and 0D models with the MC<sup>2</sup>-3 and MCNP-6 runs are summarized in Table 7. The eigenvalue results of the MC<sup>2</sup>-3 calculations for the 2D and 1D unit cell models agree well with the MCNP-6 results within 40 pcm for three cases. The eigenvalue errors of the 2D and 1D calculations are reduced by about 30 pcm compared to the 0D results. This suggests that some sort of error cancelation is involved in the 1D and 2D MOC transport calculations. For the ZPR-6 Assembly 7 fuel drawer, about 300 pcm of 2D transport effect in eigenvalue is observed when the 1D model is replaced by the 2D model that can represent the local heterogeneity more accurately. The detailed investigation of the 2D transport effect in the spectrum and broad-group cross sections is underway.

Table 7. 2D Transport Effect in k-inf for 2D Unit Cell Calculation of ZPR-6 Assembly 7 Fuel Drawer

Problem Type	k-inf			2D Effect, pcm	
	MCNP-6 (std)	MC <sup>2</sup> -3	Del_k, pcm	MCNP-6	MC <sup>2</sup> -3
2D	1.30087 (2)	1.30134	47	-	-
1D	1.29798 (2)	1.29842	44	289	292
0D	1.28358 (2)	1.28434	76	1729	1700

2D Unit Cell Model

The fuel drawer of ZRR-6/7, as illustrated in Figure 20, is formed of a square steel tube and a steel drawer containing rectangular plates of different materials such as uranium, plutonium, stainless steels, sodium, etc. Even though the actual drawer configuration has a three-dimensional (3D) nature as shown in Figure 20, the faithful 2D model can be obtained by taking a slice along the radial direction at the axial center of the 3D drawer model because the heterogeneities in the Z-direction are negligible. A preparation of the input deck for a 2D MC<sup>2</sup>-3 calculation requires extensive effort due to the complexity of the plate dimensions and compositions of the fuel drawer configuration. A shell script that can process a 3D as-built MCNP-6 model into a 2D model was developed and the 2D MC<sup>2</sup>-3 input deck for ZPR-6/7 fuel drawer was generated by making use of this shell script.

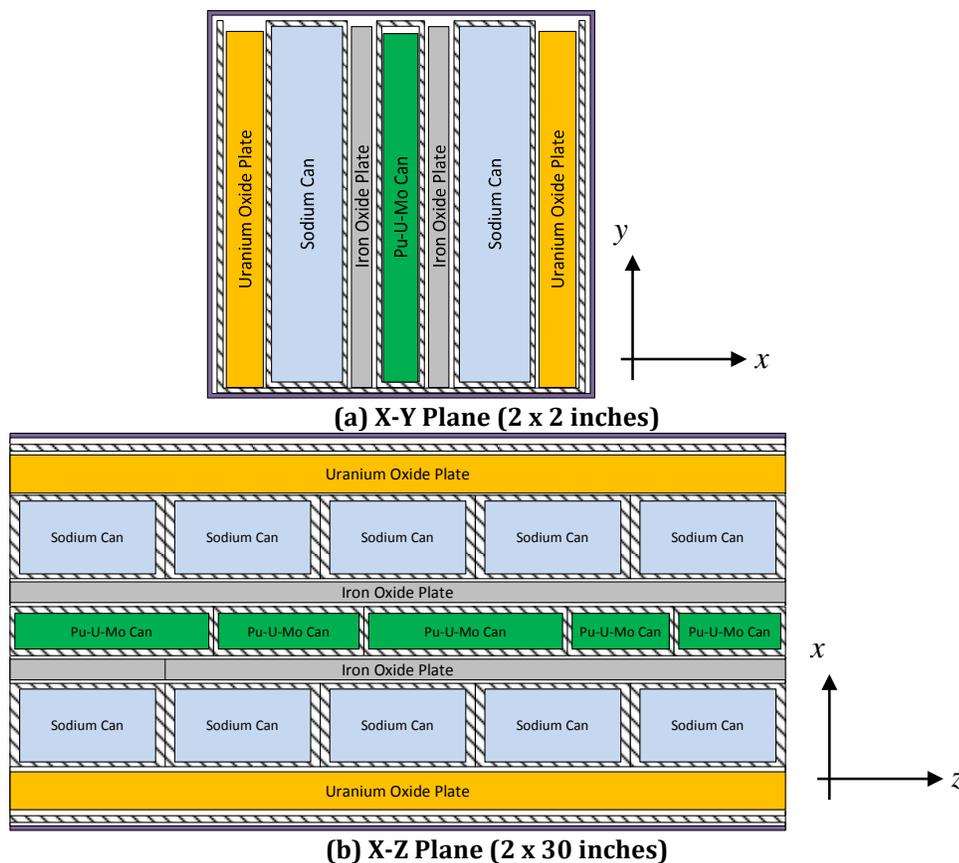


Figure 20. Configuration of ZPR-6/7 Fuel Drawer

1D Unit Cell Model

In a simple 1D model that is derived from the 2D model by homogenizing each region along the y-direction, the background cross sections of plates are modified inevitably, which leads to a substantial difference in the resulting shielded cross sections. Conventionally, a 1D model is prepared in such a way that the mass and the background cross section of fuel and sodium plates are preserved. The tube structure, plate cover and void region at the bottom of or above those

plates are smeared into similar materials at the periphery. The variation in the plate heights is removed by making all the plate heights equal to the largest one and subsequently adjusting the nuclide densities to preserve the material mass in each plate. The adjusted nuclide densities alter the optical thickness of each plate and hence perturb the escape cross section and the flux distribution. Along with the conventional model, another 1D model was derived in this study by taking a slice along the x-direction at the vertical center of the drawer. This model, which is denoted here as the 1D slice model, can retain the background cross section and the optical thickness of each plate, but it cannot preserve the material masses in the drawer.

2D Transport Effect of ZPR-6/7 Fuel Drawer

In order to investigate the 2D transport effect, MC<sup>2</sup>-3 unit cell calculations were performed for the following four unit cell models of the ZPR-6/7 fuel drawer: 2D explicit, 1D conventional, 1D slice and 0D models. The detailed descriptions of these models are provided in the previous section. The 2D MOC solver was used in the 1D problems as well as the 2D problem to incorporate the P-1 anisotropic scattering and to have consistent comparisons with 2D results. The reference calculations for those models were performed using the MCNP-6 code.

The eigenvalue results of the unit cell calculations are summarized in Table 8. The eigenvalues obtained with the 1D and 2D unit cell calculations are less than 20 pcm off from the reference MCNP-6 solutions. A local heterogeneity effect of ~1700 pcm, which is the eigenvalue difference of 0D and 2D results, is observed in the ZPR-6/7 fuel drawer. Compared to the 1D conventional model, the 2D model improves the local heterogeneity effects only by 25 pcm and 60 pcm in the MC<sup>2</sup>-3 and MCNP-6 results, respectively. In a previous study [38], a 2D transport effect of 200 pcm was obtained using a simple 1D model, which was obtained by homogenizing each plate region along the y-direction. These results suggest that the 1D conventional model can represent the local heterogeneity effects more accurately than the other 1D models since the background cross sections of plates are preserved.

Table 8. Eigenvalue Comparisons of Different Unit Cell Models of ZPR-6/7 Fuel Drawers

Model	MCNP-6	MC <sup>2</sup> -3 (Δk, pcm)	2D Effect, pcm	
			MCNP-6	MC <sup>2</sup> -3
2D	1.30087	1.30103 (16)	-	-
1D (Conventional)	1.30147	1.30128 (-19)	60	25
1D Slice*	1.35135	1.35130 (-5)	-	-
0D	1.28358	1.28405 (47)	1729	1698

\*1D slice model does not preserve the material masses of the reference 2D model

The 2D transport effect on the broad group cross section is more important than that on the unit cell eigenvalue since the broad group cross sections are used in the subsequent whole core calculations. The homogenized broad-group cross sections were calculated in the ANL 230 group structure using the 1D and 2D models. The reference broad group cross sections of the 2D explicit drawer model were obtained using the MCNP-6 code. The results of MC<sup>2</sup>-3 for the 2D

unit cell calculation were compared with the reference MCNP-6 results to verify the accuracy of 2D transport capability. The 2D transport effect on the broad group cross sections was quantified by comparing the broad group cross sections of the 2D and 1D models. Note that the broad group cross sections obtained by 1D MC<sup>2</sup>-3 calculations were compared with the reference 2D MCNP-6 results instead of the 2D MC<sup>2</sup>-3 results.

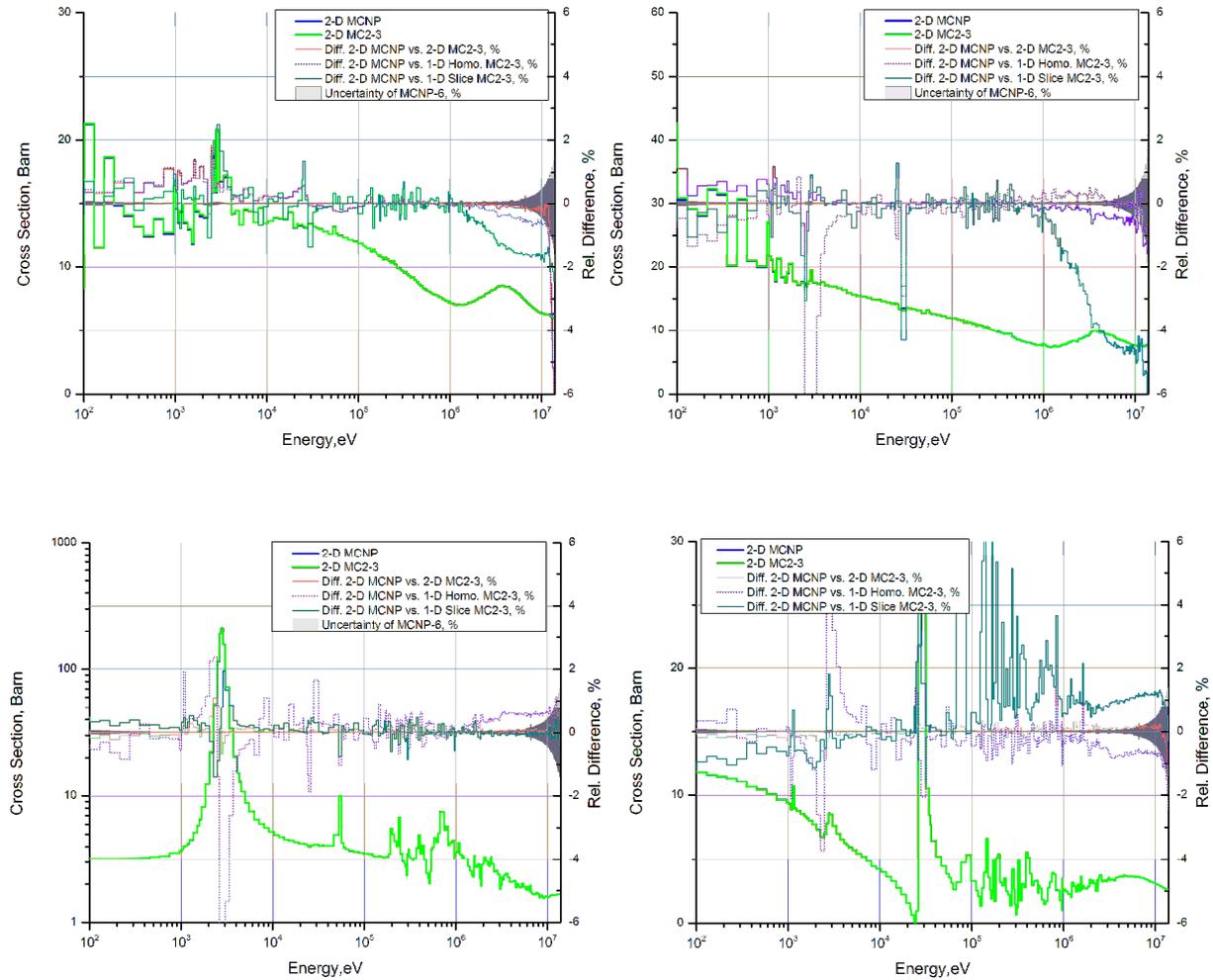


Figure 21. Comparisons of Fe-56 Total Cross Section for Different Unit Cell Models

Figure 21 shows the comparisons of the total cross sections for U-238, Pu-239, Na-23 and Fe-56. The comparisons of other principal cross sections for those isotopes are provided in Appendix A. The broad group cross sections of MC<sup>2</sup>-3 obtained with the 2D model generally agree well with the reference 2D MCNP-6 results within 1% error. However, as shown in Figure 21, noticeable differences (~2%) are observed in the total cross section of Pu-239 around 2.5 keV and 30 keV. Since these two energy points are the lower and upper bound of the unresolved resonance (URR) range of Pu-239 and the same differences are observed in the broad group cross sections of the 0D infinite medium problem, the primary cause for the observed differences appears to be the cross section discontinuities at the boundaries of the URR range. In order to identify the root

cause for this problem, the cross section library and unresolved resonance self-shielding of MC<sup>2</sup>-3 and the probability table of MCNP-6 need to be investigated.

The broad-group cross sections obtained with the 1D slice model show significant discrepancies above 1 MeV. Because the capture cross section at high energy is less than a few barns for most of the isotopes, the structure materials neglected in the slice model can affect the neutron transport in this energy range. Consequently, the 1D conventional model that preserves the material masses in the fuel drawer shows a better agreement in this energy range with the explicit model. In the energy range from 10 keV to 1 MeV, where the most of the neutrons are populated in fast reactors, the broad group cross sections of U-238, Pu-239 and Na-23 obtained with both 1D models show a good agreement within 1 % difference with the reference 2D results. The cross sections of the 1D conventional model show slightly better agreements in this energy range. The 1D slice model result of Fe-56 has a large difference in the broad group cross sections since a substantial amount of iron in the structure materials were excluded in the 1D slice model. In the results of the 1D conventional model, a noticeably large difference is observed near 2.85 keV. In this model, the plate dimensions are modified to preserve the material masses, and hence the optical thicknesses are perturbed. Since a large Na-23, scattering resonance is located at 2.85 keV, a small change in optical length that perturbs the flux distribution results in large differences in the broad group cross sections.

#### **4.4 Summary**

In order to improve the modeling capabilities of the multi-group cross generation code MC<sup>2</sup>-3 for fast reactor applications, a 2D method of characteristics solver based upon a modular ray tracing technique was successfully developed and implemented. In the MC<sup>2</sup>-3 code, the 2D transport capability is now available for the hyperfine and ultrafine group slowing-down calculations with P<sub>1</sub> anisotropic scattering sources and the local heterogeneity effect can be represent more accurately by taking into account the 2D nature of the unit cell explicitly. Preliminary verification test results for the unit cell problems of ZPR-6 Assembly 7 fuel drawer showed that the 2D transport capability is properly incorporated within the MC<sup>2</sup>-3 framework.

Using the newly implemented 2D MOC calculation capability, the 2D transport effects on the unit cell cross sections of plate-type fast critical assemblies were investigated using the ZPR-6/7 fuel drawer. MC<sup>2</sup>-3 calculations were performed for the 2D explicit, 1D conventional and 1D slice models derived from an as-built model of ZPR-6/7 fuel drawer. The effects of 2D transport calculations on the eigenvalue and broad group cross sections were investigated by comparing the MC<sup>2</sup>-3 solutions with the reference MCNP-6 solutions. The results show that the 2D transport calculations of MC<sup>2</sup>-3 can account for the local heterogeneity effect properly. It also turns out that the 1D models have the accuracy limitation in representing the local heterogeneity.

## 5. Verification and Validation of PROTEUS

### 5.1 ABTR

Last year, the initial verification of the generation of heterogeneous cross sections in MC<sup>2</sup>-3 and their use in heterogeneous PROTEUS transport calculations were performed using the ABTR core shown in Figure 22, creating pin cell, assembly, mini-core, and 2D full core geometries with varying spatial details. The comparison results for all cases up to 2D cores showed very good agreement in eigenvalue between PROTEUS and MCNP. This year, the verification tests were extended to 3D full cores so that the radial thermal expansion phenomenon of SFR could be simulated.

#### 5.1.1 Multigroup Cross Section Generation

Multigroup cross sections are prepared for three distinct calculations for the 3D ABTR core: (1) homogeneous assembly model, (2) partially homogeneous assembly model, and (3) partially homogeneous assembly model with pin heterogeneity effects. For each of the calculations, two configurations are considered: control rods fully inserted into core (“rods in”), and control rods fully withdrawn (“rods out”).

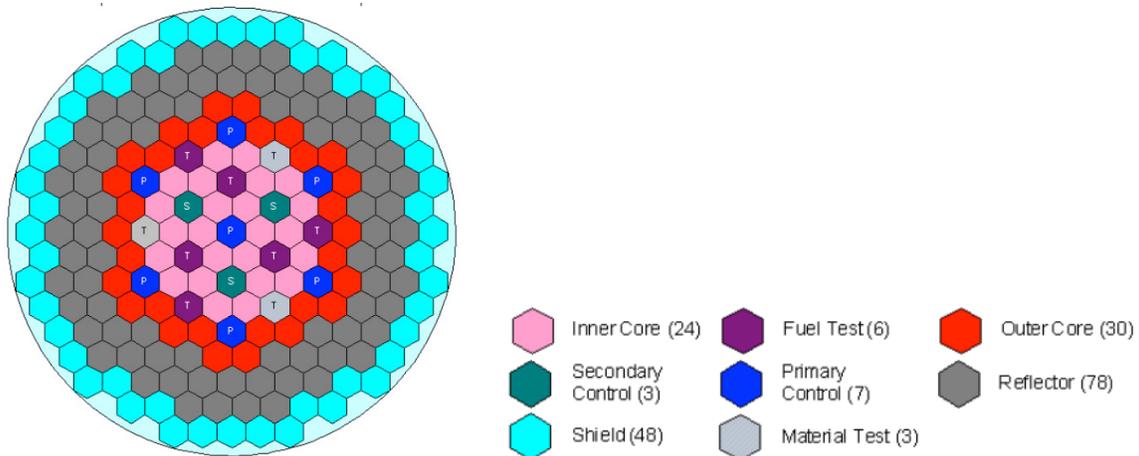


Figure 22. ABTR Core Loading

#### Homogeneous Model

A conventional homogenized assembly model was first analyzed to verify MC<sup>2</sup>-3/PROTEUS against continuous energy MCNP. Each individual assembly (fuel, control, shield and reflector) is spatially homogenized (cladding, sodium, and duct material mixed with the respective pin material). Axial heterogeneity is introduced at the natural material boundaries. The cross sections are prepared as follows:

- 1) MC<sup>2</sup>-3 with homogeneous models generates 1041 group cross sections for each homogenized assembly type (fuel, control, shield, and reflector).

- 2) TWODANT with the R-Z equivalent core model calculates 1041 group spectra for each region. The spectra are used in MC<sup>2</sup>-3 to collapse the 1041 group cross sections to the desired broad group structures.

### Partially Homogeneous Model

In the partially homogeneous model, the materials inside each assembly duct are homogenized, but the assembly duct and inter-assembly sodium channels are explicitly represented in order to facilitate multi-physics applications (structural mechanical feedback). We note that the MC<sup>2</sup>-3 calculations were limited to 1D geometries such that the hexagonal fuel and control assembly geometry are approximated by a cylinder that conserves the total volume of each of the explicit regions. For the shield and reflector assembly cross sections, the same procedures as the homogeneous model were taken as the heterogeneity effect is not significant. However, the cross sections are prepared as follows:

- 1) MC<sup>2</sup>-3 with the 1D models representing homogenized assembly interior, duct(s), and sodium channel(s) generates 1041 group cross sections. For the control rod cross sections, the control assembly is surrounded by an additional ring of fuel assemblies (supercell configuration).
- 2) TWODANT with the 1D models calculates 1041 group spectra for each region. The spectra are used in MC<sup>2</sup>-3 to collapse the 1041 group cross sections to the desired broad group structures.

In this cross section generation process, the fuel and control assemblies did not account for the global flux spectrum properly, which could introduce some error into the partially homogeneous fuel and control assembly cross sections.

### Partially Homogeneous Model with Pin Heterogeneity Effects

The partially homogeneous PROTEUS-SN model is also compared to a fully heterogeneous MCNP model by introducing a more complex cross section generation procedure that accounts for the pin heterogeneity effects. We note that the only difference between this calculation and the “partially homogeneous” calculation is the cross section generation procedure. It is desirable, although difficult, to reproduce the fully heterogeneous result with a less spatially detailed calculation by accounting for the heterogeneity effects in the cross section generation procedure.

Most of the cross sections in this model are identical to those used in the regular partially homogeneous model. Only the fuel and control absorber cross sections are generated by a slightly different second step:

- 1) MC<sup>2</sup>-3 with the 1D models (The same as the partially homogeneous model)
- 2) TWODANT with the 1D cylinder “assembly” model calculates 1041 group spectrum for each region, which is used in MC<sup>2</sup>-3 1D heterogeneous calculation to collapse 1041 group cross sections to broad group structures

The 1D heterogeneous calculation in MC<sup>2</sup>-3 takes the 1041 group spectrum from the TWODANT 1D S<sub>N</sub> calculation and then determines a spatial self-shielding correction to the spectrum for each region. MC<sup>2</sup>-3 then homogenizes and condenses all of the fuel, clad, and

coolant regions into a single macroscopic cross section for the region inside of the explicit duct in the partially homogeneous model.

Figure 23 depicts the geometry models (only fuel assembly illustrated). Table 9 lists how each calculation was performed in the multigroup generation step in MC<sup>2</sup>-3, PROTEUS-SN and what type of MCNP calculation was used as a reference solution. The ENDF/B-VII.0 data was used in both MC<sup>2</sup>-3 and MCNP. A flat temperature distribution of 300K was used for simplicity. Since MC<sup>2</sup>-3 does not have 2D capabilities, the hexagonal assemblies were approximated as cylinders for the 1D calculations.

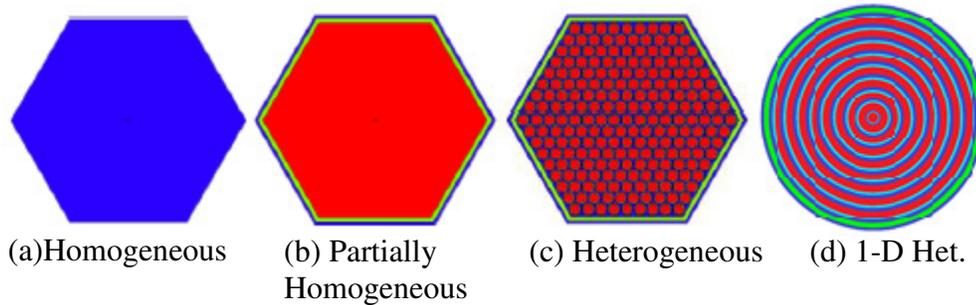


Figure 23. (a)-(c) Spatial models used to represent fuel assembly, (d) 1D equivalent model to the heterogeneous model in (c). Other assembly types are not shown (different number of ducts/pins)

Table 9. Summary of Calculation Types and Spatial Representations in Codes

Calculation	MC <sup>2</sup> -3	PROTEUS	MCNP
Homogeneous	a	a	a
Partially Homogeneous	b*	b	b
Partially Homogeneous with Pin-Heterogeneity Effects	d	b	c

\* 1D cylindrical approximation of hex assembly

### 5.1.2 Verification Tests

PROTEUS-SN calculations for the homogeneous and partially homogeneous 3D ABTR models were compared to MCNP continuous energy calculations with the same geometry. The MCNP calculations used 50,000 particles per cycle (100 inactive and 500 active cycles) and were converged within 10 pcm uncertainty. Quadratic finite elements were used in the meshes. A typical problem used 4 million spatial vertices, 64 angles, and 33 energy groups for a problem size on the order of 8 billion degrees of freedom. The eigenvalues for the control “rods out” and “rods in” configurations are stated in Table 10 and Table 11, respectively.

The homogeneous assembly results agree with MCNP within 52 pcm for both cases. Since the solution does not improve much with increasing number of energy groups (it actually becomes slightly larger in error), we conclude that the core spectrum calculation in the cross section generation step is adequate even for a small number of groups (33 groups).

For the partially homogeneous case, refinement in energy increases the accuracy of PROTEUS-SN significantly. This is expected due to increased spatial heterogeneity in the

problem. Additionally, the global flux spectrum is not accounted for in the fuel and control assembly cross sections because MC<sup>2</sup>-3 cannot currently combine local and global spectral effects simultaneously.

Table 10. PROTEUS-SN eigenvalue results, control rods out (L7T7 cubature, P1 scattering).

Rods Out	MCNP	PROTEUS 33 groups	PROTEUS 70 groups	PROTEUS 116 groups
Homogeneous	1.02634 ±0.00011	1.02567 (-67)	1.02661 (27)	1.02686 (52)
Partially Homogeneous	1.03013 ±0.00011	1.02219 (-794)	1.02587 (-426)	1.02829 (-184)
Partially Homogeneous w/pin heterogeneity effects	1.03105 ±0.00011	1.02835 (-270)	1.03181 (76)	1.03235 (130)

\*Quantity in ( ) is pcm difference from MCNP reference solution

Table 11. PROTEUS-SN eigenvalue results, control rods in (L7T7 cubature, P1 scattering).

Rods In	MCNP	PROTEUS 33 groups	PROTEUS 70 groups	PROTEUS 116 groups
Homogeneous	0.90025 ±0.00009	0.90021 (-4)	0.90067 (42)	0.90071 (46)
Partially Homogeneous	0.90829 ±0.00010	0.90437 (-392)	0.90636 (-193)	0.90747 (-82)
Partially Homogeneous w/pin heterogeneity effects	0.91044 ±0.00010	0.90793 (-251)	0.91145 (101)	0.91170 (126)

\* Quantity in ( ) is difference in pcm from MCNP reference solution

The total control rod worth calculated in MCNP for each of the three cases is, in order: 12,609 pcm, 12,184 pcm, and 12,061 pcm. In PROTEUS, 116 groups, these values are 12,615 pcm, 12,082 pcm, and 12,065 pcm. For the homogeneous assemblies and the partially homogeneous assemblies with heterogeneity effects, the difference in control rod worth between MCNP and PROTEUS is less than 10 pcm, which is within uncertainty. This is very encouraging; however, for the regular partially homogeneous case, the difference is 102 pcm. It is not clear why this large discrepancy exists: it may be related to the inconsistency in the cross section generation (i.e. lack of global spectrum for condensing the fuel/control cross sections) but this same inconsistency exists when the heterogeneity effects are added, while the difference between PROTEUS and MCNP virtually disappears.

The heterogeneity effects calculated in MCNP are given in Table 12, and those calculated in PROTEUS-SN in Table 13. While the separate duct and pin effects calculated in PROTEUS-SN seem to have little or no relation to those calculated in MCNP, the combined duct and pin heterogeneity effect is actually quite comparable between the two codes. This suggests that the MC<sup>2</sup>-3 cross section condensation is capturing most of the significant reactivity effects of the heterogeneous configuration, which is an important step towards the ultimate goal of accurate transient simulation with homogenized cross sections.

The duct heterogeneity effect calculated in MCNP (Table 12) is much more significant than the pin heterogeneity effect. In MC<sup>2</sup>-3 this is also observed with the control rods inserted, but

with the control rods out the opposite is true. The MC<sup>2</sup>-3 heterogeneity effect results are in Table 14. Except for the rods-out duct effect, the MC<sup>2</sup>-3 heterogeneity effects are significantly larger than what is calculated in MCNP. The MC<sup>2</sup>-3 models are characteristically different than the PROTEUS and MCNP models in that they are small, reflective boundary models, so there is no leakage. The full core models have leakage, which significantly alters the flux shape and spectrum. The reflected assemblies have an average k-effective of approximately 1.63, but the full core k-effective with control rods withdrawn is much lower, approximately 1.03. Thus, leakage is clearly a very significant contributor to reactivity effects. This may explain the large difference in predicted heterogeneity effects between the idealized reflective boundary models and the full 3D core model.

Table 12. Duct and Pin heterogeneity Effect Calculated from MCNP

Conf.	Homogeneous	Partially Homogeneous	Duct Het. Effect (pcm)	Heterogeneous	Pin Het. Effect (pcm)	Total Effect (pcm)
Rods in	0.90025	0.90829	804	0.91044	215	1019
Rods out	1.02634	1.03013	379	1.03105	92	471

\* Standard deviation ≤ 19 pcm

Table 13. Heterogeneity Effect Calculated from PROTEUS-SN (116 groups, L7T7, P1 scattering)

Conf.	Homogeneous	Partially Homogeneous	Duct Het. Effect (pcm)	Heterogeneous	Pin Het. Effect (pcm)	Total Effect (pcm)
Rods in	0.90071	0.90747	676	0.91170	423	1099
Rods out	1.02686	1.02829	143	1.03235	406	549

Table 14. Assembly Heterogeneity Effect Calculated from MC<sup>2</sup>-3

Region	Homogeneous	Partially Homogeneous	Duct Het. Effect (pcm)	Heterogeneous	Pin Het. Effect (pcm)
Inner Fuel	1.56140	1.56334	194	1.56615	281
Middle Fuel	1.39963	1.40150	187	1.40500	350
Outer Fuel	1.72706	1.72878	172	1.73128	250
OF Supercell	1.44162	1.45335	1173	1.45945	610
IF Supercell	1.28509	1.29681	1172	1.30287	606

Figure 24 shows a sample of the flux spectrum for 4 of the 33 energy groups at a height of 140 cm, which is approximately the active core mid-plane. The units are arbitrary; the flux is directly related to the wavelength of the color (i.e. blue is the lowest, red is the highest). For the most part, the fastest neutrons (group 5) do not make it past the reflector region without slowing down. The flux in the reflector is much higher for the lower energy groups (14 and 19). In group 19, the flux is higher in the reflector region than in the fuel. The flux in the fuel is only smaller for energy groups lower than 19, so the lower half of energy groups are much less important than the upper half for this core.

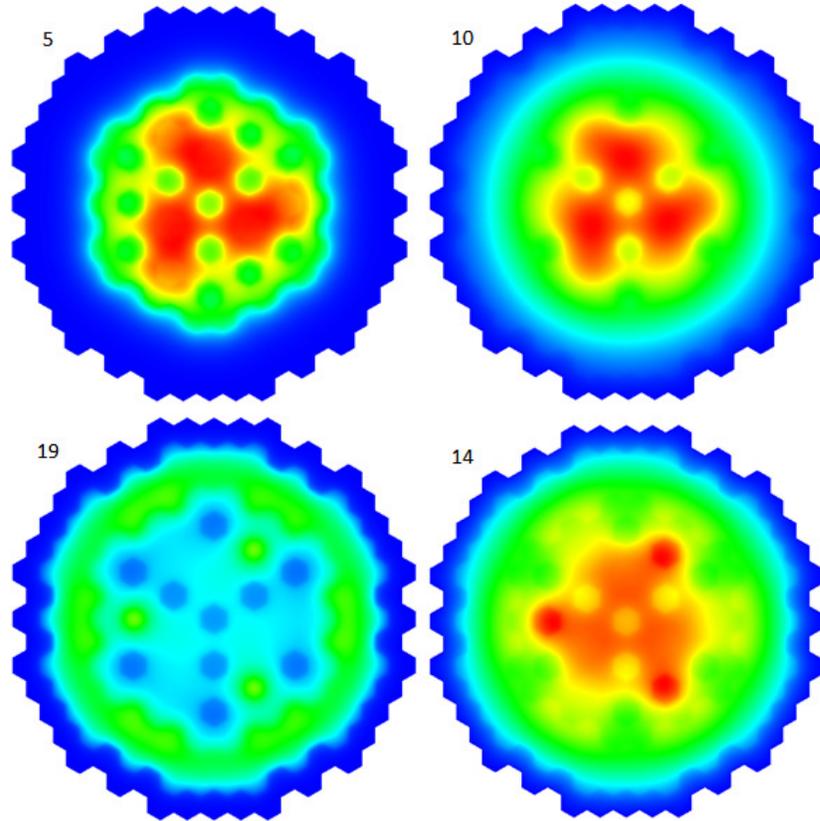


Figure 24. Scalar flux (Clockwise from top left: Group 5, 10, 14, 19 out of 33), homogeneous assemblies, 33 groups, L7T7, P1 scattering, control rods in (slice at 140 cm)

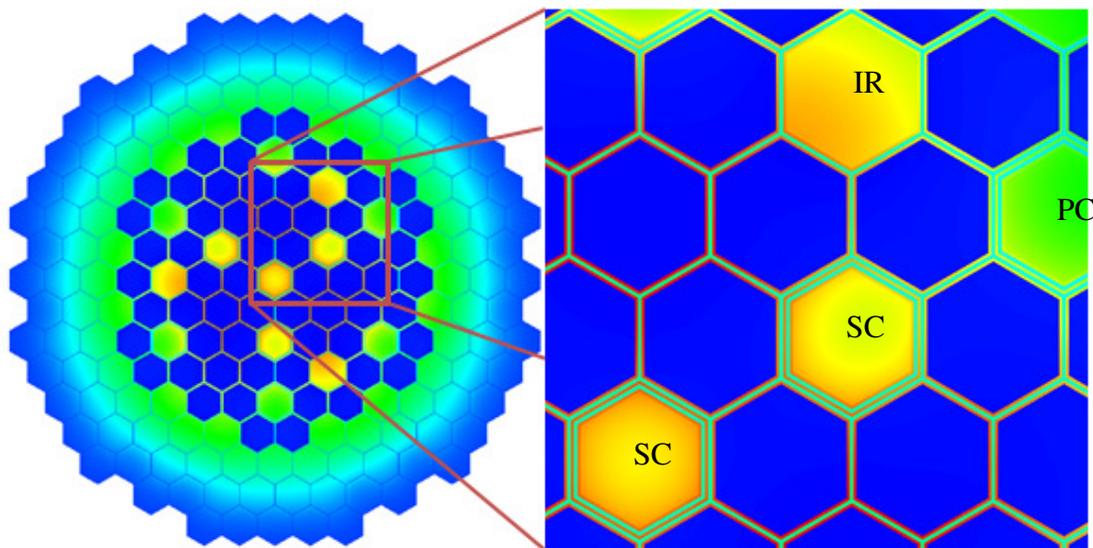


Figure 25. ABTR absorption rate for partially homogeneous assemblies with heterogeneity effects, L7T7, P1 scattering, 33 groups, control rods in (slice at 140 cm)

Figure 25 shows the absorption rate in non-fissionable materials for the partially homogeneous case with pin heterogeneity effects. The secondary control (SC) has the highest absorption rate of any assembly. The effect of the spatial self-shielding in these assemblies is apparent in the enlarged plot. The highest absorption rate of any region occurs in the ducts near the center of the core, which suggests that HT-9 is a stronger absorber than B<sub>4</sub>C, the control absorber material, for the spectrum of the ABTR. This explains why the internal “reflector” assemblies (IR) have a higher absorption rate than the outer ring of primary control assemblies (PC).

Figure 26 shows a vertical slice of the absorption rate plot in Figure 25. The assemblies are labeled in the figures: IF/MF/OF = Inner/Middle/Outer Fuel, PC/SC = Primary/Secondary Control, RR = Radial Reflector, SH = Shield. Every other assembly in this slice is intersected along the duct, so it is narrower than the adjacent assemblies, which are intersected through the widest part. The strongest absorption rates are located in the assembly ducts, especially near the center of the core. There is a significant amount of absorption well outside the active core region, both in the lower reflector and also in the assembly ducts above and below the core. This axial cross section of the core shows the strong radial absorption gradient in the control assemblies. It also shows the axial dependence of the radial gradient.

The high absorption rate in the assembly ducts highlights their neutronic significance and the importance of accurately modeling them. The partially homogeneous geometry model allows PROTEUS-SN to model the spatial configuration of the assembly ducts in a transient scenario explicitly without geometric approximation. Thus, the partially homogeneous model with pin heterogeneity effects in the cross sections is an important and unique tool for accurately modeling the behavior of the ABTR and other SFR cores with respect to assembly deformation.

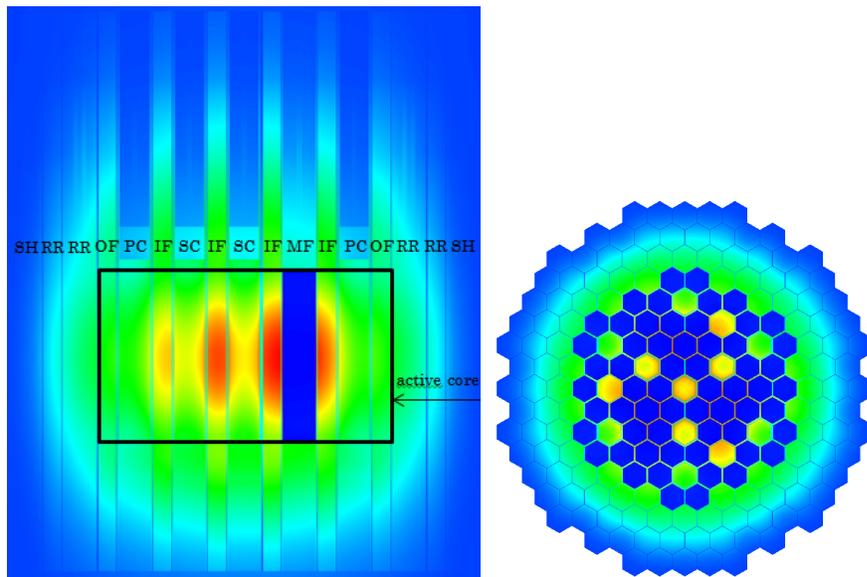


Figure 26. Absorption rate in non-fuel, vertical slice of the core midplane, partially homogeneous w/ pin heterogeneity effects, L7T7, P1 scattering, 33 groups, control rods in

### 5.1.3 Summary

PROTEUS-SN closely matches MCNP results when appropriate multigroup generation procedures are performed in MC<sup>2</sup>-3/TWODANT for spatial heterogeneity effects. Most of the PROTEUS-SN solutions with 116 groups are within 100 pcm of the MCNP reference k-effective. For the homogeneous assembly case, where the core spectrum is accounted for in the fuel and control assembly cross sections, the 33 and 70 group solutions are also close to the MCNP reference. The total heterogeneity effect estimated by PROTEUS with 116 groups is comparable to what was calculated in MCNP. The control rod worth was also very similar between the two codes: for the homogeneous and partially homogeneous cases with heterogeneity effects, the control rod worth in PROTEUS was within uncertainty on the MCNP reference.

TWODANT provides a core spectrum calculation that extends and improves the cross section condensation capabilities of MC<sup>2</sup>-3. However, some of the poor results with the coarser (33 group) energy structure demonstrate the need for unification of the spectrum calculations. The ability of MC<sup>2</sup>-3 to incorporate both the global flux spectrum from TWODANT and the local assembly-level spectral variation from a separate, internal 1D calculation is not yet verified. The implementation of that capability should greatly improve the multigroup results for cases with some assembly-level heterogeneity. This is important because a 33 group transport solution is significantly less costly than a 116 group transport solution.

## 5.2 ASTRID

With current deterministic codes, the analysis of complicated reactors is generally carried out with the use of simplified models, often omitting important geometric details. Particularly, the fuel assemblies are generally modeled through the homogenization of the constituent materials and not by an explicit pin by pin representation. The causes of these limitations are generally based on the very demanding computational resources that a detailed model requires. Additional limitations may be also associated to other issues, such as the mesh size in the case of deterministic codes based upon the nodal variational method (e.g. VARIANT).

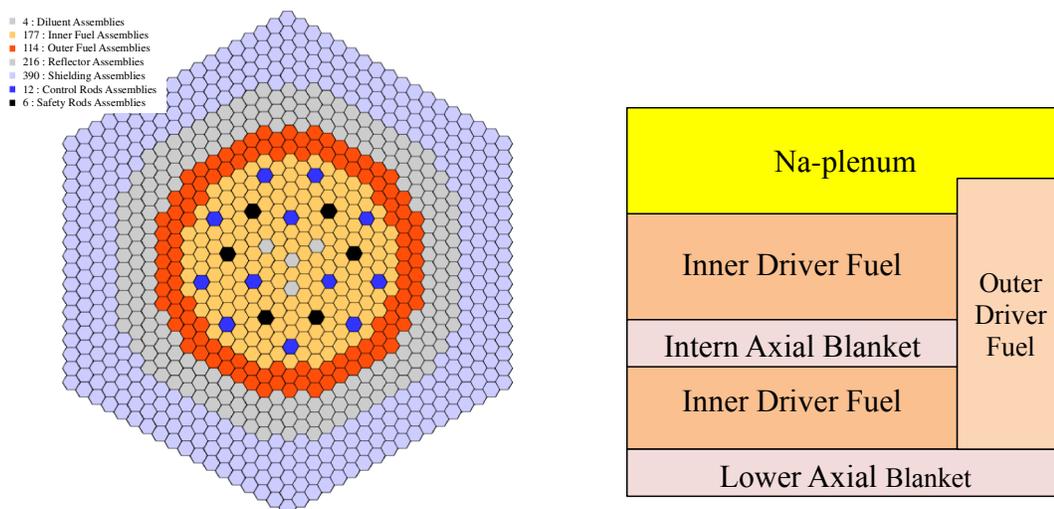


Figure 27: Radial (left) and Simplified Axial Layout (right) of ASTRID

As a code verification and validation effort, PROTEUS calculations were performed for partially homogenized systems of ASTRID, where regions of interest are represented explicitly and other regions are homogenized to reduce the problem size and required computational resources.

This section shows the preliminary application of PROTEUS to the analysis of the sodium-cooled fast reactor ASTRID. Model specifications were taken from the benchmark exercise that was proposed within a collaborative effort between DOE-INL & ANL and CEA-DEN [7]. Particularly, the goal of this study is to show the PROTEUS capabilities for the analysis of important neutronic features, such as heterogeneity effects, pin by pin power distribution, etc., that cannot be directly investigated with the deterministic codes presently in use.

### 5.2.1 Fully Assembly Homogenized Models of ASTRID

First, applications of the PROTEUS code were made to determine the multiplication factor value of a fully assembly homogenized model of the ASTRID reactor at the End of Cycle, as described later. For comparison purposes with the deterministic codes commonly used at ANL in past years, results were also obtained with the DIF3D/VARIANT code [22]. Both deterministic codes use the same 33 energy group ISOTXS cross section file generated with MC<sup>2</sup>-3 and ENDF/B-VII.0 data. At present, cross sections were simply processed with homogeneous (0D) cell calculations and without the coupling option with the TWODANT code for the collapsing procedures.

VARIANT results are obtained using a P<sub>3</sub> angular spherical harmonics approximation with a P<sub>1</sub> scattering kernel while the PROTEUS results are obtained with the standard S<sub>4</sub> angular cubature (Carlson level symmetric even-moment cubature) and P<sub>1</sub> anisotropic scattering. Additionally, the PROTEUS mesh file was generated directly from the VARIANT Hex-Z geometry with the use of a convertor tool. Results of both deterministic codes are presented in Table 15. For comparison, the k-effective values obtained by MCNP5 with ENDF/B-VII.0 data are also given in Table 15. As can be seen in Table 15, there is a ~200 pcm discrepancy between the VARIANT and PROTEUS results when using the same ISOTXS cross section set, the same anisotropic scattering order and a comparable angular approximation. It was successively demonstrated that this discrepancy is due to an inconsistency in the formulation of the fission spectrum ( $\chi$ ) in the DIF3D/VARIANT case. Thus, to ensure that the two codes make use of the exact same cross sections (particularly  $\chi$ ), PROTEUS calculations were also performed with the macroscopic cross sections (the DIF3D COMPXS file is translated into the PROTEUS ANLXS input file) which yielded k-effective values that agrees well with the VARIANT code. This also happens to be in excellent agreement with MCNP, but that is not typical. Figure 28 is also provided to demonstrate the excellent agreement on the flux spectra distributions obtained with both deterministic codes in a central zone of the inner core.

Table 15. Comparison of k-effective for a Fully Homogenized Assembly Model of ASTRID.

Code	Calculation type	$k_{\infty}$
MCNP5	Monte Carlo	1.00702 ± 0.00009
VARIANT	P3P1	1.00720 *
PROTEUS	S4P1 CARLSON_EM (using ISOTXS cross section file)	1.00508
	S4P1 CARLSON_EM (using ANLXS cross section file)	1.00713

\* A 400 pcm difference results if using a different anisotropic scattering orders (P3 vs P1) and different cross section processing in MC<sup>2</sup>-3 (heterogeneous vs homogeneous cell calculation; with and without the use of TWODANT)

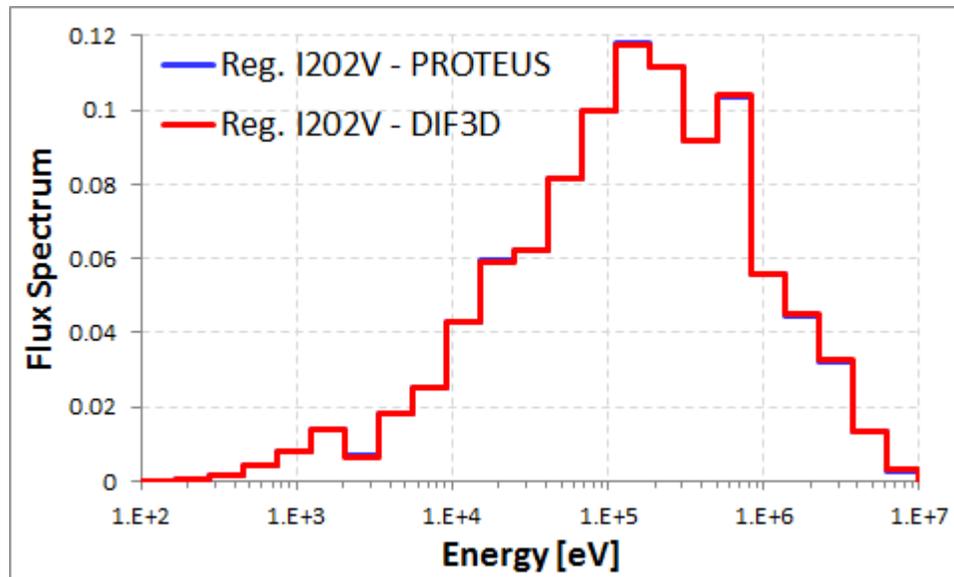


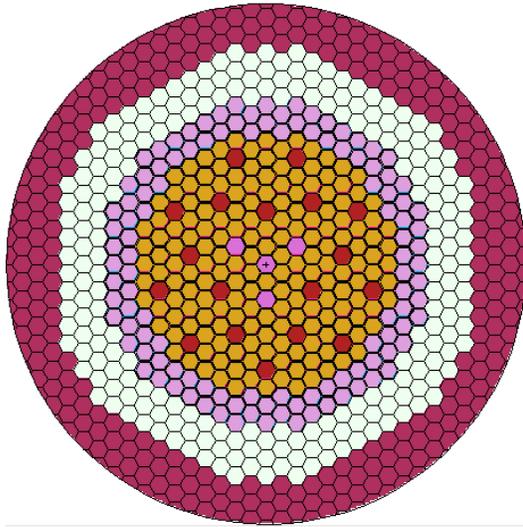
Figure 28. Flux spectra comparison in a central zone of inner core

### 5.2.2 Partially Homogenized Geometry

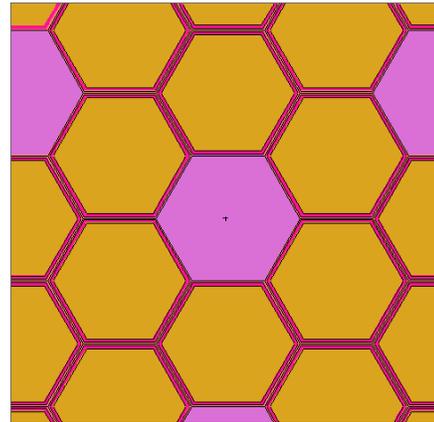
The main purpose of this work is to demonstrate some computational capabilities of the PROTEUS code that other deterministic codes do not have. Thus, applications of the PROTEUS code to partially homogenized reactor models are studied. In particular, two applications were envisaged for this kind of study: First, calculations were performed for a reactor model with the explicit representation of wrapper tube and inter-assembly sodium gap for all fuel and blanket regions (see Figure 29). Then, a second model contains an explicit pin by pin representation of a single assembly in the inner core (the heterogeneous representation regards both fuel and blanket portions of the assembly), leaving a full material homogenization in all other assemblies (see Figure 29). Both geometry models considered in this analysis cannot be handled with the deterministic codes presently in use, such as VARIANT, that are limited to the full homogenization of each reactor assembly.

#### Monte Carlo Calculations

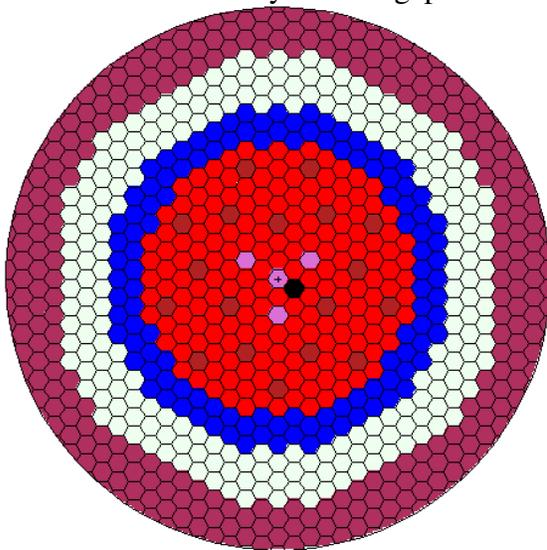
The proposed reactor models were first calculated with the MCNP5 code and results are given in Table 16. The obtained results indicate that the reactivity correction due to the explicit representation of wrapper tube and inter-assembly sodium gap with respect to the case of fully homogenized assemblies is not negligible. On the contrary, the pin by pin representation of a single driver assembly does not introduce any additional correction in terms of reactivity but it can provide information details (such as pin by pin power profile, flux distributions, etc.) that are typically impractical to obtain with the conventional deterministic codes in use.



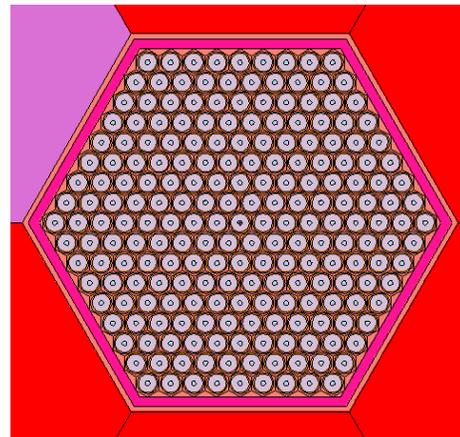
(a) explicit representation of wrapper tube and interassembly sodium gap.



(b) Detail of (a)



(c) pin by pin representation of a single assembly of inner core



(d) Detail of (c)

Figure 29. PROTEUS Models for ASTRID

Table 16.  $k_{eff}$  values for Different ASTRID Models Using MCNP5

Model	$k_{eff}$	Difference from Ref. [pcm]
Fully assembly homogenized	$1.00702 \pm 0.00009$	-
Explicit representation of duct/Na gap for all fuel and blanket regions	$1.01121 \pm 0.00008$	0.00419
Pin by pin representation of a single assembly of inner core	$1.00724 \pm 0.00009$	0.00022

2D Core Calculations with PROTEUS

Preliminary applications of PROTEUS to the partially homogenized reactor models discussed in the previous section were also performed. These applications were first carried out in 2D geometry. The mesh files were generated using the Ufmesh generation tool. Figure 30 shows some of the details of the generated meshes in the case of the explicit representation of the wrapper tube and inter-assembly sodium gap for all fuel assemblies and for the case of pin by pin representation of a single fuel assembly.

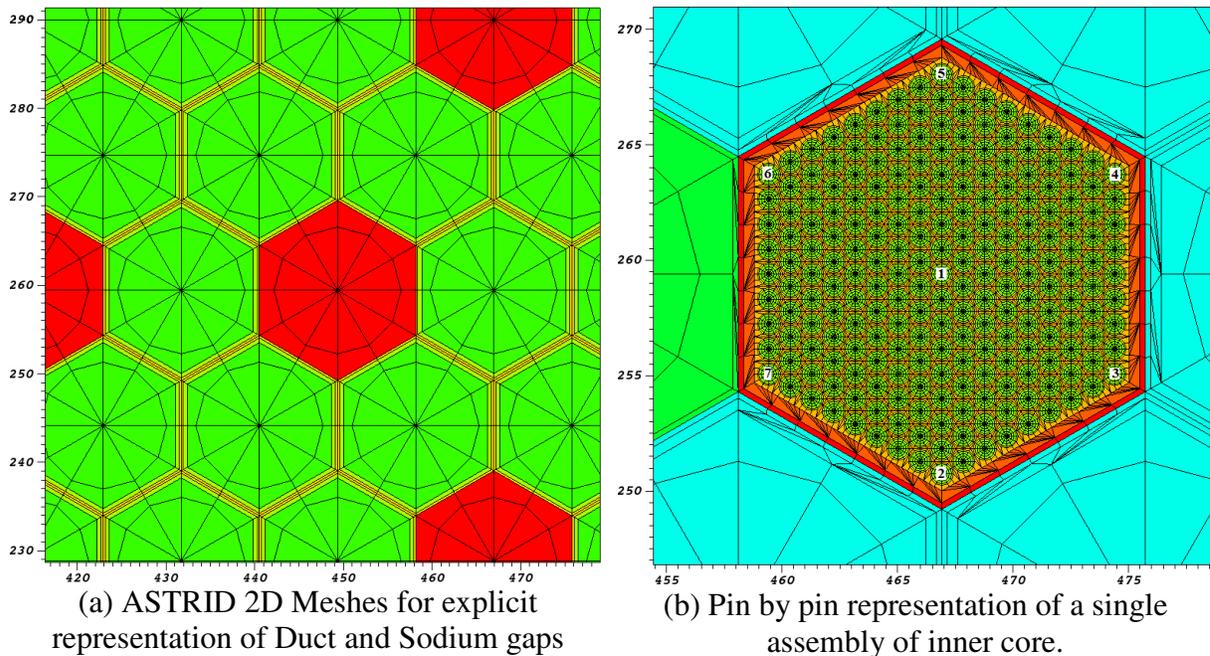


Figure 30. PROTEUS Meshes for ASTRID

Table 17 shows the preliminary results of k-effective values obtained with the PROTEUS code for both partially homogenized reactor models. For comparison, the k-effective values obtained with fully homogenized assembly models both with PROTEUS and VARIANT are also presented in Table 17. For all cases, cross sections are provided as an ISOTXS file generated with MC<sup>2</sup>-3 in 33 energy groups using ENDF/B-VII.0 nuclear data and homogeneous (0D) cell calculations are performed for each medium.

The non-negligible reactivity effect found with the MCNP5 simulation in the case of explicit representation of wrapper tube and inter-assembly sodium gap with respect to the case of fully homogenized assemblies is confirmed by the PROTEUS calculation as well. However, the magnitude of these effects is not the same as found by the Monte Carlo simulation (~800 pcm with PROTEUS in 2D compared to ~400 pcm with MCNP5 in 3D). It is then essential to run a PROTEUS calculation on the exact 3D models developed with MCNP5 so that a consistent comparison can be made between the results obtained by the two codes. Also, according to the

MCNP5 results in Table 16, Table 17 shows that the pin by pin representation of a single driver assembly does not introduce any additional correction in terms of reactivity.

Table 17. Comparison of calculated  $k_{eff}$  values for the 2D ASTRID model with DIF3D and PROTEUS

Code	Calculation type	Model	$k_{eff}$	$\Delta k$ , pcm
DIF3D	P3P1	2D with fully assembly homogenized model	1.28332	-
PROTEUS	S4P1 CARLSON_EM	2D with fully assembly homogenized model	1.28113	-
		2D with explicit description of duct/Na gap	1.28933	820
		2D with pin by pin description of a single assembly	1.28137	24

For the case with explicit pin by pin representation of a single driver assembly, power fractions and energy spectra were determined in the fuel pins located at the center and at the 6 corner of the assembly as indicated in Figure 30. Results are presented in Table 18 and in Figure 31. It can be noted that due to the reactor symmetry similar results are obtained for pins 2 and 5, 3 and 4, and 6 and 7. Table 18 shows that differences up to 5% may be found on the power fraction of the single pins within the same assembly. As shown in Figure 31, differences more or less pronounced can be found on the flux distributions as well.

It is important to underline that the results presented in this section are not intended for a specific analysis of the ASTRID neutronic features. The main goal of these applications is to demonstrate the capabilities of the code for the investigation of heterogeneity effects that are practically impossible to determine with the other deterministic codes in use. As an example, assuming that we know the assembly where the maximum power occurs, with this kind of application one can investigate the pin by pin power and flux distribution of that specific assembly, so that a reactor analysis/design can be based on more precise results than assembly averaged values.

Table 18. Power fractions for fuel pins in positions 1 to 7.

Pin	Power fraction	Difference to central pin (%)
1	2.835E-05	-
2	2.817E-05	-0.6
3	2.939E-05	3.7
4	2.939E-05	3.7
5	2.817E-05	-0.6
6	2.806E-05	-1.0
7	2.806E-05	-1.0

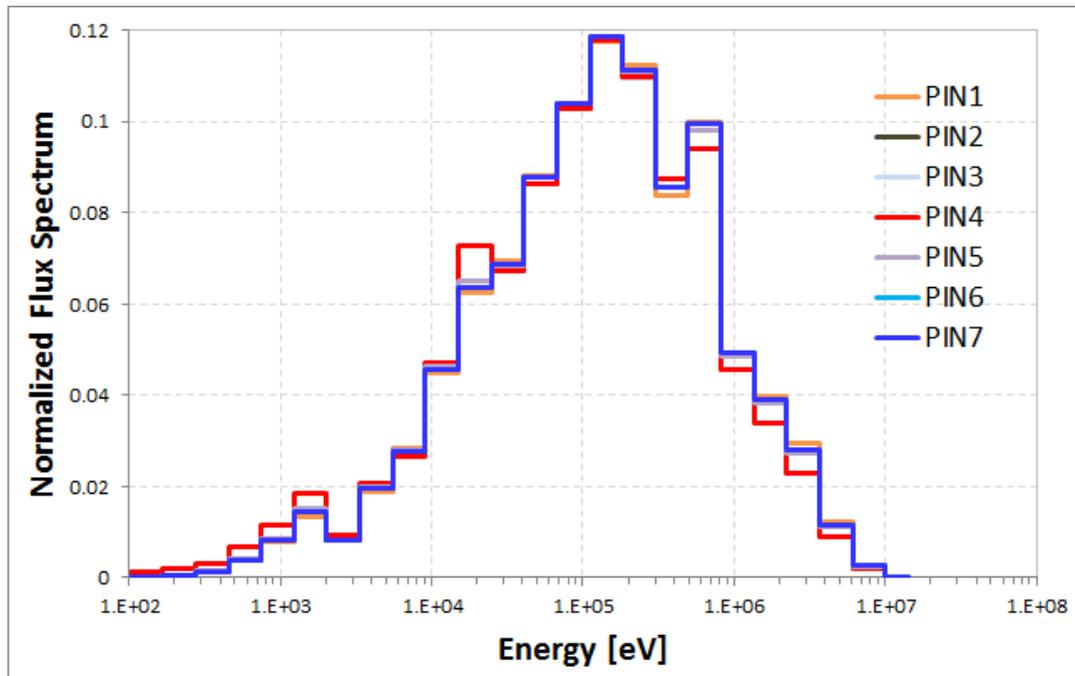


Figure 31. 33 energy-groups flux spectra for fuel pins in positions 1, 2, 3 and 7.

### 3D Core Calculations with PROTEUS

Preliminary results of PROTEUS applications to partially homogenized models of the ASTRID reactor were also obtained in 3D geometry. Using the Ufmesh input, mesh files were generated for the same reactor models that were calculated with the MCNP5 code and results are summarized in Table 19. Similarly to the calculations in 2D geometry, cross sections are provided as an ISOTXS file generated with MC<sup>2</sup>-3 in 33 energy groups using ENDF/B-VII.0 nuclear data and homogeneous (0D) cell calculations are performed for each medium.

From the results presented in Table 19, it can be noted that for the case of fully assembly homogenized reactor problems, the use of mesh files generated with the Ufmesh and GRID input capabilities and the DIF3D\_to\_PROTEUS convertor tool all produce consistent k-effective values. In Table 19 it can be also noted that the reactivity effect introduced by the explicit representation of wrapper tube and inter-assembly sodium gap with respect to the case of fully homogenized assemblies is ~1400 pcm, i.e. 1000 pcm larger than the effect estimated by the MCNP5 simulation (compare with results presented in Table 16). The results presented in Table 19 should be considered only preliminary. Further studies are required to investigate any improvement in the adopted mesh specifications. Additionally, the assembly heterogeneity effects also depend on the processed cross sections that at present have been simply obtained from separate homogeneous (0D) cell calculations performed with MC<sup>2</sup>-3 for each material (e.g. fuel pin, clad, coolant, wrapper tube). In a more accurate approach, cross sections can be processed with the heterogeneity capabilities of MC<sup>2</sup>-3 in 1D geometry.

Table 19. Comparison of calculated  $k_{eff}$  values for the 3D ASTRID model with PROTEUS

Code	Calculation type	Model	$k_{eff}$	Het. – Hom., $\Delta k$
PROTEUS	S4P1 CARLSON_EM	3D with fully assembly homogenized model	1.00408	-
		3D with explicit description of duct/Na gap	1.01869	0.01461
		3D with pin by pin description of a single assembly	1.00424	0.00016

### 5.2.3 Summary

With the deterministic codes presently in use, such as DIF3D/VARIANT, the calculations of the ASTRID reactor (like any other reactor in general) are based on fully assembly homogenized model. Under the NEAMS project, a new deterministic code, PROTEUS, has been recently developed at ANL that allows performing 3D transport calculations of more detailed reactor models that cannot be handled by the deterministic codes currently in use.

The preceding results show the application of PROTEUS to the analysis of the sodium-cooled fast reactor ASTRID. The ultimate goal of the PROTEUS would be its use on partially homogenized systems, where regions of interest are represented explicitly and other regions are homogenized to reduce the problem size and required computational resources. However, as verification purposes, the code was first applied to the fully assembly homogenized model of the ASTRID reactor and a good agreement was found with the VARIANT and MCNP5 results of  $k_{eff}$  values.

Regarding the application to partially homogenized systems, two reactor models were envisaged. The first model is characterized by the explicit representation of wrapper tube and inter-assembly sodium gap only for fuel and blanket regions. Then, a second model contains an explicit pin by pin representation of a single assembly in the inner core (the heterogeneous representation regards both fuel and blanket portions of the assembly), leaving a full material homogenization in all other assemblies. Results obtained both with PROTEUS and MCNP5 confirmed that the reactivity correction due to the explicit representation of wrapper tube and inter-assembly sodium gap with respect to the case of fully homogenized assemblies is not negligible. However the two codes do not provide consistent results with regard to the magnitude of this correction (~1400 pcm in the PROTEUS case vs ~400 pcm in the MCNP5 case). Future studies are needed in order to better address the discussed inconsistency which we believe is due to the cross section generation process. It is particularly important to note that the PROTEUS results, especially in the case of partially homogenized reactor models, are to be considered only preliminary. Further studies are required to investigate any improvement in the adopted mesh specifications. Additionally, the assembly heterogeneity effects also depend on the processed cross sections that at present have been simply obtained from separate homogeneous (0D) cell calculations performed with MC<sup>2</sup>-3 for each material (e.g. fuel pin, clad, coolant, wrapper tube). In a more accurate approach, cross sections can be processed with the heterogeneity capabilities of MC<sup>2</sup>-3 in 1D geometry. Future studies should also compare other neutronic parameters such as the sodium void worth coefficient.

On the contrary, both PROTEUS and MCNP5 results show that the pin by pin representation of a single driver assembly does not introduce any additional correction in terms of reactivity with

respect to the case of fully homogenized assemblies but it can provide information details (such as pin by pin power profile, flux distributions, etc.) that are typically impractical to obtain with the conventional deterministic codes in use.

### 5.3 TREAT

#### 5.3.1 Homogenized Fuel Assembly in Infinite Lattice

To begin investigating the impact of homogenization, MCNP simulations were performed for an infinite lattice of TREAT fuel assemblies, assuming a simplified geometry. Reflective boundary conditions were applied to the radial surfaces of the assembly, with void boundary conditions above and below the axial reflector region. In the heterogeneous model, the fuel and clad were modeled with chamfered corners, as shown in Figure 32. As a simplification, the axial reflector regions were assumed to have the same radial geometry as the fuel region, and the axial spacers were ignored, as shown in Figure 33(a).

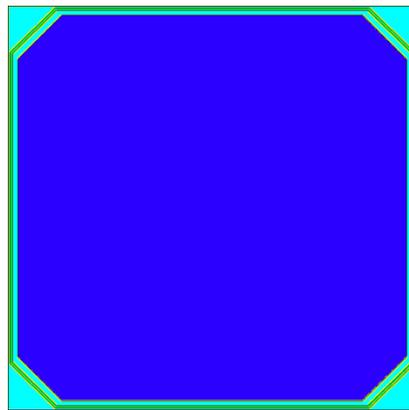


Figure 32. Cross-section view of TREAT Fuel Assembly in MCNP

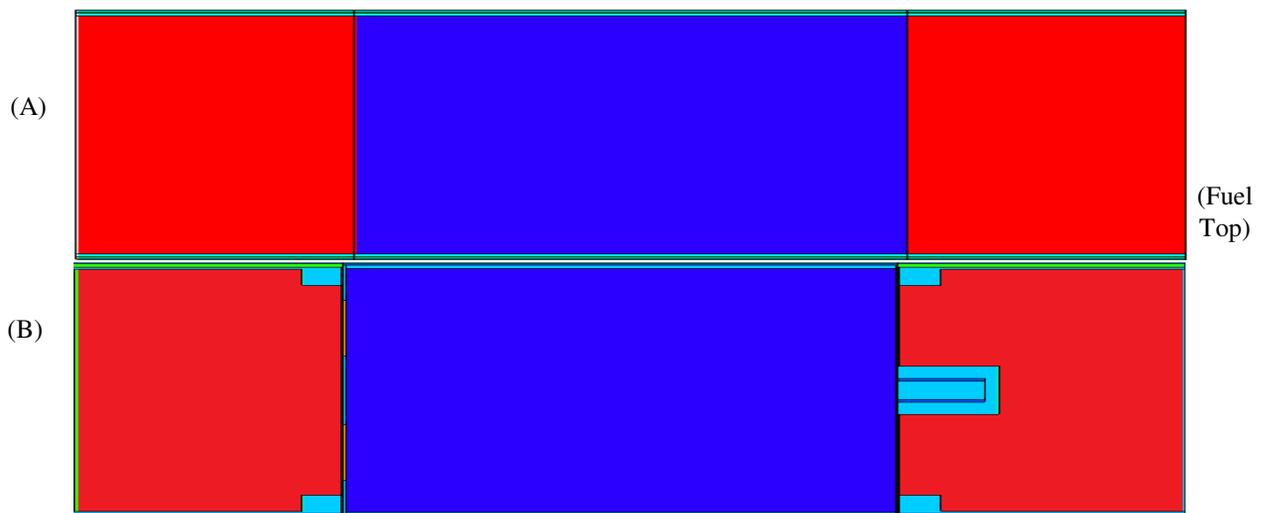


Figure 33. Lengthwise View of TREAT Fuel Assembly in MCNP, (a) Simplified Geometry and (b) True Geometry

Two cases were simulated: one assuming a zircaloy can over the whole fuel assembly, and the other assuming zircaloy cladding on the fuel and aluminum cladding on the reflector. The results are summarized in Table 20. The homogenization of the reflector region causes an under-prediction of the axial neutron leakage, increasing reactivity. This effect may be even more significant for the actual geometry of the fuel assembly, which features an outgas tube and larger gaps near the fuel-reflector interface, as shown in Figure 33(b). Homogenization of the true geometry has not yet been evaluated. In addition, the effect of homogenization has not yet been evaluated for fuel assemblies containing a control rod.

Table 20. MCNP Results for Infinite Lattice of Simplified TREAT Fuel Assemblies

Model	k	1 $\sigma$	$\Delta\rho$ , pcm	change, pcm
Full Can Zr, Heterogeneous	1.46428	0.00010	31707	
Full Can Zr, Homogenize Fuel Region, Heterogeneous Reflector Region	1.46449	0.00009	31717	10
Full Can Zr, All Regions Homogenized Radially	1.46854	0.00011	31905	198
Al Clad on Reflectors, Heterogeneous	1.45561	0.00011	31300	
Al Clad on Reflectors, Homogenize Fuel Region, Heterogeneous Reflector Region	1.45624	0.00011	31330	30
Al Clad on Reflectors, All Regions Homogenized Radially	1.46150	0.00010	31577	277

The radially homogenized, simplified geometry fuel assembly was then simulated for an MCNP-PROTEUS comparison. For consistency with the cross-section set used in the PROTEUS simulations, the MCNP inputs were changed to ENDF6.0 cross-sections for the code-to-code comparison. In addition, isotopes in the MCNP input without data available in the PROTEUS cross-section set were removed from the model, so that the PROTEUS and MCNP simulations assumed identical compositions. Once again, reflective boundary conditions were assumed for the sides of the assembly, with void boundary conditions above and below the reflector region. Two meshes were used in the PROTEUS simulations, one with 10x10 radial segmentation, and the other with 20x20 radial segmentation. Both meshes had 15 axial segments in the fuel region, and 7 in each reflector (Figure 34). Results are summarized in Table 21. Further refinement in geometry and angle can be evaluated in the future.

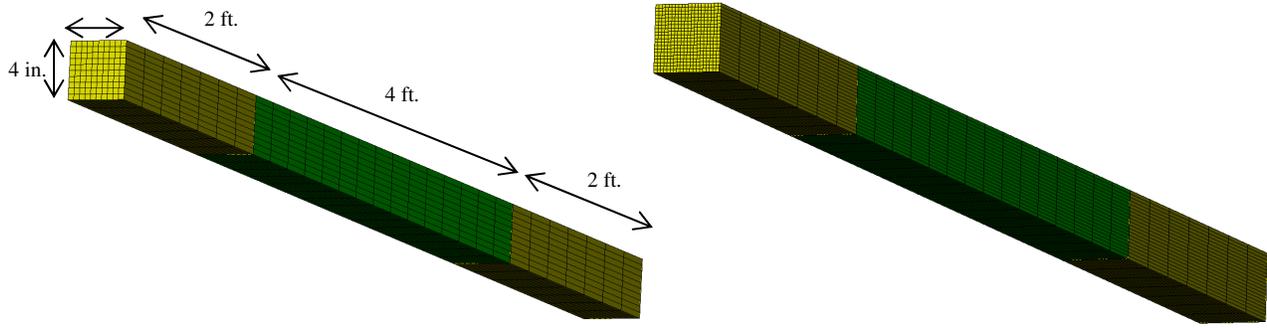


Figure 34. PROTEUS Meshes Used for Infinite Lattice of Homogenized Assemblies, (a) 10x10x29 and (b) 20x20x29

Table 21. MCNP and PROTEUS Simulations of Infinite Lattice of Homogenized, Simplified TREAT Fuel Assemblies

Model		k (1 $\sigma$ )	$\Delta\rho$ , pcm	MCNP - PROTEUS	
Whole Can Zircaloy	MCNP		1.47848 (0.00011)	32363	
	PROTEUS	LT 1 1, 10x10x29 mesh, scatter 1	1.47878	32377	-14
		LT 5 5, 10x10x29 mesh, scatter 1	1.47914	32393	-30
		LT 5 5, 20x20x29 mesh, scatter 1	1.47914	32393	-30
		LT 5 5, 20x20x29 mesh, scatter 0	1.47703	32297	66
		LT 11 11, 20x20x29 mesh, scatter 0	1.47705	32297	66
Zircaloy on Fuel, Aluminum on Reflector	MCNP		1.47054 (0.00011)	31998	
	PROTEUS	LT 5 5, 20x20x29 mesh, scatter 0	1.46862	31909	-89

### 5.3.2 Full Core Model

The previous full-core simulations of the DIF3D-converted PROTEUS model allowed for direct comparisons between DIF3D and PROTEUS. To now compare against MCNP as well, a new MCNP input was constructed which exactly models the geometry and compositions assumed in the DIF3D and PROTEUS simulations. Again ENDF6.0 cross-section libraries were used, to eliminate the impact of differences in cross-section data. In addition, the MCNP model used only

isotopes present in the PROTEUS model. The original DIF3D model featured a reduced radial reflector with an albedo boundary condition. For the current code-to-code comparisons, this geometry was still used, now assuming a void boundary condition. In the future, changes can be made to all three models to better represent the true TREAT geometry. The PROTEUS model (the same as the DIF3D model) is shown in Figure 35, and the MCNP model in Figure 36.

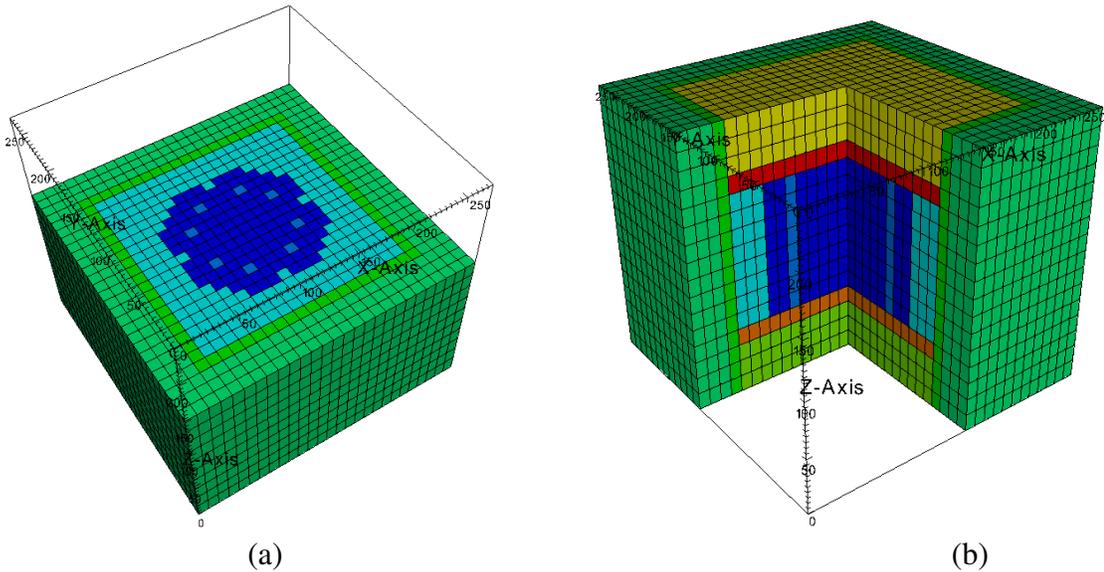


Figure 35. PROTEUS Model for TREAT, (a) Radial Slice and (b) Axial Slice

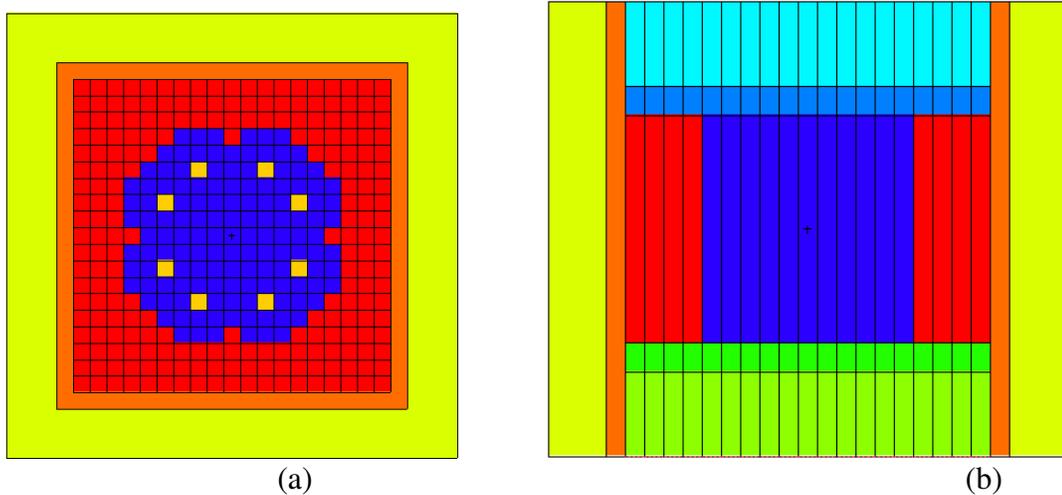


Figure 36. MCNP TREAT Core, (a) X-Y View and (b) X-Z View

PROTEUS simulations were performed for three different meshing schemes. The first used the original DIF3D mesh, which features 10.16” (i.e. the width of a single fuel assembly) cells in the x- and y-directions, with 16 axial segments. The additional simulations were performed doubling, and then quadrupling the refinement of the original mesh (as shown in Figure 37).

Results are summarized in Table 22. Additional simulations should be performed to investigate further refinement of the mesh (although this refinement may only be necessary in select areas of the geometry, rather than universally across the entire model).

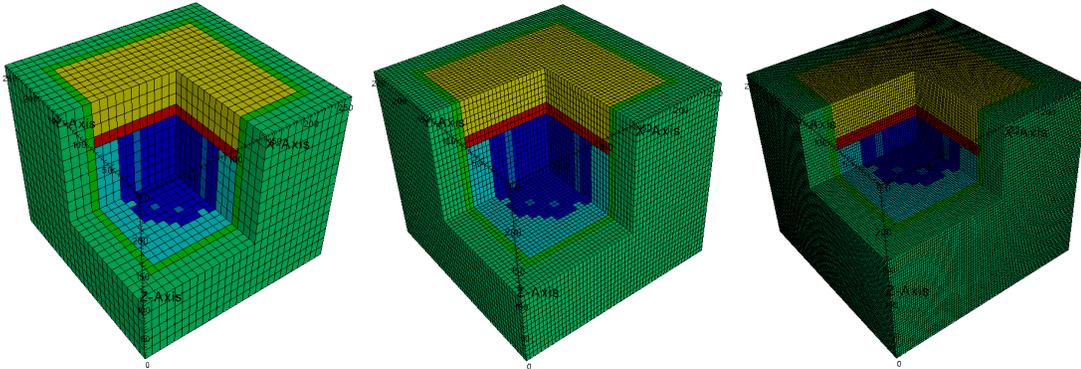


Figure 37. Meshes Used in Current PROTEUS Simulations

Table 22. PROTEUS, MCNP, and DIF3D Results for TREAT Core

MCNP			keff (1 $\sigma$ )		pcm		
			1.00872 (0.00019)		864		
PROTEUS	Scatter Order	mesh refinement	cubature		Eigenvalue		PROTEUS - MCNP
			keff	pcm			
	1	DIF3D mesh	LEG-TCHEBY	1/1	1.009725	963	99
			LEG-TCHEBY	3/2	1.009721	963	98
			CARLSON_EM	4	1.009714	962	98
		Double	LEG-TCHEBY	3/2	1.011335	1121	256
			LEG-TCHEBY	5/5	1.011329	1120	256
			LEG-TCHEBY	7/8	1.011333	1121	256
	Quadruple	LEG-TCHEBY	3/2	1.011726	1159	295	
	0	DIF3D mesh	LEG-TCHEBY	3/2	1.004608	459	-406
		Double	LEG-TCHEBY	3/2	1.006222	618	-246
		Quadruple	LEG-TCHEBY	3/2	1.006615	657	-207
LEG-TCHEBY			11/11	1.006614	657	-207	
DIF3D	Solution option			Eigenvalue		DIF3D - MCNP	
	Nodal Diffusion Theory			keff	pcm		
	Variational Nodal Transport	scatter 1	P3 flux, P3 leakage	1.012114	1197	332	
		scatter 0	P3 flux, P3 leakage	1.006964	692	-173	

Because of the high demands of the TREAT conversion project, a smaller fraction of time was spent on the TREAT PROTEUS simulations. The primary focus of effort for this project was on becoming more familiar with CUBIT, a software toolkit for generation of two- and three-dimensional geometry and finite element meshes. Because of the unique design of TREAT, models must be developed directly in CUBIT. Preliminary PROTEUS simulations for TREAT were performed using a converted DIF3D model, which features a homogenized representation of the TREAT core. Efforts are now focused on explicitly modeling the TREAT fuel assembly geometry, for comparison with MCNP simulations. An example CUBIT-generated TREAT fuel assembly mesh is shown in Figure 38.

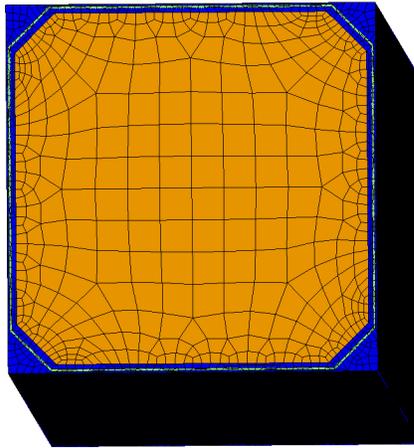


Figure 38. Example CUBIT-generated TREAT fuel assembly mesh for use with PROTEUS

## 6. Conclusions

This report summarizes the current status of NEAMS activities in FY2015. The tasks this year are (1) to improve solution methods for steady-state and transient conditions, (2) to develop features and user friendliness to increase the usability and applicability of the code, (3) to improve and verify the multigroup cross section generation scheme, (4) to perform verification and validation tests of the code using SFRs and thermal reactor cores, and (5) to support early users of PROTEUS and update the user manuals.

PROTEUS was composed of three different high-fidelity transport solvers: SN2ND, MOCFE, and MOCEX. Among them, MOCEX (2D MOC coupled with the discontinuous Galenkin method axially based on the extruded geometry in the axial direction) requires less memory and computation time but is still under verification. To provide more options for users, the development of intermediate-fidelity transport solvers (MOC 2D/1D and NODAL) were initiated, which allows the code to produce the solutions that the user needs with practical time and computing resources. Those new solvers will be completed and tested in the following years.

The current transient solution scheme of PROTEUS with the adiabatic method was reviewed to find an efficient way to improve it with the improved quasi-static (IQS) method. It was suggested that the IQS method be implemented for a first order solver which can compute the full angular flux rather than only the scalar flux (and even-parity angular flux). Therefore, an actual implementation of the IQS will be conducted to the existing MOC solvers or the first-order SN which should be developed in advance.

To improve the usability of PROTEUS, a user-friendly mesh generation capability, including UFmesh and GRID, was developed for typical Cartesian or hexagonal geometries, which generates the mesh input based on user inputs instead of using CUBIT. This allows the user to easily build a mesh file with input cards and options and thus to quickly update the mesh file by changing user inputs. As the UFmesh is generated preserving the original geometry volumes, no additional adjustment that is required for the CUBIT-generated mesh is necessary. This capability fills up the gap that RGG (MeshKit) based on CUBIT does not support.

The ANL cross section library (ACSL) has been further verified using the selected VERA PWR benchmark problems, showing good agreement with MCNP Monte Carlo solutions. Previously, the application programming interface (API) was implemented to the SN2ND and MOCEX solvers. This year, it was successfully connected to the MOCFE solver as well. To speed up the on-the-fly cross section generation, an additional resonance self-shielding option based on the Dancoff approach was tested, which allows the code to reduce the number of the FSPs to solve by more than an order of magnitude.

As an alternative and verification option to the ACSL, a procedure of generating multigroup cross sections using the Serpent Monte Carlo code was developed. The main multigroup cross sections were generated from Serpent and the higher-order  $P_N$  scattering matrices were provided from MC<sup>2</sup>-3, being produced in the ISOTXS form. Preliminary test results with the C5 PWR benchmark problem indicated very good agreement in eigenvalue between DIF3D, MCNP, and

Serpent. This approach of generating cross sections would be useful to debug and verify the multigroup cross sections for complex geometry or spectrum reactor cores.

The 2D transport capability using MOC was implemented to MC<sup>2</sup>-3 in order to better take into account the 2D effect to the resonance self-shielded multigroup cross sections generated from the code. Preliminary verification tests using ZPR-6/7 fuel drawers indicated that the 2D transport calculation of MC<sup>2</sup>-3 can account for the local heterogeneity effect properly and the 1D models have the accuracy limitation in representing the local heterogeneity.

As a continued effort of code verification and validation, three reactors were simulated and analyzed with PROTEUS. First, the 3D ABTR cores with different heterogeneity level configurations were modelled using MC2-3/PROTEUS: homogeneous assembly model, partially homogeneous (duct heterogeneous) model, and partially homogeneous model with the fuel heterogeneity effect. Note that a fully heterogeneous assembly model is not necessary because the heterogeneity effect is not significant in a fast reactor system. It was found that most of the PROTEUS-SN eigenvalues with 116 groups were within 100 pcm of the MCNP solutions. The total heterogeneity effect and control rod worths were in very good agreement with MCNP. The initial thermal expansion tests showed good agreement in the reactivity change between PROTEUS and MCNP as well.

The 3D ASTRID cores were simulated using MC<sup>2</sup>-3/PROTEUS with three different configurations similarly to the ABTR simulation: homogeneous assembly model, partially homogeneous assembly model (an explicit representation of wrapper tube and inter-assembly sodium gap), and partially homogeneous model with a fully heterogeneous assembly. The core  $k_{\text{eff}}$  for the homogeneous assembly model showed good agreement between PROTEUS and MCNP. Unlike the ABTR simulation, however, those for the partially homogeneous assembly model showed a noticeable discrepancy between the two codes, which could be attributed to the inconsistency in generating the multigroup cross sections using MC<sup>2</sup>-3 and should be further investigated in future.

The simulation of the 3D TREAT cores was initiated using PROTEUS and MCNP with the homogeneous assembly model and the heterogeneous assembly model. The eigenvalue solutions were in reasonable agreement between the two codes. For detailed investigation, 3D single assembly models with two different heterogeneous configurations were built: one including a zircaloy can smeared over the whole fuel assembly and the other including zircaloy cladding on the fuel and aluminum cladding on the reflector. The eigenvalue solutions from the two codes were in good agreement within 90 pcm. Further analysis is ongoing and the final report on TREAT will be written in a separate document in October.

As an effort of the PROTEUS user support, the PROTEUS manual was revised to include new options and changes made to the code (Revision 2.0). Technical supports were also made to the ORNL team who has been developing the depletion module for PROTEUS using the ORIGEN API. Additionally, we took the initiative to write up the MOAB API specifications for interfacing MOAB with PROTEUS to minimize current and potential problems arising from the multi-physics coupling between three physics tools via SIGMA.

The future work would be as follows:

- Develop the 1<sup>st</sup> order SN or use the MOC to implement the IQS for more accurate transient simulation,
- Verify the MOC solvers (MOCFE and MOCEX) to ensure that they are working properly in a massive parallel environment and work on getting them into a production form,
- Complete the development of the intermediate-fidelity solvers (MOC 2D/1D and NODAL),
- Invest effort on numerical acceleration for the SN and MOC solvers,
- Continue verifying the Argonne cross section libraries (ACSL) independently and via the cross section API for thermal and fast reactors,
- Continue to perform verification and validation tests of PROTEUS against ATRID, TREAT and more fast and thermal reactor cores,
- Add the fuel cycle analysis capabilities to support actual core design and analysis activities,
- Support the users of standalone PROTEUS and multi-physics simulation with other physics tools.

## REFERENCES

1. A. SIEGEL, et al, "Software Design of SHARP," Proc. of M&C+SNA 2007, Monterey, CA, April 15-17, 2007.
2. C. H. Lee and M. A. Smith, "Development Plan for NEAMS Neutronics Module," ANL/NE-13/17, Argonne National Laboratory, September 30, 2013.
3. E. R. Shemon, C. H. Lee, M. A. Smith, and A. Marin-Lafleche, "NEAMS Neutronics: Development and Validation Status," Proceedings of ICAPP 2014, Charlotte, USA, April 6-9, 2014 [CD-ROM].
4. E. R. Shemon, M. A. Smith, and C. H. Lee, "PROTEUS-SN Methodology Manual," ANL/NE-14/5, Argonne National Laboratory, June 30, 2014.
5. E. R. Shemon, M. A. Smith, C. H. Lee, and A. Marin-Lafleche "PROTEUS-SN User Manual," ANL/NE-14/6, Argonne National Laboratory, June 30, 2014.
6. Y. I. Chang, et al., "Advanced Burner Test Reactor Preconceptual Design Report," ANL-ABR-1, Argonne National Laboratory, September 2006.
7. V. Pascal, P. Sciora, F. Varaine, "Cooperative Technical Work DOE/CEA, ASTRID safety performance evaluation, Benchmark specifications," CTW DOC/CEA - ASTRID V1, August 2012.
8. W. R. Robinson, R. J. Page, A. E. Wright, "TREAT NPR Calibration Experiment AN-CAL," ANL/NPR-92/11, Argonne National Laboratory, 1992.
9. A. MARIN-LAFLECHE, M. A. Smith, and C. H. Lee, "PROTEUS-MOC: A 3D Deterministic Solver Incorporating 2D Method of Characteristics," Proc. Of International Conf. on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013), Sun Valley, Idaho, May 5-9, 2013.
10. M. A. Smith, V. S. Mahadevan, and E. R. Wolters, "Continued Research for Improvement of PROTEUS-SN," ANL/NE-13/12, Argonne National Laboratory, September 30, 2013.
11. CUBIT Website, <https://cubit.sandia.gov>.
12. R. Jain, V. Mahadevan, and R. O'bara, "Simplifying Workflow for Reactor Assembly and Full-core Modeling," Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method, Nashville, TN, April 19-13, 2015.
13. C. H. Lee, A. Marin-Lafleche, and M. A. Smith, "Development of Cross Section Library and Application Programming Interface (API)," ANL/NE-13/15, Argonne National Laboratory, September 30, 2013.
14. S. M. Dancoff and M. Ginsburg, "Surface Resonance Absorption in a Close-Packed Lattice," CP-2157, October 1944.
15. C. H. Lee and W. S. Yang, "MC<sup>2</sup>-3: Multigroup Cross Section Generation Code for Fast Reactor Analysis," ANL/NE-11-41 Rev.1, Argonne National Laboratory, January 2012.
16. E. R. Shemon, M. A. Smith, and C. H. Lee, "PROTEUS-SN User Manual," ANL/NE-14/6 (Rev 2.0), Argonne National Laboratory, July 31, 2015.

17. W. Wieselquist, "ORIGEN FY2014 End-of-year Status Report for NEAMS Including Prototype of API User's Guide," ORNL/LTR-2014/544, Oak Ridge National Laboratory, 2014.
18. V. Mahadevan, et al., "SIGMA: Scalable Interface for Geometry and Mesh Based Applications," SIAM CS&E, Software mini-symposium, Salt Lake City, March 2015,
19. S. BALAY, et al, "PETSc Webpage," <http://www.mcs.anl.gov/petsc/>, 2013.
20. S. S. Kim, et al., "Advanced Test Reactor: Serpentine Arrangement of Highly Enriched Water-Moderated Uranium-Aluminide Fuel Plates Reflected by Beryllium," NEA/NSC/DOC (95)-03/II, HEU-MET-THERM-22. Idaho National Laboratory, 2005
21. 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.
22. G. Palmiotti, C. B. Carrico, E. E. Lewis, "Variational nodal transport methods with anisotropic scattering," *Nuclear Science and Engineering*, Vol. 115, pp. 233-243, 1993.
23. M. A. Smith and E. R. Shemon, "User Manual for the PROTEUS Mesh Tools," ANL/NE-15/17 Rev.1, Argonne National Laboratory, June 1, 2015.
24. C. H. Lee, E. R. Shemon, M. A. Smith, and A. Marin-Lafleche, "FY14 Status Report on NEAMS Neutronics Activities," ANL/NE-14/10, Argonne National Laboratory, September 30, 2014.
25. K. O. Ott and D. A. Meneley, "Accuracy of the Quasi-static Treatment of Spatial Reactor Kinetics," *Nucl. Sci. Eng.*, **36**, 402-411, 1969.
26. S. Dulla, E. Mund, P. Ravetto, "The Quasi-Static Method Revisited," *Progress in Nuclear Energy*, **50**, pp 908-920, 2008.
27. F. Alcaro, et al, "Implementation of the Quasi-Static Method for Neutron Transport", Proceedings MC2011, Rio de Janeiro, Brazil, May 8-12, 2011.
28. "Benchmark on Deterministic Transport Calculations Without Spatial Homogenization," NEA/NSC/DOC(2005)16.
29. X-5 Monte Carlo Team, "MCNP – A General N-Particle Transport Code, Version 5 – Volume I: Overview and Theory," LA-UR-03-1987, Los Alamos National Laboratory, 2003.
30. R.E. MacFarlane, "NJOY 99/2001: New Capabilities in Data Processing," Proc. of the Workshop on Reactor Physics and Analysis Capabilities for Generation IV Nuclear Energy Systems, Argonne National Laboratory, February 18-19, 2003.
31. M. A. Smith, C. H. Lee, and G. Yesilyurt, "FY2012 Status Report on Subgroup Library Development," ANL/NE-12-45, Argonne National Laboratory, September 30, 2012.
32. A. T. Godfrey, "VERA Core Physics Benchmark Progression Problem Specification," CASL-U-2012-0131-001, Rev.1, December 21, 2012.
33. J. Leppanen, "Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code," VTT Technical Research Centre of Finland, June 18, 2015
34. M. J. Grimstone, J. L. Rowlands and J. M. Stevenson, "Final Report on the International Comparison of Calculation for the CADENZA Assemblies (The Pin-Plate Benchmark)," NEACRP-L-300, OECD/NEA, 1987.

35. T. Tone, "A Numerical Study of Heterogeneity Effects in Fast Reactor Critical Assemblies," *J. Nucl. Sci. Technol.*, **12**, pp. 467-478, 1975.
36. C. H. Lee and W. S. Yang, "An improved Resonance Self-Shielding Method for Heterogeneous Fast Reactor Assembly and Core Calculations," In Proc. M&C 2013, Sun Valley, ID, USA, May 5-9, 2013, on CD-ROM, 2013.
37. R. M. Lell *et al.*, "ZPR-6 Assembly 7 High  $^{240}\text{Pu}$  Core Experiments: A Fast Reactor Core with Mixed (PU, U)-Oxide Fuel and A Central High  $^{240}\text{Pu}$  Zone," ZPR-LMFR-EXP-002, NEA/NSC/DOC (2006)1, OECD/NEA (2006).
38. A. Mohamed, W. S. Yang, M. A. Smith, and C. H. Lee, "Analysis of Reaction Rate Distribution Measurements in ZPR-6 Assembly 7 Cores with MC<sup>2</sup>-3/UNIC Code System," Proc. of International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2011), Rio de Janeiro, Brazil, May 8-12, 2011.



## **Nuclear Engineering Division**

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

[www.anl.gov](http://www.anl.gov)



**U.S. DEPARTMENT OF  
ENERGY**

Argonne National Laboratory is a U.S. Department of Energy  
laboratory managed by UChicago Argonne, LLC