

Updates and Verifications of the PROTEUS Suite in FY19

Nuclear Science and 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.

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 at OSTI.GOV (<http://www.osti.gov/>), a service of the U.S. Dept. of Energy's Office of Scientific and Technical Information

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
Phone: (800) 553-NTIS (6847) or (703)
605-6000 Fax: (703) 605-6900
Email: 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
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: reports@osti.gov

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.

Updates and Verifications of the PROTEUS Suite in FY19

prepared by
Changho Lee, Yeon Sang Jung
Nuclear Science and Engineering Division, Argonne National Laboratory

September 30, 2019

ABSTRACT

In FY19 efforts were made on updating and verifying the PROTEUS suite in terms of depletion and fuel cycle capabilities, updated MSR capabilities, improving code usability, and supporting simulation of micro reactors. The fuel reloading and shuffling schemes as well as the built-in depletion solver were implemented to add the fuel cycle capability to PROTEUS-NODAL. PERSENT can run with PROTEUS-NODAL which now produce the CCCC interface files. The MSR simulation capability was further updated by making PROTEUS-NODAL coupled with SAM via the MOOSE wrapper.

The high fidelity solvers of PROTEUS were updated to support the multiphysics simulation of SFRs and micro reactors. The coupled system of PROTEUS / FLUENT / ANLHTP was developed using Python-based external drivers, demonstrating the coupled simulation for small 3D test problems with heat pipe failure transients. In addition, the gamma transport capability was implemented to compute the detailed gamma distributions that determines the primary heat sources for the fuel assembly structure and non-fueled assemblies. Verification tests using fuel assembly and 2D core problems indicated very good agreement in heating distributions between PROTEUS and VARIANT / GAMSOR solutions.

For user support, the NEAMS Workbench was updated to include PROTEUS-NODAL and further extended to handle PROTEUS-MOC for which the support of Workbench is limited, instead requiring the off-line generation of mesh files and corresponding cross section sets. The utility tools of PROTEUS were also updated to support users to model and simulate complicated and non-conventional geometry cores.

TABLE OF CONTENTS

Abstract	i
Table of Contents	ii
List of Figures	iii
List of Tables.....	iv
1. Introduction.....	1
2. Current Status of the PROTEUS Suite	3
3. Updates in the NODAL Solver of PROTEUS	7
3.1 Fuel Cycle Analysis Capability.....	7
3.1.1 Built-in Depletion	7
3.1.2 Fuel Management	8
3.2 Connection with PERSENT.....	11
3.2.1 Methodology.....	11
3.2.2 Verification Tests	12
3.3 Improvements in the MSR Capabilities	16
3.3.1 Steady-state and Transient Modeling	16
3.3.2 Coupling with SAM.....	21
4. Updates in the High Fidelity Solvers of PROTEUS	24
4.1 Coupling with ANSYS.....	24
4.1.1 Updates in PROTEUS	24
4.1.2 Coupled Simulation	25
4.1.3 Demonstration	26
4.2 Coupled Neutron and Gamma Transport.....	28
4.2.1 Gamma Transport Capability	28
4.2.2 Verification Tests	30
5. PROTEUS User Support.....	34
5.1 NEAMS Workbench Integration.....	34
5.2 MeshTool Updates	34
5.3 Post-Processing Tool Updates.....	35
6. Conclusions	37
References	39

LIST OF FIGURES

Figure 2-1. Overview of the PROTEUS Suite.....	3
Figure 3-1. Comparisons of SFR Fuel Mixture Depletion using PROTEUS-NODAL and DIF3D/REBUS	8
Figure 3-2. Illustration of Fuel Management Scheme Available in PROTEUS-NODAL.....	9
Figure 3-3. Eigenvalue Behavior of SSFR Problem from Initial Cycle to 50 th Cycle.	10
Figure 3-4. Power Distribution of SSFR Problem at Initial and Equilibrium Core.....	10
Figure 3-5. Input and Interface Files Required for PERSENT	11
Figure 3-6. 2D and 3D Test Problems for PERSENT with PROTEUS-NODAL	12
Figure 3-7. Comparison of Adjoint Flux Solutions from DIF3D and PROTEUS-NODAL for the 2D Fast Reactor Benchmark Problem.....	13
Figure 3-8. Comparison of Adjoint Flux Solutions from DIF3D and PROTEUS-NODAL for the 3D Fast Reactor Benchmark Problem.....	14
Figure 3-9. Reactivity Losses of TRU-started MSFR Due to the Fuel Drifting.....	20
Figure 3-10. Pump Start up (left) and Pump Coast Down (right) Transients without Thermal Feedback	20
Figure 3-11. Power and average core temperature rise for 50 pcm reactivity insertion	21
Figure 3-12. Coupling of PROTEUS-NODAL and SAM using the MOOSE-based Wrapper TreeFrog.....	22
Figure 3-13. Power and Average Core Temperature Rise for 50 pcm Reactivity Insertion, Resulted from the Coupling of PROTEUS-NODAL and SAM	23
Figure 4-1. Demonstration of Data Exchange of PROTEUS-MOC Coupled Calculation.	24
Figure 4-2. Data Exchange of PROTEUS, FLUENT, and ANLHTP.....	25
Figure 4-3. Python Control of PROTEUS, FLUENT, and ANLHTP	26
Figure 4-4. Temperature Distribution of Multi-Heat Pipe Problem	27
Figure 4-5. Relative Power and Heat Pipe Temperature Change with Time for One Heat Pipe Failure Transient Problem.....	28
Figure 4-6. Workflow of Gamma Transport Calculation in PROTEUS Code	29
Figure 4-7. Comparisons of Gamma Spectrum Obtained Using MC2-3 and PROTEUS codes for SFR Fuel Mixture.....	31
Figure 4-8. Gamma Distributions for AFR-100 Fuel Assembly Problem.....	31
Figure 4-9. Comparison of Local Heating Distribution between Neutron Only and Neutron/Gamma Transport Calculations	32
Figure 4-10. Core Configuration of 2D SFR Problem for Testing Gamma Transport Capability	33
Figure 4-11. Neutron and Gamma Heating Distribution of PROTEUS-SN for 2D SFR Test Problem	33
Figure 4-12. Relative Differences of Heating Distributions from PROTEUS-SN and VARIANT-GAMSOR codes for the 2D SFR Test Problem	33
Figure 5-1. Structure of the PROTEUS Integration in the PyARC and the Workbench.	34
Figure 5-2. Symmetric Geometry Generation Using the Argonne Mesh Tool.....	35
Figure 5-3. Example of Updated Post-Processing Capability.....	36

LIST OF TABLES

Table 2-1. PROTEUS Capabilities and Features	5
Table 2-2. Software in the PROTEUS Suite Developed by Argonne.....	6
Table 2-3. Software Used for the PROTEUS Suite	6
Table 3-1. Comparison of Forward and Adjoint k-effective Solutions for Selected Test Problems.....	14
Table 3-2. PERSENT Outputs with DIF3D and PROTEUS-NODAL for the 2D Homogeneous Test Problem	15
Table 3-3. PERSENT Outputs with DIF3D and PROTEUS-NODAL for the 3D Test Problem	15
Table 3-4. Eigenvalues from PROTEUS-SAM and PROTEUS with Built-in T/H.....	23
Table 4-1. Neutron and Gamma Heating Result for AFR-100 Fuel Assembly Test Case	32

1. Introduction

Under the DOE-NE NEAMS program, the PROTEUS code [1], which is a high-fidelity capable deterministic neutron transport code based on finite element discretization of the domain, has been developed, as a key component of multiphysics simulation toolkit, to accurately simulate various non-LWR types of advanced reactors including sodium-cooled fast reactors (SFRs), molten salt reactors (MSRs), and high-temperature gas-cooled reactors (HTRs). This supports the NEAMS program missions which are to develop a simulation toolkit using the leading-edge computational methods accelerating the development and deployment of nuclear power technologies.

Focuses in FY19 were on updating and verifying the PROTEUS suite in terms of depletion and fuel cycle capabilities, MSR capabilities, improving code usability, and supporting simulation of micro reactors to which are recently brought attentions of industries and departments of Defense (DOD) and Energy (DOE) for use in powering remote communities, military and civilian.

The development of PROTEUS requires not only dealing with PROTEUS itself but also updating and developing many supporting codes for meshing, cross section, post-processing, and multiphysics coupling, which is discussed in details in Section 2. Therefore, significant efforts were made to update supporting tools in this fiscal year as well.

First of all, a fuel cycle analysis capability was implemented in PROTEUS-NODAL by combining the nodal transport and depletion solvers. A depletion solver was embedded in PROTEUS-NODAL by making use of the matrix exponential solver based on the Krylov subspace method [2] and the built-in decay chain composed of about major 300 nuclides. A fuel management option was also added in order to streamline a cycle depletion calculation with flexible fuel loading and shuffling. In addition, the code was updated to produce the CCCC interface files to be connected with PERSENT [3] which is the reactivity perturbation and sensitivity analysis code PERSENT that was developed under NEAMS and is being widely used in the fast reactor design practices including VTR and PGSFR.

The MSR capabilities of PROTEUS-NODAL was further updated by improving and verifying the transient capabilities with various MSR-typical transient scenarios. The SAM [4], a MOOSE-based system analysis code, was successfully coupled with PROTEUS-NODAL to more accurately simulate the thermal-fluid behaviors of MSR cores, which required developing the MOOSE wrapper named TreeFrog [5] for PROTEUS. Potentially, this approach can be easily adapted for coupling PROTEUS with other MOOSE tools.

A coupled neutron and gamma transport capability was developed in the high-fidelity solvers of PROTEUS to support multiphysics simulations using the NEAMS tools for fast reactor analysis [6]. The accurate heating distribution for other multiphysics components can be obtained from the neutron-gamma coupled calculation by taking into account the released energy from the neutron-gamma reactions which are major heat sources for non-fueled assemblies. Therefore, the gamma transport capability was implemented in the framework of the SN and MOC solvers of PROTEUS by extending the existing transport solvers. The heating capability of PROTEUS was updated to

compute the neutron and gamma contributions explicitly using the KERMA factors available in the gamma interaction data.

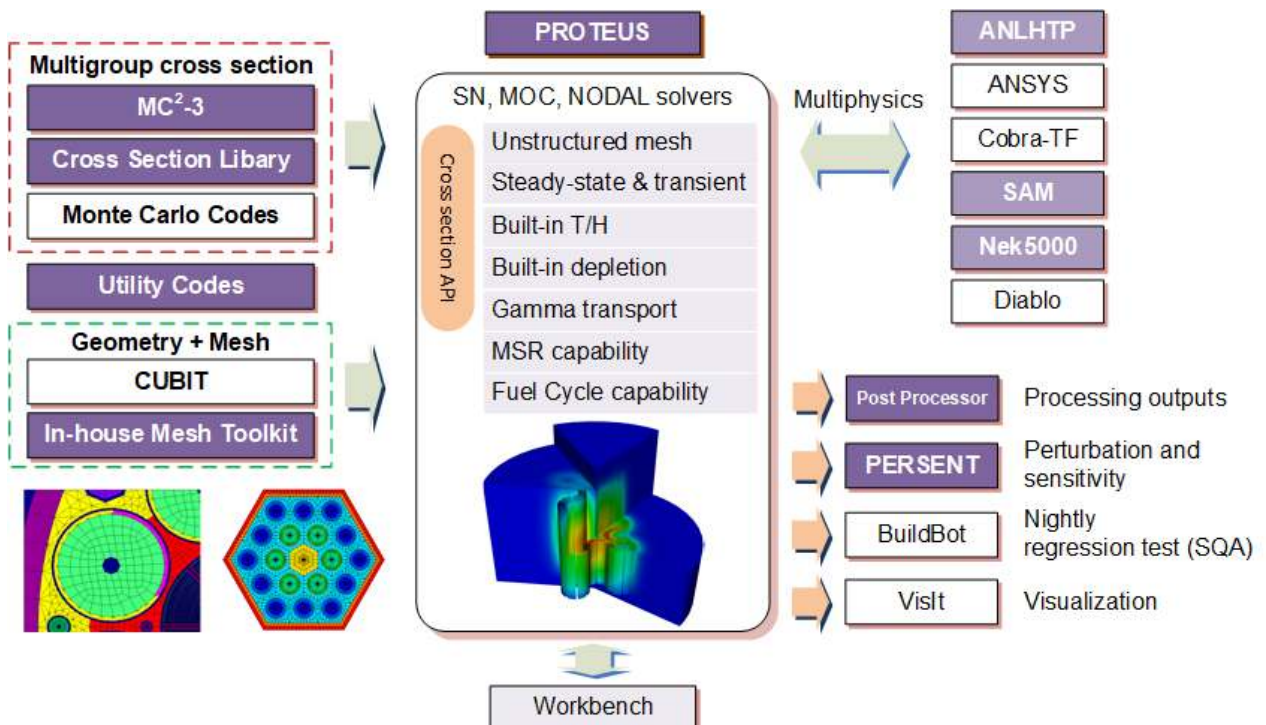
As an effort of user support, the NEAMS Workbench was updated to deal with PROTEUS by initially making use of the PyARC framework [7,8] developed for DIF3D since there are similarities between PROTEUS-NODAL and DIF3D [9] and by extending PyPROTEUS [10] for the high fidelity solvers of PROTEUS. Moreover, the mesh [11] and post-processing tools of PROTEUS were also extended to support micro reactor analysis and other applications which have complicated and non-conventional geometries.

Section 2 summarizes the current status of the PROTEUS suite including all relevant tools as well as PROTEUS. Sections 3 and 4 present the updates of the NODAL and high fidelity solvers of PROTEUS, respectively. In this sections, efforts on multiphysics simulation capabilities are discussed as well. Section 5 discusses improvements made in user support capabilities. Conclusions are made in Section 6.

2. Current Status of the PROTEUS Suite

The PROTEUS code is a high-fidelity capable neutron transport code based on finite element discretization of the domain. The S_N and MOC transport solvers are available in PROTEUS to solve heterogeneous geometry problems with almost no or minimal geometrical approximations. The nodal transport solver based on homogenized assemblies and structured geometry was also implemented in PROTEUS to provide a conventional-fidelity level of solutions in a consistent framework for use in rapid design application. In the NODAL solver, two methodology options are available: P_N and Simplified P_N (SP_N). The P_N approach is basically identical to the methodology used in DIF3D-VARIANT [12] although the release version currently handles diffusion theory on Cartesian, hexagonal, and triangular-z grids, being extended to the higher order solver. For the SP_N approach [13], a transverse integrated nodal methodology was built on the hexagonal grid model utilizing up to a SP_3 approximation.

All the three solvers are able to solve steady-state and transient problems with the built-in thermal fluid calculation capability. The gamma transport calculation is available to accurately solve for power distributions. Especially in the NODAL solver, the MSR capability was implemented along with relevant built-in thermal fluid modules in order to solve flowing fuel core problems, accounting for the redistribution of delayed neutron precursor concentrations due to fuel flow velocities inside and outside the core.



(The purple-colored boxes denote the codes developed by Argonne)

Figure 2-1. Overview of the PROTEUS Suite

Recently, the fuel cycle capability as well as depletion was implemented in the NODAL solver to meet the needs for actual reactor core design and analysis. In addition, the NODAL solver is able to generate the CCC interface files in order to run PERSENT for reactivity perturbation and sensitivity analysis.

Cross sections for PROTEUS can be generated in the ISOTXS format using MC²-3 [14] and Monte Carlo codes (Serpent [15] and OpenMC [16]). With the Monte Carlo code outputs, the GenISOTXS code [17] produces ISOTXS files which can be combined to produce the ISOPAR format with state parameters in terms of temperature, burnup, control rod, void fraction, etc. On the other hand, PROTEUS is able to generate multigroup cross sections on the fly accounting for the problem composition, geometry, and temperature conditions using the pre-calculated cross section library which basically includes the cross section tables as a function of background cross sections and temperature.

Geometry and mesh are generated using CUBIT [18] standalone or a combination of CUBIT and the Argonne mesh toolkit [11]. For conventional hexagonal and Cartesian geometry problems, the Argonne mesh toolkit is able to easily generate meshes with text-type keyword inputs. Hexahedral, quadrilateral, and tetrahedral meshes or a combination of those meshes can be used for PROTEUS as long as meshes are conformal. The SN solver uses a 3D mesh, while the MOC solver uses a 2D-based extruded mesh which is constructed and interpreted to 3D inside the code using the 3D information given from the material assignment input.

Multiphysics simulations with other physics tools such as Nek5000 [19], Diablo[20], Cobra-TF (CTF) [21], SAM [4], and ANSYS [22] were attempted by developing interface routines for coupling (via FORTRAN or Python). Currently, Nek5000 and Diablo work with PROTEUS-SN, Cobra-TF and ANSYS can be coupled with PROTEUS-MOC, and SAM (a MOOSE-based code) works with PROTEUS-NODAL. In particular, the MOOSE wrapper named TreeFrog was developed to connect PROTEUS with SAM. The coupling approach that we employed for the coupling of SAM and PROTEUS-NODAL will be extended for coupling with other MOOSE-based codes in future.

Post processing tools were also developed to process the PROTEUS outputs, editing out data of user interest and plotting them using the visualization tools such as VisIt [23] and ParaView [24]. Recently, the NEAMS Workbench was updated with PyPROTEUS [10] which helps users create PROTEUS inputs, run the code, and post-process code outputs for the high fidelity solvers as well as the NODAL solver of PROTEUS.

PROTEUS has been verified and validated using various numerical and experimental benchmark problems including fast reactors (FRs), pressurized water reactors (PWRs), high temperature reactors (HTRs), and molten salt reactors (MSRs). Recently, micro reactors including MegaPower heat-pipe-cooled reactor [25,26] and Holos-Gen HTR [27] were simulated using PROTEUS with other physics tools.

The current status of PROTEUS capabilities and features discussed above are summarized in Table 2-1. In order to support PROTEUS, many codes in terms of cross section generation, mesh generation, data conversion, post-processing data, and multiphysics simulation have been developed, as listed in Table 2-2. The user manuals of all the codes were developed as well. All the codes have been developed and managed using the SVN or Git software version control system. Table 2-3 lists the computer codes what were used for PROTEUS.

Table 2-1. PROTEUS Capabilities and Features

Feature	SN	MOC	NODAL
Solver	SS, TR, Adjoint (2 nd order)	SS, TR, Adjoint	SS, TR, Adjoint PN (P1), SP3
Gamma transport	Yes	Yes	No
Acceleration	DSA	CMFD	CMFD
Mesh	3D hex/quad/tri (mixed)	Extruded mesh hex/quad/tri (mixed)	Structured PN (Cartesian, Hex, Tri) SP3 (Hex)
Built-in TH	Not tested yet	Yes	Yes
Built-in depletion	No (ORIGEN- incomplete)	No	Yes
MSR capability	No	No	Yes
Cross section	CSAPI ISOTXS / ISOPAR	CSAPI ISOTXS / ISOPAR	ISOTXS / ISOPAR
Multiphysics	Nek5000 Diablo (BISON- incomplete)	CTF ANSYS	SAM (via TreeFrog)
Application	ATR, SFR, etc.	TREAT, RCF, MSR, Micro reactors, VERA, VHTR, C5G7-TD, etc.	SFR, MSR, C5G7, etc.
Fuel cycle capability	No	No	Yes
Link to PERSENT	No	No	Yes
Link to Workbench	Not tested yet	Yes	Yes
Link to VisIt, ParaView	Yes	Yes	Yes
Post processor	No	Yes	N/A
Develop. Repository	SVN/Git	SVN/Git	Git
User manual	Yes	Yes	Yes

Table 2-2. Software in the PROTEUS Suite Developed by Argonne

Software	Description
PROTEUS (CSAPI)	High-fidelity-capable neutron transport calculation for SS and TR (Cross section API which works for SN and MOC)
MC ² -3	Multigroup cross section generation (being used for VTR design)
PERSENT	Reactivity perturbation and sensitivity analysis (being used for VTR design)
GenISOTXS	ISOTXS generation using Serpent or OpenMC outputs
GenISOPAR	ISOPAR generation with ISOTXS files of different states
GenCSLIB	Cross section library generation for CSAPI
MeshTool	Mesh generation toolkit for Cartesian and hex geometries
MOCEX-PPT	Post processing of PROTEUS-MOC output
Exodus II convertor	Conversion of Exodus II file to PROTEUS mesh format
PyPROTEUS	PROTEUS user interface for NEAMS Workbench
TreeFrog	MOOSE interface between PROTEUS and SAM
ANLHTP	Heat pipe performance analysis (resurrection of the developed in the 1980s)
Python scripts	Support PROTEUS MOC input generation, PROTEUS / ANSYS coupling

Table 2-3. Software Used for the PROTEUS Suite

Software	Description
VisIt	Visualization (LLNL)
Paraview	Visualization (Kitware, open-source)
CUBIT	Mesh generation (SNL)
Serpent	Monte Carlo neutron transport (VTT)
OpenMC	Monte Carlo neutron transport (ANL/MIT)
Nek5000	CFD (ANL)
Diablo	Structural mechanics (LLNL)
ANSYS	Structural mechanics, CFD
SAM	T/H and system analysis (ANL)
CTF	COBRA-TF sub-channel analysis
ORIGEN	Depletion (ORNL). Connection to PROTEUS is incomplete
Warthog	MOOSE interface (ORNL) between PROTEUS and BISON, incomplete

3. Updates in the NODAL Solver of PROTEUS

3.1 Fuel Cycle Analysis Capability

In FY19, the fuel cycle analysis capability was developed in PROTEUS-NODAL to extend its application domains so as to predict the evolution of core characteristics during fuel cycles. For performing the fuel cycle calculation, PROTEUS-NODAL was updated to include depletion, fuel reloading, and shuffling functions and update the transport solvers accordingly. The details of these new functionalities are discussed in this section.

3.1.1 Built-in Depletion

For given core configuration and cycle length, a depletion calculation computes the core composition changes as a function of time by taking into account the region-wise fluxes and associated reaction rates that vary with time as well. Under the assumption that fluxes and reaction rates remain unchanged within the time step, the following Bateman equation is solved for each depletion time step:

$$\frac{\partial}{\partial t} \mathbf{N}(t) = \mathbf{A}(\phi_n, \sigma_n, \lambda_n) \mathbf{N}(t), t \in [t_n, t_{n+1}], \quad (3-1)$$

where n is the index of time step, and N and A are the number density vector and transmutation matrix, respectively. The indices for energy group and regions are omitted for brevity. The transmutation matrix is identical to the conventional one that contains the generation and removal of isotopes through radioactive decay and neutron reactions as:

$$\begin{aligned} A_{i,j}(t_n) &= \sum_x \gamma_{ij}^x \sigma_j^x \phi_n + \gamma_{ij} \lambda_j \quad (i \neq j), \\ A_{i,i}(t_n) &= -\sigma_i^a \phi_n - \lambda_i, \end{aligned} \quad (3-2)$$

where i and j are indices for isotope of interest and associated isotopes through a reaction or a decay, respectively, and x is the index for reaction. Using the fluxes and reaction rates from transport calculations embedded to depletion solver, the transmutation matrix can be set up, and Eq. (3-1) can be solved by using the well-established matrix exponential solver as:

$$\mathbf{N}(t_n) = \exp(\mathbf{A}_n \Delta t_n) \mathbf{N}(t_{n-1}). \quad (3-3)$$

By computing Eq. (3-3) for each region as marching the time steps, the changes of fuel compositions can be computed as a function of time from the beginning of cycles (BOCs) to end of cycles (EOCs). The neutron transport calculations are also incorporated in order to obtain the core characteristics such as reactivity and to update reaction rates of each region.

In the implementation for PROTEUS-NODAL, the transmutation matrices were constructed based upon the pre-defined decay chain data that are composed of around 300 major isotopes and key reactions and decays. The mapping between isotopes defined in the ISOTXS file and the decay chain can be automatically prepared, which removes a tedious preparation work of user-defined ones. The resulting matrix is solved using the matrix exponential solver based on the Krylov

subspace method. The details of this numerical algorithm can be found in Reference [2]. The depletion and transport solvers were connected by employing the predictor-corrector scheme.

For verifying the newly implemented depletion solver, a simple mixture based on the typical SFR fuel assemblies was depleted using PROTEUS-NODAL and DIF3D/REBUS. Figure 3-1 compares the number density changes of U-235 and Pu-239 with time that were obtained from both calculations. The changes of number densities over depletion agreed well within only small differences due to the decay chain differences. This results indicate that the built-in depletion solver was correctly implemented.

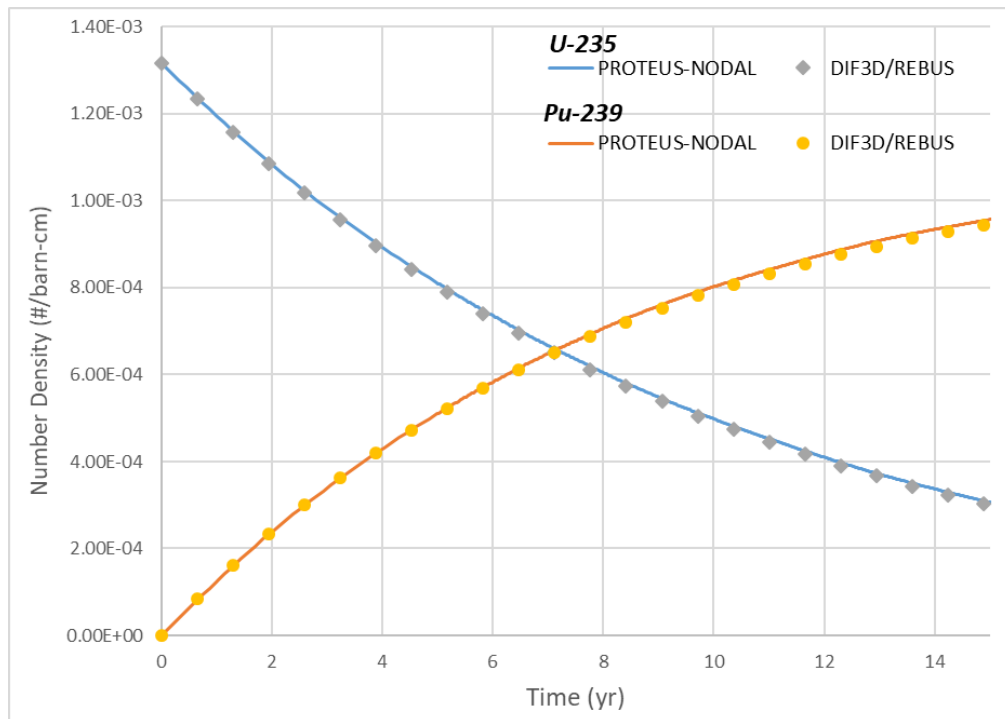


Figure 3-1. Comparisons of SFR Fuel Mixture Depletion using PROTEUS-NODAL and DIF3D/REBUS

3.1.2 Fuel Management

A fuel management option of fuel cycle analysis codes allows a user to perform sophisticated cycle depletion calculations, which is an essential feature for fuel cycle analysis for fast reactors. The fuel management option was implemented in the PROTEUS-NODAL solver and connected to the built-in depletion capability in order to complete the implementation of the fuel cycle analysis capability.

The fuel management options implemented in PROTEUS-NODAL can handle the fuel shuffling and reloading schemes. As shown in Figure 3-2(a), the reloading scheme enables to remove and insert fuel assemblies at each BOC. Movements of fuel assemblies can be modelled through the

fuel shuffling scheme. An example of fuel shuffling is illustrated in Figure 3-2(b). For realizing these two fuel management schemes, the PROTEUS-NODAL code uses the core inventory database file that contains depleted fuel assembly information of the previous fuel cycles. This database is updated at each EOC. It is utilized in the subsequent fuel cycle to configure the initial core inventories built upon a user-defined fuel management scheme that is provided through an additional input file.

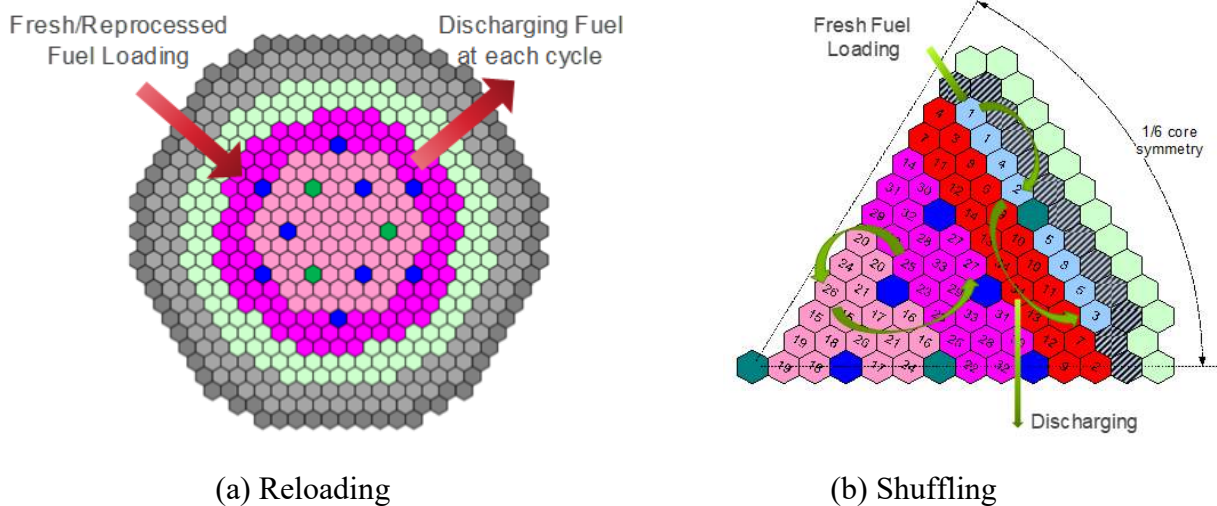


Figure 3-2. Illustration of Fuel Management Scheme Available in PROTEUS-NODAL

Preliminary tests were performed to verify the fuel cycle analysis capability of PROTEUS-NODAL using the once-through sustainable SFR core design (SSFR) which has the complex 34-batch fuel shuffling scheme [28]. This core is a breed-and-burn core design in which the fresh fuel assembly with natural uranium are loaded on the out-most active core region and is burned for 34 cycles with shuffling. For the initial core configuration, the fuel assemblies with three different enriched uranium contents are configured in the core, but the subsequent cycles maintain the criticality by replacing the burnt fuels with fresh natural uranium fuels. After initial 34 cycles, the entire initially-loaded enriched fuel assemblies are discharged, and the core ends up with fresh and burnt natural uranium assemblies. The detail of core design can be found in Reference [28].

In the calculations, the 34-batch fuel shuffling scheme shown in Figure 3-2(b) was explicitly modelled through the fuel management option of the NODAL solver, and the 50-cycle depletion calculations were performed to obtain the equilibrium core configuration. The obtained results are summarized in Figure 3-3, showing the time evolution of the eigenvalue at BOC and EOC. In the initial 20 cycles, the notable changes in eigenvalue are observed as the enriched fuel assemblies are shuffled and discharged, which can have significant impact on the core reactivity. After 20 cycles, the burnt natural uranium assemblies are filled with most of the active core locations, which leads to the smooth transient to equilibrium core status. In this calculation, the core configuration almost reached the equilibrium core after 40 cycles. The assembly-wise power distributions at the initial

fresh and equilibrium cores are shown in Figure 3-4. The significant changes are indicated in the power distribution due to the transition from initial to equilibrium cores.

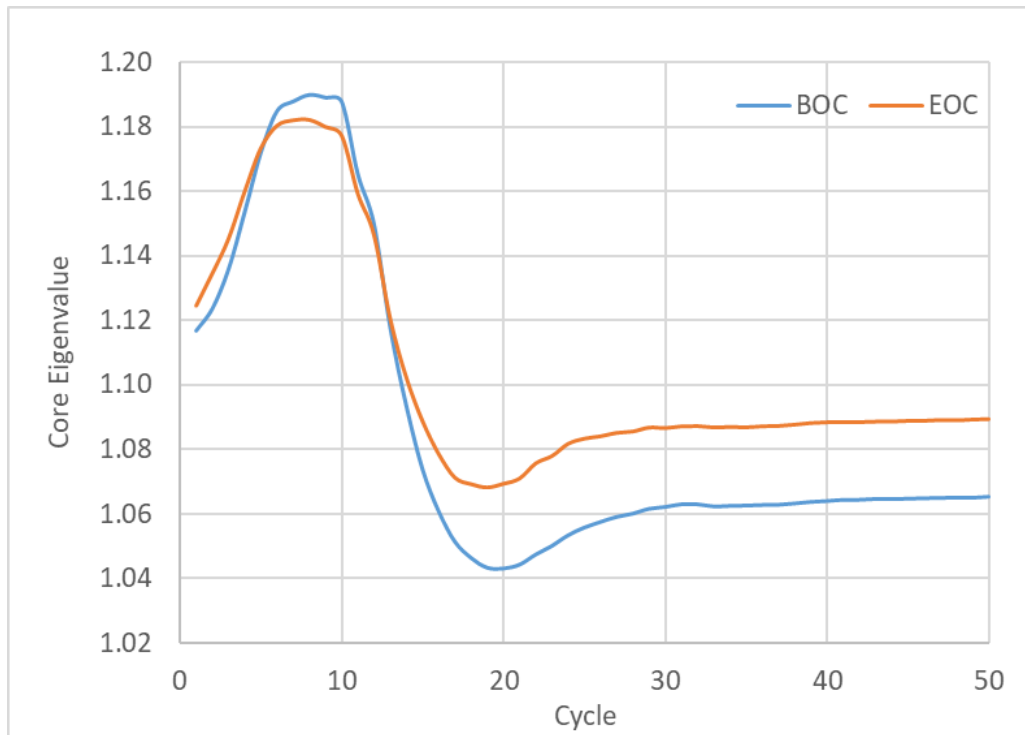


Figure 3-3. Eigenvalue Behavior of SSFR Problem from Initial Cycle to 50th Cycle.

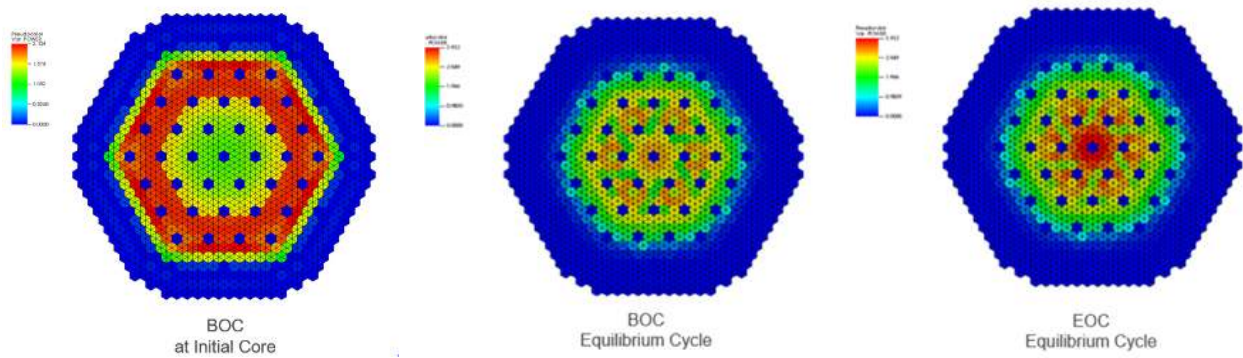


Figure 3-4. Power Distribution of SSFR Problem at Initial and Equilibrium Core

3.2 Connection with PERSENT

3.2.1 Methodology

The PERSENT code [3] is a reactivity perturbation and sensitivity analysis code based upon the variational nodal transport option of DIF3D-VARIANT which combines spherical harmonic angular trial functions with orthogonal polynomial spatial trial functions within each node. For perturbation theory problems, PERSENT generates and views the detailed spatial contributions to any given reactivity worth or kinetics parameter of interest. For sensitivity and uncertainty calculations, PERSENT generates the desired results in a fraction of the computational effort required if using a finite difference method for obtaining the results.

For reactivity perturbation calculations, PERSENT requires the data files below in the CCCC format. Most of the interface files are outputs of DIF3D, except for ISOTXS and DLAYXS which are inputs of the code.

- GEODST : geometry description
- NDXSFR : nuclide density, data, cross section referencing
- LABELS : region and composition labels, area data, half heights, nuclide set labels, alias zone labels, control-rod model data, burnup dependent cross section data
- ZNATDN : zone atomic densities of nuclides
- COMPXS : macroscopic composition cross sections
- NHFLUX : forward nodal flux-moments and interface partial currents
- NAFLUX : adjoint nodal flux-moments and interface partial currents
- ISOTXS : multigroup neutron cross section
- DLAYXS : multigroup delayed neutron precursor data

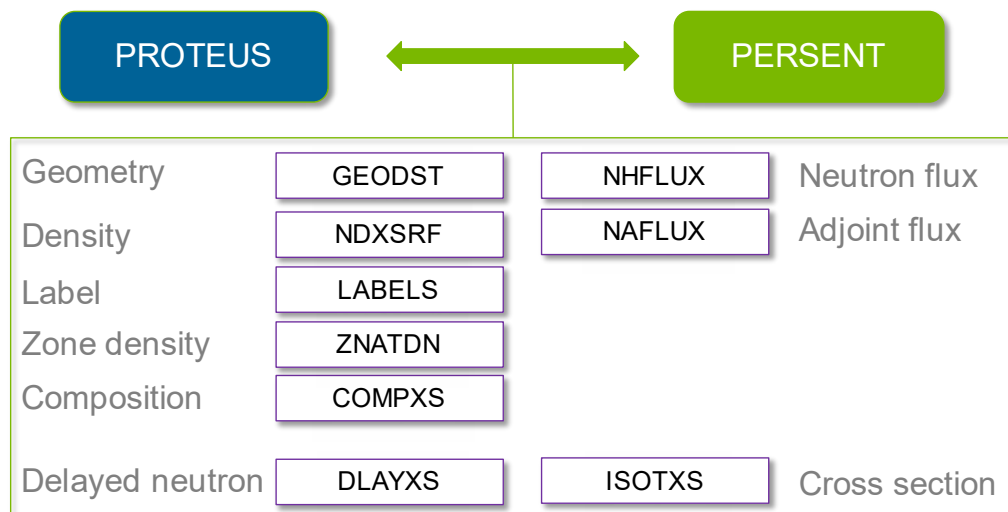


Figure 3-5. Input and Interface Files Required for PERSENT

Rather than updating PERSENT which keeps updated to support the conventional tools including DIF3D, we decided that it would be more reasonable to update PROTEUS-NODAL to produce the CCCC-format interface files.

To solve for the effective prompt neutron life time and effective delayed neutron fraction in PERSENT, the adjoint flux solution is required. Therefore, the PROTEUS-NODAL code was updated to produce the adjoint solution. The 3D steady-state P_1 equation with the transport approximation is given as:

$$-D_g \nabla^2 \phi_g(r) + \Sigma_{rg} \phi_g(r) = \lambda \chi_g \sum_{g'=1}^G v \Sigma_{fg'} \phi_{g'}(r) + \sum_{g' \neq g}^G \Sigma_{sg' \rightarrow g} \phi_{g'}(r), \quad (3-4)$$

where $\phi_g(r)$ denotes the forward neutron flux for group g with $g = 1, \dots, G$ and $\Sigma_{\alpha g}(r)$ is a standard form of removal, fission or scattering cross section. The adjoint operator becomes the transpose of the direct operator, leading to:

$$-D_g \nabla^2 \phi_g^*(r) + \Sigma_{rg} \phi_g^*(r) = \lambda v \Sigma_{fg} \sum_{g'=1}^G \chi_{g'} \phi_{g'}^*(r) + \sum_{g' \neq g}^G \Sigma_{sg \rightarrow g'} \phi_{g'}^*(r), \quad (3-5)$$

where $\phi_g^*(r)$ denotes the adjoint neutron flux. Basically, the same algorithm used to solve the forward equation can be applied for the adjoint equation by transposing the source term in the right hand side of the equation.

3.2.2 Verification Tests

The PROTEUS-NODAL code in connection with PERSENT was tested using multiple test problems. Chosen were two fast reactor problems with 120-degree periodic boundary conditions: a 2D core with 3 energy groups and a 3D core with 21 energy groups, as illustrated in Figure 3-6. For the 3D core, note that the axial height is 480.2 cm with the active core height of ~114.9 cm and top and bottom reflectors.

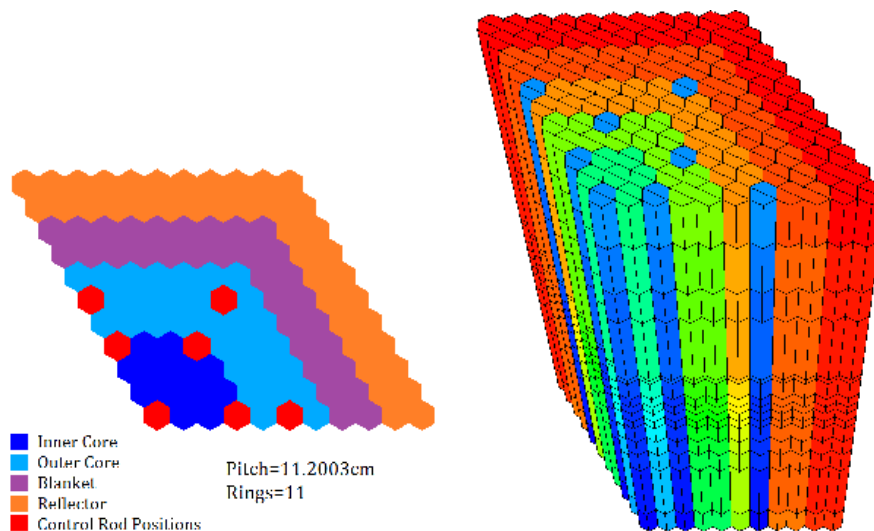


Figure 3-6. 2D and 3D Test Problems for PERSENT with PROTEUS-NODAL

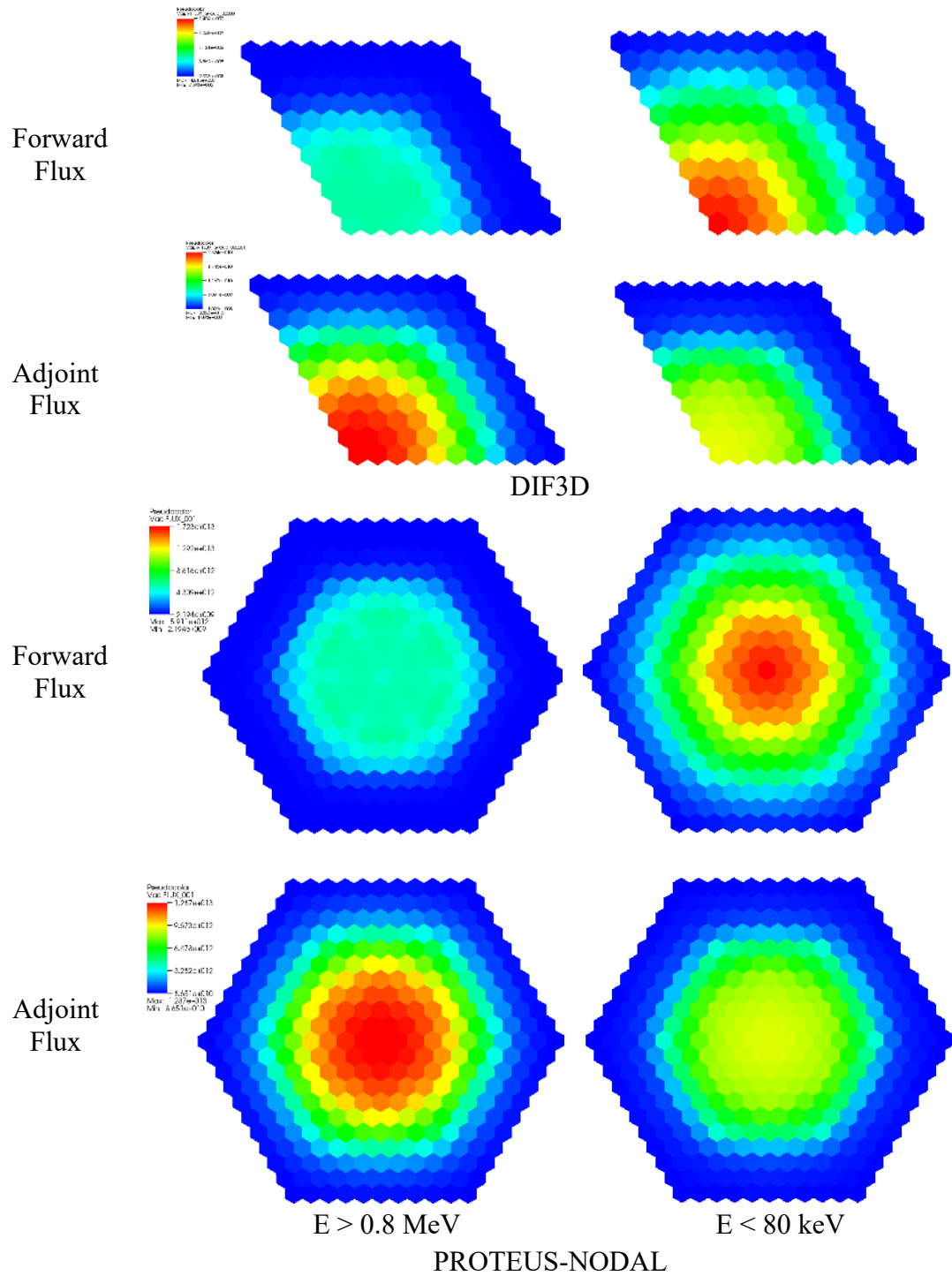


Figure 3-7. Comparison of Adjoint Flux Solutions from DIF3D and PROTEUS-NODAL for the 2D Fast Reactor Benchmark Problem

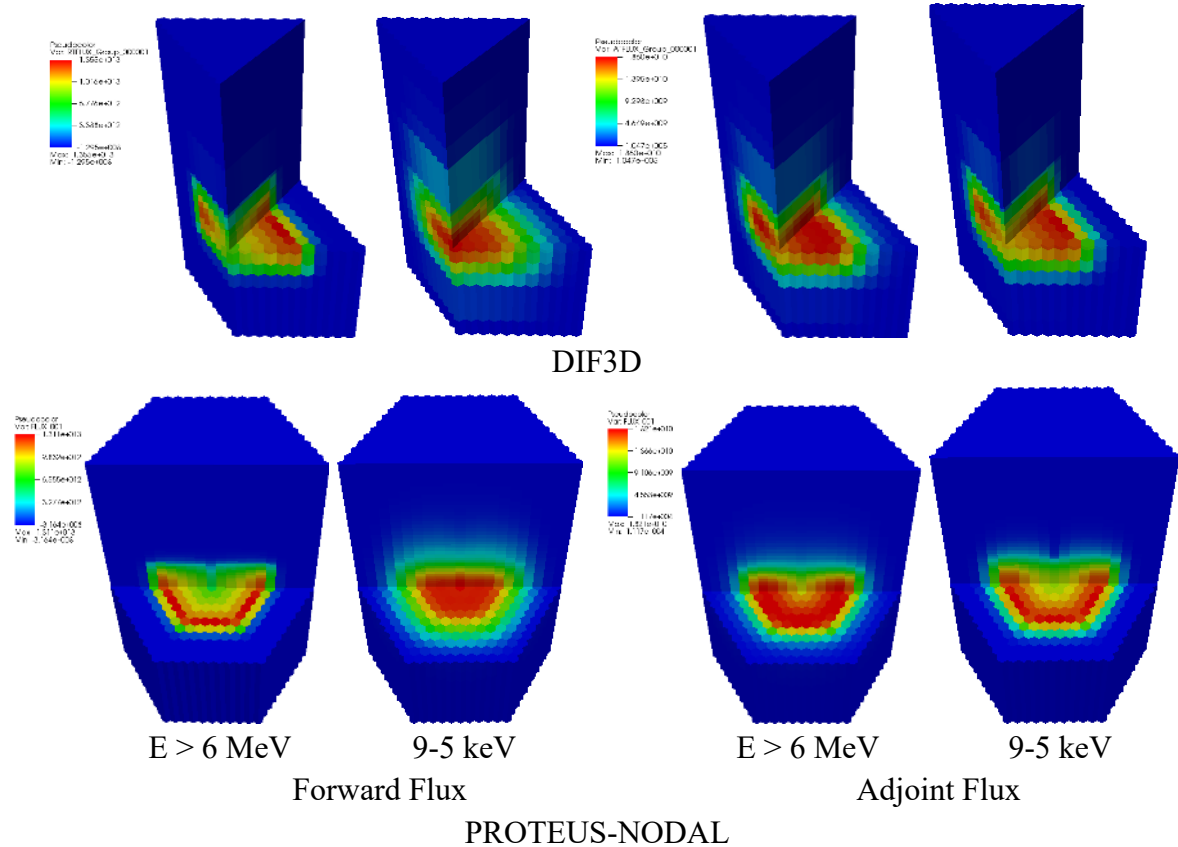


Figure 3-8. Comparison of Adjoint Flux Solutions from DIF3D and PROTEUS-NODAL for the 3D Fast Reactor Benchmark Problem

Table 3-1. Comparison of Forward and Adjoint k-effective Solutions for Selected Test Problems

Test Problem	k-effective		Δk
	Forward	Adjoint	
Hex-60 2D	1.3653637	1.3653880	0.000024
Hex-360 2D	1.3653558	1.3653788	0.000023
Hex-120 2D	1.1635450	1.1635606	0.000016
Hex-60 3D	1.1149954	1.1149214	-0.000074

First of all, the k-effective solutions were compared between forward and adjoint flux calculations of PROTEUS-NODAL, confirming that they were very close to each other, as shown in Table 3-1. In addition, the group-wise adjoint fluxes were compared between DIF3D and PROTEUS-NODAL in Figure 3-7 and Figure 3-8 for 1/3 hex 2D and 3D cores, respectively, showing the similar trends of adjoint fluxes for those cores. The adjoint flux profiles appear to be reasonable, showing relatively larger values at the core center and higher energy. Since eigenvalues from forward and adjoint solutions should basically be identical, a further refinement study will be conducted to find out sources of the differences even though small.

An initial verification test was performed using a simple homogeneous problem with the vacuum boundary condition and three energy groups, which is composed of five isotopes (U-238, Pu-239, O-16, Fe-56, and Na-23), to ensure that PERSENT works correctly with the CCCC interface files produced from PROTEUS-NODAL. Then, the 3D fast reactor test problem discussed above was tested using PERSENT with DIF3D and PROTEUS-NODAL. The effective delayed neutron fractions, beta, and point kinetics parameters resulted from PERSENT with two different flux solvers are listed in Table 3-2 and Table 3-3, showing overall good agreements in the solutions between the two different solution paths. For the 3D test problem, however, we could see a noticeable difference of 10% for the effective betas for U-238 resulting in a difference of 3.6 % in the total effective beta of the core. Further verification tests with various test problems for all reactivity and sensitivity options of PERSENT are in progress.

Table 3-2. PERSENT Outputs with DIF3D and PROTEUS-NODAL for the 2D Homogeneous Test Problem

Parameter			DIF3D	NODAL
Eigenvalue			1.289995	1.290030
Beta	Pu-239	Family 1	9.63804E-04	9.63820E-04
		Family 2	8.72220E-04	8.72210E-04
	U-238	Family 1	4.34051E-04	4.34233E-04
		Family 2	9.34047E-04	9.34413E-04
	Point Kinetics	Family 1	1.39786E-03	1.39805E-03
		Family 2	1.80627E-03	1.80662E-03
Generation Time			4.28996E-07	4.29019E-07
Prompt Lifetime			5.53403E-07	5.53447E-07

Table 3-3. PERSENT Outputs with DIF3D and PROTEUS-NODAL for the 3D Test Problem

Parameter			DIF3D	NODAL
Eigenvalue			1.055872	1.054458
Beta	Pu-239	Family 1	9.53777E-04	9.68904E-04
		Family 2	8.60983E-04	8.78929E-04
	U-238	Family 1	4.53814E-04	4.04911E-04
		Family 2	9.75970E-04	8.73513E-04
	Point Kinetics	Family 1	1.40759E-03	1.37381E-03
		Family 2	1.80627E-03	1.75244E-03
	Total			3.24454E-03

3.3 Improvements in the MSR Capabilities

3.3.1 Steady-state and Transient Modeling

Steady-state and Transient Solution Capability

The MSR fuel is dissolved into the liquid salt that is circulated throughout the whole primary system. Some of the delayed neutron precursors generated in the core may decay in a position of low importance or even out of the core, which will reduce the effective delayed neutron fraction of the reactor. As a result, the multiplication factor of MSR will be less than of a reactor with solid conventional fuel, when all the other conditions are the same.

In the modeling of MSR, the motion of the fuel must be considered in solving the delayed neutron precursor equations to account for their decay outside the core and their distribution in the core, which is totally different from the stationary fuel. The steady-state neutron diffusion equation for MSR can be written:

$$-\nabla \cdot D(\vec{r})\nabla \phi_g(\vec{r}) + \Sigma_{tg}(\vec{r})\phi_g(\vec{r}) = \sum_{g'=1}^G \Sigma_{sg' \rightarrow g}(\vec{r})\phi_{g'}(\vec{r}) + \lambda \chi_{pg}(\vec{r}) \sum_{g'=1}^G \nu_p \Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r}) + \sum_{k=1}^K \chi_{dkg}(\vec{r})\lambda_k C_k(\vec{r}), \quad g = 1, 2, \dots, G \quad (3-6)$$

$$\nabla \cdot [\vec{u}(\vec{r})C_k(\vec{r})] + \lambda_k C_k(\vec{r}) = \lambda \sum_{g'=1}^G \nu_{dk} \Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r}), \quad k = 1, 2, \dots, K \quad (3-7)$$

with the boundary condition below:

$$C_k(r, 0) = \frac{\int_{A_{out}} dAu(r, H)C_k(r, H)e^{-\lambda_k \tau}}{\int_{A_{in}} dAu(r, 0)}. \quad (3-8)$$

For a stationary fuel, with the diffusion theory approximation, the kinetics equations for prompt neutrons and delayed neutron precursor concentrations can be written as:

$$\frac{1}{\nu_g} \frac{\partial}{\partial t} \phi_g(\vec{r}, t) - \nabla \cdot D(\vec{r}, t)\nabla \phi_g(\vec{r}, t) + \Sigma_{tg}(\vec{r}, t)\phi_g(\vec{r}, t) = q_g(\vec{r}, t), \quad g = 1, 2, \dots, G. \quad (3-9)$$

$$\frac{\partial}{\partial t} C_k(\vec{r}, t) + \lambda_k C_k(\vec{r}, t) = \lambda \psi_k(\vec{r}, t), \quad k = 1, 2, \dots, K$$

where

$$q_g(\vec{r}, t) = \sum_{g'=1}^G \Sigma_{sg' \rightarrow g}(\vec{r}, t)\phi_{g'}(\vec{r}, t) + \lambda \chi_{pg}(\vec{r}, t) \sum_{g'=1}^G \nu_p \Sigma_{fg'}(\vec{r}, t)\phi_{g'}(\vec{r}, t) + \lambda \sum_{k=1}^K \chi_{dkg}(\vec{r}, t)\lambda_k C_k(\vec{r}, t), \quad (3-10)$$

$$\psi_k(\vec{r}, t) = \sum_{g'=1}^G v_{dk} \Sigma_{fg'}(\vec{r}, t) \phi_{g'}(\vec{r}, t)$$

The time derivative term in the neutron flux equation is discretized using the implicit Euler method, and it is solved with the transient fixed source problem (TFSP) approach at each time step:

$$\frac{\phi_g^n(\vec{r}) - \phi_g^{n-1}(\vec{r})}{v_g \Delta t_n} - \nabla \cdot D_g^n(\vec{r}) \nabla \phi_g^n(\vec{r}) + \Sigma_{fg}^n(\vec{r}) \phi_g^n(\vec{r}) = q_g^n(\vec{r}), \quad (3-11)$$

where n is the time step index, and Δt_n is the time step size.

The time dependency of the precursor concentration equation is typically solved by applying the second-order analytic integration method [29]. Based on this approach, the delayed fission source distribution is assumed to have a quadratic shape on time using the delayed fission sources at the current and the two previous time nodes. However, for the flowing fuel case this approach is not valid because of the flow convection term in the delayed neutron precursor equation. Instead, the time-dependent delayed precursor equation Eq. (3-10) should be solved analytically. In PROTEUS-NODAL the semi-analytical scheme was implemented based on discretizing the time-dependent precursor equation and finding the analytical solution of the ordinary differential equations for the precursor concentrations at the end of each time step.

Thermal-fluid Analysis Capability

In order to account for thermal feedback effects, a single-phase, parallel-channel thermal-hydraulics model was added to PROTEUS-NODAL. In this model, the thermal-fluidic behavior of the core is described by representing the core with one-dimensional parallel channels. The total flow rate is split among the parallel flow channels to satisfy the equal pressure drop boundary conditions. For a given flow rate, the thermal-hydraulics calculation for each channel is performed as described below.

Ignoring the axial heat conduction and the shear forces due to velocity gradients in the fuel salt, the mass, momentum, and energy balance equations for one-dimensional flow in a vertical channel can be written as [30]:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial z} = 0, \quad (3-12)$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial}{\partial z}(\rho u^2) = -\frac{\partial P}{\partial z} - \rho g - \left(\frac{\partial P}{\partial z} \right)_{fric}, \quad (3-13)$$

$$\frac{\partial(\rho h)}{\partial t} + \frac{\partial(\rho u h)}{\partial z} = q_{salt}^m, \quad (3-14)$$

where ρ , u , h , and P are the density, velocity, temperature, and pressure of the fuel salt, respectively, and q_{salt}^m is the heat source produced in the fuel salt. The parameter g is the gravitational acceleration. The friction pressure gradient with the wall shear stress and the momentum flux is accounted for

as well. The three balance equations are solved directly to determine T/H solutions.

Heat Exchanger Modeling Capability

In the reactor system, the fuel salt exiting the core flows through the primary system before returning to the core inlet. To describe the core behavior during transients more accurately, the heat exchanger in the primary system is also included in the thermal-hydraulics model of molten salt fast reactor (MSFR) [31]. The heat exchanger has been modeled with a lumped parameter approach to simulate the heat removal by the intermediate loop, considering it as one node. The fuel salt temperatures at the core inlet and outlet are directly coupled with the fuel salt temperatures at the heat exchanger outlet and inlet, respectively. The other components of the primary loop such as primary pumps, inlet and outlet plenums are not considered in this model.

Assuming the variation of temperatures of all materials in the heat exchanger during a transient is proportional to the steady-state temperatures, the heat exchanger can be characterized by the average fuel salt temperature \bar{T}_{HX} , of which variation is determined by:

$$C_{HX} \frac{d\bar{T}_{HX}}{dt} = \dot{m}_{HX} c_p (T_{in}^{HX} - T_{out}^{HX}) + \dot{Q}_{decay} - \dot{Q}_{HX}, \quad (3-15)$$

where $C_{HX} = m_{HX} c_p$ is the heat capacity of the heat exchanger, c_p is the specific heat of the fuel salt at the corresponding temperature, and \dot{m}_{HX} and m_{HX} are the fuel salt mass flow rate and the mass in the heat exchanger, respectively. T_{in}^{HX} and T_{out}^{HX} are the inlet and outlet temperatures of the heat exchanger of the of the fuel salt, \dot{Q}_{decay} is the decay heat generated by the fuel salt in the heat exchanger, and \dot{Q}_{HX} is the heat removal rate by the intermediate fluid in the heat exchanger.

Decay Heat Modeling Capability

In MSR the decay heat equation is different from the case of the solid fuel, where it should account for the flow of the fuel salt outside the core that's lead to the release of the decay heat outside the core due to the radioactive decay of fission products. A simple decay heat model is included in PROTEUS-NODAL code based on gathering all the decay heat generated by the isotopes into K groups, and solving the following equation in the core:

$$\frac{\partial}{\partial t} h_k(r, z, t) + \frac{\partial}{\partial z} [u(z) h_k(r, z, t)] + \lambda_k h_k(r, z, t) = f_k \lambda_k Q_{tot}(r, z, t), \quad k = 1, 2, \dots, K. \quad (3-16)$$

where h_k defined as the precursor concentration times energy release for decay heat in group k . f_k , and λ_k , are decay heat fraction and decay heat constant for group k , respectively, averaged over all the total groups and isotopes of the fuel salt. Q_{tot} is the total volumetric heat source produced in the core.

This equation can be solved in a similar way as in the delayed precursor neutron equation with discretizing the equation in time and solving the resulting ordinary differential equation analytically. For the decay heat in the outer loop there is no heat generated from fission, so the following equation is considered:

$$\frac{\partial}{\partial t} h_k(r, z, t) + \frac{\partial}{\partial z} [u(z) h_k(r, z, t)] + \lambda_k h_k(r, z, t) = 0. \quad (3-17)$$

This equation should be solved to provide a heat source for the heat exchanger. The fission, decay, and total volumetric heat sources in the core can be expressed as:

$$Q_{decay}(r, z, t) = \sum_{k=1}^K \lambda_k h_k(r, z, t), \quad (3-18)$$

$$Q_{fiss}(r, z, t) = \left[1 - \sum_{k=1}^K f_k \right] \sum_{g=1}^G \varepsilon \Sigma_{fg}(r, z, t) \phi_g(r, z, t), \quad (3-19)$$

$$Q_{tot}(r, z, t) = Q_{fiss}(r, z, t) + Q_{decay}(r, z, t). \quad (3-20)$$

Verification Tests

The capability of MSR solver to model delayed neutron precursor drifting was tested by varying the time spent by the fuel in the core (fuel salt speed) and that out the core (decay time). It is expected that when all the other conditions are the same, the effective multiplication factor of the flowing fuel is smaller than that of the stationary fuel due to the decay of the delayed neutron precursors in the regions of less importance or outside the core. This will reduce the effective delayed neutron fraction of the core, which in turn decreases the effective multiplication factor. In this section, the effects of the flowing fuel on the effective multiplication factor and delayed neutrons distribution are presented for the TRU-started MSFR.

The fuel transient time for the MSFR design is 4.0 s, and the fuel salt spends 2.0 s in the core region and 2.0 s outside the core. For this reference case, the reactivity difference is about 157 pcm, which is about half of the effective delayed neutron fraction of the TRU-started MSFR (about 300 pcm). Figure 3-9 shows the trends of the effective multiplication factor as a function of the time spent in the core for different decay times. As the fuel salt spends a longer time in the core region (as the fuel velocity in the core goes to zero), the effective multiplication factor increases and it converges to the stationary fuel value.

Figure 3-10 shows the power evolutions of pump start-up transients to different final velocities (left figure) and those of pump coast-down transients from different initial velocities (right figure). In a pump start-up transient, the power decreases more rapidly with increasing velocity (i.e., with decreasing transit time in the core). The fuel coming back to the core contains the delayed neutron precursors produced before. As a result, the precursor concentration in the fuel coming back to the core can be higher or lower than those in the fuel flowing out of the core, depending on the core power when they were generated relative to the current power. Therefore, they can introduce positive and negative reactivity alternatively, and the power shows an oscillatory behavior. A pump coast-down transient increases the power monotonically, and the power increases more rapidly with increasing initial flow velocity.

The unprotected transient over power (UTOP) scenario assumes a step insertion of positive reactivity and the subsequent failure of the reactor protection system to shut down the reactor. A

step reactivity was inserted by injection of fissile material in the core (increasing fuel salt concentration) while maintaining the nominal fuel salt flow in the core. Two UTOP transients were simulated with different reactivity insertions at the full power states: 50 pcm (super-critical) and 200 pcm (super-prompt critical). Among those cases, Figure 3-11 compares the time evolutions of the core power and the average temperature increase across the core obtained from the MSR transient solver of PROTEUS-NODAL with the TUDelft results, showing the results of the super-critical transient with 50 pcm reactivity insertion.

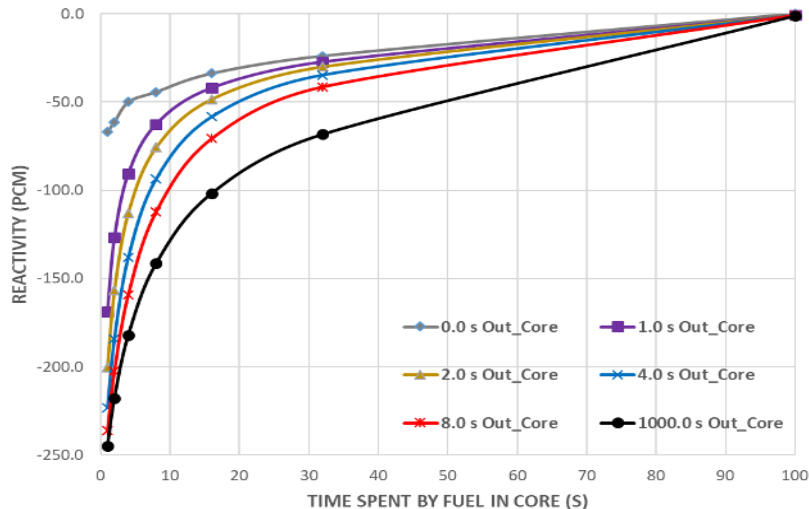


Figure 3-9. Reactivity Losses of TRU-started MSFR Due to the Fuel Drifting

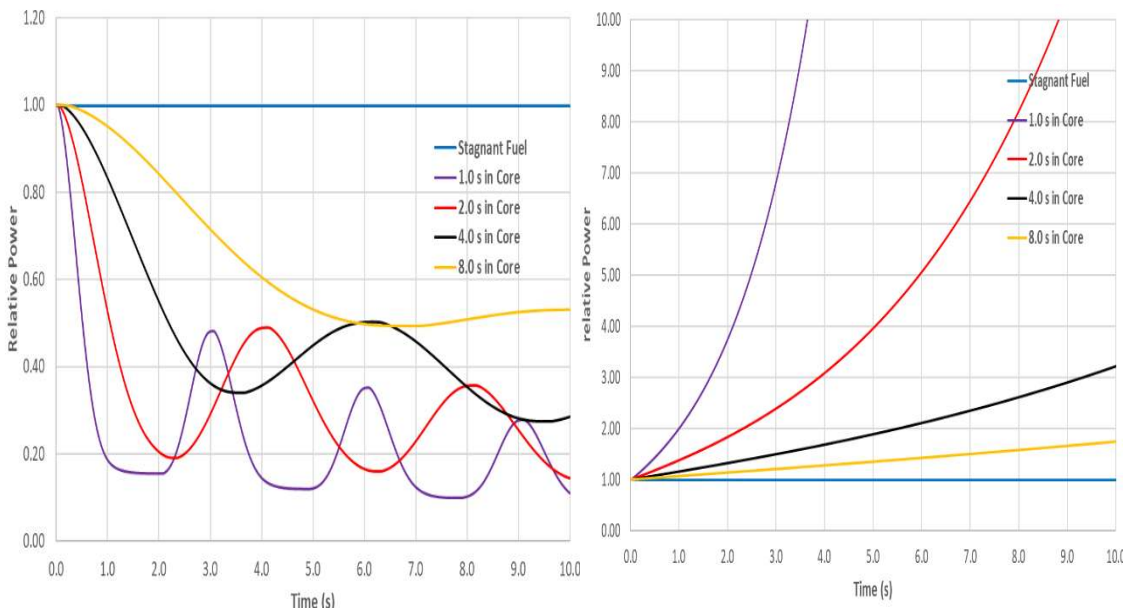


Figure 3-10. Pump Start up (left) and Pump Coast Down (right) Transients without Thermal Feedback

The super-critical transient results in Figure 3-11 show that the power increases initially and attains its maximum (~1.5 times of the nominal power) around 0.8 ms into the transient. The increased power increases the fuel salt temperature which in turn introduces a negative reactivity because of Doppler effect and reduced fuel salt density. Because of the negative reactivity feedback, the power starts to decrease at 0.8 ms and approaches the asymptotic power around 1.0 s. Around 3.0 s into the transient, the power experiences a step decrease again because of the heated fuel salt flowing back into the core. It can be seen that the power and temperature solutions of PROTEUS-NODAL agree well with the reference TUDelft solutions. However, the peak power predicted by PROTEUS-NODAL is slightly lower than the reference solution. This can be attributed to the different feedback coefficients. The total reactivity feedback coefficient of PROTEUS-NODAL (-7.68 pcm/K) was more negative than TUDelft's value of -6.79 pcm/K. The more negative reactivity feedback coefficient reduces the asymptotic core temperature rise by about 2.0K.

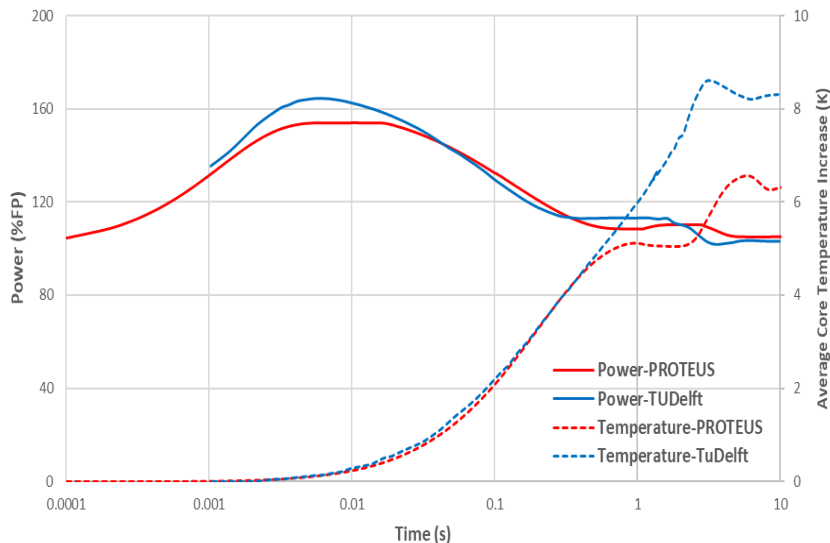


Figure 3-11. Power and average core temperature rise for 50 pcm reactivity insertion

3.3.2 Coupling with SAM

PROTEUS-NODAL was coupled with the system analysis code SAM for the application to steady state and transient analyses of molten salt reactors (MSRs) with flowing fuel. Since SAM is a MOOSE-based code, we developed a MOOSE-based wrapper, called TreeFrog, for PROTEUS-NODAL. A utility code was developed which converts the finite element mesh of the PROTEUS format to the Exodus II format so that the converted mesh can be imported by the MOOSE-based wrapper to transfer data between PROTEUS-NODAL and SAM.

The data transfer between PROTEUS-NODAL and SAM was made through the “ExternalProblem” object and the transfer system between the MOOSE “MultiApps” system. It was successfully tested to pass temperature, velocity, and density from SAM to PROTEUS-NODAL with matched axial meshes. Interpolation was performed for unmatched axial meshes. The

transfer of the power density from PROTEUS-NODAL to SAM was developed as well. Not that power densities from PROTEUS-NODAL were passed to the “AuxVariable” object in the MOOSE framework, and a new heat source object was implemented in SAM to read the transferred power density. The success of coupling of can be adapted with minor changes to other MOOSE-based tools such as BISON [32].

With this data transfer capability, the stead-state coupled neutronics and thermal-hydraulics equations were solved with the Picard iteration using the MOOSE “Executioner” object “SteadyWithPicardCheck.” The coupled transient solver was developed as well. The results of the coupled null-transient calculation well approached to the coupled steady state calculation results.

The restart capability of a coupled transient simulation from a steady state solution was implemented. A restart file containing neutronics solutions was generated at a steady-state condition and was imported for a coupled transient calculation. The restart calculation for SAM is handled by the MOOSE restart modules utilizing “checkpoint files.” The restart capability in the coupled calculation was successfully tested for a null-transient test. The transient coupling is available only when using the same time step size in both PROTEUS-NODAL and SAM.

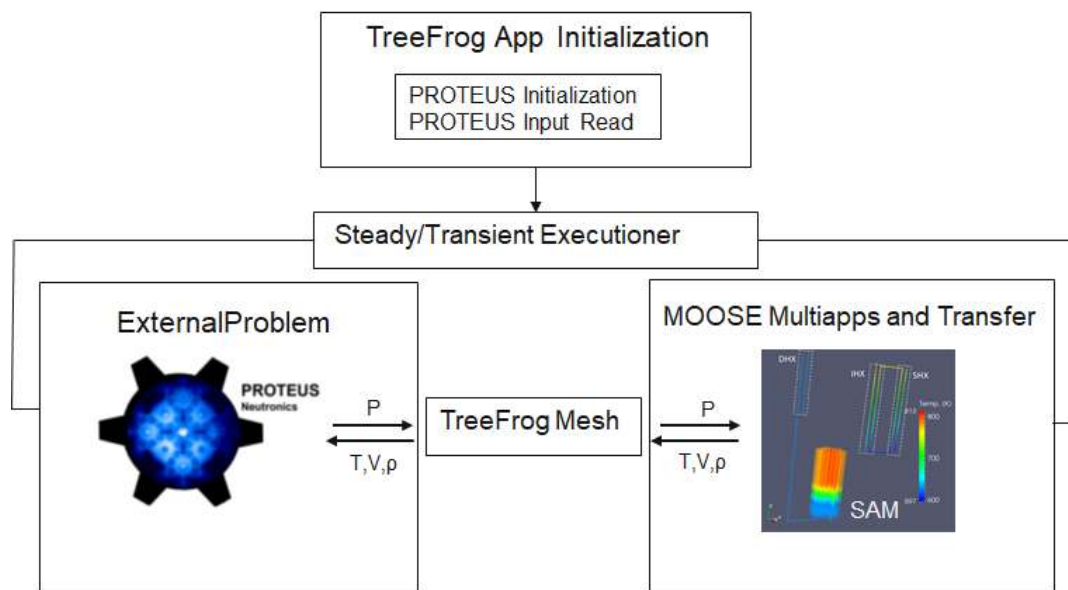


Figure 3-12. Coupling of PROTEUS-NODAL and SAM using the MOOSE-based Wrapper TreeFrog

The coupled system of PROTEUS-NODAL and SAM was tested against the following unprotected accident scenarios of the MSFR benchmark problem: unprotected loss of flow (ULOF), unprotected loss of heat sink (ULOHS), and unprotected transient over-power (UTOP) accidents. By comparing the reactor power and core-averaged temperature histories with the standalone PROTEUS-NODAL results with the built-in simple thermal hydraulics model, it was confirmed

that the coupled system worked properly. Figure 3-13 shows the result from the coupled system for the case with a reactivity insertion of 50 pcm, which is comparable with the PROTEUS-NODAL results with the built-in T/H solver as shown in Figure 3-11.

Table 3-4. Eigenvalues from PROTEUS-SAM and PROTEUS with Built-in T/H

Parameter	Coupled PROTEUS-SAM	PROTEUS with Built-in T/H	Difference (pcm)
k-eff	0.98055	0.98128	73

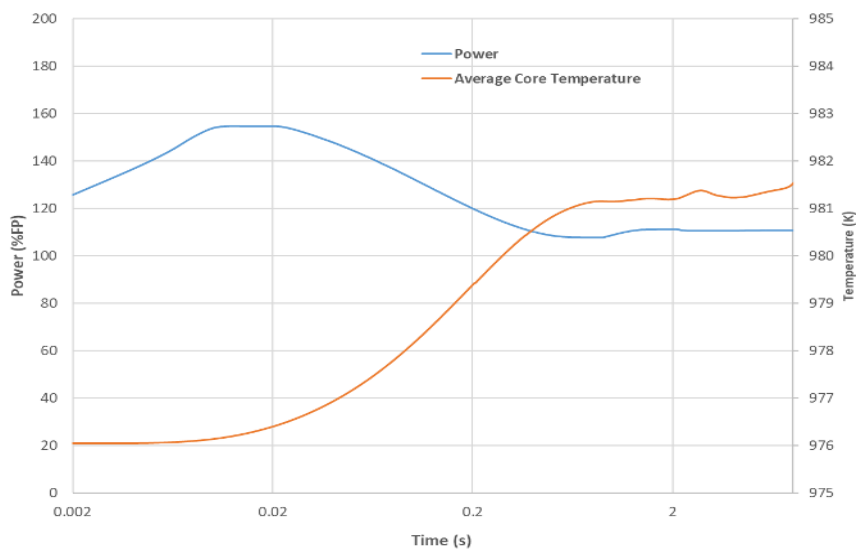


Figure 3-13. Power and Average Core Temperature Rise for 50 pcm Reactivity Insertion, Resulted from the Coupling of PROTEUS-NODAL and SAM

4. Updates in the High Fidelity Solvers of PROTEUS

4.1 Coupling with ANSYS

Most micro reactors are based on unconventional and irregular geometry and configurations as well as thermal expansion behavior (for some micro reactors) that are very challenging or impossible to model using the existing conventional reactor analysis tools other than Monte Carlo tools. However, due to the disadvantages of Monte Carlo tools in terms of uncertainty for transient analysis and limitations in thermal expansion simulation, deterministic simulation approaches are preferred and desirable for some time. Since the high-fidelity solvers of PROTEUS are based on finite-element unstructured meshes, they can be readily used for simulating micro reactors having irregular geometries.

For modeling the thermal and expansion behaviors of heat pipe cooled micro reactors, the multiphysics simulation capability was implemented by coupling PROTEUS and the thermal mechanical code ANSYS [22]. Due to the limited licensing of the ANSYS package, we initially attempted the coupling of PROTEUS and FLUENT [33], instead of ANSYS-mechanical [34] by using the Python external drivers which coordinates data transfer as well as controls the two codes. This section details the coupling scheme and test results.

4.1.1 Updates in PROTEUS

PROTEUS-MOC was updated to support the Python driver for the coupled simulation with FLUENT. Coupling interface routines were implemented in PROTEUS-MOC in a way to communicate with the Python driver through the MPI library. Using interface routines, the Python driver is able to control PROTEUS-MOC to perform a certain segment of calculation flow or to wait while FLUENT generates temperature solutions in the subsequent calculation flow.

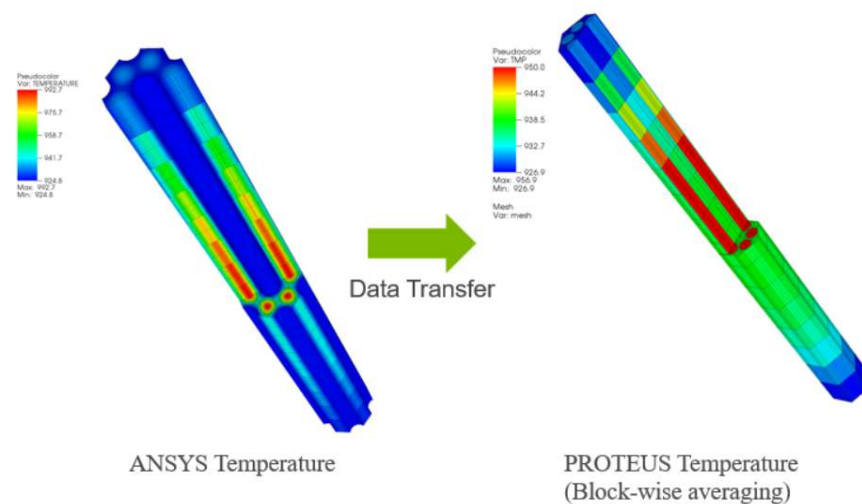


Figure 4-1. Demonstration of Data Exchange of PROTEUS-MOC Coupled Calculation.

For data exchanges with other coupling components, the import and export routines of heat source and temperature distributions were implemented in PROTEUS-MOC. Using this feature, the power distribution determined by PROTEUS-MOC can be used as the heat source distribution in the subsequent FLUENT calculation. Then, the temperature distribution from FLUENT can be incorporated in the PROTEUS-MOC calculation as illustrated in Figure 4-1.

4.1.2 Coupled Simulation

For multiphysics simulation of micro reactors, three physics codes were coupled together as shown in Figure 4-2: PROTEUS for neutronics analysis, FLUENT for thermal analysis, and ANLHTP for heat pipe performance analysis. An efficient way of coupling different codes would be to develop a driver and compile all codes together in order to control physics components by calling them directly and exchanging data required for individual physics calculations via data memories. Since, however, FLUENT is a commercial code, it is difficult to compile it with external codes. Therefore, power and temperature data are transferred between PROTEUS and FLUENT via files.

In addition, we found that the Exodus mesh generated by FLUENT is hard to be made readable by PROTEUS because it is difficult to map region names (too many regions to manage) with compositions and identify boundary surfaces for setting up boundary conditions. In this study, therefore, meshes are constructed for each of PROTEUS and FLUENT, transferring data between the two codes based on their own meshes and requiring the interpolation from one to the other meshes. FLUENT interpolates the power data transferred from PROTEUS using its built-in interpolation module, whereas the PROTEUS Python driver interpolates the temperature data given from FLUENT and provides the interpolated data to PROTEUS. The MPI-based Python is used to transfer the interpolated data to PROTEUS via a memory instead of a file, once data are read and interpolated by the Python driver.

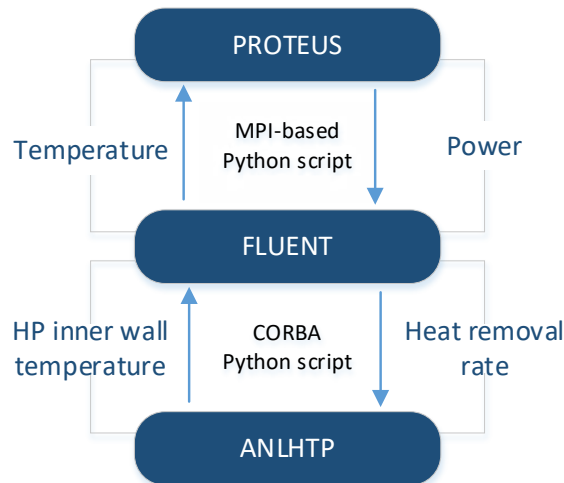


Figure 4-2. Data Exchange of PROTEUS, FLUENT, and ANLHTP

The coupling of PROTEUS and FLUENT is controlled by two separate Python drivers. The PROTEUS Python driver controls the overall system as well as PROTEUS, while the FLUENT Python driver controls the coupling of FLUENT and ANLHTP. Two Python drivers run three codes at the same time, having them execute or wait while the other codes are running and generating input data. CORBA [35] allows us to control FLUENT externally, having the code execute or wait until the input data are updated by ANLHTP (temperatures at the wick-vapor interface of heat pipes) or PROTEUS (power). Figure 4-3 shows the coupling scheme of the three codes controlled by the two Python drivers.

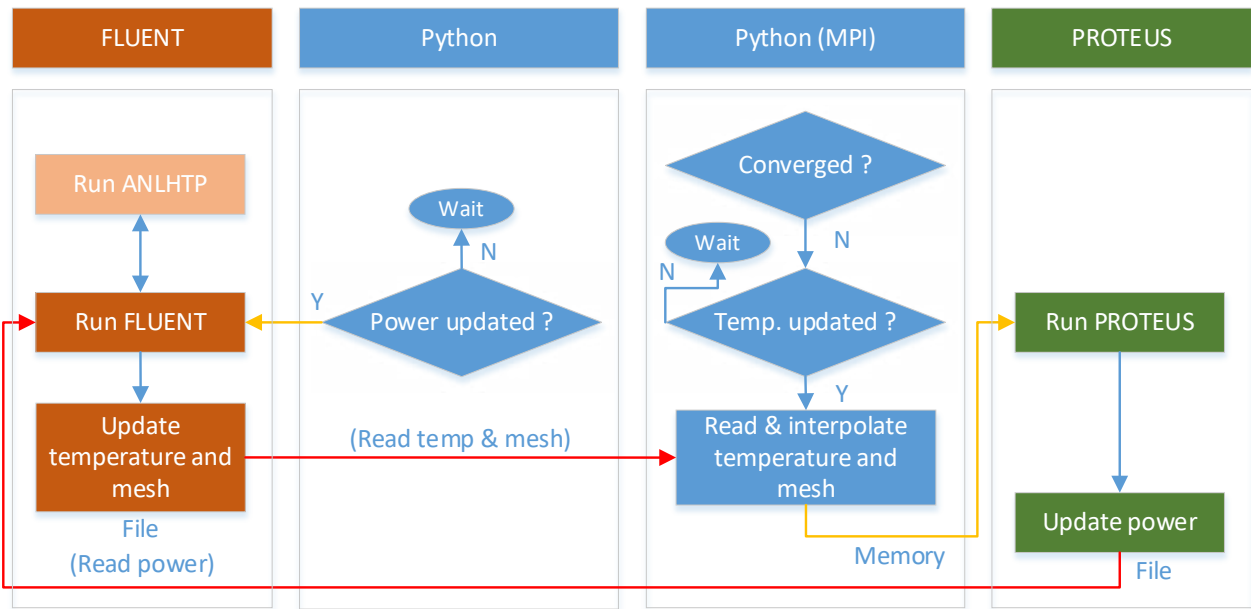


Figure 4-3. Python Control of PROTEUS, FLUENT, and ANLHTP

4.1.3 Demonstration

A conceptual unit assembly problem developed based on the MegaPower heat pipe cooled reactor core was analyzed in order to demonstrate the capability of the coupled codes. In this conceptual problem, seven heat pipes surround six fuel rods in the monolith, of which powers vary as shown in Figure 4-4. The mesh and geometry used in the calculation for the seven heat pipes are shown in Figure 4-7. The domain was divided into 30 axial layers for the monolith and fuel rods. The total power was 14,400 W and ~14% lower power was applied to the fuel rods 2 and 5 in order to generate asymmetric temperature distributions. In the axial direction, the power has a cosine shaped profile. Two meshes were tested to confirm mesh convergence; one has 171,720 cells and the other 71,680 cells. The heat pipe inner wall temperatures obtained by the two meshes showed a difference of less than 0.1 K and the former was selected for this calculation.

First of all, a steady-state calculation was performed using the three codes. The thermal feedback effect was accounted for in the power calculation of PROTEUS, and the power feedback was considered in the temperature calculation of FLUENT. When the steady-state solution was converged, the transient calculation initiated by the HP2 failure was started. As shown in the

FLUENT / ANLHTP calculation above, the failure of HP2 led to the increase of the temperature at HP2 until ~ 170 sec which was propagated to the neighboring regions with the temperature increase. However, due to the Doppler feedback, the power decreased as the fuel temperatures increased, leading to the reduction of the temperature increase rate and finally the decrease of temperatures starting at ~ 170 sec. Since the power kept decreasing, the HP temperatures decreased below the temperatures at the steady-state condition. As temperatures decreased, the decrease rate of power was reduced and consequently the decrease rate of temperatures was slowed down as well. A negative reactivity feedback was initiated due to the temperature increase at the HP2 failure, whose magnitude was reduced due to the temperature decrease. Since the negative reactivity was reduced to almost zero, the power and temperature of the benchmark problem were converged to the different steady-state solutions.

Figure 4-5 shows the power and temperature changes with time after the transient started. The final temperature of HP2 became higher than the initial before the transient started and the temperatures of the other heat pipes were lower than those at the initial steady-state condition. The transient results from the coupled simulation of PROTEUS / FLUENT / ANLHTP appear to be reasonable qualitatively, indicating that the coupled system were implemented correctly. Further verification tests will be conducted for different transient cases as well as larger or whole-core benchmark problems. Test results will be verified quantitatively as well.

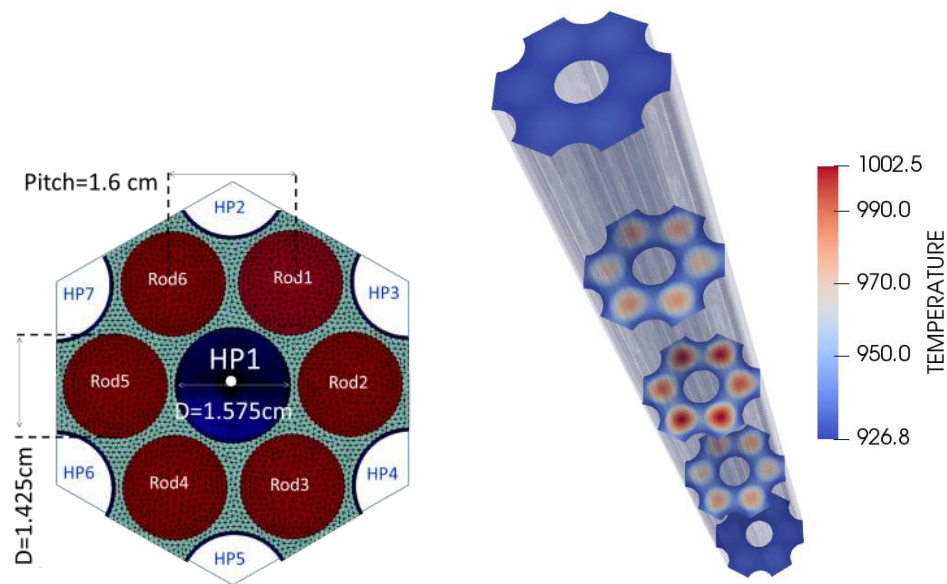


Figure 4-4. Temperature Distribution of Multi-Heat Pipe Problem

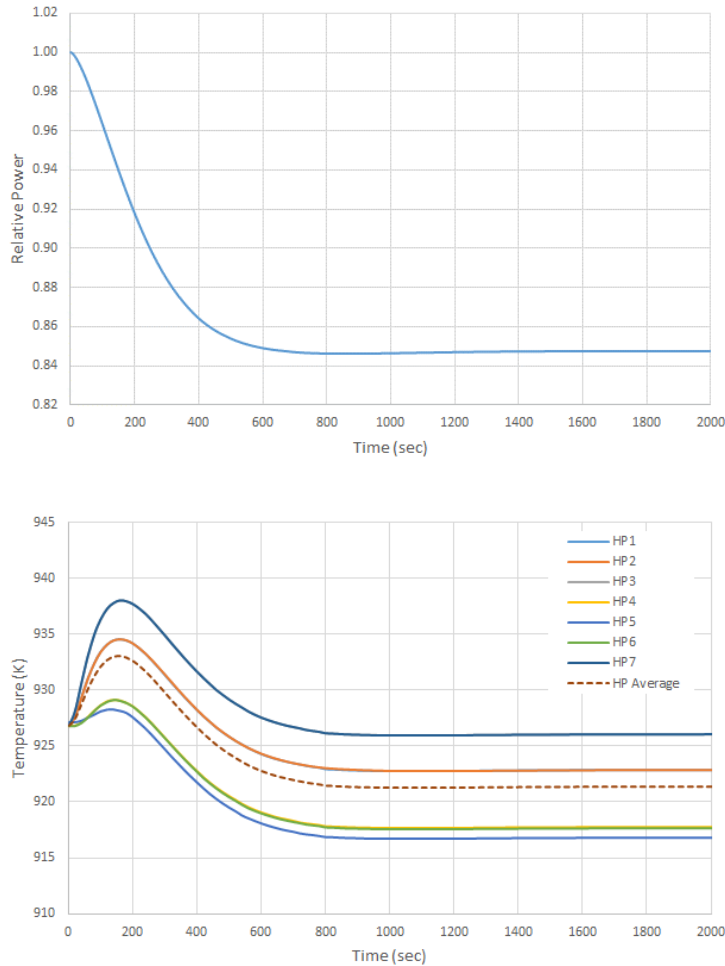


Figure 4-5. Relative Power and Heat Pipe Temperature Change with Time for One Heat Pipe Failure Transient Problem

4.2 Coupled Neutron and Gamma Transport

In order to support the multiphysics simulations using the NEAMS tools for fast reactor analysis [6], the gamma transport capability was implemented in the high fidelity solvers of PROTEUS to perform the coupled neutron and gamma transport calculations. Through the coupled calculations, the detailed gamma distribution can be obtained, and the portion of energy release of neutron reaction carried by gamma particles can be accurately accounted for in the resulting heating distribution. Therefore, the accurately estimated heat distribution can be utilized in the multiphysics simulations. The details of gamma transport capability that is newly implemented in the SN and MOC solvers of PROTEUS is addressed in this section.

4.2.1 Gamma Transport Capability

Since both neutron and gamma transport solvers solve the identical form of the Boltzmann transport equation, the gamma transport capability could be realized in the framework of the SN

and MOC solvers of PROTEUS by extending the existing transport solvers. As summarized in Figure 4-6, new functions required for the gamma transport calculation were implemented and coordinated with associated routines such as the fixed source solver.

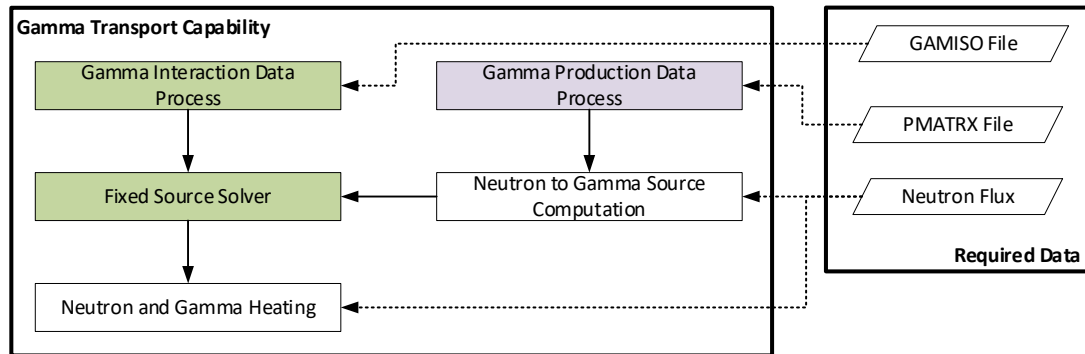


Figure 4-6. Workflow of Gamma Transport Calculation in PROTEUS Code

The gamma interaction data is used as the cross section of the neutron calculations and the gamma particles produced by neutron reaction such as absorption and fission are set as the fixed source that drives transport of gamma flux for the entire problem domain. Thus, the transport solver for neutron flux can be used to calculate the gamma flux distribution by setting the gamma interaction data and the gamma source as the cross section and the fixed source terms. The fixed source solver for the gamma transport problem was implemented by making use of the existing eigenvalue and transient transport solver. For given gamma interaction data and source distribution, the gamma distribution in angle, space, and energy can be obtained by invoking this solver. The parallelization in angle and space for the fixed source transport calculation was made work and tested as well. The anisotropic scattering can be explicitly treated in the gamma transport calculation.

The gamma interaction data processing routines were added to PROTEUS which reads the gamma interaction data from the GAMISO file generated from MC²-3 and assigns the data to meshes. It could be done quickly because the ISOTXS routine could be reused for processing the GAMISO file. The gamma source routines were added to PROTEUS and its function is to prepare the fixed gamma source distribution by combining neutron fluxes and isotope-wise gamma production matrices from the neutron transport calculation and the PMATRX file, respectively. The overall code structure of PROTEUS was also updated in order to better support the neutron and gamma coupled calculations in terms of data management, calculation flow control, etc. The development of gamma transport capability was completed by properly coordinating the developed gamma source routine and other associated routines such as the fixed source solver. The heating capability of PROTEUS was updated to compute the neutron and gamma contributions explicitly using the KERMA factor available in the PMATRX file. The output processing routines was also updated to include the mesh-wise neutron and gamma heating data in the main output file.

4.2.2 Verification Tests

To verify the gamma transport capability, a homogeneous problem for typical SFR fuel pin was solved using the gamma interaction data generated from MC²-3. The identical gamma spectra was expected in the PROTEUS and MC²-3 due to the characteristics of homogeneous problem. Figure 4-7 compares the gamma spectra obtained in the MC²-3 and PROTEUS-MOC calculations, indicating that the gamma spectra used in MC²-3 could be retrieved in the PROTEUS calculations, as expected.

The updated heating capability was tested for the 2D AFR-100 assembly problem. The gamma flux distribution obtained in the coupled neutron and gamma transport calculation are shown in Figure 4-8. Table 4-1 and Figure 4-9 compare the heating distributions obtained from the neutron only and coupled transport calculations. As shown in Table 4-1, the coupled calculation resulted in the gamma heating to be ~10% of the total heating which is reasonable and typical in fast reactors. The notable differences were observed in the non-fuel regions (coolant, clad and duct) because the heating of these regions are contributed from the gamma energy deposition.

Verification tests for the overall gamma transport capability was performed using a 2D SFR core problem which is composed of 4 rings of homogeneous fuel assemblies surrounded with 2 rings of reflectors as shown in Figure 4-10. The additional VARIANT / GAMSOR calculation was performed for comparison since the VARIANT / GAMSOR calculation can produce the reliable solution for this problem. The obtained results from the PROTEUS-SN calculation is plotted in Figure 4-11 and Figure 4-12. The neutron and gamma heating distributions of the test problem produced from PROTEUS-SN and VARIANT / GAMSOR calculations matched well within 1% relative differences as shown in Figure 4-12. For the PROTEUS-MOC solver, slightly larger differences were observed in the heating distribution comparison because of the flat source approximation used in the MOC solver. Note that the relatively coarse mesh structures were used in the MOC and SN calculations, and these errors can be further reduced by refining the mesh used in the MOC calculation. These verification tests indicate that the gamma transport capability was correctly implemented in PROTEUS.

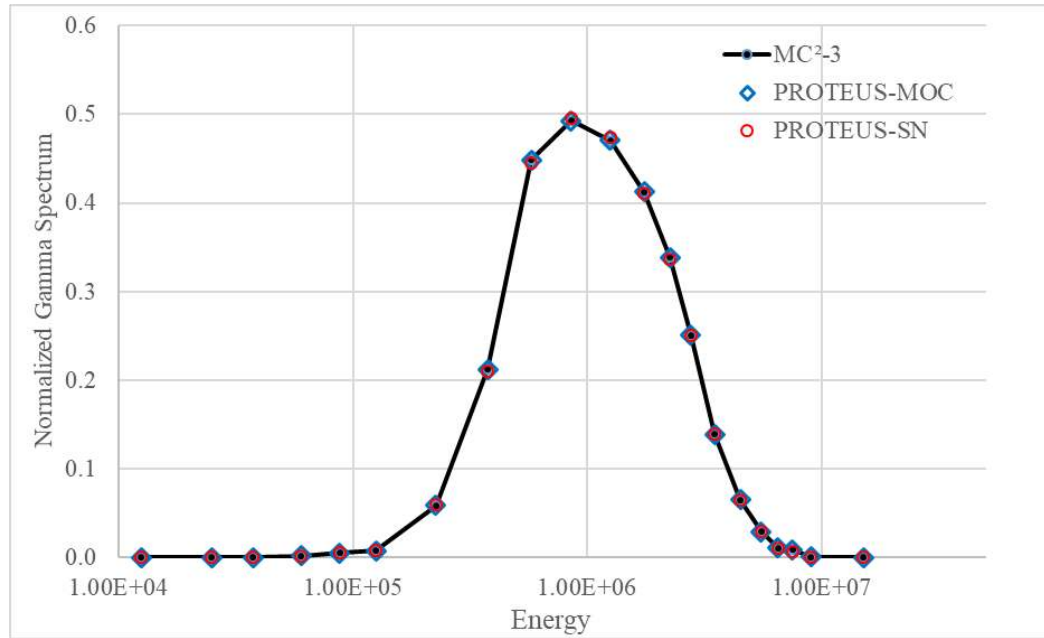


Figure 4-7. Comparisons of Gamma Spectrum Obtained Using MC2-3 and PROTEUS codes for SFR Fuel Mixture

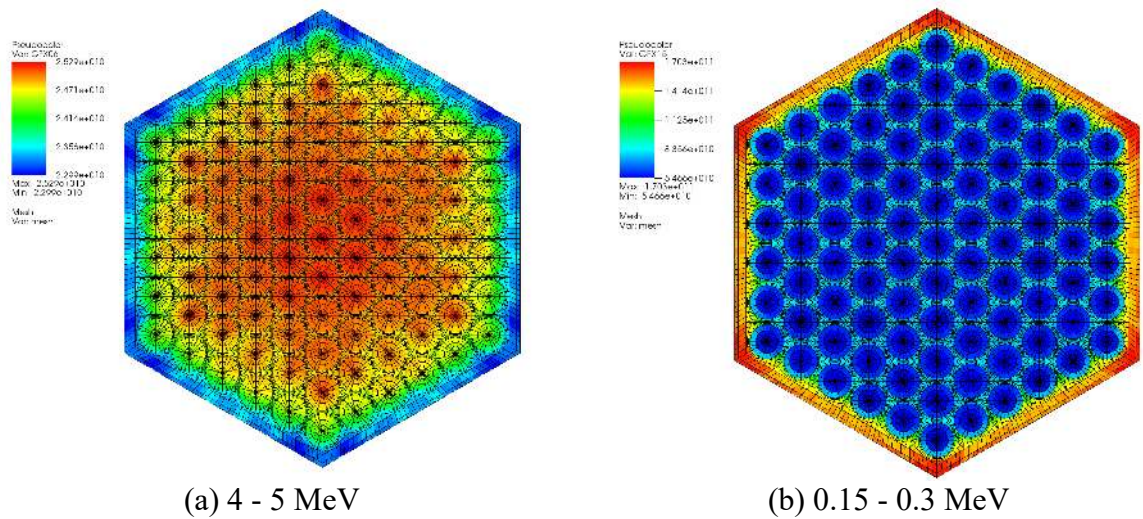


Figure 4-8. Gamma Distributions for AFR-100 Fuel Assembly Problem

Table 4-1. Neutron and Gamma Heating Result for AFR-100 Fuel Assembly Test Case

Region	Neutron Transport Only	Neutron/Gamma Transport		
		Neutron Heating	Gamma Heating	Neutron + Gamma Heating
Fuel	1.00E+00	9.01E-01	9.19E-02	0.99308
Clad	1.73E-04	1.25E-04	1.74E-03	0.00187
Coolant	1.08E-04	1.11E-03	2.02E-03	0.00313
Duct	1.35E-04	9.51E-05	1.30E-03	0.00140
Gap	1.91E-05	1.90E-04	3.42E-04	0.00053
Total	1.00000	0.90272	0.09728	1.00000

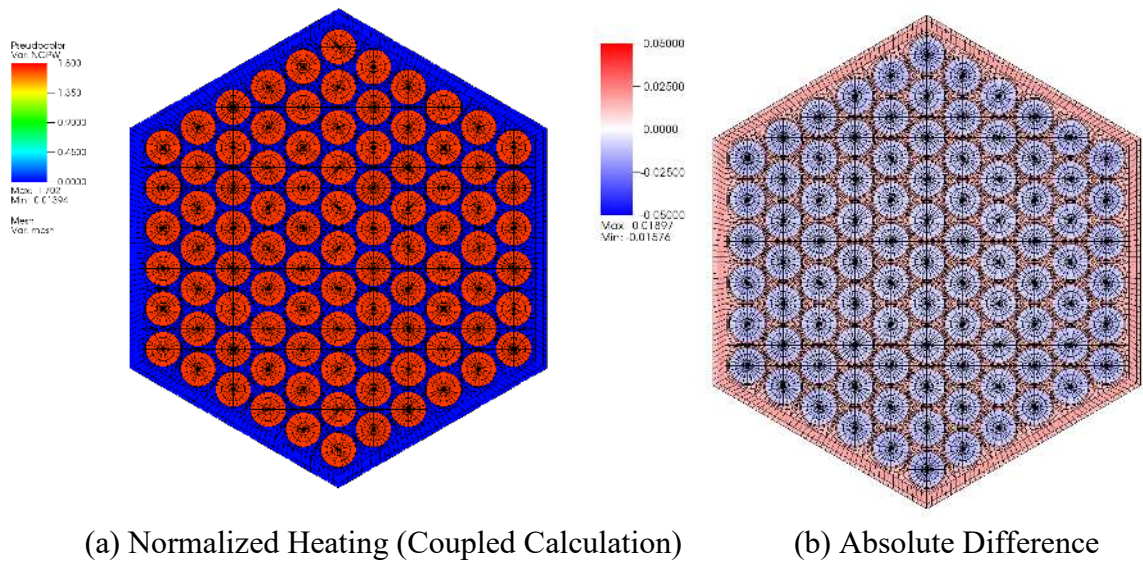


Figure 4-9. Comparison of Local Heating Distribution between Neutron Only and Neutron/Gamma Transport Calculations

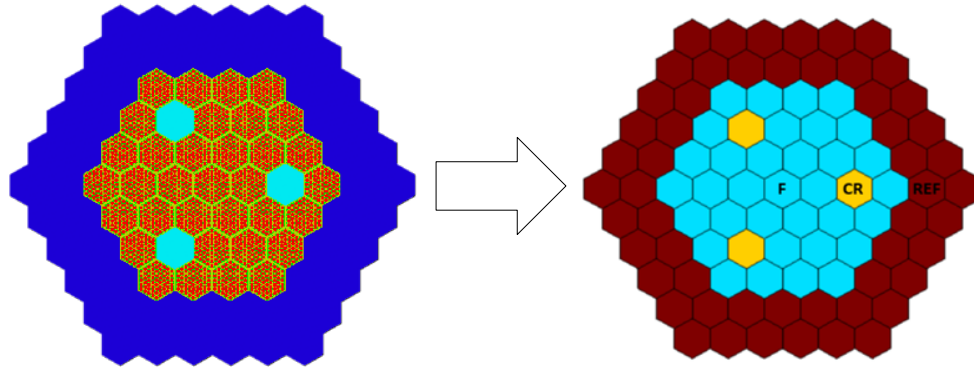
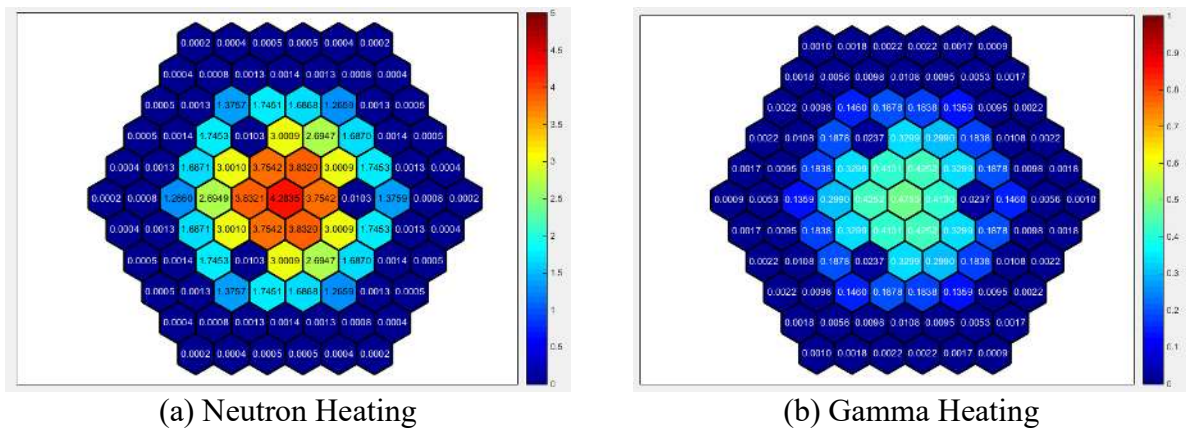


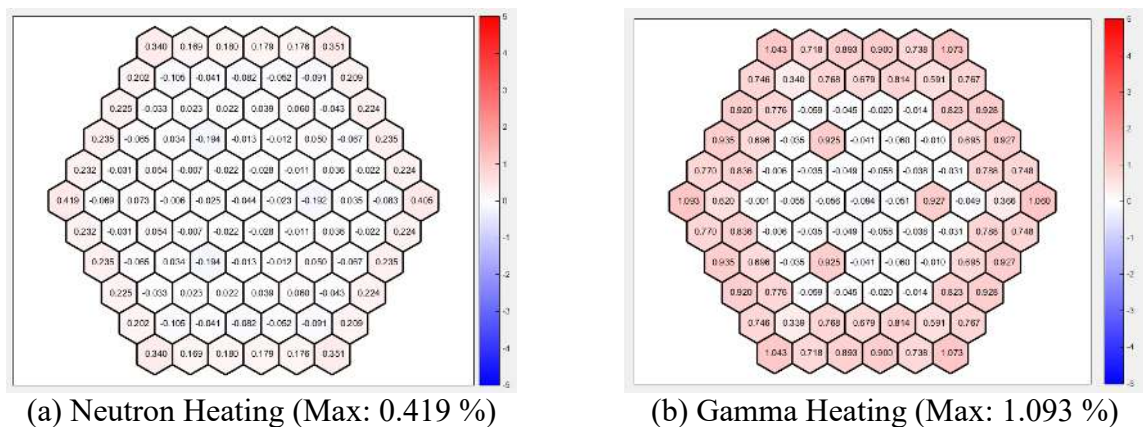
Figure 4-10. Core Configuration of 2D SFR Problem for Testing Gamma Transport Capability



(a) Neutron Heating

(b) Gamma Heating

Figure 4-11. Neutron and Gamma Heating Distribution of PROTEUS-SN for 2D SFR Test Problem



(a) Neutron Heating (Max: 0.419 %)

(b) Gamma Heating (Max: 1.093 %)

Figure 4-12. Relative Differences of Heating Distributions from PROTEUS-SN and VARIANT-GAMSOR codes for the 2D SFR Test Problem

5. PROTEUS User Support

5.1 NEAMS Workbench Integration

In FY19, an integration of PROTEUS into the NEAMS Workbench interface was initiated to improve the code usability by taking advantage of the PyARC framework. For the PROTEUS integration, the extension of the PyARC module referring to its PyPROTEUS sub-module was developed for connecting the Workbench interfaces. Figure 5-1 illustrates how the Workbench interface is connected with PROTEUS through the PyPROTEUS/PyARC wrappers. The followings are the integration status of PROTEUS in FY19:

- *NODAL*: Fully integrated for steady-state calculations. The integration supports all the features of the Workbench/PyARC framework (input generation, workflow management, post-processing).
- *MOC*: Partially integrated for steady-state and transient calculations. It requires off-line mesh and cross section generation since this is currently not supported under the PyARC common user interface model creation.
- *SN*: Not integrated yet, but insignificant efforts are expected.

The PROTEUS-MOC integration does not currently use the PyARC geometry description logic and instead relies on pre-generated off-line mesh and associated cross section generations because the PyARC common input logic for geometry creation does not support an unstructured finite element mesh generation. The details of NEAMS-Workbench integration work can be found in Reference [10].

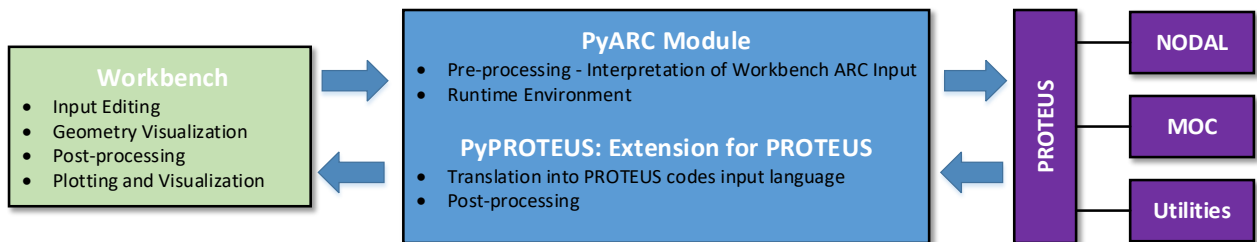


Figure 5-1. Structure of the PROTEUS Integration in the PyARC and the Workbench.

5.2 MeshTool Updates

The MeshTool [11] was updated to provide a convenient way to generate symmetric geometries as shown in Figure 5-2. Before this update of MeshTool, these type of geometries could not be easily generated using the Ufmesh format [11] due to the partial assemblies located along the symmetric lines. The updated version of mesh tool provides a cutting function that can remove a certain portion of the existing mesh file. Using this mesh cutting function, symmetric geometries can be easily generated as illustrated in Figure 5-2.

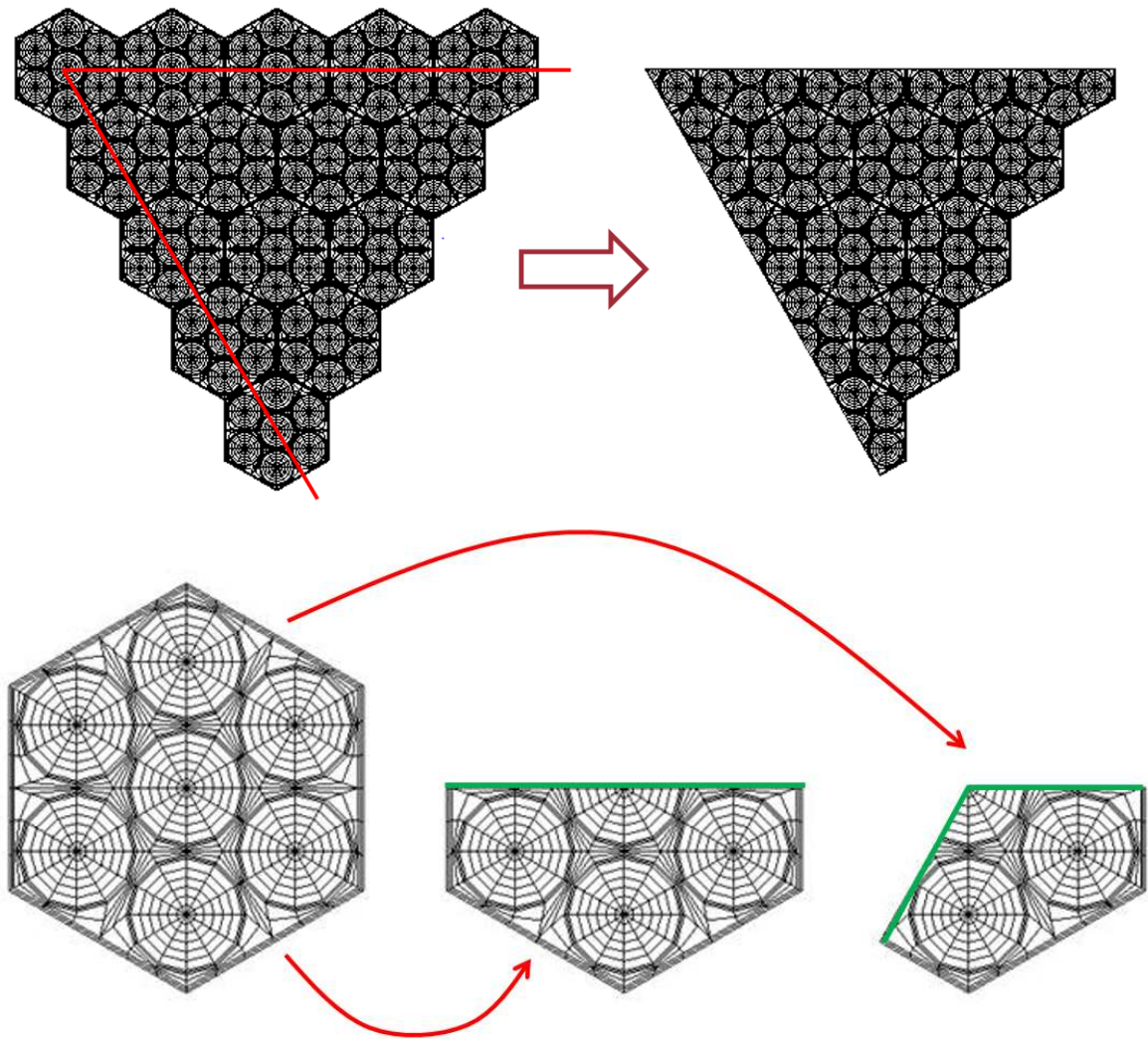
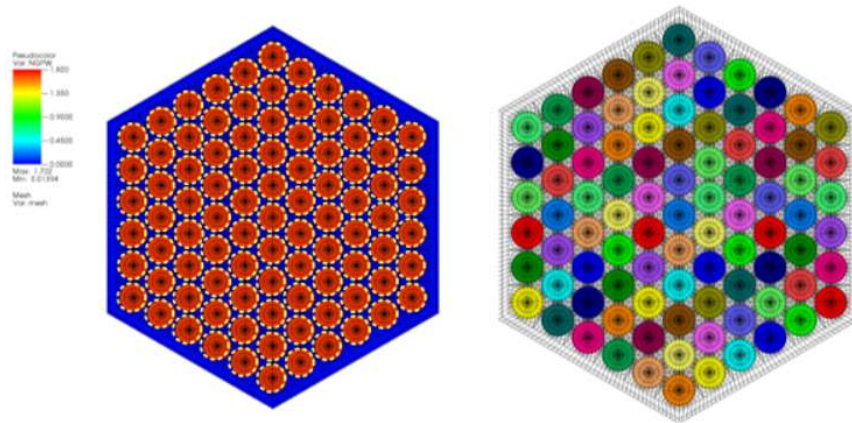


Figure 5-2. Symmetric Geometry Generation Using the Argonne Mesh Tool

5.3 Post-Processing Tool Updates

The post-processing of PROTEUS-MOC could process the region-wise data only for limited configurations such as the Cartesian and hexagonal lattices. The post-processing tool of the MOC solver was updated to enable extracting data for regions defined through the combination of surfaces. This new capability allows users to extract the flux and power information for unstructured regions, which will be useful for the micro reactor calculations which have unconventional and irregular core configuration. Note that the current version only supports limited regions that can be constructed only through a combination of circle and planes. As illustrated in Figure 5-3, the pin-wise flux and power distribution of a SFR assembly can be obtained using the updated post-processing tool.



PROTEUS-MOC Power Dist.

Processing Regions

1.467	1.479	1.481	1.482	1.482	1.482	----	----	----	----	----
1.479	1.475	1.481	1.483	1.484	1.484	1.482	----	----	----	----
1.481	1.480	1.479	1.482	1.483	1.484	1.484	1.481	----	----	----
1.481	1.482	1.481	1.481	1.483	1.484	1.483	1.482	1.480	----	----
1.479	1.480	1.481	1.479	1.481	1.481	1.482	1.482	1.480	1.478	----
1.467	1.475	1.479	1.481	1.481	1.482	1.481	1.481	1.479	1.474	1.466
----	1.479	1.481	1.482	1.483	1.481	1.481	1.479	1.481	1.480	1.478
----	----	1.481	1.483	1.484	1.484	1.482	1.482	1.481	1.482	1.481
----	----	----	1.482	1.484	1.485	1.484	1.482	1.480	1.480	1.481
----	----	----	----	1.482	1.484	1.484	1.483	1.481	1.476	1.479
----	----	----	----	----	1.482	1.482	1.482	1.481	1.479	1.468

Processed Pin Power

Figure 5-3. Example of Updated Post-Processing Capability

6. Conclusions

In FY19, the computational capabilities of PROTEUS-NODAL were updated in terms of fuel cycle capability, link to PERSENT, and updated MSR capabilities. For the fuel cycle capability, the built-in depletion solver was implemented and connected to the nodal transport solvers to perform the cycle depletion calculation. The fuel management option was added, which allows a user to define fuel reloading and shuffling schemes. The fuel cycle analysis capability was tested for a SFR problem with the sophisticated 34-batch shuffling scheme, demonstrating reasonable solutions with burnup. PROTEUS-NODAL was updated to produce the CCCC interface files which are required to run PERSENT. The adjoint flux solver was added as well. Preliminary tests with PROTEUS-NODAL / PERSENT showed reactivity solutions comparable to those with DIF3D / PERSENT. The MSR capabilities were updated with coupling with SAM, comparing the preliminary transient solutions of the coupled system with those with the built-in T/H solver.

The improvements of PROTEUS-MOC in this year focused on the capability extension to support the multiphysics simulation of SFRs and micro reactors. Especially for micro reactor modeling and simulation, the coupled system of PROTEUS and ANSYS were developed to simulate a heat pipe cooled micro reactor. Python-based external drivers were developed to properly coordinate the overall workflow including data transfer and to control individual calculation steps. Currently, PROTEUS-MOC can be coupled with FLUENT for steady-state and transient conditions, instead of ANSYS-mechanical, due to the code access limitation of ANSYS. The heat pipe analysis code ANLHTP was successfully resurrected and verified. An initial demonstration of the coupled system of PROTEUS / FLUENT / ANLHTP was successfully demonstrated for small 3D test problems with heat pipe failure transients.

For supporting multiphysics simulation for fast reactor analysis using the NEAMS tools, the coupled neutron and gamma transport capability was developed in the SN and MOC solvers of PROTEUS. In typical fast reactors, around 10 % of total power is generated from the neutron-gamma reactions and redistributed in the core through the transport of gamma particles. The gamma transport capability was implemented to compute the detailed gamma distributions that determines the primary heat sources for the fuel assembly structure and non-fueled assemblies such as control blocks and reflectors. The heating calculation was updated to explicitly consider the gamma contribution. Verification tests using fuel assembly and 2D core problems indicated very good agreement in heating distributions between PROTEUS and VARIANT / GAMSOR solutions.

The integration of PROTEUS to the NEAMS Workbench was initiated to improve the usability of PROTEUS by leveraging the user-friendly interface provided by Workbench. This year, the NODAL and MOC solvers were connected to Workbench interfaces by extending the PyARC module. Currently, the entire workflow of the NODAL solver from cross section generation to post-processing is supported in Workbench framework. For the MOC solver, Workbench can only assist the PROTEUS input generation, simulation execution, and post processing, while the off-line generation of mesh files and corresponding cross section sets are still required. Along with the

Workbench integration work, the utility tools of PROTEUS were also extended to help users easily model and simulate complicated and non-conventional geometry cores.

REFERENCES

1. A. SIEGEL et al, "Software Design of SHARP," Proc. of M&C+SNA 2007, Monterey, CA, April 15-17, 2007.
2. A. YAMAMOTO, M. TATSUMI and N. SUGIMURA, "Numerical Solution of Stiff Burnup Equation with Short Half Lived Nuclides by the Krylov Subspace Method," *Journal of Nuclear Science and Technology*, **44**, 147-154, 2007.
3. M. A. SMITH, W. S. YANG, A. MOHAMED and E. E. LEWIS, "Perturbation and Sensitivity Tool Based on the VARIANT Option of DIF3D," ANS Transactions 107, San Diego, Nov. 11-15 (2012).
4. R. HU, "SAM Theory Manual," ANL/NE-17/4, Argonne National Laboratory, March 2017.
5. W. S. Yang, Private Communication, August (2019).
6. E. R. SHEMON, Y. YU, Y. S. JUNG, B. FENG and T. K. KIM, "Extension and Demonstration of NEAMS Multiphysics Tools to Lead-Cooled, Sodium-Cooled, and Molten Salt Fast Reactor Applications," ANL/NSE-19/30, Argonne National Laboratory, September 30, 2019.
7. N. STAUFF, N. GAUGHAN, and T. K. KIM, "ARC integration into the NEAMS Workbench," ANL/NE-17/31, Argonne National Laboratory, September 30, 2017.
8. N. STAUFF and T. K. KIM, "Updated Status of the ART Neutronic Fast Reactor Tools Integration to the NEAMS Workbench," ANL/NEAMS-18/1, Argonne National Laboratory, September 30, 2018.
9. K. L. DERSTINE, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite Difference Diffusion Theory Problems," ANL-82-64, Argonne National Laboratory (1984).
10. N. STAUFF, P. LARTAUD, Y. S. JUNG, P. SEURIN, C. H. LEE, K. ZENG and J. HOU, "Status of the NEAMS and ARC Neutronic Fast Reactor Tools Integration To The NEAMS Workbench," ANL/NEAMS-19/1, Argonne National Laboratory, September 30, 2019.
11. M. A. SMITH and E. R. SHEMON, "User Manual for the PROTEUS Mesh Tools," ANL/NE-15/17 (Rev.2), Argonne National Laboratory, August 2016.
12. G. PALMIOTTI et al, "Variational nodal transport methods with anisotropic scattering," *Nuclear Science and Engineering*, **115**, pp. 233-243 (1993).
13. E. M. GELBARD, "Application of Spherical Harmonics Methods to Reactor Problems," WAPD-BT-20, Bettis, Atomic Power Laboratory, 1960.
14. C. H. LEE, Y. S. JUNG, and W. S. YANG, "MC2-3: Multigroup Cross Section Generation Code for Fast Reactor Analysis," ANL/NE-11-41 (Rev.3), Argonne National Laboratory, August 31, 2018.
15. J. LEPPANEN, "Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code," VTT Technical Research Centre of Finland, June 18, 2015.
16. P. K. ROMANO., "OpenMC: A State-of-the-Art Monte Carlo Code for Research and Development," *Ann. Nucl. Energy*, **82**, 90-97, 2015.
17. N. E. STAUFF, P. K. ROMANO, C. H. LEE, and T. K. KIM, "Verification of Mixed Stochastic/Deterministic Approach for Fast and Thermal Reactor Analysis," ICAPP 2017, Fukui and Kyoto, Japan, April 24-27, 2017.
18. CUBIT Web Page, www.cubit.sandia.gov.
19. NEK5000 Web page, <https://www.paraview.org/>.
20. D. PARSONS et al., "DIABLO User Manual," UCRL-SM-234927, Lawrence Livermore National Laboratory, September, 2007.

21. R. K. SALKO, M. M. AVRAMOVA, "CTF Theory Manual," Reactor Dynamics and Fuel Management Group, Pennsylvania State University, March 10, 2015.
22. ANSYS Web page, www.ansys.com.
23. VisIt Web Page, <https://wci.llnl.gov/simulatison/computer-codes/visit>.
24. ParaView Web Page, <https://www.paraview.org>.
25. P. R. MCCLURE, D. I. POSTON, V. R. DASARI, and R. S. REID, "Design of Megawatt Power Level Heat Pipe Reactors," LA-UR-15-28840, Los Alamos National Laboratory (2015).
26. J. W. STERBENTZ et al., "Reactor (5 MW) for Reliable Power at Remote Sites Assessment Report Using Phenomena Identification and Ranking Tables (PIRTs)," INL/EXT-16-40741, Rev.1, Idaho National Laboratory (2017).
27. N. E. STAUFF, C. H. LEE, P. SHRIWISE, Y. MIIAO, R. HU, P. VEGENDLA, and T. FEI, "Neutronic Design and Analysis of the Holos-Quad Concept," ANL/NSE-19/8, Argonne National Laboratory, June 5, 2019.
28. T. K. KIM and T. A. TAIWO, "Once-through Sustainable Sodium-cooled Fast Reactor," The 5th NEA Information Exchange Meeting on Actinide and Fission Product Partitioning and Transmutation, Manchester, UK, September 30 – October 3, 2018.
29. H. G. JOO et al., "PARCS: Purdue Advanced Reactor Core Simulation," PU/NE-98-26, Purdue University, 1998.
30. N.E. TODREAS and M.S. KAZIMI, *Nuclear Systems Vol. I: Thermal Hydraulic Fundamentals*, Chapter 4, 2nd Ed., CRC Press, 2012.
31. M. BROVCHENKO et al., "Optimization of the pre-conceptual design of the MSFR: Evaluation and Viability of Liquid Fuel Fast Reactor System," (EVOL) Project. Deliverable EVOL D2 (2), 2013.
32. R. L. Williamson et al., "Validating the BISON fuel performance code to LWR experiments", *Journal of Nuclear Materials*, **301**, 232-244 (2016).
33. ANSYS-FLUENT Web Page, <https://www.ansys.com/products/fluids/ansys-fluent>.
34. ANSYS-Mechanical Enterprise Web Page, <https://www.ansys.com/products/structures/ansys-mechanical-enterprise>.
35. CORBA Web page, www.corba.com.



Nuclear Science and Engineering Division

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

www.anl.gov



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