ANL/NE-14/8

Verification & Validation Plan for PROTEUS

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.

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
Alexandra, 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

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.

ANL/NE-14/8

Verification and Validation Plan for PROTEUS

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

August 31, 2014

ABSTRACT

This document is a verification and validation plan for the DOE NEAMS neutronics code, PROTEUS, developed at Argonne National Laboratory. PROTEUS is the Reactor Product Line SHARP neutron transport module. The verification and validation requirements presented here are applicable to a specific computational model: evaluation of passive safety features resulting from multiphysics, multiscale reactor dynamics during unprotected loss of flow (ULOF) transients in sodium fast reactor (SFR) cores. In this report, we present an overview of the PROTEUS code, the computational model of interest, and software verification and validation requirements. We also discuss and recommend experimental comparisons and future work.

TABLE OF CONTENTS

Abstract i
Table of Contents ii
List of Figures iii
List of Tables iii
Acronyms 1
1. Introduction 2
   1.1 Overview of PROTEUS 2
   1.2 Overview of Computational Model 2
   1.3 Users, Responsible Parties, and NEAMS Role 5
2. Description of Computational Model 7
   2.1 Phenomena Identification 7
   2.2 Description of Physics and Empirical Models 8
   2.3 Required Toolkit Elements and Libraries 10
   2.4 Consequence of Failure, Quality Rigor Level, Risk Grading 10
   2.5 Effect of Hardware on Validation 11
   2.6 Problem Setup and Analysis 11
3. Model Verification and Validation 12
   3.1 Software Verification 12
      3.1.1 Software Configuration Management 12
      3.1.2 Software Version Control 12
      3.1.3 Regression, Benchmark Tests, and Bug Reporting 13
      3.1.4 Documentation 13
      3.1.5 Static Analysis 14
   3.2 Verification with Analytical Benchmarks 14
   3.3 Verification with Code-to-Code Comparison 14
   3.4 Relevant Experimental Data for Validation 14
      3.4.1 ZPPR-15 Experiments 16
      3.4.2 EBRI-II Shutdown Heat Removal Tests 17
      3.4.3 FFTF Inherent Safety Test Series 17
   3.5 Unit, Component, Subsystem, System, Integral tests, etc. 17
   3.6 Validation Matrix and Gaps 18
   3.7 Schedule and Priorities 19
4. Sensitivity Analysis and Uncertainty Quantification 20
   4.1 Sensitivity Analysis 20
   4.2 Uncertainty Quantification 20
5. Path Forward 21
References 22

LIST OF FIGURES

Figure 1. ABTR Design: Vertical View of Primary System Showing Pool and Assemblies.... 3
Figure 2. ABTR Design: Top View of Core Showing Assembly Map. 4
Figure 3. ABTR Design: Schematic of Core Geometry with Support and Restraint System.... 4

LIST OF TABLES

Table 1. List of Phenomena for Computational Model of Interest. 8
Table 2. Neutronics Simulation Quantities of Interest..... 8
Table 3. Relevant Experiments for Validation..... 15
Table 4. Examples of Neutronics Unit, Component, Subsystem and System Tests..... 18
Table 5. Validation Matrix..... 18

Acronyms

ABTR = Advanced Burner Test Reactor

ANL = Argonne National Laboratory

API = Application Programming Interface

BOL = Beginning of Life

CRBR = Clinch River Breeder Reactor

DOE= Department of Energy

FFTF = Fast Flux Test Facility

FRUIT = Fortran Unit Testing Suite

GEM = Gas Expansion Module

LOHS = Loss of Heat Sink

MMS = Method of Manufactured Solutions

NEAMS=Nuclear Energy Advanced Modeling and Simulation

QRL = Quality Rigor Level

SFR = Sodium-cooled Fast Reactor

SHRT = Shutdown Heat Removal Test

S_N = Discrete Ordinate Transport Method

SS = Stainless Steel

SVN = Subversion

ULOF = Unprotected Loss of Flow

UQ = Uncertainty Quantification

V&V = Verification and Validation

ZPPR = Zero Power Physics Reactor

ZPR = Zero Power Reactor

1. Introduction

1.1 Overview of PROTEUS

The DOE NEAMS neutronics code PROTEUS is a three-dimensional, highly scalable, high-fidelity neutron transport code [1,2] developed at Argonne National Laboratory (ANL). PROTEUS is the neutron transport module in the multi-physics Reactor Product Line SHARP toolkit. The code is applicable to all spectrum reactor transport calculations, particularly those in which a high degree of fidelity is needed either to represent spatial detail or to resolve solution gradients. PROTEUS solves the second-order formulation of the transport equation using the continuous Galerkin finite element method in space, the discrete ordinates (S_N) approximation in angle, and the multigroup approximation in energy. PROTEUS's parallel methodology permits the efficient decomposition of the problem by both space and angle, allowing large problems to run efficiently on hundreds of thousands of cores. PROTEUS can also be used in serial or on smaller compute clusters (10's to 100's of cores) for smaller homogenized problems, although it is generally more computationally expensive than traditional homogenized methodology codes. Multigroup cross sections are typically generated externally using the MC^2-3 code [3] for fast reactor systems or internally using the cross section application programming interface (API) [4] for thermal or fast reactor systems.

PROTEUS calculates the eigenvalue and multigroup flux spatial distributions for both the forward and adjoint problems. The flux distributions are also processed into reaction rates and power distributions. For time-dependent solutions, an adiabatic method has been implemented to solve for the power and kinetics parameters (neuron generation time and delayed neutron fraction) at each time step. Fixed source capabilities were also recently added for both steady-state and kinetics modes.

1.2 Overview of Computational Model

This verification and validation (V&V) plan is presented for a specific use case called a "computational model" [5,6]. In this plan, we focus on the target use case [7] of evaluation of passive safety features resulting from multiphysics, multiscale reactor dynamics during unprotected loss of flow (ULOF) transients in sodium-cooled fast reactor (SFR) cores. In particular, we are interested in using PROTEUS to perform steady-state and transient neutronics calculations for SFR reactor designs such as the Advanced Burner Test Reactor (ABTR) [8].

ABTR is a conceptual advanced sodium-cooled nuclear reactor designed by Argonne National Laboratory. ABTR is rated for a thermal power of 250 MW with an electric output of approximately 95 MW (38% thermal efficiency). The reactor core contains 199 assemblies (54 driver fuel assemblies with 3 types of enrichments). The reference fuel design uses weapons-grade plutonium-based ternary metal (U-TRU-10Zr). There are 7 primary and 3 secondary control rod assemblies for providing reactivity control. In addition, 9 test assembly locations are provided (six for fuel tests and three for material tests).

All primary system components are submerged in a sodium pool-type configuration as illustrated in Figure 1, which shows the elevation view of the primary system. The sodium coolant inlet and outlet temperatures are 355°C and 510°C, respectively. The cold pool level is 10.16 m above the bottom of the pressure vessel. The hot pool level is at 12.20 m elevation above the bottom of the pressure vessel (2.04 m above the cold pool level). The core consists of hexagonally oriented assemblies surrounded by the core barrel. The assembly loading map of the core is depicted in Figure 2.

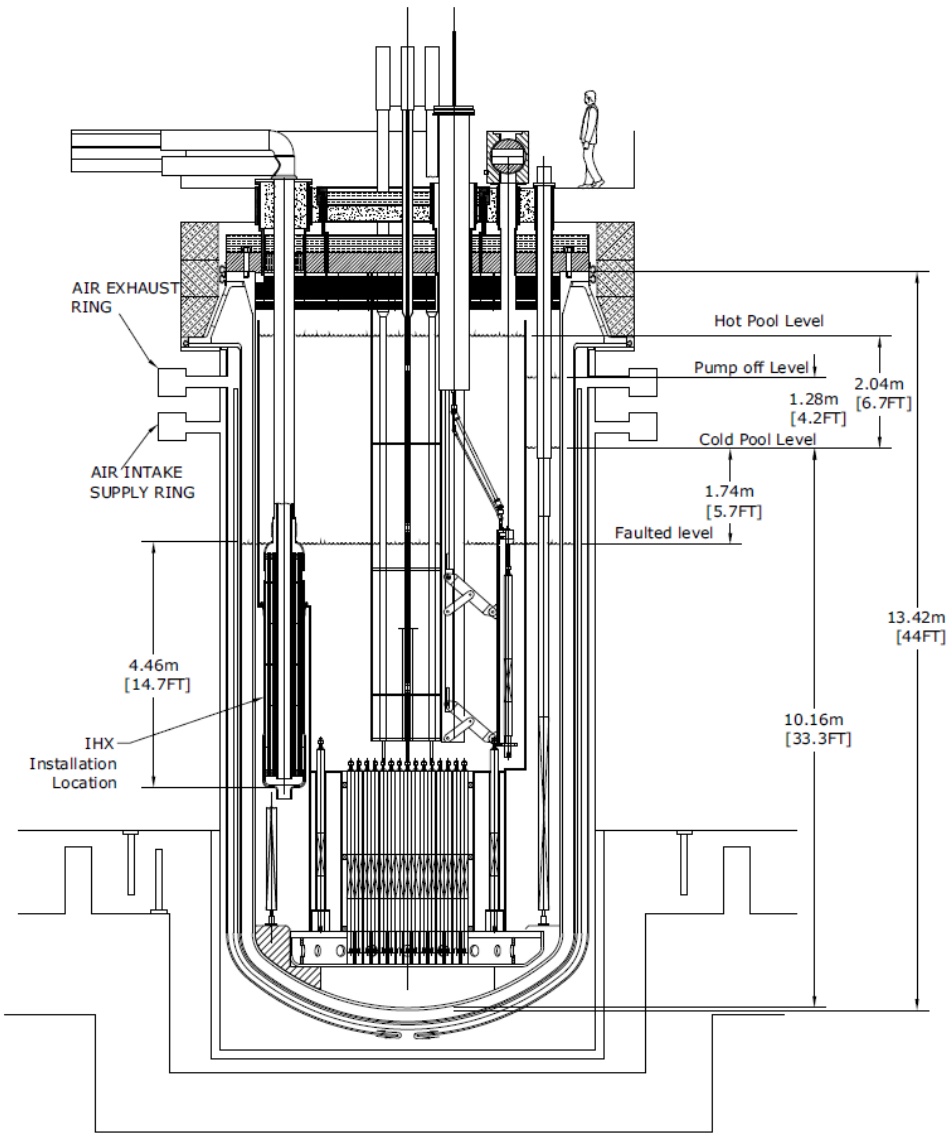

Figure 1. ABTR Design: Vertical View of Primary System Showing Pool and Assemblies.

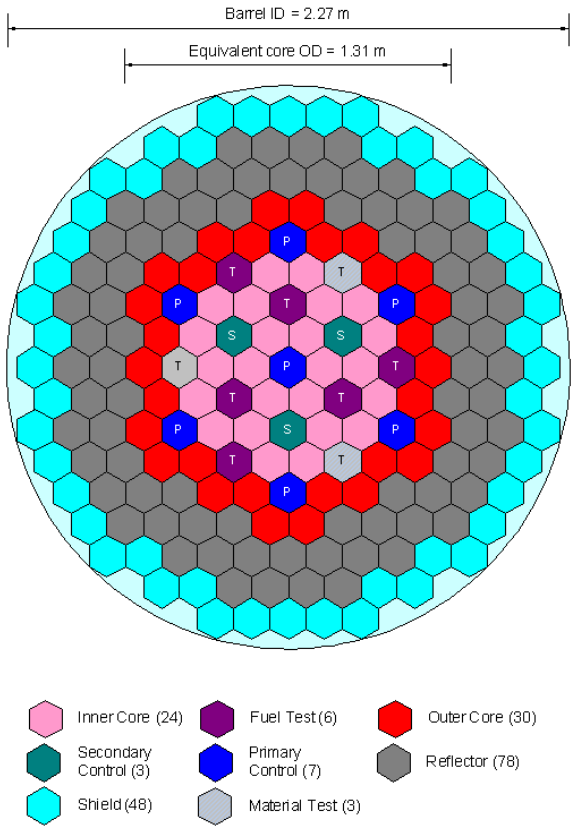

Figure 2. ABTR Design: Top View of Core Showing Assembly Map.

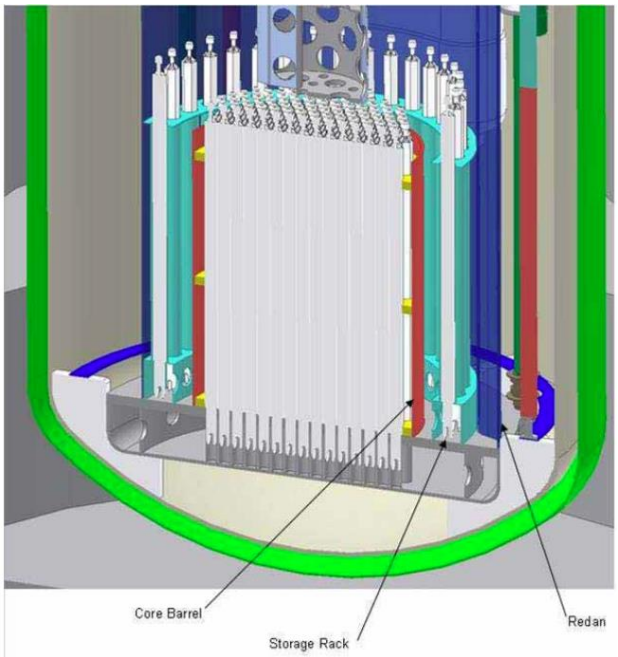

Figure 3. ABTR Design: Schematic of Core Geometry with Support and Restraint System.

The ABTR uses a limited free-bow core restraint system designed to shape the assembly ducts into a configuration which provides negative reactivity feedback. The ABTR was recently chosen as the focus of a demonstration problem for the NEAMS Reactor Product Line SHARP Toolkit [9,10], which includes neutronics, thermal fluid, and structural mechanics physics modules coupled together in a framework. A schematic of the core restraint system is shown in Figure 3.

During an unprotected loss of flow transient in a SFR, the sodium mass flow rate through the core decreases causing the sodium coolant and the fuel temperatures to rise. Initially, the rising sodium coolant temperature causes an increase in the power due to the reduced density and reduced moderating power of the coolant. The rising temperature of the system causes mechanical deformations from thermal expansion and, because of the restraint system, bowing of the lattice of assemblies. The core bowing provides a negative reactivity feedback due to increased leakage (larger core volume with fixed mass of heavy metal leads to a lower criticality state in fast spectrum systems), which decreases the power production. The increase in fuel temperature alters the material cross sections and increases resonance absorption due to the Doppler broadening effect. Therefore, ULOF results in the activation of passive safety mechanisms which provide a natural limit to the temperature and power levels that can occur during a transient.

To accurately model this system, PROTEUS must be able to accurately model deformed (unstructured) geometry as well as to account for temperature feedback in the neutron cross section data. PROTEUS itself is not, however, responsible for calculating the deformations or temperature changes: these are the responsibility of the coupled physics modules Diablo [11] and Nek5000 [12]. The problem of interest is inherently multi-physics (requiring coupling to both thermal hydraulic and structural mechanical physics modules). However, this V&V plan is focused on examining the accuracy of PROTEUS as a single-physics code given an assumed geometry deformation and temperature distribution that reasonably correlate with the PROTEUS computed power distribution. The accuracy of PROTEUS within the SHARP multi-physics framework should be discussed in a SHARP V&V plan.

For this computational model, PROTEUS should be able to predict core eigenvalue and detail reaction rate distributions (such as radial pin powers) within experimental uncertainty for all configurations (cold initial state, hot undeformed geometry, and hot deformed geometry) given an assumed deformed geometry and assumed temperature profile. For transient calculations, it is of particular importance to properly predict detailed fission rate (power) distributions to ensure that no structural materials exceed their failure points.

1.3 Users, Responsible Parties, and NEAMS Role

This V&V Plan is intended to serve research scientists who are modeling or designing sodium cooled fast reactors which have a limited free bow structural mechanical feedback mechanism.

The validation studies described in this V&V plan should be carried out under NEAMS auspices and funding. The NEAMS role is fundamental in performing sufficient verification and validation in order to demonstrate capability and accuracy for early users.

2. Description of Computational Model

2.1 Phenomena Identification

We are interested in validating PROTEUS for modeling unprotected loss of flow events in metal-fuel SFRs with a core restraint system such as the ABTR. The ABTR is a conceptual advanced sodium-cooled nuclear reactor using weapons-grade plutonium ternary metal (U-TRU-10Zr) for its reference fuel. The ABTR design uses a core restraint design concept similar to the Clinch River Breeder Reactor (CRBR) or the Fast Flux Test Facility (FFTF) which limits the assembly bowing under high temperature conditions to an optimal shape.

During an unprotected loss of flow transient in a SFR, the sodium mass flow rate through the core decreases causing the sodium coolant and the fuel temperatures to rise. Initially, the rising sodium coolant temperature causes an increase in the power due to the reduced density and reduced moderating power of the coolant. The rising temperature of the system causes mechanical deformations from thermal expansion and, because of the restraint system, bowing of the lattice of assemblies. The core bowing provides a negative reactivity feedback due to increased leakage (larger core volume with fixed mass of heavy metal leads to a lower criticality state in fast spectrum systems), which decreases the power production. The increase in fuel temperature alters the material cross sections and increases resonance absorption due to the Doppler broadening effect. Therefore, ULOF results in the activation of passive safety mechanisms which provide a natural limit to the temperature and power levels that can occur during a transient. We summarize the key physical phenomena in Table 1.

In order to properly predict the sodium void reactivity feedback, thermal expansion reactivity feedback, and Doppler temperature feedback, PROTEUS must be able to accurately model deformed geometry and account for temperature feedback in the interaction cross sections. The geometrical deformations and temperature changes must be calculated in other physics modules based on the power distribution computed by PROTEUS, but they can be assumed to take the form of some type of distribution for this “single-physics” validation exercise.

Table 2 lists the desired neutronics simulation quantities in order of increasing detail and difficulty that should be predicted to assess the impact of the phenomena in Table 1. Essentially, the eigenvalue, flux, and power distributions should be calculated for various states of temperature and geometry (varying between cold initial state and hot deformed geometry). The maximum pin power location should also be predicted in order to pinpoint peak temperatures for material considerations. To model time-dependent phenomena, sufficiently small time steps should be taken such that the neutronics and thermal hydraulics solutions are converged on the correct power and temperature. However, that problem requires multi-physics coupling via the SHARP toolkit and is not considered here (it should be considered in a SHARP V&V plan).

Table 1. List of Phenomena for Computational Model of Interest.

Phenomenon	Description
Sodium Void Reactivity Feedback	As the temperature of the sodium coolant increases, the density of the sodium decreases. This provides decreased moderation which leads to a positive reactivity feedback in a SFR (better fission/capture ratio).
Thermal Expansion Reactivity Feedback	As the temperature of structural and fuel materials increases, they geometrically expand. This causes the core as a whole to bow (or flower, depending upon the core restraint system). The larger core volume with a fixed mass yields a negative reactivity feedback.
Doppler Temperature Feedback	As the temperature of the fuel and structural materials increase, the resonances of the neutron cross section data broaden which leads to more resonance absorption of neutrons due to reduced energy self-shielding.
Time Dependent Behavior	The fission process itself has a small component of neutrons that are time delayed (lagged) and do not appear promptly in time. These delayed neutrons are the primary means by which reactor systems can be held at a stable power level. With regard to coupled multi-physics behavior, it is important to properly account for the time lagged behavior and distribution of these neutrons to ensure that proper power peaking is properly predicted. No transient will yield a uniform bowing with respect to time, and thus the transition in power over time will also not be uniform.

Table 2. Neutronics Simulation Quantities of Interest.

Priority	Description
1	Eigenvalue
2	Assembly-averaged reaction rates and power + axial distribution
3	Pin-cell-averaged reaction rates and power + axial distribution
4	Radial and axial power distribution of each fuel rod in a core with simplified geometry (e.g., no spacers)
5	Radial and axial power distribution for each fuel rod in a core with exact geometry
6	Above quantities with effect of temperature feedback
7	Above quantities with effect of structural mechanical feedback

2.2 Description of Physics and Empirical Models

The PROTEUS code solves the neutron transport equation to yield average characteristics of the neutron population such as the neutron flux distribution and k-effective (a measure of criticality). The approximations in PROTEUS are generally related to the user's choice of discretization in the space, angle, and energy phase spaces. Use of the multigroup approximation is inherent in the code. Convergence studies in each phase space must be performed by the user to ensure a sufficient level of fidelity has been chosen. Additionally, PROTEUS incorporates an

approximate time-dependent method (adiabatic kinetics) which is less accurate than full spatial kinetics. While this approach is likely appropriate for most transients of interest, there are numerous transients for which it is known to be insufficient. PROTEUS also does not perform a gamma transport calculation and therefore gives a power distribution based only upon the fission rate distribution. The gamma rays that are normally produced by fission (and those by neutron-gamma reactions) are assumed to be absorbed in the reacting material . This primarily over predicts the power in the fuel regions as a significant amount of power (2-3%) is deposited in the steel reflectors and shielding that surround the reactor core via gamma heating.

In a multi-physics calculation, we are specifically interested in capturing the multi-physics reactivity feedback due to temperature-dependent cross section data as well as structural-mechanical feedback due to assembly bowing. The model would ideally be performed with varying levels of spatial detail: homogenized, partially homogenized, or fully heterogeneous, depending on the available computing resources. Homogenization is typically used in specific physical regions or when lower spatial fidelity of the solution is required. PROTEUS can theoretically model any level of spatial heterogeneity including fuel pins, grid spacers, and assembly ducts, but due to computational expense and low impact on the neutronics solution, typically PROTEUS calculations do not consider modeling details such as the wire-wrap around the fuel pins or geometric details far from the active core.

Due to extensive historical use, the assembly-homogenized method is well validated and verified for reactor analysis, and there is extensive literature on the appropriate way to generate multigroup cross section data. Assembly-homogenized codes like DIF3D [13] are used routinely for fast reactor analysis, and the use of PROTEUS for such problems should be straightforward. However, the partially homogenized and fully heterogeneous methods have not previously been extensively explored in the literature due to lack of unstructured geometry transport codes. Furthermore, appropriate cross section generation tools for heterogeneous geometry must also be developed.

PROTEUS depends on multigroup cross section input data, which are typically generated for SFRs by the MC^2-3 code. The generation of multigroup cross section data is a multi-level, multi-step procedure which collapses the point-wise energy-dependent cross section data into a set of finite values, called multigroup constants, which are defined for each “energy group” (finite ranges in energy space). The values of the multigroup constants for each group are determined by a series of consistent calculations which attempt to preserve reaction rates in the geometry over the energy range in the group. While a homogenized geometry cross section methodology has been widely used in the past, the more recent application of heterogeneous geometry cross sections requires additional studies. To apply temperature feedback, the cross sections can be generated in MC^2-3 for assumed temperature distributions and then interpolated by PROTEUS at runtime. This introduces some amount of error based upon the assumed temperature distribution (relative changes in fuel, structure, and coolant temperature).

2.3 Required Toolkit Elements and Libraries

The PROTEUS code is part of the NEAMS Reactor Product Line SHARP Toolkit. PROTEUS itself depends on various libraries:

- MPICH: message passing/parallel processing [14]
- METIS: online mesh decomposition [15]
- PETSc: parallel linear algebra solver [16]
- HDF5: parallel and portable I/O (optional) [17]
- MOAB: multiphysics calculations (optional) [18]

Finite element mesh software such as Cubit [19] or the MeshKit Reactor Geometry Generator [20] which also utilizes Cubit is required to create the initial EXODUS II-type finite element mesh input for PROTEUS. PROTEUS includes an EXODUS II mesh converter and can therefore indirectly utilize EXODUS II-type meshes. PROTEUS also allows users to manually create meshes and has significant checks in place to verify the correctness of such meshes. Additionally, a means of generating accurate multigroup cross sections such as the MC^2-3 code is also required. Generation of accurate multigroup cross sections is a fundamental issue in deterministic transport simulations, so the multigroup cross section input data must also be verified.

2.4 Consequence of Failure, Quality Rigor Level, Risk Grading

PROTEUS simulations are used to gain a basic understanding of the neutronics behavior including finely detailed local flux and power solutions. However, the use of inadequate cross section generation techniques or inadequate space-angle-energy discretization is known to bias the resulting solution. Verification of heterogeneous cross sections and discretization convergence studies are required to verify convergence to the correct solution. Additionally, lack of appropriate physics models (coupled neutron gamma transport, advanced kinetics) can also degrade the quality of the neutronics solution.

PROTEUS is currently a Quality Rigor Level 3 (QRL) code, which is a “research and development activity that is exploratory, preliminary, or investigative in nature” [5]. As a QRL 3 code, PROTEUS must meet verification requirements such regular regression testing, error tracking, and configuration management with documentation. It must also perform unit test validation and sensitivity studies. PROTEUS currently meets these standards, however, the goal is to build up the V&V of PROTEUS such that it meets the more rigorous QRL 2 requirements.

The consequences of failure are poor prediction of the physics phenomena, including overestimation or underestimation of the structural mechanical reactivity feedback and temperature feedback. Additionally, in a coupled multi-physics scenario, inaccurate solution transfer would affect the other physics modules. However, since QRL 3 codes are understood to be exploratory in nature, any conclusions drawn from PROTEUS in its current state should stay within QRL 3 expectations, and not used in a QRL 2 manner to inform important decisions.

Again, the goal is to transition PROTEUS to a QRL 2 code whereby it will be much better verified, validated, documented, etc.

2.5 Effect of Hardware on Validation

PROTEUS contains Fortran 90 source code with C preprocessing directives. The code and its dependent libraries are typically compiled in Linux/Unix environments using the Intel 10.1+ compiler. The code has been successfully tested on various hardware architectures running a UNIX OS including Intel Xeon, Cray XT5 (AMD Opteron), IBM BlueGene/P (PowerPC 450) and BlueGene/Q (PowerPC A2). The code uses pure MPI (message passing interface) and does not yet utilize threading. Therefore the code, at this point, does not take advantage of hardware threads or GPU devices.

A series of benchmark tests and basic regression tests are provided with the distribution in order to verify proper installation and functionality with the given hardware and compiler. After each installation on a new machine, the benchmark tests and regression tests should be performed using a simple command. A script will execute all of the tests and report any differences from the given "reference" solution provided with the installation. If differences are reported, the development team investigates the cause of the problem which can result from different implementations of MPI on the hardware, or varying levels of rigor in the local compiler which may cause errors. Typically, once the PROTEUS dependencies (METIS, PETSc, etc.) have been individually installed successfully, PROTEUS passes all of its benchmark tests. In the rare cases that a problem is found, it should be detected immediately by the tests.

2.6 Problem Setup and Analysis

The PROTEUS code requires a finite element mesh, angular cubature, and material multigroup cross sections data. The finite element mesh and angular cubature can be refined relatively straightforwardly by increasing the resolution to reduce discretization errors. The multigroup cross section generation process, however, has never been used for heterogeneous geometry problems, and requires significant research to examine its viability for large heterogeneous problems. Heterogeneous cross section generation must be done in conjunction with high-fidelity code development as errors in the input cross section data have a large influence on the accuracy of the transport solution.

With the exception of the multigroup cross section data, the input files to PROTEUS are generally ASCII (text-based) files and portable to any platform. We note that the multigroup cross section data is typically generated as binary file, and a Binary-to-ASCII conversion tool is available to convert port the cross section data to/from any platform.

PROTEUS generates output files are in portable ASCII and HDF5 formats. The HDF5 file contains the full solution (flux, power, absorption, mesh everywhere in the problem) and can be visualized and queried using the VisIt software tool [21] which has a special "UNIC" plugin for reading PROTEUS-generated files.

3. Model Verification and Validation

3.1 Software Verification

Software verification is the process of ensuring that the software satisfies the expected requirements and was built according to expectation. It answers the question whether the software is solving the problem as intended by the programmer. Software verification is important in order to eliminate mistakes or bugs in the code as a source of error. In this section we discuss ways to ensure software verification of PROTEUS.

3.1.1 Software Configuration Management

Currently, various files are modified by hand to configure PROTEUS with different packages or capabilities. The files which can be modified to change configuration are:

- PROTEUS_Preprocess.h contains C-preprocessing directives via variable definitions. For example, one can choose to enable HDF5, MOAB, and/or the cross section API.
- Makefile.arch contains architecture-specific compiler locations and library paths for easy and consistent compilation on a given architecture.
- Makefile contains targets and instructions for compilation.

Improvements to this procedure should be made such that the user does not have to modify files in different locations to achieve a single purpose. Additionally, the compile-time configuration used to compile the code should be recorded in a log and/or outputted in the code for debugging purposes.

3.1.2 Software Version Control

PROTEUS is hosted in a private, access-controlled Subversion (SVN) repository at Argonne National Laboratory which we refer to here as the primary repository. The expert developers of PROTEUS work with this primary repository, and all changes to the source code are committed with comments and tracked in the primary repository. Exported versions are hosted on a backed-up filesystem. However, no consistent export naming procedure (i.e. Version 1.0, with date and corresponding documentation) has been established. This should be improved such that released versions are easily tracked.

Additionally, a secondary repository, known as the SHARP repository, holds a static exported version of PROTEUS which is available to a wider range of people. That repository should be updated to pull in the latest version of PROTEUS, rather than existing independently. Currently, SHARP developers (who typically deal with the MOAB routines) are modifying parts of PROTEUS in the SHARP repository, and these changes are being made to outdated code and not being propagated back to the primary repository. This leads to essentially two branches of the code being developed which is not good practice.

3.1.3 Regression, Benchmark Tests, and Bug Reporting

The code is sometimes ported to new platforms and compilers, and updated to add or modify features. In order to ensure the code is working properly after (a) installation on a new architecture, (b) compilation with a different compiler, or (c) modifications to the source code, a series of regression tests and benchmark tests should be performed.

Regression tests should perform unit testing on basic functionality of all the subroutines. The goal is to test all functions used by PROTEUS in a bottom-up fashion. Starting with testing low-level functionality (such as basic operations like sorting) and working all the way up to the high-level routines can eliminate specific subroutines as sources of potential error, such that the error must be in the logical flow of the program.

PROTEUS currently includes a regression test which performs unit tests for some of the basic routines using a framework called FRUIT (FoRtran UnIt Testing) available online [22]. FRUIT provides assertion subroutines and functionality to report how many tests were successful. While a few of the basic modules are currently being tested, we are far from goal of testing all modules. Additionally, PROTEUS needs to undergo regression testing on a nightly basis.

Benchmark tests are a series of full transport problems which test the integrated behavior of the code. Reference solutions, known either from analytical solutions, another code, or some other technique should be provided in order to compare the most recent solution. All differences between the recent solution and the reference solution should be reported as a failed test. Sometimes, round-off errors cause negligible differences from the solution – these can and should be ignored by using a smart comparison script.

PROTEUS currently includes a series of basic benchmarks problems for steady-state, kinetics, and the cross section API. However, the reference solutions for these tests are not necessarily from analytical solutions or from another code. We should improve the benchmark testing suite to verify that all parts of the code are tested and the provided reference solutions are accurate.

Problems or bugs should be reported directly to developers or to nera-software@anl.gov (developer email list). A ticket in the SVN TRAC system should be created to describe the bug. After resolution of the problem, the ticketing system shall be updated to describe the solution, refer to the SVN revision number containing the fix, and close the ticket. Once the code is more widely distributed, we need to follow this procedure more closely and track the version numbers which contain bug fixes. Updated versions should be transmitted or at least communicated to the users when applicable.

3.1.4 Documentation

A user manual [23] and methodology manual [24] for PROTEUS are provided with the distribution, or upon request. The methodology manual describes the physics being solved, and methodology used in PROTEUS. The user manual describes how to obtain, install, and simulate problems using PROTEUS. Additionally, training presentations are available which cover the

same material. File format descriptions are also available for the two structured input files (cross section data and finite element mesh).

3.1.5 Static Analysis

Static analysis is the process of checking that software meets requirements by doing a physical inspection. Static analysis is not currently performed, as the dynamic analysis (regression and benchmark tests) fully cover the required testing. However, static analysis could be implemented easily if there was a need.

3.2 Verification with Analytical Benchmarks

Some analytical or semi-analytical benchmarks exist [25,26] to test the mechanics of transport codes for nuclear engineering applications. These benchmarks typically contain simple geometries and/or conditions such that it is possible to obtain analytic transport solutions. The benchmarks range from infinite medium slowing-down (continuous energy, no space dependence) to multigroup transport in 3D (multigroup approximation, 3D space dependence). While these benchmarks cannot be expected to represent the computational model at hand, they do provide confidence that the code algorithms are representing the basic physics correctly. Additionally, as PROTEUS now has a fixed source capability, the Method of Manufactured Solutions (MMS) could be used to generate analytic solutions to simple problems.

3.3 Verification with Code-to-Code Comparison

Due to the limited availability of experimental data for particular regimes of interest, newer software can be verified against other more established codes or reference solutions. Monte Carlo transport is typically used to generate a reference solution due to the absence of discretization errors and ability to represent other geometries. Specific problems can be devised to test the physics of interest. For example, a model can be created for the ABTR and compared against the Monte Carlo stochastic solution to assess the entirety of discretization errors introduced by the deterministic code. Monte Carlo can also be used in multigroup mode using the same cross sections as the deterministic model to further assess the errors introduced by the multigroup cross section generation procedure. Furthermore, well-established deterministic codes using the same (or different) methodology can be used to verify how much the solution differs. Official benchmarks [27] contain numerous code entries and reference solutions which can be compared against. One recent benchmark [28] includes spatial heterogeneity.

3.4 Relevant Experimental Data for Validation

Software validation is the process of determining how well the given simulation software accurately models the real-world problem of interest. It requires comparison of numerical results to real data (typically measured by experiment or known analytically) for each use case or computational model of interest. Software validation is a complex exercise which expands in breadth as problems with different physics regimes are added to the list of use cases. Here we

discuss the required software validation only for the computational model of interest: using PROTEUS to model neutronics effects in a SFR under transients such as ULOF.

Several experiments have already been performed that provide data relevant to our problem of interest and are listed in Table 3. In particular, significant validation data exists for the Zero Power Reactor (ZPR) and Zero Power Physics Reactor (ZPPR) series of experiments performed at Argonne National Laboratory. Of these experiments, the ZPPR-15 experimental configuration is most relevant to this validation exercise. It also has the best available detailed specification and measurement data. The EBR-II and FFTF datasets are closely aligned with the computational model of interest, but the corresponding neutronics specifications and measurements are more limited and/or have an unknown status.

Table 3. Relevant Experiments for Validation.

Facility	Tests	Brief Description
ZPPR-15, ANL, USA	Phase A L015 Phase B L088 Phase C L166 Phase D L185	Criticality measurements for different phases (core compositions) of metal-fueled sodium-cooled IFR.
ZPPR-15, ANL, USA	Phase A L016-L022 Phase B L091-L094 Phase C L167-L169 Phase D L184, L189-L190	Sodium void worth experiments for different phases (core compositions) of metal-fueled sodium-cooled IFR.
ZPPR-15, ANL, USA	Phase A L025-L028 Phase B L096, L100, L102-L106 Phase D L193-L195, L198-L199	Control rod worth (B_4C) experiments for different phases (core compositions) of metal-fueled sodium-cooled IFR.
ZPPR-15, ANL, USA	Phase A L034, L048 Phase B L079, L080, L115, L123, L134, L140, L143, L144 Phase C L171 Phase D L203, L215	Flux and reaction rate measurements on different foils (plutonium, enriched uranium, and depleted uranium) for different phases (core compositions) of metal-fueled sodium-cooled IFR.
ZPPR-15, ANL, USA	Phase A L061 Phase B L117 Phase D L206	Fuel axial expansion tests for different phases (core compositions) of metal-fueled sodium-cooled IFR.
ZPPR-15, ANL, USA	Phase A L043 Phase B L139 Phase D L207	Doppler feedback due to heating a test material in a single drawer of metal-fueled sodium-cooled IFR.
*EBR-II, ANL,	SHRT-17 and SHRT-	Shutdown heat removal test to demonstrate

Facility	Tests	Brief Description
USA	45R test series (1984-1986)	passive safety features including natural circulation decay heat removal and passive shutdown in ULOF and LOHS events.
**FFTF, Hanford, WA, USA	Inherent Safety Test Series (1986)	Test facility for breeder reactor program principally supporting oxide fuel development for CRBR. Tested natural circulation shutdown heat removal and passive power reduction in unprotected ULOF.
**Neutronics specification and measurement data is limited/incomplete* ***Availability and completeness of data is unknown at this time*		

3.4.1 ZPPR-15 Experiments

Argonne National Laboratory's ZPPR-15 series of experiments consisted of large, metal fuel, sodium cooled fast reactor tests performed for the Integral Fast reactor program. The series included 4 phases (termed A, B, C, and D) differing by their core compositions. Phase A contained stainless steel, sodium, plutonium and depleted uranium fuel (no zirconium). Phase B contained zirconium in addition to the Phase A materials. Phase C transitioned to 50% enriched uranium + 50% plutonium fuel. Phase D transitioned to 90% enriched uranium + 10% plutonium fuel (BOL design). The fuel composition in Phase B is the most similar to the ABTR core design. Measurements of sodium void worth, control rod worth, foil flux and reaction rates, Doppler effect, and fuel expansion effects were taken in various combinations for Phases A-D. As-built models (explicit geometry specifications) and evaluated measurement data are available for many of the different tests although the latter results are still being processed [29].

The flux and reaction rate foil measurement data provide a good baseline to begin validation of PROTEUS for the reference configuration using various models (homogeneous, partially homogeneous, and heterogeneous geometry). Additionally, the control rod worth experiment is directly relevant as the ABTR uses B_4C control rods similar to the experiment.

The sodium void experiment was performed by measuring a reference configuration where a given drawer contains sodium, then removing the sodium and re-measuring the data. The experimental conditions are more complex than the real-life scenario: a sodium-filled plate is replaced with an empty plate, which introduces neutron streaming that is difficult to model. Regardless, the effect of changes in the sodium coolant is important for SFRs.

Most interestingly, experiments were performed to evaluate the effect of thermal expansion of fuel plates in ZPPR. The plates were not heated but instead manually pulled from the back of the drawer to provide small geometrical variations in the distance between plates. While the geometry change is different in the experiment from that would occur in a hexagonal lattice core, this experimental data would be useful to validate against.

Finally, Doppler experiments were performed whereby a "Doppler mechanism" which heats surrounding material was installed in a specific drawer and the change in reactivity measured. While these experiments could be useful, the measured reactivity was extremely small (< 1 pcm) and may be within the noise of discretization errors. We will likely need to find another experiment which raises the temperature of a larger portion of the reactor.

3.4.2 EBRI-II Shutdown Heat Removal Tests

Argonne National Laboratory's Experimental Breeder Reactor II (EBR-II) was a liquid metal reactor with a sodium-bonded metallic fuel core. The SHRT-17 test [30] in 1984 demonstrated the effectiveness of natural circulation in EBR-II under severe loss-of-flow test conditions. The follow-up SHRT-45R [30,31] test performed in 1986 was similar to SHRT-17 except the plant protection system (PPS) was disabled to prevent it from initiating a control rod scram. Both SHRT-17 and SHRT-45R were initiated by tripping the primary coolant pumps and the intermediate loop pump to simulate a loss-of-flow accident. In SHRT-45R, the loss of forced coolant flow caused the reactor temperature to rise temporarily but eventually shut down due to negative reactivity feedback from thermal expansion. A variety of temperature and flow data exists for these tests, but the neutronics data is likely to be incomplete (fuel composition at time of experiment estimated only by modeling, and few neutronics parameters were measured). It is important to note that the core was rearranged such that a small, but negative sodium density coefficient resulted which ensured a positive and safe experiment outcome.

3.4.3 FFTF Inherent Safety Test Series

A series of tests was performed at the 400-MW (thermal) Fast Flux Test Facility (FFTF) to demonstrate the passive safety characteristics of liquid-metal-cooled fast reactors [32]. In 1981, FFTF tests were performed to measure decay heat removal by sodium natural circulation. The 1986 test series demonstrated passive reactor shutdown during a loss-of-flow event when several inherent shutdown devices called gas expansion modules (GEMs) were installed in the reactor. However, these tests also provided additional data on the natural circulation performance of the primary system, in particular the reactor core, and thus add to the data base available for checking the validity of available analytical tools. The exact type and amount of data collected is unknown at this time, but the data is owned by DOE [32]. There is an active program supported by Advanced Reactor Technology (NE-74) to retrieve the FFTF data with the objective of initiating either a national or international benchmark activity. The FFTF tests demonstrated the contribution of radial core expansion feedback to shutting down the fission chain reaction.

3.5 Unit, Component, Subsystem, System, Integral tests, etc.

A hierarchy of tests should be performed to comprehensively validate the PROTEUS code, known as unit, component, subsystem, system, and integral tests. Unit tests are typically simple math problems where the solution is known analytically. A component test typically contains single physics effects (neutronics) and could consist of a reactor core with no multi-physics feedback, for example. A subsystem test example would be a reactor core with multi-physics

feedback. A system test typically models the entire system of the nuclear reactor. Integral tests in neutronics typically measure the reactor period. The following table gives examples of unit, component, subsystem, and system tests.

Table 4. Examples of Neutronics Unit, Component, Subsystem and System Tests.

Unit	Component	Subsystem	System
Simplified verification of the solution algorithm by dissection and combined MMS and analytical math problems.	SFR core with only neutronics modeled	SFR core with temperature and structural mechanical feedback	Full SFR system as built

In this V&V plan, we have made suggestions for the Unit and Component type validation cases, as well as some tests which could be considered Subsystem tests, as their phenomena are multiphysics in nature. However, the tests we propose in this report do not require PROTEUS to be coupled to thermal hydraulics or structural mechanics codes. The Subsystem and System tests should be detailed in a SHARP Toolkit V&V plan.

3.6 Validation Matrix and Gaps

Table 5 is an example validation matrix describing which tests can be used to validate, or partially validate, the desired phenomena from Table 1.

Table 5. Validation Matrix.

Phenomena addressed by		**Test facility**	ZPPR-15	EBR-II	FFTF
		Test Number	Various Assemblies, Loadings (see Table 3)	SHRT-17 and SRT-45	TBD
1	Sodium Void Reactivity Feedback		Full	TBD	TBD
2	Thermal Expansion Reactivity Feedback		Partial	TBD	TBD
3	Doppler Temperature Feedback		Partial	TBD	TBD
4	Time Dependent Behavior		Partial	TBD	TBD

* TBD: to be determined

The validation matrix is somewhat incomplete as the ZPPR-15 experiments do not fully test Doppler temperature feedback or thermal expansion in the way that it occurs in the full core, hexagonal lattice SFR. While we can pull data from experiments, these are typically smaller and have considerably different spectrum than the target ABTR. With respect to FFTF, it is by far the most appropriate reactor system but the complete lack of available experimental details precludes

our ability to either exclude or include it. It is unlikely that new experiments would be performed to gather measurements for these phenomena, although they would be desirable.

3.7 Schedule and Priorities

The validation of PROTEUS for each experiment is an extensive task and consists of the following steps:

1) Identification of experiments beyond those listed in this report
2) Collection of experiment specification and measurement data
3) Requesting computer time, if necessary
4) Building the basic model in PROTEUS
5) Initial testing of model and optimization of model parameters
6) Summarizing results in report

This report identifies some relevant experiments, although other experiments may come to our attention with further, in-depth, research. The next task will consist of gathering the actual experimental specification and measurements in order to characterize all details of the model: this step could take the longest since it is not necessarily within the control of the PROTEUS team. Following this step, a basic model in PROTEUS can be built and optimized to best agree with measured data. Building and optimizing the model can take several months due to the dependence on good material cross section data, geometry, mesh, and angular cubature. Additionally, large problems require large computational resources. Consistent support for validation must be provided such that the PROTEUS team can request appropriate computer time in advance. Finally, a report should be written to communicate the findings.

4. Sensitivity Analysis and Uncertainty Quantification

4.1 Sensitivity Analysis

Sensitivity analysis in neutronics is the quantification of how much the neutronics solution (flux, power, eigenvalue) change as certain input parameters change. The neutronics solution can change with respect to the following key physical inputs: cross section data (due to fundamental data uncertainty or due to multigroup approximation), temperature, geometry, and composition.

Material cross section data can have significant uncertainties depending on the energy and isotope of interest to the problem. Sensitivity analysis should be carried out to determine the effect of experimental uncertainty on the final solution. Various techniques are available to measure sensitivity to cross section data including perturbation theory using the adjoint solution.

Additionally, the impact of using different temperatures, structural deformations, and compositions will result in a different neutronics solution which should be quantified via sensitivity analysis.

4.2 Uncertainty Quantification

Uncertainty is divided into two categories: calculation and experimental uncertainties. The experimental uncertainty is composed of the uncertainties of measurement, material, and geometry, i.e., due to the inherent experimental uncertainty in measurements, and due to the manufacturing tolerances for any given structural component (size, density, and composition). The calculation uncertainty may be evaluated from the statistical distribution of differences between measurement and prediction. The prediction accuracy depends upon the computational models and methodologies that the prediction code system employs. It is desirable to be able to quantify the uncertainty in a given neutronics solution based on this input uncertainties. However, little work has been done in this error to propagate all of these uncertainties through the system of equations to see their effect. The state of the art UQ techniques will need to be analyzed to see which best apply to multi-physics reactor calculations.

5. Path Forward

The first task in V&V of PROTEUS is to verify and validate the use of multigroup cross section generation procedures in heterogeneous calculations. Without accurate cross section data, heterogeneous calculations will not obtain more accurate solutions than the traditional homogenized methodologies. We have already begun to develop on-line cross section generation procedures (using the cross section API) and generalized cross sections but these must still be extensively verified and validated. Changes to the methodology will be required if sufficient accuracy cannot be obtained with current methods.

An additional path is to verify the transport method in PROTEUS independently of the multigroup cross section data by comparing PROTEUS calculations against Monte Carlo reference solutions which use the same multigroup cross section data. Such comparisons have been performed in the past and proven instructive for quantifying the source of error from the cross section constants versus the transport methodology. However, ultimately the generation of accurate multigroup cross sections is key to the predictive accuracy of the transport calculations.

Validation should begin with various ZPPR-15 tests, especially Phase B, which are very detailed and contain physics relevant to the computational model of interest. However, the ZPPR-15 experiments use a plate-type geometry which significantly differs from the proposed computational model, and do not exhibit natural circulation or rod bowing. Additional experiments which more closely resemble the proposed computational model will be required. In order to move forward with verification and validation of PROTEUS for this specific computational model, more information should be requested about the EBR-II and FFTF series of experiments to determine their relevance and feasibility for validation exercises. If EBR-II and FFTF cases include useful and complete data, these should be obtained and examined in addition to the stated ZPPR tests. If they are not available or are unsuitable for validation exercises, a new set of experiments must be proposed in order to perform validation for this work.

REFERENCES

1. 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].
2. M. A. Smith, et al, "Recent Research Progress on UNIC at Argonne National Laboratory," *Proc. International Conference on Mathematics, Computational Methods and Reactor Physics*, Saratoga Springs, New York, May 3-7, 2009, American Nuclear Society, 2009 (CD-ROM).
3. C. H. Lee and W. S. Yang, "MC^2-3: Multigroup Cross Section Generation Code for Fast Reactor Analysis," ANL/NE-11-41, Argonne National Laboratory, January 2012.
4. 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.
5. R. Versluis, "NEAMS Software Verification and Validation Plan Requirements, Version 0," August 20, 2013. Available via http://energy.gov/ne/downloads/nuclear-energy-advanced-modeling-and-simulation-neams-software-verification-and.
6. R. Versluis, Personal communication, April 2014.
7. W. D. Pointer et al., "Developing a Comprehensive Software Suite for Advanced Reactor Performance and Safety Analysis," 2013 IAEA Conference on Fast Reactors, Paris, France, March 4-7, 2013.
8. Y. I. Chang, P. J. Finck, and C. Grandy, "Advanced Burner Test Reactor Preconceptual Design Report," Argonne National Laboratory report ANL-ABR-1 (ANL-AFCI-173), September 2006.
9. V. Mahadevan, et al, "SHARP Coupled Multiphysics Toolkit for Reactor Analysis," *Proceedings of ICAPP 2014*, Charlotte, USA, April 6-9, 2014 [CD-ROM].
10. A. Siegel, et al, "Software Design of SHARP," in *Proceedings of the Joint International Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications (M&C + SNA)*, American Nuclear Society, April 2007.
11. D. Parsons, et al, "Diablo User Manual," Lawrence Livermore National Laboratory Report UCRL-SM-234927, Sept 2007.
12. P. F. Fischer, J. W. Lottes, S. G. Kerkemier, Nek5000 Web Page, http://nek5000.mcs.anl.gov, 2008.
13. K. L. Derstine, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Diffusion Theory Problems," ANL-82-64, Argonne National Laboratory, 1982.
14. MPICH2 Website, www.mcs.anl.gov/mpich2.

15. G. Karypis and V. Kumar, "A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs," SIAM Journal on Scientific Computing, Vol. 20, No. 1, pp. 359—392, 1999.

16. S. Balay, J. Brown, et al, PETSc Web page, http://www.mcs.anl.gov/petsc, 2013.

17. The HDF Group. Hierarchical Data Format, version 5, 1997-2014. http://www.hdfgroup.org/HDF5/

18. T. J. Tautges, H. J. Kim, A. Caceres, and R. Jain, "Coupled Multi-Physics simulation frameworks for reactor simulation: A Bottom-Up approach," in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C)*, American Nuclear Society, Rio de Janeiro, Brazil, May 2011.

19. CUBIT Website, https://cubit.sandia.gov.

20. R. Jain and T. J. Tautges, "NEAMS MeshKit: Nuclear Reactor Mesh Generation Solutions," *Proceedings of International Congress on the Advances in Nuclear Power Plants (ICAPP)*, American Nuclear Society, Charlotte, NC, April 6-9, 2014.

21. H. Childs, et al, "A Contract-Based System for Large Data Visualization," Proceedings of IEEE Visualization 2005, Minneapolis, MN, pp 190-198, 2005.

22. FORTRAN Unit Test Framework (FRUIT) Website, sourcefourge.net/projects/fortranxunit/

23. E. R. Shemon, M. A. Smith, C. H. Lee, and A. Marin-Lafleche, "PROTEUS-SN User Manual (Revision 1.0)," ANL/NE-14/6, August 4, 2014.

24. E. R. Shemon, M. A. Smith, and C. H. Lee, "PROTEUS-SN Methodology Manual," ANL/NE-14/5, Argonne National Laboratory, June 30, 2014.

25. B. D. Ganapol, *Analytical Benchmarks for Nuclear Engineering Applications, Case Studies in Neutron Transport Theory*, NEA/DB(2008)1, Paris, France, ISBN 978-92-64-99056-2, OECD 2008, NEA No. 6292.

26. Argonne Code Center, *Benchmark Problem Book: Numerical Determination of the Space, Time, Angle or Energy Distribution of Particles in an Assembly Prepared by the Benchmark Problem Committee of the Mathematics and Computation Division of the American Nuclear Society*, ANL-7416 (TID-4500), Argonne National Laboratory, Argonne, IL, February 1968.

27. Nuclear Science and Data Bank Publication Website, http://www.oecd-nea.org/tools/publication?query=benchmark&div=SCI-DB&lang=English&period=100y&sort=title&filter=1#p3135.

28. *Benchmark on Deterministic Transport Calculations Without Spatial Homogenization, A 2-D/3-D MOX Fuel Assembly Benchmark*, NEA/NSC/DOC(2003)16, Paris, France, ISBN 92-64-02139-6, OECD 2003.

29. R. Lell, Personal communication, Argonne National Laboratory, August 2014.
30. T. Sumner and T. Y. C. Wei, Unpublished Information, Argonne National Laboratory, May 31, 2012.
31. T. Fei, A. Mohamed, and T. K. Kim, Unpublished Information, Argonne National Laboratory, January 23, 2013.
32. T. M. Burke, et al, "Results of the 1986 FFTF Inherent Safety Tests," *Trans. Am. Nucl. Soc.*, **54**, p 249-250, 7-11 June 1987.

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

www.anl.gov

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