FY17 Status Report on NEAMS Neutronics Activities

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

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.
FY17 Status Report on NEAMS Neutronics Activities

prepared by
C. H. Lee, Y. S. Jung, and M. A. Smith
Nuclear Engineering Division, Argonne National Laboratory

September 30, 2017
EXECUTIVE ABSTRACT

Under the U.S. DOE NEAMS program, the high-fidelity neutronics code system has been developed to support the multiphysics modeling and simulation capability named SHARP. The neutronics code system includes the high-fidelity neutronics code PROTEUS, the cross section library and preprocessing tools, the multigroup cross section generation code MC²-3, the in-house meshing generation tool, the perturbation and sensitivity analysis code PERSENT, and post-processing tools.

The main objectives of the NEAMS neutronics activities in FY17 are to continue development of an advanced nodal solver in PROTEUS for use in nuclear reactor design and analysis projects, implement a simplified sub-channel based thermal-hydraulic (T/H) capability into PROTEUS to efficiently compute the thermal feedback, improve the performance of PROTEUS-MOCEX using numerical acceleration and code optimization, improve the cross section generation tools including MC²-3, and continue to perform verification and validation tests for PROTEUS.

The PROTEUS-MOCEX code was significantly updated to improve both performance and usability by implementing the CMFD acceleration scheme and optimizing the transport solver for use on large scale computing machines. Test results using the 3D C5G7 and TREAT problems indicated that the overall performance of the updated PROTEUS-MOCEX runs over 10 times faster than the previous version. The restart capability was also implemented to PROTEUS-MOCEX to enhance the code usability when running the code for large-scale 3D problems. Output processing tools were also developed for improved user support.

The PROTEUS-NODAL code was developed based upon the DIF3D-VARIANT methodology to provide a rapid and accurate calculation capability for users focusing on the scoping and design type analysis activities of nuclear reactors. The diffusion option now works for Cartesian, hexagonal and triangular-z geometries. In addition, the nodal simplified P₃ (SP₃) solution option in combination with the radial triangle based polynomial expansion method and the axial nodal expansion method was added to expand the functionlity past simple diffusion theory. Verification tests with the ABTR and MONJU fast reactor benchmark problems indicated that the PROTEUS-NODAL results agreed well with the reference DIF3D-VARIANT solutions for eigenvalues fuel assembly powers.

Because Nek5000 requires substantial computational resources, a built-in T/H solver based on a simplified sub-channel model was added to PROTEUS. This capability was made available to both PROTEUS-NODAL and PROTEUS-MOCEX. To allow cross section variance with respect to state parameters including temperature, a tabulated cross section data set named ISOPAR was created for PROTEUS along with a utility code to generate ISOPAR data files from base ISOTXS files.
As part of work to continue improving the cross section generation capability, the thermal cross section capability in $\text{MC}^2$-3 was updated using finer groups in the thermal energy range below 5 eV. Verification tests with various types of pin and assembly problems for light water reactors showed that $\text{MC}^2$-3 with the updated thermal cross section library could predict eigenvalues and power distributions more accurately.

For code verification and validation efforts, several thermal reactor problems were modeled using PROTEUS-MOCEX. Through the 3D C5G7 numerical benchmark problems, we could confirm that the MOCEX solutions were converging to the Monte Carlo solutions as the angle and space discretization was refined. Further detailed calculations using MOCEX for TREAT M8CAL core, the RCF core at RPI, and VHTR assemblies demonstrated that MOCEX could predict eigenvalues and power distributions very accurately for challenging problems with large amounts of neutron streaming.
TABLE OF CONTENTS

Executive Abstract ........................................................................................................................... i
Table of Contents ........................................................................................................................... iii
List of Figures ................................................................................................................................ iv
List of Tables.................................................................................................................................. vi
1. Introduction ............................................................................................................................... 1
2. Updates in PROTEUS ............................................................................................................... 4
   2.1 MOCEX Solver ..................................................................................................................... 4
      2.1.1 CMFD Acceleration ....................................................................................................... 4
         2.1.1.1 CMFD Formulations for MOCEX ..................................................................... 4
         2.1.1.2 Modification of MOCEX Solver ............................................................................. 5
         2.1.1.3 Implementation CMFD Solver ............................................................................... 9
         2.1.1.4 Incorporation of CMFD and MOCEX Solvers .................................................. 11
         2.1.1.5 Performance Examinations ............................................................................... 13
      2.1.2 Optimization in Blue Gene/Q ..................................................................................... 17
      2.1.3 Restart Capability ....................................................................................................... 17
   2.2 NODAL Solver ..................................................................................................................... 18
      2.2.1 NODAL-Pn .................................................................................................................... 18
         2.2.1.1 Methodology ......................................................................................................... 19
         2.2.1.2 Modification of the interface current orientation ............................................... 20
         2.2.1.3 Modification for 3D Sweeping .............................................................................. 21
         2.2.1.4 Truncation of the FMGRES vector space allowances ....................................... 22
         2.2.1.5 Verification Tests ................................................................................................. 23
      2.2.2 NODAL-SP3 .................................................................................................................. 24
         2.2.2.1 Methodology ......................................................................................................... 25
         2.2.2.2 Verification Tests ................................................................................................. 28
         2.2.2.3 CMFD Acceleration ............................................................................................. 35
   2.3 Thermal Feedback Capability .............................................................................................. 37
   2.4 Transient capability ............................................................................................................. 42
3. Updates in MC^2-3 .................................................................................................................... 47
4. Utility Codes ............................................................................................................................. 51
   4.1 Cross Section Parameterization Code ............................................................................... 51
   4.2 Post-Processing Code for PROTEUS-MOCEX ............................................................... 52
5. Verification and Validation Tests ............................................................................................ 55
   5.1 C5G7 Benchmark Problem ............................................................................................... 55
   5.2 TREAT Cores ..................................................................................................................... 59
   5.3 Reactor Critical Facility (RCF) Cores .............................................................................. 63
   5.4 VHTR Assemblies ............................................................................................................... 65
6. Conclusions ............................................................................................................................... 68
References ..................................................................................................................................... 70
LIST OF FIGURES

Figure 1. Overview of the NEAMS Neutronics Code Suite ................................................................. 1
Figure 2. Illustration of Fine and Coarse Mesh Structures (Black Color: Fine Mesh, Red Color: Coarse Mesh) .................................................................................................................. 6
Figure 3. Examples of Domain Decomposition Based on Fine and Coarse Mesh Structures ... 7
Figure 4. Illustration of Surface Current Tally Process ..................................................................... 8
Figure 5. Flux Distribution and Neutron Balance of 2D UO$_2$ Assembly Problem (Group 7 out of 7) ......................................................................................................................................... 8
Figure 6. Flux Distribution and Neutron Balance of 2D MOX Assembly Problem (Group 7 out of 7) ......................................................................................................................................... 9
Figure 7. Comparisons of CMFD and Transport Solutions for 2D C5G7 Benchmarks (Relative Error, %) ...................................................................................................................................................... 10
Figure 8. Comparisons of CMFD and Transport Solutions for 3D C5G7 Benchmarks (Relative Error, %) ...................................................................................................................................................... 11
Figure 9. Calculation Flow of 2D/3D with CMFD Acceleration .......................................................... 13
Figure 10. Comparisons of MOCEX Pin Power Distributions w/ and w/o CMFD Acceleration for C5G7 Benchmark Problems (Relative Error in %) ................................................................. 15
Figure 11. Coarse Mesh Structure for TREAT-M8CAL Calculation (Blue: Coarse Mesh, Black: Fine Mesh) .................................................................................................................................................. 16
Figure 12. Comparisons of MOCEX Convergence Behavior w/ and w/o CMFD acceleration for TREAT-M8CAL Problem .................................................................................................................. 16
Figure 13. Comparisons of MOCEX Fission Source Distributions w/ and w/o CMFD acceleration for TREAT-M8CAL Problem .................................................................................................................. 17
Figure 14. DIF3D-VARIANT Surface Trial Function Alignment ......................................................... 20
Figure 15. PROTEUS-NODAL Surface Trial Function Alignment .................................................... 21
Figure 16. Configuration (left) and Thermal Flux Solutions (right) of a Quarter-core of the C5G7 Problem .................................................................................................................................................. 24
Figure 17. Discretization of Hexagonal Node in the TPEN Method .................................................... 26
Figure 18. Unknowns of Local Problem for Hexagon Node in TPEN Method .................................... 27
Figure 19. Calculation Flow of PROTEUS-NODAL-SP3 Solver ....................................................... 28
Figure 20. Assembly-wise Power Distribution for ABTR Benchmark Problem ............................... 31
Figure 21. Assembly-wise Power Comparison for ABTR Benchmark Problem (Relative Error in %) .................................................................................................................................................. 31
Figure 22. Assembly-wise Power Distribution for 200C Core Configuration of Monju Benchmark Problem .................................................................................................................................................. 32
Figure 23. Assembly-wise Power Comparisons for 200C Core Configuration of Monju Benchmark Problem (Relative Error in %) .................................................................................................................................................. 32
Figure 24. Assembly-wise Power Distribution for C1-out Core Configuration of Monju Benchmark Problem .................................................................................................................................................. 33
Figure 25. Assembly-wise Power Comparison for C1-out Core Configuration of Monju Benchmark Problem (Relative Error in %) .................................................................................................................................................. 33
Figure 26. Assembly-wise Power Distribution for C1-in Core Configuration of Monju Benchmark Problem .................................................................................................................................................. 34
Figure 27. Assembly-wise Power Comparison for C1-in Core Configuration of Monju Benchmark Problem (Relative Error in %) .................................................................................................................................................. 34
Figure 28. Comparisons of Fission Convergence Behavior with and without CMFD Acceleration ................................................................. 36
Figure 29. Comparisons of Assembly-wise Power Distribution Obtained with and without CMFD Acceleration for ABTR Benchmark problem (Relative Difference in %) ...... 36
Figure 30. Temperature Profile over the Fuel Pin .......................................................................................................................... 39
Figure 31. Input and Output Files of the MOCEX and NODAL Solvers of PROTEUS ................................................................. 39
Figure 32. Power, Fuel Temperature, and Moderator Temperature Distributions from MOCEX with T/H Feedback ................................................................. 41
Figure 33. Change of Total and Fission Cross Sections due to Temperature Change ................................................................. 41
Figure 34. TWIGL Transient Benchmark Problem .............................................................................................................. 43
Figure 35. Results from PROTEUS for TWIGL Transient Benchmark Problem ................................................................. 44
Figure 36. 2D C5G7-TD Benchmark Problem .............................................................................................................. 45
Figure 37. Power Changes with Time (left) and Power Differences (right) between MPACT and PROTEUS for TD1 Exercises of the C5G7-TD Benchmark Problem ................. 46
Figure 38. Power Changes with Time (left) and Power Differences (right) between MPACT and PROTEUS for TD2 Exercises of C5G7-TD Benchmark Problem ................................................................. 46
Figure 39. Power Changes with Time (left) and Power Differences (right) between MPACT and PROTEUS for TD3 Exercises of C5G7-TD Benchmark Problem ................................................................. 47
Figure 40. UO2 (left) and MOX (right) Neutron Spectra Obtained from MC2-3 and MCNP6 49
Figure 41. ISOPAR File Structure Generated from GenISOPAR Using Multiple ISOTXS Files .............................................................................................................................. 51
Figure 42. Examples of MOCEX Visualization (TREAT-M8CAL) .............................................................................................................. 53
Figure 43. Example of Coarse Mesh Editing Function in MOCEX Post-processing Tool (RCF) .............................................................................................................................. 54
Figure 44. Mesh Structures of MOCEX Models for C5G7 Benchmark Problems .............................................................................................................................. 56
Figure 45. Thermal Flux Distribution of C5G7 Benchmark Problems .............................................................................................................................. 56
Figure 46. Power Distribution of C5G7 Benchmark Problems .............................................................................................................................. 56
Figure 47. Pin Power Comparisons for Unrodded Configuration of C5G7 Benchmark Problems at Upper 1/3 of Fuel Region .............................................................................................................................. 58
Figure 48. Pin Power Comparisons for Rodded A Configuration of C5G7 Benchmark Problems at Upper 1/3 of Fuel Region .............................................................................................................................. 58
Figure 49. Pin Power Comparisons for Rodded B Configuration of C5G7 Benchmark Problems at Upper 1/3 of Fuel Region .............................................................................................................................. 58
Figure 50. 2D Projected Core Mesh of M8CAL .............................................................................................................................. 60
Figure 51. Core configuration of Full 3D M8CAL (Simplified Model) .............................................................................................................................. 61
Figure 52. 3D Fast (left) and Thermal (right) Flux Solutions of PROTEUS for Full 3D M8CAL ............................................................................................................................................. 61
Figure 53. Axially Integrated Fuel Element Power (top row) and Percent Difference (bottom row) Distributions for MinCC (left) and M8CAL (right) ............................................................................................................................................. 62
Figure 54. Neutron Flux Spectra Obtained from Pin and 2D Cores of RCF .............................................................................................................................. 63
Figure 55. Fast (left) and Thermal (right) Flux Solutions from MOCEX for 2D RCF Cores with and without Control Rods ............................................................................................................................................. 64
Figure 56. Power Comparison between MOCEX and Serpent for 2D RCF Cores with and without Control Rods ............................................................................................................................................. 65
Figure 57. Core Configuration (left) and Thermal Flux Distributions (right) for the 3D RCF Core ............................................................................................................................................. 65
Figure 58. Mesh Structures of MOCEX Models for VHTR Assembly Problems ....................... 66
Figure 59. Flux Distributions of 2D VHTR Assembly Problem ............................................. 67
Figure 60. Flux Distributions of 3D VHTR Assembly Problem ............................................. 67

LIST OF TABLES

Table 1. Computing Parameters of MOCEX Calculation with CMFD for C5G7 Benchmark Problems ....................................................................................................................... 14
Table 2. Performance Improvements of MOCEX with CMFD Acceleration for C5G7 Benchmark Problems ................................................................................................................. 14
Table 3. Computing Parameters of MOCEX Calculation with CMFD for TREAT-M8CAL Problem ........................................................................................................................ 16
Table 4. Performance improvements of MOCEX with CMFD acceleration for TREAT-M8CAL Problem .................................................................................................................... 16
Table 5. Performance Improvement of MOCEX for 3D C5G7 problems in BG/Q ............... 17
Table 6. Eigenvalues from DIF3D-VARIANT and PROTEUS-NODAL for the C5G7 Benchmark Problems ......................................................................................................................... 24
Table 7. DIF3D-VARIANT Angular Expansions and Anisotropic Scattering Options for ABTR and Monju Benchmark Problems ........................................................................................................ 28
Table 8. Eigenvalue Results for ABTR Benchmark Problem ...................................................... 30
Table 9. Eigenvalue Results for 200C Core Configuration of Monju Benchmark Problem .... 30
Table 10. Eigenvalue Results for C1-out Core Configuration of Monju Benchmark Problem ... 30
Table 11. Eigenvalue Results for C1-in Core Configuration of Monju Benchmark Problem .... 30
Table 12. Summary of Performance Examination of CMFD Acceleration for NODAL-SP3 Solver ................................................................................................................................. 35
Table 13. Comparison of the Initial Steady-State Solution for TWIGL Transient Benchmark Problem .......................................................................................................................... 43
Table 14. Comparison of the TWIGL Transient Benchmark Results ............................................. 44
Table 15. Comparison of Initial Steady-State Solution for C5G7 Transient Benchmark Results ................................................................................................................................. 45
Table 16. Description of C5G7-TD Benchmark Problems .......................................................... 45
Table 17. Eigenvalues of MC$_2$-3 and MCNP6 for Homogenized Compositions of UO$_2$ and MOX Fuel Pin Cell Problems ...................................................................................... 49
Table 18. Eigenvalues of MC$_2$-3 and MCNP6 for 2D UO$_2$ and MOX Fuel Pin Cell Problems ................................................................................................................................. 50
Table 19. Eigenvalues of MC$_2$-3 and KENO-VI for 2D pin cell (P1A through P1D) and lattice (P2A through P2D) problems in HZP condition at BOC of VERA benchmarks........ 50
Table 20. Eigenvalue Results of C5G7 Benchmark Problems .................................................. 57
Table 21. Summary of Pin Power Results of C5G7 Benchmark Problems .................................. 57
Table 22. Summary of Performance Examinations for TREAT Experiments ............................... 62
Table 23. Eigenvalue Solutions from MOCEX and Serpent ....................................................... 64
Table 24. Eigenvalue Results of VHTR Assembly Problems .................................................... 66
1. Introduction

Under the U.S. DOE NEAMS program, the high-fidelity multiphysics modeling and simulation capability SHARP [1] for nuclear reactor design and analysis has been developed. SHARP is a suite of physics simulation software modules – mainly composed of PROTEUS [2], Nek5000 [3], and Diablo [4] - and computational framework components that enables users to accurately evaluate the physical processes of nuclear reactors including neutron transport, thermal fluid, and fuel and structure behaviors. Among the SHARP components, the goal of the NEAMS neutronics effort is to develop a neutronics toolkit for use primarily on sodium-cooled fast reactors (SFRs) and an extended use to other reactor types. The neutronics toolkit includes the high-fidelity deterministic neutron transport code PROTEUS and many supporting tools such as a cross section generation code MC$^2$-3 [4], a cross section library generation code, alternative cross section generation tools, mesh generation and conversion utilities, an automated regression test tool BuildBot, and a reactivity perturbation and sensitivity analysis code PERSENT [6], etc., as shown in Figure 1.

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

Figure 1. Overview of the NEAMS Neutronics Code Suite
The main objectives of the FY17 NEAMS neutronics workpackage are 1) to develop an advanced nodal solver in PROTEUS for more routine nuclear reactor design and analysis, 2) to implement a simplified sub-channel based thermal-hydraulic (T/H) module for the NODAL and MOCEX solvers of PROTEUS for a rapid calculation of the thermal feedback effect, 3) to improve the performance of PROTEUS-MOCEX using numerical acceleration and code optimization, 4) to possibly improve the cross section generation tools including MC\(^2\)-3, and 5) to continue to perform verification and validation tests of PROTEUS.

Since NEAMS initially focused on developing tools to simulate challenging problems that could not be tackled using conventional tools, PROTEUS was focused on combining unstructured finite element meshing with the discrete ordinate (S\(_N\)) or the method of characteristics (MOC) paradigms to obviate the various modeling approximations that could not be used in the conventional schemes. Massive parallelism was seen as a key need to counter the huge computational costs that arise from using unstructured finite element meshing. In recent years, however, NEAMS has been putting emphasis on developing more practical tools that industrial partners can use in their advanced reactor design, analysis, and licensing in the near term. To satisfy the recent needs, in this year, we successfully built a production quality nodal solver, PROTEUS-NODAL, and added a simplified sub-channel based T/H capability to it.

The PROTEUS-NODAL code is based upon the DIF3D-VARIANT methodology [7], but with much more improved capabilities. The initial work was to only focus on a diffusion capability for Cartesian, hexagonal, and triangular-z geometries. To expand functionality, a nodal simplified P\(_3\) (SP\(_3\)) solution option was also added. Both diffusion and SP\(_3\) capabilities provide comparable solutions to DIF3D-VARIANT P\(_N\) transport solutions of several problems noting that PROTEUS-NODAL also reproduces the DIF3D-VARIANT diffusion and SP\(_N\) solutions.

The coupled PROTEUS and Nek5000 simulations often require excessive computational resources. In many cases, the detailed temperature information that Nek5000 provides to PROTEUS is unnecessary because the cross section data is not very sensitive to small changes temperature and thus simple averages from a sub-channel type scheme would normally suffice. This is especially true for power and flux distributions. To help reduce the computational effort of the coupled calculation, a built-in T/H solver based on simple conduction, convection equations, and thermal correlations was implemented, which can be used for both PROTEUS-NODAL and PROTEUS-MOCEX.

Conventional reactor analysis requires that the multi-group cross sections be updated with changes of state parameters such as temperature, density, burnup, etc. In PROTEUS-MOCEX, we can rely upon the cross section application program interface (API) fed with the updated state parameters to handle the feedback correctly. In nodal solvers, like PROTEUS-NODAL, the conventional approach focuses on functionalizing the multi-group cross section data with the various state parameters. To handle this, we modified PROTEUS-NODAL to handle a tabulated data set and created a code to tabulate the multi-group cross section data.
In parallel with developing the NODAL solver, an effort was made to improve the computational performance of PROTEUS, especially for the MOCEX solver. A focus was put on implementing the coarse mesh finite difference (CMFD) acceleration scheme in order to reduce the excessive computational burden of a whole core transport calculation. The performance improvement with the CMFD acceleration was tested for the 3D C5G7 benchmark problem and the M8CAL core of TREAT experiments. In addition, extensive updates were made to MOCEX to enhance its usability and performance, which include the implementation of the restart capability and the optimization of the MOCEX transport solver on the Blue Gene/Q (BG/Q) computing architecture.

In order to support PROTEUS users and make verification and validation analysis easier, a post-processing tool for the MOCEX outputs was developed as well. The initial version of the post-processing tool allows users to visualize the detailed MOCEX solutions and to easily edit the pin- and assembly-wise quantities such as fluxes and powers.

As part of a cross section improvement effort, the thermal cross section capability in MC\(^2\)-3 was updated this year using the 1,700 thermal groups in the thermal energy range below 5 eV. This should be better than the current thermal cross section capability based on equivalence theory whose accuracy was not guaranteed for non-LWR cases.

For code verification and validation efforts, various thermal reactor problems were modeled and simulated using PROTEUS-MOCEX. First, the 3D C5G7 problems were calculated with PROTEUS-MOCEX whose eigenvalues and pin power solutions were compared with MCNP solutions. Simulation and analysis for the M8CAL core of TREAT [9] and the Reactor Criticality Facility (RCF) core at PRI [10] were continued this year. The 2D and 3D VHTR assembly problems based on the NGNP design were solved in order to demonstrate the accuracy of MOCEX for VHTR analyses which are challenging for most deterministic codes.

Finally, work was carried out to implement a transient capability to the SN and MOC solvers of PROTEUS by the NEUP team [8]. Preliminary verification tests were conducted using the TWIGL and 2D C5G7-TD transient benchmark problems.

Sections 2, 3, and 4 present improvements made to PROTEUS, MC\(^2\)-3, and utility codes in turn. Section 5 discusses verification and validation test results from PROTEUS. Conclusions are discussed in Section 6.
2. Updates in PROTEUS

In FY17, the PROTEUS code has been significantly updated in terms of performance and capability. The computational performance of PROTEUS was greatly improved using an efficient acceleration method and by optimizing the transport sweeping routines for the BG/Q machine. More importantly, the advanced NODAL solver was successfully implemented to satisfy the immediate needs for the practical design use of the code. For an efficient multiphysics calculation, a simple built-in thermal calculation capability was implemented as well, which can be used for both the MOCEx and NODAL solvers. In addition, an attempt was made to improve the accuracy of the cross section library based on the multigroup cross sections generated from a Monte Carlo code. Those updates will be discussed in details in this section. The progress made on the implementation of the transient capability made by the NEUP team [8] was briefly described in this section.

2.1 MOCEx Solver

2.1.1 CMFD Acceleration

In FY16, a fully consistent coarse mesh finite difference (CMFD) formulation for the 2D/3D MOC solver was devised and its applicability was investigated by developing a small test code based on the 2D/3D MOC solver of MC2-3 [11]. In this year, the CMFD acceleration scheme was successfully implemented into PROTEUS-MOCEX and its performance was tested for the 3D C5G7 benchmark problem [12] and the M8CAL core of TREAT experiments [10].

2.1.1.1 CMFD Formulations for MOCEX

As an acceleration scheme for eigenvalue calculation, the conventional CMFD method has been widely adopted in various diffusion and transport solvers with a conservative property in spatial discretization. On the homogenized coarse meshes, the CMFD scheme solves a discretized neutron balance equation with a diffusion-type expression for interfacial currents:

\[ \sum_{k} J_{i \rightarrow k} + \Sigma_{i} \phi_{i} V_{i} = q_{i} V_{i}, \]  

\[ J_{i \rightarrow k} = -\tilde{D}_{i \rightarrow k} (\phi_{k} - \phi_{i}) - \hat{D}_{i \rightarrow k} (\phi_{i} + \phi_{k}), \]

where \( i, k \) = indices for the coarse mesh connected through a common interface,

\( \phi_{i} \) = average flux of coarse mesh \( i \),

\( J_{i \rightarrow k} \) = surface average current from coarse mesh \( i \) to \( k \),

\( \tilde{D} \) = coupling coefficient yielded from the finite difference formulation for diffusion equation,

\( \hat{D} \) = current correction factor to preserve the interface current from the solution of neutronics solver of interest.
The coarse mesh-wise scalar flux can be obtained by solving Eq. (2-1) in conjunction with the scattering and fission source iterations. Using the resulting CMFD solution, the higher-order fluxes are updated by applying the coarse flux ratio of CMFD and previous higher-order solution. The updated higher-order solution is served as the initial guess for the subsequent higher-order calculation. Since the higher eigenmodes are relatively quickly damped through the coarse-mesh solution, the number of fission source iterations in the higher-order calculation can be considerably reduced by incorporating the CMFD solution. Consequently, 3D problems with a large domain size can be efficiently solved by alternatively performing the higher-order and CMFD calculations.

The 2D/3D MOC formulation of MOCEX results in discrepancies of neutron balances due to the non-conservative discretization property of the Galerkin method applied to the axial variable. A blind application of the conventional CMFD scheme fails to accelerate 2D/3D MOC calculations due to a presence of persistent error in the fission source iteration [13]. This error results from the inconsistency between the CMFD and 2D/3D MOC solutions. As a remedy to enforce the neutron balance of coarse mesh solution with the 2D/3D solution, a fictitious cross section, referred to the pseudo absorption cross section (PAXS) hereafter, is introduced in the CMFD neutron balance equation as:

$$\sum_k J_{i\rightarrow k} + (\Sigma_{i}^f + \Sigma_{i}^p) \Phi_i V_i = q_i V_i, \quad (2-3)$$

where $\Sigma_p$ is the PAXS defined for individual coarse meshes and energy groups. Since the neutron imbalance in each computational mesh approaches a certain value as the solution converges, the PAXSs are iteratively updated during the 2D/3D MOC calculation.

### 2.1.1.2 Modification of MOCEX Solver

The prerequisites for performing the CMFD acceleration are the coarse mesh structure and the associated averaged flux, surface current and the group constants over individual coarse meshes. To prepare these prerequisites, the MOCEX routines were modified such a way to incorporate the coarse mesh structure within the unstructured finite element framework of PROTEUS and to compute the surface currents on coarse mesh surfaces in within-group transport calculations.

#### Incorporation of Coarse Mesh Structure

MOCEX performs the transport calculation using the unstructured finite elements as the base computing grid. An enormous effort is required to devise a robust and reliable algorithm to automatically extract a coarse mesh structure from arbitrary geometry. Alternatively, an appropriate coarse mesh structure is provided to MOCEX as an additional mesh input. A user generates these coarse meshes that are consistent with the fine meshes by using the mesh tools available such as CUBIT [14] and the in-house mesh toolkit [15]. The generated coarse mesh structure is imported in MOCEX using the same format of the fine mesh file. Figure 2 illustrates the fine and coarse mesh structures. As shown in Figure 2, the coarse meshes are prepared such a way that their surfaces are perfectly aligned along surfaces of fine meshes.
A mapping that projects the fine meshes onto the coarse meshes is needed to perform the CMFD acceleration. This mapping is prepared by making use of the coordinate information of two mesh structures. For the coarse mesh structure, surface equations of individual interfaces are constructed, and then individual coarse meshes are redefined in terms of combinatorial geometry scheme. The combinatorial geometry scheme allows us to easily determine that a certain point is located either inside or outside of a certain region. For each fine mesh, the fine mesh mapping to coarse mesh are determined based on the center of mass coordinate by making use of the combinatorial geometry descriptions of coarse mesh structures. In addition, the mapping between the fine mesh and coarse mesh surfaces are derived from the mesh mapping.

For the parallel computation, the domain decomposition routine was also modified to incorporate the coarse mesh structure. If the radial domain is decomposed based on the fine mesh structure as shown in Figure 3(a), the sophisticated communications are needed to compute the coarse mesh quantities. This leads to the cumbersome effort in its implementation and to the considerable computational burden in the CMFD calculation. To circumvent these difficulties, the domain decomposition routine was also updated such a way that the global domain is decomposed into the multiple local domains based on the coarse mesh structure, as shown in Figure 3(b). It was also confirmed that the modification of the domain decomposition scheme does not alter the converged MOCEX solution.
Surface Current Tally

In the CMFD acceleration, the average currents on the coarse mesh surfaces are required to specify a coupling relation between two adjacent coarse meshes, which is conventionally referred to the CMFD relation. The axial surface currents can be easily obtained by processing the element-wise angular flux distribution. However, the angular flux distributions on the fine mesh surfaces are not stored in the solution vector. For the radial coarse mesh surfaces, a capability of on-the-fly current tally was implemented in the MOCEX solver. To perform the surface current tally efficiently, the outgoing angular fluxes of each element along a single trajectory are stored in a temporary array during the MOCEX calculation. After finishing the MOCEX calculation for a single trajectory, the angular flux values on the coarse mesh surfaces are extracted from the temporary array to compute the surface currents. Moreover, the array indices for the angular fluxes of interest are pre-computed and stored by making use of the trajectory and coarse mesh information. This algorithm, illustrated in Figure 4, can minimize the computational cost of surface current tally because it can eliminate the computations of if-statements in the ray tracing calculation. To realize this process in the MOC calculation, the following three functions were implemented:

- Construct a surface mapping between fine and coarse mesh structures.
- Prepare a list of the trajectory location (index) that passes the coarse mesh surfaces and the associated coarse mesh surface indices.
- Compute a surface current based on the angular flux data on the coarse mesh surfaces.

In the 2D 17x17 PWR assembly calculations, the computational time of single transport sweeping increased about 5% when the surface currents were computed for every pin cell surfaces. In addition, a capability of computing the surface fluxes on the local domain boundaries was implemented. When the spatial domain is partitioned for parallel run, the obtained surface fluxes are utilized to update the outgoing angular fluxes on the broken trajectories in the CMFD acceleration.
Temporary Array

Figure 4. Illustration of Surface Current Tally Process

The current tally capability of MOCEX was verified by the neutron balance of individual coarse meshes using the surface current and coarse mesh averaged flux and source. For the MOCEX solutions for 2D problems, note that the neutron balance should be strictly satisfied in every fine and coarse meshes. The 2D UO$_2$ and MOX assembly problems derived from the C5G7 benchmark specification were solved and the neutron balance of each pin cell (coarse mesh) was examined. For this verification test, a subroutine that computes the neutron balance error defined in Eq. (2-4) was implemented in MOCEX.

$$\text{Neutron Balance Error} = 1 - \frac{1}{Q} \left( \sum_{i} J_{i} + \Sigma_{i} \phi \right).$$  \hspace{1cm} (2-4)

As shown in Figure 5 and Figure 6, the MOCEX solutions essentially satisfy the neutron balances of individual coarse meshes within the truncation error of the double precisions. These results confirm that this capability was properly implemented.

(a) Flux  \hspace{1cm} (b) Neutron Balance

Figure 5. Flux Distribution and Neutron Balance of 2D UO$_2$ Assembly Problem (Group 7 out of 7)
2.1.1.3 Implementation CMFD Solver

The implementation of CMFD solver was followed by the MOCEX modification. This work involved the implementations of the dynamic homogenization, the sparse matrix storage, and the data communications between the spatially partitioned domains. Along with these capabilities, a module that solves the CMFD linear system using the GMRES method was also implemented. Prior to applying the developed CMFD solver to the convergence acceleration of MOCEX, the equivalence of the MOCEX transport and CMFD problems was investigated by means of integrated tests for CMFD solver.

**CMFD Matrix Construction**

In order to construct the CMFD problem from the MOCEX solution for the fine mesh structure, the following quantities are required:

- Homogenized cross sections for the individual coarse meshes
- CMFD coupling coefficients between the two adjacent coarse meshes
- Pseudo absorption cross sections (PAXSs)

The capabilities of computing the above quantities were implemented in MOCEX. In the determination of the homogenized cross sections and CMFD coupling coefficients, the conventional procedures were implemented by making use of the MOCEX modules for the coarse mesh processing and the surface tally. Since the 2D/3D MOC formulation involves the non-conservative discretization for axial variable, the PAXSs defined for each coarse mesh and group are essential to ensure the equivalence between the coarse mesh and fine mesh solutions.

In MOCEX calculations with the CMFD acceleration, those input quantities of CMFD solver are repeatedly updated using the available transport solution at the end of each outer iteration, which is referred to the dynamic homogenization [16]. Once these prerequisite data are prepared, the following CMFD equation can be formed for each coarse mesh:

\[
-\sum _{k} (\hat{D}_{i\rightarrow k} + \hat{D}_{k\rightarrow i}) \bar{\Sigma }_{k} + \left[ \sum _{k} (\hat{D}_{i\rightarrow k} + \hat{D}_{k\rightarrow i}) + (\Sigma _{i} + \Sigma _{p}) \right] V_{i} \bar{\Theta }_{i} = \bar{Q}_{i}. \tag{2-5}
\]
Matrix Storage and Linear System Solver

A sparse matrix module tailored for storing and handling the CMFD linear system was implemented. Currently, the sparse matrix module can accommodate CMFD linear systems that are partitioned by the spatial domain decomposition. It also provides a matrix-vector multiplication using the MPI communication. In order to perform the matrix-vector multiplication for the partitioned matrix efficiently, a relevant MPI communication subroutine was additionally implemented such a way that the coarse mesh on the local domain boundaries are transferred to their adjacent domains through the non-blocking MPI communication. Once a linear system is properly constructed, it can be solved by invoking the existing GMRES solver in conjunction with the sparse matrix-vector multiplication function.

Equivalence Test

As an effort of verifying the integrated functionality of CMFD module, the equivalence of CMFD and MOCEX formulations was investigated using the C5G7 benchmark problems. The CMFD problems were constructed for each energy group using the fully converged transport solutions of MOCEX. The resulting CMFD problems were solved with the fixed coarse mesh sources determined from the transport solution without the within-group source iteration. The obtained coarse mesh fluxes were compared against the reference coarse mesh fluxes that were directly obtained from the transport solution. Figure 7 and Figure 8 compare two flux distributions for the 2D and 3D C5G7 benchmark problems respectively. The flux distributions obtained from the CMFD problem are essentially identical to the reference distributions. These comparisons indicate that the CMFD formulation is established in a consistent way and the associated modules are functioned as intended.

![Figure 7. Comparisons of CMFD and Transport Solutions for 2D C5G7 Benchmarks (Relative Error, %)](image)
2.1.1.4 Incorporation of CMFD and MOCEX Solvers

The CMFD eigenvalue solution routine was implemented by employing the conventional fission source iteration scheme, and the inverse homogenization process was implemented to reflect the obtained CMFD solution to the subsequent 2D/3D transport calculation. An under-relaxation scheme was also added to the solution process of the CMFD acceleration in order to stabilize the convergence of CMFD acceleration. The implementation of CMFD capability was completed by adopting and implementing the well-established alternating solution process.

**CMFD Eigenvalue Calculation**

In order to damp the higher eigenmodes effectively using the coarse mesh solution, the CMFD acceleration is casted into its own eigenvalue problem. The CMFD eigenvalue problem is solved by employing the conventional fission source iterations and the group sweepings. Note that the higher-order scattering source is not explicitly included in the solution process because the CMFD calculation solves the diffusion-type equation. Instead, the CMFD calculation accounts for the anisotropic scattering effect indirectly through the homogenized cross sections and current coupling coefficients.

**Inverse Homogenization**

The inverse homogenization reflects the CMFD solution to the subsequent 2D/3D MOC calculation. The well-established inverse homogenization schemes used in the 2D/1D MOC codes were adopted and implemented in the CMFD module. Using the coarse mesh flux ratio of the previous transport and newly obtained CMFD solution, the fine-mesh fluxes within each coarse mesh are updated as:

\[
\phi_i^{\text{new}} = \phi_i^{\text{old}} \frac{\bar{\phi}_i^{\text{new}}}{\bar{\phi}_i^{\text{old}}} \quad \text{for } i \in k, \tag{2-6}
\]
where $\phi_i$ and $\bar{\phi}_k$ are the scalar fluxes defined on the $i$-th fine mesh and the $k$-th coarse mesh respectively. In the transport calculation with the decomposed domain, the outgoing angular fluxes on the local domain boundaries are utilized as the incoming angular fluxes of adjacent local domains. The outgoing angular fluxes on the local domain boundaries are updated by making use of the $P_1$ approximation of angular flux as:

$$\phi_i^{\text{new}} = \phi_i^{\text{old}} \frac{\bar{\phi}_m^{\text{old}} + 3 \Omega_n \mathcal{J}^{\text{new}}}{\phi_k^{\text{old}} + 3 \Omega_n \mathcal{J}^{\text{old}}} \quad \text{for } l \in m,$$  

(2-7)

where $\phi_i$ is the outgoing angular flux of the $l$-th trajectory, and $\bar{\phi}_m$ and $\mathcal{J}_m$ are the flux and current of the $m$-th coarse mesh surface, respectively. [17] When Eq. (2-7) yields negative angular fluxes, the following relation, which is based on the DP$_1$ approximation, is employed alternatively:

$$\phi_i^{\text{new}} = \phi_i^{\text{old}} \frac{\bar{\phi}_m^{\text{new}} \exp(3 \Omega_n \mathcal{J}^{\text{new}} / \phi_k^{\text{old}})}{\phi_i^{\text{old}} \exp(3 \Omega_n \mathcal{J}^{\text{old}} / \phi_k^{\text{old}})} \quad \text{for } l \in m.$$

(2-8)

**Stabilization of CMFD Convergence**

In order to stabilize the convergence of CMFD acceleration, the under-relaxation scheme was added to the solution process of CMFD acceleration. Specifically, the under-relaxation scheme is applied to the inverse homogenization process. In the inverse homogenization with the under-relaxation scheme the scalar flux is updated as:

$$\phi_i = (1 - \omega)\phi_i^{\text{old}} + \omega \phi_i^{\text{new}} \bar{\phi}_k^{\text{new}} \quad \text{for } i \in k,$$

(2-9)

where $\omega$ is the relaxation parameter. The outgoing angular fluxes are also under-relaxed in the same manner. The relaxation parameter can be manually selected through an input deck and the default one is 0.5. The numerical tests revealed that the CMFD acceleration scheme can be unstable especially for the 2D/3D MOC calculations with the coarse axial mesh structure. These instabilities can be effectively mitigated by employing the under-relaxation scheme.

If large voided regions exist in a problem domain, the CMFD matrices can be ill-posed due to extremely large diffusion constants. Therefore, a special remedy for the voided region was additionally implemented. For the voided region, a considerably large diffusion constant was assigned to the coarse mesh and the current coupling coefficients of the CMFD formulation were adjusted such a way to preserve the neutron balance. Additionally, the $P_1$ are DP$_1$ approximations used in Eqs. (2-7) and (2-8) are inappropriate for updating the angular flux due to the pronounced transport effect. To prevent the CMFD acceleration from any instability, the outgoing angular flux is simply updated as:

$$\phi_i^{\text{new}} = \phi_i^{\text{old}} \bar{\phi}_k^{\text{new}} \quad \text{for } i \in k.$$

(2-10)
Solution Process of MOCEX Calculation with CMFD Acceleration

The well-established alternating solution process of CMFD and transport calculations was realized in the MOCEX solver by coordinating the relevant functions in the CMFD modules. The overall calculation procedure involving three layers of nested iterations is illustrated in Figure 9. When the CMFD acceleration is applied to the MOCEX solver, the CMFD system is formed with the partially converged transport solution, being solved using the inverse power method involving the nested iterations on fission and scattering sources. The fine mesh 2D/3D MOC solution is updated by applying the inverse homogenization process. The updated fine mesh solution is served as an initial guess of next within-group calculations.

The CMFD acceleration secures the equivalence with the 2D/3D MOC calculation through the PAXS. Therefore, it does not introduce a bias in the 2D/3D MOC solution which can alter the eigenvalue and flux distribution. The 2D/3D MOC calculation yields the identical solution regardless of CMFD acceleration usage.

Figure 9. Calculation Flow of 2D/3D with CMFD Acceleration

2.1.1.5 Performance Examinations

In order to investigate the performance of CMFD acceleration in the MOCEX calculation, two test problems were prepared: the 3D C5G7 benchmark problems [12] and the M8CAL core
configuration of TREAT experiments [11]. The detailed descriptions of MOCEX models can be found in Section 5. These problems were solved using MOCEX with and without the CMFD acceleration in the Blue Gene/Q (BG/Q) machine [18]. The performance of CMFD acceleration was studied by comparing the obtained solutions, convergence behaviors and computing times.

**C5G7 Benchmark Problems**

As a preliminary performance examination of the CMFD acceleration, the MOCEX calculations were performed for the C5G7 benchmark problems with and without the CMFD acceleration. In the CMFD calculations, the individual pin cells were selected as a single coarse mesh. Table 1 provides the detailed information of MOCEX calculations such as the computing options and the mesh structures. To measure the performance of CMFD, the MOCEX calculations were additionally performed without the CMFD acceleration. Table 2 summarizes the observed performance improvement of CMFD acceleration and Figure 10 compares the pin power distribution obtained with and with CMFD acceleration. As shown in Table 2 and Figure 10, the CMFD acceleration reduces the overall computational time by a factor of 4 without introducing the bias on the eigenvalue and power distribution.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># Elements</td>
<td>472,449</td>
</tr>
<tr>
<td># Planes</td>
<td>28</td>
</tr>
<tr>
<td>Cubature (# Directions)</td>
<td>L9T25 (384)</td>
</tr>
<tr>
<td>Ray Spacing</td>
<td>0.025 cm</td>
</tr>
<tr>
<td>Decomposition (Radial/Axial/Angle)</td>
<td>73 / 7 / 32</td>
</tr>
<tr>
<td># Processors</td>
<td>16,320 (1,020 nodes)</td>
</tr>
<tr>
<td>Convergence Criterion</td>
<td>1.E-6</td>
</tr>
</tbody>
</table>

**Table 2. Performance Improvements of MOCEX with CMFD Acceleration for C5G7 Benchmark Problems**

<table>
<thead>
<tr>
<th>Case</th>
<th>MOCEX Calculation</th>
<th>F.S. Iterations</th>
<th>Computing Time, Sec</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unrodded</td>
<td>w/o CFMD</td>
<td>54</td>
<td>8142</td>
<td>4.28</td>
</tr>
<tr>
<td></td>
<td>w/ CMFD</td>
<td>13</td>
<td>1902</td>
<td></td>
</tr>
<tr>
<td>Rodded A</td>
<td>w/o CFMD</td>
<td>55</td>
<td>8289</td>
<td>4.35</td>
</tr>
<tr>
<td></td>
<td>w/ CMFD</td>
<td>13</td>
<td>1902</td>
<td></td>
</tr>
<tr>
<td>Rodded B</td>
<td>w/o CMFD</td>
<td>56</td>
<td>8453</td>
<td>4.40</td>
</tr>
<tr>
<td></td>
<td>w/ CMFD</td>
<td>8453</td>
<td>1910</td>
<td></td>
</tr>
</tbody>
</table>
TREAT-M8CAL Benchmark Problem

MOCEX with the CMFD acceleration was applied to the M8CAL benchmark problem, which was derived from the actual M8CAL core of TREAT experiments. Its core configuration has the large voided channels toward the hodoscope section located at the core periphery. In the CMFD calculation, the coarse meshes were configured using the assembly structure of M8CAL core as shown in Figure 11. The detailed information of MOCEX run is provided in Table 3. The MOCEX performances with and without the CMFD acceleration are summarized in Table 4 and Figure 12. As shown in Figure 12, the fission source and flux errors decrease monotonically without experiencing a numerical instability even in the presence of the strong neutron streaming along the voided channel. It can be also seen that the CMFD acceleration significantly reduces the total computing time by about a factor of 6. To check the equivalence of CMFD acceleration, the assembly-wise fission source distributions obtained with CMFD acceleration was compared to the reference values from the unaccelerated MOCEX run. The obtained result is plotted in Figure 13 along with the reference fission distribution. The fission source distribution obtained with the CMFD acceleration is identical to the reference solution obtained with no CMFD acceleration within 0.015% differences. Note these extremely small discrepancies can be further reduced by tightening the convergence criterions. These results confirm the CMFD acceleration scheme equipped in MOCEX can significantly improve the convergence of MOCEX calculation without introducing any noticeable error.
Figure 11. Coarse Mesh Structure for TREAT-M8CAL Calculation (Blue: Coarse Mesh, Black: Fine Mesh)

Table 3. Computing Parameters of MOCEX Calculation with CMFD for TREAT-M8CAL Problem

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># Elements</td>
<td>352,536</td>
</tr>
<tr>
<td># Planes</td>
<td>24</td>
</tr>
<tr>
<td>Cubature (# Directions)</td>
<td>L7T9 (160)</td>
</tr>
<tr>
<td>Ray Spacing</td>
<td>0.1 cm</td>
</tr>
<tr>
<td>Decomposition (Radial/Axial/Angle)</td>
<td>68 / 12 / 20</td>
</tr>
<tr>
<td># Processors</td>
<td>16,320 (1,020 nodes)</td>
</tr>
<tr>
<td>Convergence Criterion</td>
<td>1.E-5</td>
</tr>
</tbody>
</table>

Table 4. Performance improvements of MOCEX with CMFD acceleration for TREAT-M8CAL Problem

<table>
<thead>
<tr>
<th>MOCEX Calculation</th>
<th>Fission Source Iterations</th>
<th>Computing Time, Sec</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o CFMD</td>
<td>73</td>
<td>162.0</td>
<td></td>
</tr>
<tr>
<td>w/ CMFD</td>
<td>13</td>
<td>28.1</td>
<td>5.76</td>
</tr>
</tbody>
</table>

Figure 12. Comparisons of MOCEX Convergence Behavior w/ and w/o CMFD acceleration for TREAT-M8CAL Problem
2.1.2 Optimization in Blue Gene/Q

The transport sweeping routines of MOCEX, which take most of the computational time, were extensively optimized by accounting for the following three aspects: vectorization, minimization of computation and efficient usage of cache memory. Along with these efforts, the computations of exponential and trigonometric functions were optimized by using the Mathematical Acceleration Subsystem (MASS) library available in the BG/Q machines. The performance improvement from the optimizations was tested by solving the 3D C5G7 problems with and without the optimizations and the obtained results are summarized in Table 5. The optimized MOCEX code is 2.2 times faster than the original one in the transport sweeping calculation and the overall computation time is reduced about 54%.

<table>
<thead>
<tr>
<th>Component</th>
<th>Computing Time*, Sec</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o Optimization</td>
<td>w/ Optimization</td>
</tr>
<tr>
<td>Overall</td>
<td>1223.4</td>
<td>664.2 (54%)</td>
</tr>
<tr>
<td>Transport Sweeping Calculation</td>
<td>917.3</td>
<td>426.0 (46%)</td>
</tr>
</tbody>
</table>

*1920 processors were used in the calculations (16 processors per computing nodes and 120 computing node)

2.1.3 Restart Capability

A restart capability, which allows to export and import a solution vector of MOCEX via restart file, was newly implemented in MOCEX. Using this capability, the MOCEX code can write the solution vector obtained in the latest iteration to the restart file. MOCEX can proceed the
transport calculation from the solution vector available in the restart file. It is beneficial for running a large-scale 3D problem in a computing machine with a running time restriction. For realizing this capability, a restart file should contain sufficient data to retrieve the solution vector of preceding run. In MOCEX, the solution vector accommodates the following components:

- Mesh-wise angular flux
- Trajectory-wise outgoing angular fluxes on the local domain boundaries
- Mesh surface-wise incoming angular flux of each surface on the global domain boundaries

Here, a local domain boundary refers to a set of interface surfaces formed between the partitioned domains. The straightforward way is to export and import the entire solution vector along with the eigenvalue through the restart file. This approach takes substantial processing time and uses excessive disk storage for the restart file. Regarding practicality, a minimum data should be processed through the restart file. It is noted that any solution vector can be reconstructed by performing the transport sweepings only if the relevant source distribution can be retrieved from the restart file. The source distribution is determined by the P_N flux moments and eigenvalue in the MOCEX calculation. Currently, MOCEX provides two options for restart calculation: processing entire solution vector or group- and mesh-wise P_N flux moments. With the first option, the MOCEX calculation can be immediately continued from the solution obtained in the last iteration of previous run. When the second option is employed, the fixed source problems for each energy group are formed and solved using the within-group transport solver. With a few iterations in the fixed source calculation, the solution vector including the angular fluxes on the local and global domain boundaries can be obtained.

To reduce the processing time further, the data export and import routines for the restart capability were implemented such a way to use the single precision. Compared to the usage of double precision, the file I/O and the MPI communication time can be reduced along with a factor of two reduction of restart file size. Due to the truncation error of single precision, there is a potential penalty on performance but it turns out that this error can be easily eliminated with a few additional within-group iterations in the initial transport sweeping.

### 2.2 NODAL Solver

#### 2.2.1 NODAL-P\_N

The NODAL software development was initiated in FY10 in response to the efforts by INL to create a replacement set of software for ARC targeting sodium cooled fast spectrum systems. The goal was to modernize the DIF3D-FD (finite difference) and DIF3D-VARIANT methodologies under the common PROTEUS (NODAL) framework. As part of PROTEUS, NODAL would have access to the fine meshing details of the homogeneous problem it was running such that the flux reconstruction was both consistent and seamless and the homogenization could be applied as an acceleration of the fine mesh transport methodologies.
The focus of the previous work was to research energy parallelism in diffusion theory to determine whether the 230 group diffusion theory calculations being carried out for fast spectrum reactor calculations could be done faster. The end conclusion from that work was that energy parallelism could be effective up to a few tens of processors (60% max scaling), but it requires a very complex code compared with the previous version to make FGMRES and a low cost preconditioner work efficiently. The end product of the previous work was not a production quality code like that of DIF3D as it required significant user involvement in deciding the iteration settings for each problem to get optimal parallel performance. Thus most of the work spent on NODAL in this year was to clean up the code and get it closer to a production quality piece of software.

### 2.2.1.1 Methodology

The NODAL methodology is a rewrite of the DIF3D-VARIANT methodology. In this methodology, we take the multigroup transport equation

\[
\hat{\Omega} \cdot \nabla \psi_g (\vec{r}, \hat{\Omega}) + \Sigma_{t,g} (\vec{r}) \psi_g (\vec{r}, \hat{\Omega}) = W_g (\vec{r}, \hat{\Omega}) + S_g (\vec{r}, \hat{\Omega})
\]

\[
W_g (\vec{r}, \hat{\Omega}) = \int \Sigma_{s,g \rightarrow g} (\vec{r}, \hat{\Omega} \cdot \hat{\Omega}') \psi_g (\vec{r}, \hat{\Omega}') d\hat{\Omega}',
\]

where \( g = 1, \ldots, G \) and, using an \( P_1 \) angular approximation theory, recast it into the commonly known diffusion equation written in a functional form

\[
\int dV \left[ \hat{\nabla} \lambda (\vec{r}) D_g (\vec{r}) \hat{\nabla} \phi_g (\vec{r}) + \lambda (\vec{r}) \Sigma_{t,g} (\vec{r}) \phi_g (\vec{r}) \right]
\]

\[
= \int dV \left[ \lambda (\vec{r}) \Sigma_{s,g \rightarrow g} (\vec{r}) \phi_g (\vec{r}) + \lambda (\vec{r}) S_g (\vec{r}) \right] - \sum_x \int d\Gamma_x \sum_k \hat{n}_{x,k} V K_x A_x (\vec{r}) \chi_{x,y} (\vec{r}),
\]

where the notation is mostly standard except for the boundary condition term appearing on the right hand side of this equation. In this case, these are necessary to define the interface current (odd parity flux) contributions to the flux solution within each spatial mesh. By spatially discretizing this system, we can collapse the system into the matrix relationship

\[
\sum_{K,L} D_{g} P_{K,L} \phi_g + \Sigma_{t,g} F \phi_g = \Sigma_{s,g \rightarrow g} F \phi_g + FS_g - \sum_{\gamma} M_{\gamma} \chi_{g,\gamma}.
\]

We can collect the matrices to write the compact form

\[
A_{g} \phi_{g} = FS_g - \sum_{\gamma} M_{\gamma} \chi_{g,\gamma},
\]

which we can solve for the scalar flux

\[
\phi_{g} = A_{g}^{-1} FS_g - \sum_{\gamma} A_{g}^{-1} M_{\gamma} \chi_{g,\gamma}.
\]

To put this into a response matrix form, we impose continuity on the scalar flux between spatial meshes and write
Because we need to solve the global system of nodes simultaneously, we assume partial currents of the form

\[ j^\pm_g = \frac{1}{4} \varphi_g \pm \frac{1}{2} \chi_g, \quad (2-17) \]

which leads to the response matrix relationships of

\[ j^+_g = R_g j^-_g + B_g S_g. \quad (2-18) \]

In PROTEUS-NODAL, the above system is setup assuming homogeneous cross sections within each spatial mesh although those cross sections can have a polynomial functionalization within the node. With this approach, orthogonal spatial basis functions internal and on the surfaces of the node are ideal when it comes to defining the spatial matrices.

2.2.1.2 Modification of the interface current orientation

The entire solver was simplified by altering the coordinate space of the current vectors on each surface from that originally used in DIF3D-VARIANT. Figure 14 shows the conventional approach used in DIF3D-VARIANT for surface trial function alignment. For certain periodic rotations this setup is optimal as it requires no manipulation of the currents to preserve the mapping and was done because DIF3D-VARIANT has very few geometry options. However, as documented in the updated DIF3D-VARIANT manual, the actual treatment of those current rotations was only recently done properly [19].

![Figure 14. DIF3D-VARIANT Surface Trial Function Alignment](image-url)
As mentioned above, PROTEUS-NODAL was vastly simplified by modifying the surface trial function orientation to that seen in Figure 15. By using this approach, the complex remapping of the partial current in various periodic geometries could be eliminated only leaving a few symmetry geometry options. Of course the problem with this action is that it also requires every outgoing current to be remapped to define the incoming current on the adjacent node.

Both DIF3D-VARIANT and PROTEUS-NODAL relied upon a separate subroutine to apply the reflective, vacuum, periodic, and symmetry boundary conditions. In DIF3D-VARIANT the subroutine is called after all of the red and black nodes have been solved in each iteration. In NODAL, the subroutine was called after each red and black node was solved (i.e. hundreds of times more). After some work, it was decided that the current remapping allowed the subroutine to be completely eliminated in favor of a simple if branch at the end of each node operation which checks whether that node has boundary conditions to be applied. This check was also required in DIF3D-VARIANT, it just stored it as a list of surfaces to apply boundary conditions to followed by a flag indicating which type to apply. In the updated PROTEUS-NODAL, the boundary albedo is used to simultaneously apply periodic, reflected, and vacuum boundary conditions to all node surfaces. This approach should be optimal as reflected, vacuum, extrapolated, and periodic together constitute just four albedo vector-vector operations that have to be applied.

2.2.1.3 Modification for 3D Sweeping

In DIF3D-VARIANT, the matrix-vector operations are manually unrolled for each surface and the red-black iteration is applied hierarchically in the axial and radial directions. In PROTEUS-NODAL, the red-black iteration is applied in 3D and we allow the compiler to determine the optimal matrix-vector operation. While we can show that the DIF3D-VARIANT approach is ideal in some reactor problems, using the full 3D sweeping in the partitioned matrix acceleration,
matrix of PROTEUS-NODAL is always more efficient because of the better memory access pattern for the acceleration system and elimination of numerous intermediate vectors.

The DIF3D-VARIANT code actually chose the two level red-black scheme because of the memory constraints in the 1980s. With modern computers, we can handle a full red-black sweep over the entire domain for a given energy group as the total number of surface currents (3·x·y·z < 25000 for most cases) and surface moments (3 for DIF3D-Nodal and 6 for DIF3D-VARIANT in diffusion < 150,000) leads to a small vector memory space (< 1.1 MB) which is within the typical cache size of most modern processors. More of an issue is the response matrix storage, which for 10,000 unique nodes leads to 12 MB of memory. Given both of these are still under the 32 MB level 3 cache, and unlikely to be less than the typical 2 MB of level 2 cache even with the two level iterative scheme, we can improve the convergence by removing the extra iteration needed in the two level scheme. It is important to note that in most fast reactor problems this alone should eliminate 25% to 50% of the computational effort.

It is important to note that this approach was tried before in DIF3D-VARIANT with a negative outcome. In the two level scheme of DIF3D-VARIANT, the response matrix storage was not split to explicitly handle the separated axial and radial components (i.e. the ideal approach). Instead, the memory access pattern had to be setup to access the response matrix components using offsets when computing the axial current based source for each radial plane, solving each radial system, and updating the outgoing axial currents. Regardless, upon inspection, the two-level approach was found to constitute fewer response matrix related flops per iteration. Thus the conclusion was that the performance of the full red-black sweep was not as good as the two level approach but this did not consider the restrictions placed on the cache sizes which were much smaller at the time of the tests (2-4 MB instead of todays 32 MB). While we implemented the full 3D sweep, we did not have time to contrast it against DIF3D-VARIANT this year. If in the future we find that two-level scheme is vastly inferior to the 3D one, then we can switch to using the two-level scheme again.

### 2.2.1.4 Truncation of the FMGRES vector space allowances

Continuing, in DIF3D-VARIANT, there was a heavy focus placed on minimizing the vector storage of surface currents which arose due to the DIF3D-Nodal origination (where rotation of current moments was not an issue). In PROTEUS-NODAL, the same convention was followed but because of the desire to use a Krylov solver (GMRES or FGMRES), an additional layer of expansion/contraction was necessary as the surface currents are multiply defined in some geometry setups (e.g. periodic and symmetry domains). In the previous work, the expansion/contraction approach resulted in a complex if branch operation in the main iterative (repeating) segment of the NODAL solver. In the updated PROTEUS-NODAL, both the DIF3D-VARIANT and the previous approaches were discarded as storing the full vector resulted in the minimal if branching during the iterative solver phase. While this does increase the memory usage slightly, it maximizes the potential success for threading as it allows the current vector components of each node to be independent memory targets.
Most of the rest of the changes focused on code cleanup and truncation. Because of the rather minor success of energy parallelism and usage of GMRES/FGMRES, all aspects of the Krylov solver were removed as was the parallelism in energy concept. At the moment, the new version is thus completely serial although we are targeting OpenMP threading to allow it to better utilize the cores on a given node.

2.2.1.5 Verification Tests

Given the numerous changes to the software, the PROTEUS-NODAL code was successfully verified to handle Cartesian, hexagonal, and triangular-Z geometries. For Cartesian we constructed verification checks for 1D, 2D, & 3D for all regular, symmetry (45), and periodic (90 & 180 degree) options. We similarly setup verification checks for the hexagonal geometry options in 2D and 3D which include regular, symmetry (30, 60, 120), periodic (60, 90, 120, 180). Triangular-Z was the last option to be setup as we did not have a reference calculation ability (i.e. DIF3D-VARIANT) and many of the tests of symmetry and periodic conditions are 60 degrees out of phase with the hexagonal geometry option.

With regards to the solver today, it is still lacking the region edits, restart, solution visualization and other common features of a production code (like a manual). One particular concern is the lack of acceleration on the fission source iteration (power method). While the coarse mesh rebalance (CMR) was originally implemented in DIF3D-VARIANT, it was found to be very ineffective for hexagonal geometries and did not work for the adjoint calculation due to an implementation bug that could not be isolated.

The specific problem with the hexagonal geometry CMR is the use of hexagonal rings. In this approach, each consecutive coarse mesh is considerably larger in volume/area than the previous coarse mesh and smears regions together regardless of the local flux solution (i.e. all six sectors should have similar flux levels for the acceleration to work well). A comparison with Tchebychev indicated that CMR on hexagonal fast reactor geometries was generally not faster with most cases being nearly identical in performance. CMR should of course be ideal acceleration for the nodal methodology which was demonstrated easily using the Cartesian geometry option of DIF3D-VARIANT. We chose not to implement Tchebychev acceleration into PROTEUS-NODAL and have done some work on how to implement something like CMR into PROTEUS-NODAL. At present we have lined up a PhD student to attempt an alternative acceleration scheme to CMR into the updated PROTEUS-NODAL and hope to report on that work next year.

The current version of PROTEUS-NODAL correctly solves multigroup forward problems in the same or less computational effort than DIF3D-VARIANT on problems. In many cases PROTEUS-NODAL finishes 10-20 times faster than DIF3D-VARIANT but this is believed to be more due to the numerous binary interface files that DIF3D-VARIANT creates during execution than the improvements in the solver algorithm discussed above. Since neither code has an effective acceleration on the fission source iteration (i.e. CMR), both codes can be observed to run poorly on large dominance ratio thermal and fast spectrum problems.
Simple verification tests were performed using the well-known 2D and 3D C5G7 benchmark problems with a quarter-core symmetry, as shown in Figure 16. As listed in Table 6, eigenvalues from DIF3D-VARIANT and PROTEUS-NODAL are almost identical for the 2D and 3D C5G7 problems, indicating that the PROTEUS-NODAL was implemented successful for the Cartesian geometry with the reflective and vacuum boundary conditions. Verification tests for the hexagonal geometry will be discussed in the following sections.

Figure 16. Configuration (left) and Thermal Flux Solutions (right) of a Quarter-core of the C5G7 Problem

<table>
<thead>
<tr>
<th>Configuration</th>
<th>DIF3D-VARIANT</th>
<th>PROTEUS-NODAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>1.18622</td>
<td>1.18627</td>
</tr>
<tr>
<td>3D</td>
<td>0.95043</td>
<td>0.95043</td>
</tr>
</tbody>
</table>

2.2.2 **NODAL-SP$_3$**

The SP$_3$ equation, which is an approximated version of the P$_3$ equation, is known for capturing the major transport effects in both fast and thermal reactors. [20] With affordable additional computation cost, it can render significantly improved accuracy over the conventional diffusion equation. Moreover, it can be readily expanded for the full spatial kinetics by adopting the well-established methods for the diffusion equation. In order to support steady-state and transient analyses for routine design practices, the SP$_3$ nodal option was added to the PROTEUS-NODAL solver this year. Currently the SP$_3$ nodal option can provide the steady-state solution of SP$_3$ equation on the 3D hexagonal geometries. It is planned to be extended to the transient calculation next year.
In the SP3 nodal capability, the SP3 equation is efficiently solved by combining the 2D triangle based polynomial expansion nodal (TPEN) method and 1D nodal expansion method (NEM) for radial and axial directions, respectively. The convergences of eigenvalue and fission source are accelerated by employing the diffusion CMFD method. In order to verify the accuracy, the ABTR [23] and Monju [24] benchmark problems were solved by using the PROTEUS-NODAL and DIF3D-VARIANT solvers and their solutions were compared in terms of eigenvalue and assembly-wise power distribution.

2.2.2.1 Methodology

The 3D steady-state SP3 equation with the transport approximation is given as:

\[
\begin{bmatrix}
-D_i & -2D_i & 0 \\
0 & -D_i & 0 \\
0 & 0 & -D_i \\
\end{bmatrix}
\begin{bmatrix}
\nabla^2 \phi_i^0 (\vec{r}) \\
\nabla^2 \phi_i^2 (\vec{r}) \\
\phi_i^2 (\vec{r}) \\
\end{bmatrix}
+ 
\begin{bmatrix}
\Sigma_{rg,i} \\
\Sigma_{rg,i} \\
\Sigma_{rg,i} \\
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
-2 \Sigma_{rg,i} \\
\end{bmatrix}
\begin{bmatrix}
\phi_i^0 (\vec{r}) \\
\phi_i^2 (\vec{r}) \\
q_i^0 (\vec{r}) \\
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
\phi_i^2 (\vec{r}) \\
q_i^2 (\vec{r}) \\
\end{bmatrix},
\]

where \( \phi_i^k \) denotes the \( k \)-th flux moment on the \( i \)-th mesh, and the group index is omitted for brevity. The diffusion constant for the 2\(^{nd} \) flux moment, \( D_i^2 \), is defined as:

\[
D_i^2 = \frac{3}{\Sigma_{rg,i}}.
\]

By introducing a new solution variable \( \hat{\phi}_i = \phi_i^1 + 2\phi_i^2 \), Eq. (2-19) can be further simplified as:

\[
\begin{bmatrix}
-D_i & 0 & 0 \\
0 & -D_i & 0 \\
0 & 0 & -D_i \\
\end{bmatrix}
\begin{bmatrix}
\nabla^2 \hat{\phi}_i (\vec{r}) \\
\nabla^2 \phi_i^2 (\vec{r}) \\
\phi_i^2 (\vec{r}) \\
\end{bmatrix}
+ 
\begin{bmatrix}
\Sigma_{rg,i} \\
\Sigma_{rg,i} \\
\Sigma_{rg,i} \\
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
-2 \Sigma_{rg,i} \\
\end{bmatrix}
\begin{bmatrix}
\hat{\phi}_i (\vec{r}) \\
\phi_i^2 (\vec{r}) \\
q_i^0 (\vec{r}) \\
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
\phi_i^2 (\vec{r}) \\
q_i^2 (\vec{r}) \\
\end{bmatrix},
\]

The currents for the 0\(^{th} \) and 2\(^{nd} \) flux moments can be determined using the following relations:

\[
J_i^0 (\vec{r}) = -D_i \nabla^2 \hat{\phi}_i (\vec{r}) = -D_i \left( \nabla^2 \phi_i^0 (\vec{r}) + \nabla^2 \phi_i^2 (\vec{r}) \right),
\]

\[
J_i^2 (\vec{r}) = -D_i \nabla^2 \phi_i^2 (\vec{r}).
\]
solving the local problem for the hexagon using the local boundary conditions for surface currents and vertex fluxes. In this local problem, a hexagon node is divided into 6 triangular sub-nodes as shown in Figure 17 and an intra-nodal flux distribution is represented with a cubic polynomial. The intra-nodal flux distributions of six sub-meshes are simultaneously determined using the surface average incoming currents and vertex fluxes of hexagonal mesh along with additional constraints. Note that a single polynomial is used to represent the intra-nodal solution in the DIF3D-VARIANT method. In the TPEN method, the additional interfacial conditions on the internal surfaces are involved and this can adversely affect the solution accuracy. However, the TPEN approach is beneficial in that the flux variation within the homogenized node can be represented with a relatively concise form.

\[ \phi^m = c^m_0 + a^m_1 x + a^m_2 y + b^m_1 x^2 + b^m_2 u^2 + b^m_3 p^2 + c^m_4 x^3 + c^m_5 u^3 + c^m_6 p^3. \]  

In the procedure to yield the response matrix, the 9 expansion coefficients in Eq. (2-24) can be alternatively expressed in terms of the following 9 quantities defined on each triangular node: the average, x-moment and y-moment fluxes, surface average fluxes for three surfaces and vertex fluxes on three vertices. Consequently, a single hexagon has the 54 unknowns in total for each flux moment. By noting that the vertex fluxes of hexagon are given and the continuity conditions are imposed on the triangle vertices and the internal surfaces, as shown in Figure 18, the number of unknowns to be determined reduce to the following 31 quantities: average, x-moment and y-moment fluxes for the 6 triangular meshes, surface average fluxes for the 6 hexagon surfaces, surface average fluxes for the 6 internal surfaces and vertex fluxes on the hexagon center.
To determine the above quantities uniquely, the following 25 constraints are imposed along with the incoming partial currents for the six hexagon surfaces:

- Neutron balance equation for each triangular meshes (6 constraints)
- x- and y-weighed residual equations for each triangle meshes (12 constraints)
- Surface current continuity for each internal surface (6 constraints)
- Source-free condition in the hexagon center (1 constraint)

The detailed derivation of response matrix and its solution technique can be found elsewhere [21, 22].

In the 1D SP3 NEM for the axial direction, the solution of each axial mesh is expanded using the 4th order orthogonal polynomials and its coefficients are determined through the conventional one-node nodal technique. In the current implementation of 1D NEM module of NODAL-SP3 solver, the following conditions/constraints are used to determine the intra-nodal flux shape for axial meshes:

- Node average flux
- Partial incoming currents at the top and bottom surfaces
- x- and y-weighed residual equations for the axial mesh

The axial nodal solver uses the node average flux and radial transverse leakage from the radial solution and it updates the axial currents on the top and bottom of hexagonal meshes. Hence, the axial nodal solver can be viewed as the embedded procedure on the radial solver to account for the axial streaming effect. The overall calculation procedure of NODAL-SP3 solver is illustrated in Figure 19.

Figure 18. Unknowns of Local Problem for Hexagon Node in TPEN Method
2.2.2.2 Verification Tests

For the verification of NODAL-SP3 solver in PROTEUS, the ABTR and Monju benchmark problems were solved and the obtained solutions were compared against the DIF3D-VARIANT results in terms of eigenvalue and assembly-wise power distribution. Using the MC$^2$-3, the region-wise cross section sets were generated for the NODAL-SP3 and DIF3D-VARIANT calculations in the ANL 33 group structure. The DIF3D-VARIANT calculations were performed with the angular expansions and anisotropic scattering options given in Table 7 and the DIF3D-VARIANT results using P$_3$ angular expansions with the explicit P$_3$ anisotropic scattering options (P5-P3), referred to the reference solutions hereafter, were used in the comparisons of power distributions. In the NODAL-SP3 calculations, the transport approximation was used for the anisotropic scattering treatment. For brevity, the DIF3D-VARIANT code is denoted as DIF3D in this section.

Table 7. DIF3D-VARIANT Angular Expansions and Anisotropic Scattering Options for ABTR and Monju Benchmark Problems

<table>
<thead>
<tr>
<th>DIF3D Option</th>
<th>Angular Expansion</th>
<th>Anisotropic Scattering</th>
</tr>
</thead>
<tbody>
<tr>
<td>P5-P3 (Reference)</td>
<td>P$_5$</td>
<td>P$_3$</td>
</tr>
<tr>
<td>P5-P0</td>
<td>P$_5$</td>
<td>Transport approx.</td>
</tr>
<tr>
<td>P1-P0 (Diffusion)</td>
<td>P$_1$</td>
<td>Transport approx.</td>
</tr>
</tbody>
</table>
For the ABTR benchmark problem, the obtained eigenvalue results are summarized in Table 8. The eigenvalue results obtained with the diffusion approximation are more than 1000 pcm off from the reference value. The NODAL-SP3 solution has good agreement in eigenvalue with about 200 pcm difference, which suggests that the SP3 equation can properly capture the major transport effect in fast reactors. The comparison between the DIF3D results with the P5-P3 and P5-P0 options indicates that the transport approximation causes around a 70 pcm error in eigenvalue. Along with eigenvalue, the accuracy of assembly-wise power distributions was investigated. Figure 20 and Figure 21 compare the assembly-wise power distributions. The assembly-power distribution from the NODAL-SP3 solution agrees very well with the reference solution. In the fuel assemblies, the NODAL-SP3 solver predicted powers within 1%. The assemblies having relatively large errors are the non-fuel assemblies, such as control block, shield and reflector, in which relative errors are notably exaggerated due to small power values.

In the verification tests using the Monju benchmark problem, the three core configurations derived from the initial startup core design were analyzed. The summary of obtained eigenvalue results are provided in Table 9 and the comparisons of assembly-wise power distributions are plotted in Figure 22 through Figure 27. As shown in Table 9 to Table 11, the SP3 results show good agreement in eigenvalue with the reference solutions within 100 pcm for three cases. It is noted that the diffusion and SP3 equations give better eigenvalue solutions compared to the previous ABTR problem because of the reduced leakage effect from a larger domain size. The assembly-wise powers in the active core regions were obtained with less than 1% difference from the reference solution as shown in Figure 22 through Figure 27. Similar to the previous benchmark problem, the errors in the blanket region are substantially exaggerated while the absolute differences are very small. Even in the presence of the control rod, as shown in Table 11 and Figure 27, the NODAL-SP3 solver provides the faithful solution without introducing noticeably additional errors.
Table 8. Eigenvalue Results for ABTR Benchmark Problem

<table>
<thead>
<tr>
<th>Code</th>
<th>Eigenvalue</th>
<th>Diff., pcm</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF3D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P5-P3 (Ref.)</td>
<td>1.02452</td>
<td>0</td>
</tr>
<tr>
<td>P5-P0</td>
<td>1.02522</td>
<td>70</td>
</tr>
<tr>
<td>P1-P0</td>
<td>1.01043</td>
<td>-1409</td>
</tr>
<tr>
<td>PROTEUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NODAL-SP3</td>
<td>1.02260</td>
<td>-192</td>
</tr>
<tr>
<td>NODAL-P1</td>
<td>1.01077</td>
<td>-1375</td>
</tr>
</tbody>
</table>

Table 9. Eigenvalue Results for 200C Core Configuration of Monju Benchmark Problem

<table>
<thead>
<tr>
<th>Code</th>
<th>Eigenvalue</th>
<th>Diff., pcm</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF3D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P5-P3 (Ref.)</td>
<td>0.99576</td>
<td>0</td>
</tr>
<tr>
<td>P5-P0</td>
<td>0.99667</td>
<td>90</td>
</tr>
<tr>
<td>P1-P0</td>
<td>0.98873</td>
<td>-704</td>
</tr>
<tr>
<td>PROTEUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NODAL-SP3</td>
<td>0.99632</td>
<td>55</td>
</tr>
<tr>
<td>NODAL-P1</td>
<td>0.98914</td>
<td>-662</td>
</tr>
</tbody>
</table>

Table 10. Eigenvalue Results for C1-out Core Configuration of Monju Benchmark Problem

<table>
<thead>
<tr>
<th>Code</th>
<th>Eigenvalue</th>
<th>Diff., pcm</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF3D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P5-P3 (Ref.)</td>
<td>0.99866</td>
<td>0</td>
</tr>
<tr>
<td>P5-P0</td>
<td>0.99956</td>
<td>89</td>
</tr>
<tr>
<td>P1-P0</td>
<td>0.99169</td>
<td>-697</td>
</tr>
<tr>
<td>PROTEUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NODAL-SP3</td>
<td>0.99960</td>
<td>94</td>
</tr>
<tr>
<td>NODAL-P1</td>
<td>0.99210</td>
<td>-656</td>
</tr>
</tbody>
</table>

Table 11. Eigenvalue Results for C1-in Core Configuration of Monju Benchmark Problem

<table>
<thead>
<tr>
<th>Code</th>
<th>Eigenvalue</th>
<th>Diff., pcm</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF3D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P5-P3 (Ref.)</td>
<td>0.98814</td>
<td>0</td>
</tr>
<tr>
<td>P5-P0</td>
<td>0.98905</td>
<td>90</td>
</tr>
<tr>
<td>P1-P0</td>
<td>0.98089</td>
<td>-728</td>
</tr>
<tr>
<td>PROTEUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP3</td>
<td>0.98921</td>
<td>107</td>
</tr>
<tr>
<td>NODAL-P1</td>
<td>0.98310</td>
<td>-684</td>
</tr>
</tbody>
</table>
Figure 20. Assembly-wise Power Distribution for ABTR Benchmark Problem

Figure 21. Assembly-wise Power Comparison for ABTR Benchmark Problem (Relative Error in %)
Figure 22. Assembly-wise Power Distribution for 200C Core Configuration of Monju Benchmark Problem

Figure 23. Assembly-wise Power Comparisons for 200C Core Configuration of Monju Benchmark Problem (Relative Error in %)
Figure 24. Assembly-wise Power Distribution for C1-out Core Configuration of Monju Benchmark Problem

Figure 25. Assembly-wise Power Comparison for C1-out Core Configuration of Monju Benchmark Problem (Relative Error in %)
Figure 26. Assembly-wise Power Distribution for C1-in Core Configuration of Monju Benchmark Problem

Figure 27. Assembly-wise Power Comparison for C1-in Core Configuration of Monju Benchmark Problem (Relative Error in %)
2.2.2.3 CMFD Acceleration

The CMFD acceleration scheme was implemented to promote the NODAL-SP3 solver to the production-level tool. The conventional CMFD solver based on the diffusion equation was successfully implemented for the hexagonal-geometries such that the convergence of fission source and eigenvalue are accelerated through the alternating solution process of SP3 nodal and CMFD calculations. Since the prerequisites for CMFD acceleration is readily available, the SP3 nodal problem is simply casted into the CMFD problem. The CMFD capability is now properly functioned in the PROTEUS-NODAL framework. It can be promptly extended to incorporate a fixed source problem in the transient calculations, which will make the transient calculation more effective and practical.

For the performance examination, the NODAL-SP3 calculations were performed with and without the CMFD acceleration. The obtained results using the ABTR and Monju benchmark problems are summarized in Table 12 and Figure 28. A convergence criterion of $10^{-5}$ was imposed on the eigenvalue and fission source. As shown Table 12, the obtained eigenvalues with and with the CMFD acceleration are essentially identical within the imposed convergence criterion. The CMFD acceleration improves the performance by about a factor of two for the ABTR case and a factor of three for the Monju cases. With the CMFD acceleration, the numbers of fission source iterations are around 12 for the ABTR and Monju problems regardless of the notable differences in domain size. Figure 28 shows that the NODAL-SP3 calculations were completed without experiencing an unstable behavior of the fission source convergence. Figure 29 compares the power distribution obtained with and without the CMFD acceleration, indicating that the CMFD acceleration results in the identical power distribution of the unaccelerated case.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>ABTR</th>
<th>Monju</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200C</td>
<td>C1-out</td>
</tr>
<tr>
<td>Eigenvalue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o CMFD</td>
<td>1.0226001</td>
<td>0.9963166</td>
</tr>
<tr>
<td>w/ CMFD</td>
<td>1.0225993</td>
<td>0.9963172</td>
</tr>
<tr>
<td>Diff.,pcm</td>
<td>0.08</td>
<td>0.06</td>
</tr>
<tr>
<td>Fission source</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iterations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o CMFD</td>
<td>23</td>
<td>33</td>
</tr>
<tr>
<td>w/ CMFD</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>Reduction</td>
<td>1.91</td>
<td>2.62</td>
</tr>
</tbody>
</table>
Figure 28. Comparisons of Fission Convergence Behavior with and without CMFD Acceleration

Figure 29. Comparisons of Assembly-wise Power Distribution Obtained with and without CMFD Acceleration for ABTR Benchmark problem (Relative Difference in %)
2.3 Thermal Feedback Capability

PROTEUS-SN calculates power and neutron flux distributions and Nek5000 solves for velocity and temperature distributions in a nuclear reactor core. PROTEUS and Nek5000 are tightly coupled together within the SHARP Toolkit and power and temperature information can be seamlessly exchange online during the simulation while iterating between the two codes.

Each of PROTEUS and Nek5000 needs significant time and computational resource, and multiphysics calculations with the two codes require more excessive computational resources and time until the solutions are fully converged in an iterative manner.

In reality, the cross section regions are larger than the element meshes. The temperature changes in the element mesh level or in the region smaller than the cross section mesh are too small to affect neutronics solutions. A high-fidelity code like Nek5000 may provide too detailed temperature information to PROTEUS which may not be necessary for the neutronics solutions of PROTEUS. In many cases, detailed temperature distributions are not needed as well.

A simplified sub-channel based temperature calculation capability was developed for PROTEUS to be able to simulate typical LWR and SFR problems without Nek5000 coupling. We expect that core integral solutions such as a core eigenvalue and a reactivity worth from PROTEUS with the simplified temperature calculation capability would not be different from those from PROTEUS with Nek5000 even though local fluxes and powers may be different.

An additional benefit of the simplified temperature calculation capability would be to reduce the number of iterations for the coupled simulation of PROTEUS and Nek5000 by using the converged solution of PROTEUS with the simplified thermal feedback.

The simple temperature calculation method is based on the assumption of no boiling, constant pressure, no axial heat conduction, and no cross flow to neighboring pins or assemblies (closed channel). It is also assumed that the power generated from the fuel is completely transferred to the coolant. The coolant density and enthalpy are tabulated with temperature. The constitutive relations provided as a form of polynomial at a given pressure are applicable to the hot operating condition.

In a thermal hydraulic node, the radial temperature distribution within a fuel pin is calculated to determine the Doppler effect. A finite difference scheme is used to obtain the radial temperature distribution. The heat conduction and heat convection equations are solved independently.

In a thermal hydraulic channel, we assume that the coolant channel is closed in the radial direction, the coolant flows from bottom to top and the heat is generated and transferred from the fuel. The mass and energy conservation equations are solved using the steady-state one-dimensional (1D) formulation.

\[
\frac{1}{\Delta z} \left( \rho v \big|_{z_{out}} - \rho v \big|_{z_{in}} \right) = 0, \tag{2-25}
\]
\[ \frac{1}{\Delta z} \left( \rho v h_i - \rho v h_0 \right) = \bar{q}_c + \frac{D_c}{A_c} \bar{q}_f = \bar{q}, \quad (2-26) \]

where \( \rho \) = coolant density [kg/m³],
\( v \) = coolant velocity [m/s],
\( h \) = coolant enthalpy [J/kg],
\( D_c \) = heated perimeter [m],
\( A_c \) = cross section of the channel [m²],
\( \bar{q}_c \) = z-averaged heat source from the coolant [W/m³],
\( \bar{q}_f \) = z-averaged heat flux from the fuel [W/m²].

In a thermal hydraulic channel, the thermal properties of the coolant are known at the bottom of the channel and the heat fluxes at the fuel are determined from the heat conduction calculation. Therefore, the thermal properties of a channel including the coolant and fuel are solved for by sweeping from the bottom to the top of the channel.

The heat conduction in the fuel and clad is solved using the following equation:

\[ \frac{1}{r} \frac{\partial}{\partial r} \left( k(T) r \frac{\partial T}{\partial r} \right) = 0, \quad (2-27) \]

where \( k(T) \) = heat conductivity as a function of temperature [W/m-⁰C]. The heat convection based on Newton’s law is used for the coolant and gas gap as follows. The temperature notations are illustrated in Figure 30.

\[ \bar{q}_{\text{gap}} = -\left( k_{\text{gap}} \frac{\partial T}{\partial r} \right)_s = h_{\text{gap}} (T_{fs} - T_{ci}), \quad (2-28) \]

\[ \bar{q}_{\text{clad}} = -\left( k_{\text{clad}} \frac{\partial T}{\partial r} \right)_{\text{clad}} = h_{\text{clad}} (T_{co} - T_{\text{bulk}}), \quad (2-29) \]

where \( h_{\text{gap}} \) : Heat transfer coefficient at the gap,
\( h_{\text{clad}} \) : Heat transfer coefficient at the wall,
\( T_{fs} \) : Temperature at the outer surface of the fuel pellet,
\( T_{ci} \) : Temperature at the inner surface of the clad,
\( T_{co} \) : Temperature at the cladding outer surface,
$T_{\text{bulk}}$ : Bulk temperature of the coolant.

Figure 30. Temperature Profile over the Fuel Pin

To summary, the flow calculation in the coolant is performed before conducting the heat conduction calculation in the fuel. The heat source is calculated based on the relative power distribution from the neutronics calculation. The thermal properties of the coolant are determined using the heat flux and the inlet temperature. The coolant temperature distribution along the channel is calculated and then the radial temperature distribution of the fuel is calculated using the coolant bulk temperature as a boundary condition of the fuel conduction equation.

Figure 31. Input and Output Files of the MOCEX and NODAL Solvers of PROTEUS
The thermal calculation routines were implemented to PROTEUS to be used for both the NODAL and MOCEX solvers with minor modifications. For both solvers, an additional T/H input file was created, as shown in Figure 31, to provide the detailed thermal conditions of the problem (pressure, inlet temperature, mass flowrate, detailed pin geometries, etc.). The MOCEX solver requires two additional input files that include information on pin geometries and T/H regions (fuel, gap, cladding, and coolant) within a pin.

A coarse mesh file for the CMFD acceleration is used to obtain the information on pin geometries because the CMFD mesh is based on pin geometries in most cases. Using the CMFD mesh file and the T/H input file, the T/H routines calculate effective temperatures for fuel, cladding, and coolant regions and coolant densities within a pin. A T/H assignment file is used to identify which meshes belong to fuel, cladding, or coolant region in a pin. For guide tubes (GTs) or instrument thimbles (ITs) where power is not generated, moderator temperatures are determined by averaging moderator temperatures from the neighboring pins.

Once the temperature and density distributions are determined, multigroup cross sections are updated using the cross section data tabulated with state parameters. With the updated multigroup cross sections, the eigenvalue problem is solved again and this thermal feedback iteration is continued until all of eigenvalue, fission sources, scalar fluxes, and angular fluxes are converged within the user-specified criteria. For the cross section update with the state parameter changes, an ISOPAR type of cross sections should be provided.

Tests for thermal feedback calculations were performed with the C5G7 benchmark problem for which 7 group cross sections were generated in an ISOPAR format with the change of fuel and moderator temperatures. Figure 32 shows the results of power, fuel and moderator temperatures. The water regions outside were removed in the figures, in which temperatures remained unchanged. Higher powers were shown in the MOX assemblies and an axial peak power was at the center of the axial height. Fuel and moderator temperatures showed the same trend as the power distribution in the radial direction as expected. The highest fuel temperature was shown at slightly above the center of the axial height which is due to the monotonically increasing moderator temperature distribution in the axial direction as shown in Figure 32. Unlike the power and fuel temperatures which are zeros at the GT and IT locations, the moderator temperatures are smooth with non-zero values because those at the GT and IT locations are determined by averaging the neighboring moderator temperatures as aforementioned.

For further verification checks, the cross section changes in percent from the initial values are plotted. As expected, relatively large changes in total cross sections were observed at the top and center regions while relatively large changes in fission cross sections were shown at the MOX pin locations, as shown in Figure 33. Further verification tests will be performed in future with code-to-code comparisons.
Power Fuel Temperature Moderator Temperature

Figure 32. Power, Fuel Temperature, and Moderator Temperature Distributions from MOCEX with T/H Feedback

% Change of Total Cross Section % Change of Fission Cross Section

Figure 33. Change of Total and Fission Cross Sections due to Temperature Change
2.4 Transient capability

PROTEUS-SN (SN2ND) has a transient capability based on the adiabatic method which employs various approximations to reduce the time-dependent transport equation into the point kinetics equations for the amplitude function and the steady state equation for the shape function. In the adiabatic transient method, the equations are coupled through the point kinetics parameters which are defined in terms of the angular flux solution. The point kinetics equations are solved for entire time domain while the steady state equation is recomputed intermittently to update the point kinetics parameters. The adiabatic method can be readily realized in existing transport or diffusion solvers but its applicability is inevitably limited for specific cases.

Recently, a full transient capability has been implemented to the SN2ND and MOCEX solvers of PROTEUS by the NEUP team [8]. Even though the implementation of the transient capability is not part of the planned tasks this year, its progress including preliminary results is briefly described in this section.

As the first step of the implementation, the time-dependent transport equation is casted into the transient fixed source problem (TFSP). In the TFSP formulation, the time-dependent term is moved to the right hand side (RHS) of the equation such that a transient equation is solved as a standard fixed-source steady-state equation at each time step.

\[
\Omega \cdot \nabla \psi^n(\mathbf{r}, \Omega, E) + \sum_q^n \psi^n(\mathbf{r}, \Omega, E) = 
\int_0^{\Delta t} \int_{E}^{E + \Delta E} \sum_q^n(\mathbf{r}, \Omega \cdot \Omega', E' \rightarrow E) \phi^n(\mathbf{r}, \Omega', E') d\Omega' dE' + \frac{\chi^n_p(\mathbf{r}, E) S^n_f(\mathbf{r})}{4\pi} + \frac{S^n_{tr}(\mathbf{r}, E)}{4\pi},
\]

where \( \psi^n, \phi^n \) = neutron angular and scalar fluxes, respectively, at the n-th time step,

\( \sum_q^n, \sum_v^n \) = total and scattering cross sections, respectively, at the n-th time step,

\( \chi^n_p, S^n_f \) = prompt neutron spectrum and fission source, respectively, at the n-th time step,

\( S^n_{tr} \) = transient source at the n-th time step.

The transient source appearing in Eq. (2-30) is composed of the time derivative term of angular flux named the \( 1/v \) term and delayed neutron sources. Using the isotropic approximation, the \( 1/v \) term is represented without angular dependency as:

\[
\frac{\psi^n(\mathbf{r}, \Omega, E) - \psi^{n-1}(\mathbf{r}, \Omega, E)}{v(E) \Delta t_n} \approx \frac{\phi^n(\mathbf{r}, E) - \phi^{n-1}(\mathbf{r}, E)}{4\pi v(E) \Delta t_n}.
\]

By employing the second precursor integration technique [25], the delayed neutron source term can be represented in terms of precursor density of previous time step and the fission source of current and two previous time steps.

The transient capability of SN2ND and MOCEX were tested using the 2D TWIGL [26] and 2D C5G7 [12] transient benchmark problems. TWIGL is a small reactor with three different
homogeneous regions as shown in Figure 34. TWIGL is a small reactor with three different homogeneous regions as shown in Figure 34. The TWIGL benchmark problem is composed of 3 regions as shown in the figure, in which the reactivity is linearly inserted until 0.2 sec, changed at 0.2 sec and linearly reduced until 0.4 sec, and then changed again at 0.4 sec.

![Figure 34. TWIGL Transient Benchmark Problem](image)

As listed in Table 13, eigenvalues for the initial steady-state condition are in very good agreement each other among SN2ND, DeCART, and MPACT [27] within 10 pcm. Relative power changes with time for TWIGL core power history are compared between PROTEUS solutions with different transport and transient solvers, as shown in Figure 35. The SN2ND and MOCEX solutions with TFSP matched well, which are off from the SN2ND solution with the adiabatic approximation. The results in Table 14 indicate that SN2D solution with the adiabatic approximation is inaccurate because the solutions from all other codes and solvers agree well in terms of all of peak power, asymptotic power, and integrated power. This indicates that the transient capability with TFSP to SN2ND and MOCEX was successful implemented.

<table>
<thead>
<tr>
<th>Code</th>
<th>DeCART</th>
<th>MPACT</th>
<th>PROTEUS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SN2ND</td>
</tr>
<tr>
<td>k-eff</td>
<td>0.91605</td>
<td>0.91601</td>
<td>0.91609</td>
</tr>
</tbody>
</table>
The transient capability of PROTEUS was tested using the well-known C5G7-TD OECD/NEA benchmark problems whose geometry (see Figure 36) and transient scenarios are more complicated than TWIGL. Among the 2D and 3D C5G7-TD benchmark problems, we tested only three 2D exercises (TD1 - TD3) which include 12 cases. The eigenvalues for the initial steady-state condition from PROTEUS and MPACT agree well with the MCNP solution within 10 pcm, showing that the PROTEUS eigenvalue is a little bit closer to MCNP, as shown in Table 15.

TD1 simulates transient events in the 2D configuration driven by the insertion of one or several control rod banks in the first 1 sec and withdrawal of those control rods in the next 1 sec. Five cases were considered in this exercise. TD2 simulates transient events in the 2D configuration driven by the insertion and withdrawal of control rods but with a higher amplitude. TD3 simulates transient events in the 2D configuration driven by a linear decrease of moderator density in the fuel assemblies following by a linear increase back to the original state. Four cases were considered in the exercise, each with a different value for the minimum fractional density during the transient. Brief descriptions are digested in Table 16.
Table 15. Comparison of Initial Steady-State Solution for C5G7 Transient Benchmark Results

<table>
<thead>
<tr>
<th>Code</th>
<th>DeCART</th>
<th>MPACT</th>
<th>MOCEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-eff</td>
<td>$1.18646 \pm 0.07%$</td>
<td>$1.18666$</td>
<td>$1.18651$</td>
</tr>
</tbody>
</table>

Table 16. Description of C5G7-TD Benchmark Problems

<table>
<thead>
<tr>
<th>Exercise ID</th>
<th>Number of Problems</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD1</td>
<td>5</td>
<td>Linear insertion of 1% of rod length, followed by linear removal of full 1% of rod</td>
</tr>
<tr>
<td>TD2</td>
<td>3</td>
<td>Linear insertion of 10% of rod length, followed by linear removal of full 10% of rod</td>
</tr>
<tr>
<td>TD3</td>
<td>4</td>
<td>Linear decrease of water density, followed by linear increase to original density</td>
</tr>
</tbody>
</table>

For TD1, the power history results from MOCEX for all 5 cases are plotted in Figure 37 in comparison with MPACT. A transient time step of 10ms was used for both MOCEX and MPACT. The MOCEX solutions are nearly identical to those from MPACT for each case. Detailed numerical comparisons are listed in Figure 37, showing that the maximum difference in power between MPACT and PROTEUS is 0.2%.

For TD2, the power history solutions from MOCEX during the transient events are plotted in Figure 38, compared with the MPACT solutions. A transient time step of 10 ms was used for both
MOCEX and MPACT. As expected, the power history results are nearly identical between PROTEUS-MOCEX and MPACT. Numerical comparisons are listed in Figure 38, showing that the maximum difference in power between MPACT and PROTEUS is 0.8% which is larger than TD1 but still small.

For TD3, the power history results from MOCEX for all 4 cases are plotted in Figure 39 in comparison with the MPACT solutions. Like the previous cases, a transient time step of 10 ms was used for both MOCEX and MPACT. The time-dependent powers from both codes are almost identical. Detailed numerical comparisons are listed in Figure 39, showing that the maximum difference in power between MPACT and PROTEUS is 0.9%.

<table>
<thead>
<tr>
<th>Case ID</th>
<th>RMS Difference, %</th>
<th>Max. Difference, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD 1-1</td>
<td>0.051</td>
<td>0.20</td>
</tr>
<tr>
<td>TD 1-2</td>
<td>0.034</td>
<td>0.11</td>
</tr>
<tr>
<td>TD 1-3</td>
<td>0.025</td>
<td>0.03</td>
</tr>
<tr>
<td>TD1-4</td>
<td>0.060</td>
<td>0.11</td>
</tr>
<tr>
<td>TD 1-5</td>
<td>0.073</td>
<td>0.24</td>
</tr>
</tbody>
</table>

Figure 37. Power Changes with Time (left) and Power Differences (right) between MPACT and PROTEUS for TD1 Exercises of the C5G7-TD Benchmark Problem

<table>
<thead>
<tr>
<th>Case ID</th>
<th>RMS Difference, %</th>
<th>Max. Difference, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD 2-1</td>
<td>0.096</td>
<td>0.77</td>
</tr>
<tr>
<td>TD 2-2</td>
<td>0.107</td>
<td>0.27</td>
</tr>
<tr>
<td>TD 2-3</td>
<td>0.085</td>
<td>0.19</td>
</tr>
</tbody>
</table>

Figure 38. Power Changes with Time (left) and Power Differences (right) between MPACT and PROTEUS for TD2 Exercises of C5G7-TD Benchmark Problem
3. **Updates in MC\textsuperscript{2-3}**

Preliminary thermal cross section libraries for the energy range from 10-5 eV to 5 eV are prepared in a 1700-group structure. The energy range from 0.1 eV to 5 eV that includes important thermal resonances is divided into 1625 groups, and the energy range from 10-5 eV to 0.1 eV, where the cross sections vary smoothly, is represented by 75 groups. The 1625-group structure is selected to represent the thermal resonances in the energy range from 0.1 eV to eV almost pointwise. As a result, the total number of ultrafine groups of MC\textsuperscript{2-3} becomes 3,483: 1,783 fast groups from 5 eV to 14.2 MeV and 1,700 thermal groups from 10-5 eV to 5 eV.

The NJOY code is used to generate the thermal scattering matrices and the interaction cross section libraries at the infinite dilute condition and target temperatures. A utility code for MC\textsuperscript{2-3} sub-library prepares the NJOY input files and the input file for another utility code to process the NJOY output files. The thermal scattering matrices and cross section libraries are generated using the RECONR, BROADR, THERMR and GROUPR modules of NJOY. The NJOY output-processing tool converts the output files of NJOY (in the GENDF format) in the formats of the thermal cross section libraries of MC\textsuperscript{2-3}. Since only the thermal scattering matrices are prepared with the NJOY code, the elastic and inelastic scattering cross sections from fast groups to thermal groups are prepared using the MC\textsuperscript{2-3} algorithms [4] except for hydrogen. For this, the routines of MC\textsuperscript{2-3} have been modified to comply with the thermal group structure. For hydrogen, the analytic scattering kernel is directly used.

The MC\textsuperscript{2-3} code solves the consistent P\textsubscript{1} transport equation to determine the fundamental mode spectrum in a homogeneous problem, whereas the collision probability method (CPM) is used to solve 1D slab and cylindrical geometry problems. As mentioned in the introduction, a 2D MOC solver has been implemented to solve two-dimensional pin cell and lattice problems. In this MOC solver, anisotropic scattering is modeled up to P\textsubscript{3}, whereas the current CPM module of MC\textsuperscript{2-3} is limited to the isotropic scattering source with transport correction.
For the calculation of thermal spectrum, the 2D MOC transport solver as well as the homogeneous and 1D CPM solvers has been extended to comply with the thermal group cross sections and to perform the upscattering iterations in the thermal energy range. The Gauss-Seidel method is used for the upscattering iteration.

As mentioned above, the spectrum calculation is performed in the ultrafine group (UFG) or hyperfine group (HFG) level. In the UFG calculation, the self-shielded cross sections are determined by the numerical integration of pointwise cross section based on the NR approximation.

Since the NR approximation assumes that the resonance width is much smaller than the average energy loss per scattering, the NR approximation is only valid above a few hundred eV. The errors in the ultrafine group cross sections due to the NR approximation can be eliminated by the HFG calculation in the resolved resonance range. In this case, the self-shielded UFG cross sections are recalculated using the HFG flux distribution. The isotropic and anisotropic elastic scattering sources are directly incorporated in the HFG calculation, but the fission, inelastic and (n,2n) sources are interpolated from the corresponding UFG sources.

As an initial verification test of the thermal spectrum calculation capability of MC2-3, various homogeneous composition and 2D pin cell problems were solved, and the results were compared with the MCNP6 solutions. Both MC2-3 and MCNP6 calculations were performed using the ENDF/B VII.0 data. The MCNP6 calculations were performed with 1,000 active cycles and 100,000 histories per cycle. Table 17 compares the k-infinity values determined with MC2-3 and MCNP6 for the homogenized compositions of UO2 and MOX pin cell problems. Here, MC2-3 w/ HFG indicates the MC2-3 UFG solution with the self-shielded UFG cross sections recalculated with the HFG flux calculation. It can be seen that the MC2-3 results agree well with the MCNP6 solutions and that the HFG calculation improves the accuracy by eliminating the limitation of NR approximation.

It was observed that in the unresolved resonance range, MC2-3 yields a smaller absorption cross section of Pu238 and a larger fission cross section of Pu239 than MCNP6. These differences are attributed to different unresolved resonance self-shielding methods of MC2-3 and MCNP6 (i.e., the direct integration method vs. the probability table method). These differences in Pu isotope cross sections make the difference in k-infinity between MC2-3 and MCNP6 larger for the MOX problem than the UO2 problem. Figure 40 compare the 3,483-group spectra obtained with MC2-3 and MCNP6. It can be seen that the spectra calculated with MC2-3 w/ HFG agree well with the MCNP6 solutions.

Table 18 presents the k-infinity values calculated with MC2-3 and MCNP6 for 2D UO2 and MOX fuel pin cell problems. The 2D MOC calculations of MC2-3 were performed with a ray spacing of 0.025 cm, 32 azimuthal angles and 4 polar angles. It can be seen that the k-infinity values of MC2-3 agree well with the MCNP6 results. It is noted that in the MOX problem, MC2-3 results agree better with the MCNP6 solutions for the 2D problem than the homogenized problem. It is considered due to the error cancellation in the fuel region of the 2D problem. It is
also noted that the MC$^2$-3 results approach the MCNP6 solutions with increasing order of anisotropic scattering.

Table 19 presents the k-infinity values of MC$^2$-3 and KENO-VI for the 2D pin cell and lattice problems in hot zero power (HZP) condition at the beginning of cycle (BOC) of the VERA benchmark problems [28]. The 2D MOC calculations of MC$^2$-3 were performed with a ray spacing of 0.05 cm, 16 azimuthal angles and 4 polar angles. Since the UFG cross section errors induced from the NR approximation are small for UO$_2$ fuel, the HFG calculations were not invoked for the VERA benchmark problems. The P$_2$ anisotropic scattering was used based on the pin cell results shown in Table 19 since the P$_2$ anisotropic scattering is sufficient for typical LWR fuel pin cells.

![Figure 40. UO2 (left) and MOX (right) Neutron Spectra Obtained from MC$^2$-3 and MCNP6](image)

Table 17. Eigenvalues of MC$^2$-3 and MCNP6 for Homogenized Compositions of UO$_2$ and MOX Fuel Pin Cell Problems

<table>
<thead>
<tr>
<th>Case</th>
<th>Fuel/moderator density (g/cm$^3$), temperature</th>
<th>Code</th>
<th>$k_{\infty}$</th>
<th>$\Delta k$ (pcm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSNP UO$_2$ fuel pin [29]</td>
<td>10.061/0.660 300K/300K</td>
<td>MCNP6</td>
<td>1.08949 (6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC$^2$-3 UFG</td>
<td>1.09091</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC$^2$-3 HFG</td>
<td>1.08989</td>
<td>40</td>
</tr>
<tr>
<td>LWR MOX fuel pin [30]</td>
<td>10.300/0.660 300K/300K</td>
<td>MCNP6</td>
<td>1.13218 (6)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC$^2$-3 UFG</td>
<td>1.13462</td>
<td>237</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC$^2$-3 HFG</td>
<td>1.13378</td>
<td>152</td>
</tr>
</tbody>
</table>
### Table 18. Eigenvalues of MC²-3 and MCNP6 for 2D UO₂ and MOX Fuel Pin Cell Problems

<table>
<thead>
<tr>
<th>Case</th>
<th>Fuel/moderator density (g/cm³), temperature</th>
<th>Code</th>
<th>$k_\infty$</th>
<th>$\Delta k$ (pcm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSNP UO₂ fuel pin [29]</td>
<td>10.061/0.660 300K/300K</td>
<td>MCNP6</td>
<td>1.16679 (5)</td>
<td>-103</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₁) UFG</td>
<td>1.16576</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₁) HFG</td>
<td>1.16580</td>
<td>-99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂) UFG</td>
<td>1.16609</td>
<td>-70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂) HFG</td>
<td>1.16614</td>
<td>-65</td>
</tr>
<tr>
<td>LWR MOX fuel pin [30]</td>
<td>10.300/0.660 300K/300K</td>
<td>MCNP6</td>
<td>1.22609 (6)</td>
<td>-219</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₁) UFG</td>
<td>1.22390</td>
<td>-201</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₁) HFG</td>
<td>1.22408</td>
<td>-40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂) UFG</td>
<td>1.22569</td>
<td>-23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂) HFG</td>
<td>1.22586</td>
<td></td>
</tr>
</tbody>
</table>

### Table 19. Eigenvalues of MC²-3 and KENO-VI for 2D pin cell (P1A through P1D) and lattice (P2A through P2D) problems in HZP condition at BOC of VERA benchmarks

<table>
<thead>
<tr>
<th>Case</th>
<th>Fuel/moderator densities (g/cm³) and temperatures</th>
<th>Code</th>
<th>$k_\infty$</th>
<th>$\Delta k$ (pcm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1A</td>
<td>10.257/0.743 565K/565K</td>
<td>KENO-VI</td>
<td>1.18654 (2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂)</td>
<td>1.18667</td>
<td>13</td>
</tr>
<tr>
<td>P1B</td>
<td>10.257/0.661 600K/600K</td>
<td>KENO-VI</td>
<td>1.18182 (2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂)</td>
<td>1.18180</td>
<td>1</td>
</tr>
<tr>
<td>P1C</td>
<td>10.257/0.661 900K/600K</td>
<td>KENO-VI</td>
<td>1.17161 (2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂)</td>
<td>1.17078</td>
<td>-83</td>
</tr>
<tr>
<td>P1D</td>
<td>10.257/0.661 1200K/600K</td>
<td>KENO-VI</td>
<td>1.16310 (2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂)</td>
<td>1.16186</td>
<td>-124</td>
</tr>
<tr>
<td>P2A</td>
<td>10.257/0.743 565K/565K</td>
<td>KENO-VI</td>
<td>1.18193 (2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂)</td>
<td>1.18105</td>
<td>-88</td>
</tr>
<tr>
<td>P2B</td>
<td>10.257/0.661 600K/600K</td>
<td>KENO-VI</td>
<td>1.18284 (2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂)</td>
<td>1.18213</td>
<td>-71</td>
</tr>
<tr>
<td>P2C</td>
<td>10.257/0.661 900K/600K</td>
<td>KENO-VI</td>
<td>1.17350 (2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂)</td>
<td>1.17246</td>
<td>-104</td>
</tr>
<tr>
<td>P2D</td>
<td>10.257/0.661 1200K/600K</td>
<td>KENO-VI</td>
<td>1.16568 (2)</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC²-3 (P₂)</td>
<td>1.16425</td>
<td>-143</td>
</tr>
</tbody>
</table>
4. Utility Codes

4.1 Cross Section Parameterization Code

In the thermal feedback calculation, a set of cross sections tabulated with a function of state parameters is necessary to calculate cross sections with state parameters (temperature, density, control rod presence, etc., as applicable) online with no interruption during the iteration for solution convergence. In the past, a cross section set called ISOTAB [31] was developed to include cross sections tabulated with state parameters. However, it is based on isotopes which all are often hard to name with the limitation of 6 to 8 characters when a large core problem is solved with burnup. The isotope-based cross sections were in fact directed by the DIF3D need.

For use in the PROTEUS-NODAL, we developed a new type of cross section structure that is based on compositions, each of which contains a group of isotopes with fixed isotope names. In the structure, composition names should be difference one another, which is much easier to maintain different compositions necessary for any problems because the number of compositions are much smaller than the number of isotopes. To distinguish the new cross section set from ISOTAB, it was named as ISOPAR whose structure is shown in Figure 41.

![ISOPAR File Structure Generated from GenISOPAR Using Multiple ISOTXS Files](image)

The GenISOPAR code was developed to generate an ISOPAR using multiple ISOTXS files with different state parameters. ISOPAR is basically in the same format of ISOTXS but additionally contains a header block at the beginning of every composition which includes the
number of state parameters such as fuel temperature (K), moderator temperature (K), moderator density (g/cm$^3$), void fraction, boron concentration (ppm), control rod presence (0 or 1), and burnup (MWD/MTU) – as well as state parameter indices and values. The reference state at each burnup is specified in the header block.

An additional difference of ISOTAB from ISOTXS is to use the same fixed isotope names (e.g., U235, U238, PU239, etc.) in a composition, to use the number of compositions at the header line of the file instead of the number of isotopes. A disadvantage of this tabulation approach is to need a larger amount of cross section data compared to the functionalization approach.

4.2 Post-Processing Code for PROTEUS-MOCEX

A post-processing tool of MOCEX is necessary to streamline comprehensive analyses by providing a convenient way to process data of interest and to visualize detailed 3D solutions after MOCEX calculations are completed. In FY17, the preliminary version of post-processing tool was developed by utilizing on various functions in PROTEUS mesh toolkit [15]. The output editing module of MOCEX was also extended such a way to export the obtained MOCEX results into a binary file format acceptable for the post-processing tool.

Overview of Post-Processing Tool

The MOCEX code provides the element-wise group fluxes, fission source and power as the outcomes of calculation via an output file referred to MOCEX solution file. Additionally, the core configuration data is also provided in the solution file for input verification purpose. For processing the data available in the MOCEX solution file, the post-processing tool equips three functions:

- Visualizing MOCEX results such flux and power distribution
- Editing block-wise quantities based on the region tags attached in finite elements
- Extracting pin-wise or assembly-wise quantities by making use of the user-defined coarse mesh structure

For executing the post-processing tool, the MOCEX solution file and the relevant 2D mesh file are required along with a control input. The control input file contains information for output data of interest, data editing options and axial mesh structure used in the preceding MOCEX run. Based on the axial mesh structure and the 2D mesh file, the post processing tool reconstructs the 3D extruded geometry and then assigns the MOCEX solution data onto the corresponding 3D elements, which forms the database to be processed. Based upon these database, the above three functionalities were preliminarily implemented in this year.

Visualization of MOCEX Solution

The post-processing tool can convert a MOCEX solution file into the VTK file format so that VisIt [32] can visualize the MOCEX solution. In the implementation, the PROTEUS mesh toolkit was utilized to convert NTmesh into the VTK mesh format. By using the various options
provided in VisIt, the MOCEX solutions can be further processed. Figure 42 shows examples of MOCEX visualization using this capability.

The post-processing tool provides the option for problem domain partitioning, which is useful for visualizing a large domain in PC environment. With this option, a domain to be visualized is decomposed into several sub-domains by employing the METIS library. Then, each sub-domain and associated data are written into the separate VTK file. When these VTK files are visualized into a single 3D object, the processing time is significantly reduced due to the eliminated connectivity information among the sub-domains.

Figure 42. Examples of MOCEX Visualization (TREAT-M8CAL)

Data Condensing to Coarse Mesh Structure

To provide a simple way to extract the data of interest from the MOCEX solution file, the post-processing tool equips a function to condense output data based upon a coarse mesh structure. This function can be mainly utilized for yielding the pin- or assembly-wise quantities by choosing the pin or assembly configuration as the coarse mesh structure. These resulting quantities, such as assembly-wise power distribution, can be used for the V&V of MOCEX by comparing with the measured data or the reference Monte Carlo solutions. Figure 43 is an example of this processing capability for RCF calculation [11]. Alternatively, these coarse-mesh quantities can be obtained in a cumbersome way by making use of the tag name attached to each mesh block. It usually requires a user to prepare an addition script tailored for specific MOCEX run. Hence, this function can significantly reduce an effort to post-data processing.

A radial coarse mesh structure for data processing is defined using the mesh formats supported in PROTEUS code. The structured coarse mesh structure can be easily generated using the UFmesh or GRID format of mesh toolkit. The axial coarse mesh structure is specified in control input. This function was implemented such a way that the MOCEX solution defined on the
detailed mesh structure is projected onto the user-defined coarse mesh structure. Currently, the PROTEUS mesh format does not explicitly support a hexagonal element. Instead, a hexagon is represented into six triangular or two quadrilateral elements. Thus, an additional process is to yield the hexagon-averaged values. The hexagonal element will be added to the PROTEUS mesh format in the next year.

(a) Original Data   (b) Condensed Data

Figure 43. Example of Coarse Mesh Editing Function in MOCEX Post-processing Tool (RCF)
5. Verification and Validation Tests

As a code verification and validation effort, the numerical benchmark problems (C5G7) and experiment-based benchmark problems (TREAT and RCF) were solved using PROTEUS-MOCEx whose solutions were compared with Monte Carlo solutions. The verification tests were initiated last year, and further progress was made this year. Details are presented in this section.

5.1 C5G7 Benchmark Problem

In order to verify the solution accuracy of PROTEUS-MOCEx, the 3D C5G7 benchmark problems were analyzed by solving and comparing eigenvalues and pin powers with the reference MCNP solutions. [33]

MOCEX Model

The solution of MOCEX converges to that of a Monte Carlo code as discretization of angle and space is refined. In order to obtain the faithful result of MOCEX, the very refined discretization was used for the spatial and the angular domains of the C5G7 problem. Due to symmetry of the problem in terms of geometry and composition, a quarter-core radially and half-core axially is used. In the radial domain, each pin is divided into 3 and 2 annular rings annular rings for fuel and coolant regions, respectively, with 32 azimuthal sectors. The 2D meshes for UO\textsubscript{2}, MOX and reflector assemblies were separately generated using the UF mesh tool and they are merged into the 2D core mesh. Each 2D plane has the total 472,449 meshes. In the axial domain, the active core and top reflector regions are uniformly divided into 18 and 10 planes, respectively. Due to the strong flux variation along the axial direction caused by control rod insertions, this problem requires to use refined axial mesh structure. Figure 44 shows the 2D assembly and the axial plane models. Using the Legendre-Tchebychev cubature, the angular flux of each mesh is represented with the 384 directions that include 48 azimuthal and 8 polar divisions on the sphere. The mesh structure and angular flux discretization were determined through the sensitivity studies for the 2D and 3D C5G7 problems. It was also checked that MOCEX solutions were fully converged in terms of space and angle by using the meshes and angles above.

The 3D problem contains about 10 billion unknowns in each energy group, which can be solved using MOCEX in about 20 minutes with 1020 processors of the IBM BG/Q with the CMFD acceleration. For unrodded and rodded B cases, the resulting thermal fluxes (Group 7) and pin power distributions are visualized in Figure 45 and Figure 46, respectively.
Figure 44. Mesh Structures of MOCEX Models for C5G7 Benchmark Problems

(a) Radial mesh

(b) Axial mesh

Figure 45. Thermal Flux Distribution of C5G7 Benchmark Problems

(a) Unrodded

(b) Rodded B

Figure 46. Power Distribution of C5G7 Benchmark Problems

(a) Unrodded

(b) Rodded B
Comparisons with Reference MCNP Solutions

Table 20 shows the obtained eigenvalues for three cases of the C5G7 benchmark problems. For the unrodded case, the eigenvalue obtained using MOCEX has only 12 pcm differences compared to the MCNP solutions. For the rodded cases, eigenvalue differences slightly increase but they are still less than 30 pcm. Noting that the MCNP results have 6 pcm uncertainties, we can conclude that the obtained eigenvalues agree very well with the MCNP solutions.

The pin power distribution obtained using MOCEX was compared against the detailed MCNP solutions, in which the reference pin-wise powers are available for three axial regions as shown in Figure 44(b). The pin power comparisons are summarized in Table 21. The pin powers of MOCEX agree very well with the MCNP solutions within 1.4 % for the unrodded and rodded cases. For three cases, Figure 47 through Figure 49 show the pin power errors at upper 1/3 regions and the associated uncertainty of reference MCNP solutions. Relatively large errors appear at the core periphery regions where the powers are very small and the associated uncertainties of MCNP solutions are substantially large. The C5G7 benchmark results indicated that the 3D transport solution of MOCEX are very accurate compared to the Monte Carlo solutions when the same cross sections were used for both codes.

Table 20. Eigenvalue Results of C5G7 Benchmark Problems

<table>
<thead>
<tr>
<th>Case</th>
<th>Code</th>
<th>MCNP (1-σ)</th>
<th>MOCEX</th>
<th>Diff, pcm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unrodded</td>
<td></td>
<td>1.14308 (6)</td>
<td>1.14296</td>
<td>-12</td>
</tr>
<tr>
<td>Rodded A</td>
<td></td>
<td>1.12821 (6)</td>
<td>1.12801</td>
<td>-20</td>
</tr>
<tr>
<td>Rodded B</td>
<td></td>
<td>1.07777 (6)</td>
<td>1.07750</td>
<td>-27</td>
</tr>
</tbody>
</table>

Table 21. Summary of Pin Power Results of C5G7 Benchmark Problems

<table>
<thead>
<tr>
<th>Case</th>
<th>Axial Region</th>
<th>MCNP 1-σ, %</th>
<th>MOCEX Error, %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max.</td>
<td>RMS</td>
<td>Max.</td>
</tr>
<tr>
<td>Unrodded</td>
<td>Lower 1/3 of Fuel</td>
<td>0.200</td>
<td>0.171</td>
</tr>
<tr>
<td></td>
<td>Middle 1/3 of Fuel</td>
<td>0.320</td>
<td>0.190</td>
</tr>
<tr>
<td></td>
<td>Upper 1/3 of Fuel</td>
<td>0.430</td>
<td>0.255</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>0.192</td>
<td>0.114</td>
</tr>
<tr>
<td>Rodded A</td>
<td>Lower 1/3 of Fuel</td>
<td>0.210</td>
<td>0.163</td>
</tr>
<tr>
<td></td>
<td>Middle 1/3 of Fuel</td>
<td>0.250</td>
<td>0.186</td>
</tr>
<tr>
<td></td>
<td>Upper 1/3 of Fuel</td>
<td>0.330</td>
<td>0.266</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>0.143</td>
<td>0.111</td>
</tr>
<tr>
<td>Rodded B</td>
<td>Lower 1/3 of Fuel</td>
<td>0.230</td>
<td>0.150</td>
</tr>
<tr>
<td></td>
<td>Middle 1/3 of Fuel</td>
<td>0.200</td>
<td>0.184</td>
</tr>
<tr>
<td></td>
<td>Upper 1/3 of Fuel</td>
<td>0.380</td>
<td>0.290</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>0.157</td>
<td>0.108</td>
</tr>
</tbody>
</table>
Figure 47. Pin Power Comparisons for Unrodded Configuration of C5G7 Benchmark Problems at Upper 1/3 of Fuel Region

(a) Rel. Error, % (Max: 1.40)  
(b) MCNP 1-\(\sigma\), % (Max: 0.43)

Figure 48. Pin Power Comparisons for Rodded A Configuration of C5G7 Benchmark Problems at Upper 1/3 of Fuel Region

(a) Rel. Error, % (Max: 1.00)  
(b) MCNP 1-\(\sigma\), % (Max: 0.33)

Figure 49. Pin Power Comparisons for Rodded B Configuration of C5G7 Benchmark Problems at Upper 1/3 of Fuel Region

(a) Rel. Error, % (Max: 1.24)  
(b) MCNP 1-\(\sigma\), % (Max: 0.38)
5.2 TREAT Cores

For PROTEUS calculations with MOCEX, meshes were generated using the meshing toolkit [15] of the PROTEUS package combined with the CUBIT mesh generation software [14]. Meshes for standard hexagonal or Cartesian types can be generated using the PROTEUS meshing tool alone, but components with complex or irregular geometries are generated using CUBIT and then are merged together using the PROTEUS meshing tool. For TREAT, the fuel, control rod, and zirconium-clad or aluminum-clad dummy elements with chamfered corners were modeled and meshed using CUBIT. These were then merged together to construct partial- or whole-core problems using the PROTEUS meshing tool. The meshes for the permanent graphite reflector (PGR), as well as the air gap between the core and PGR, were also generated using the PROTEUS meshing toolkit. Since it is based on the extruded geometry formulation, MOCEX requires 2D projected meshes only with which 3D geometry representation is given via a user input.

The meshes used for fuel and control rod element components are shown in Figure 50. The normal mesh was generated for initial verification tests of simple geometry models, but later on the 2D projected meshes were generated to simulate actual 3D cores with axially non-uniform geometries. For example, the normal meshes include only 4 regions for the fuel element and only 9 regions for the control rod element, while the 2D projected meshes are composed of 8 regions for the fuel element and 17 regions for the control rod element. In particular, the experiment vehicle located at the core center of M8CAL was separately generated in a three-element size and merged to complete the 3D whole-core M8CAL mesh. The hodoscope channel in the permanent graphite block was separately defined and merged as well.

As aforementioned, the graphite blocks above and below the hodoscope channel were represented using an additional input file containing the mapping between geometry and composition. Figure 50 illustrates the 2D whole-core mesh for M8CAL.

In this study, macroscopic cross sections were generated using Serpent [34] to focus on verification of the transport solution accuracy of MOCEX. An 11 group structure was determined based on preliminary studies, starting from the 23 energy group structure used for VHTR analysis [35] and making a group optimization effort to reduce the number of energy groups maintaining most of the accuracy of the 23 group solutions. The GenISOTXS code [36], which was developed by ANL to process cross sections generated from Serpent or OpenMC Monte Carlo codes, was used to verify macroscopic cross sections from Serpent and to convert them to the ISOTXS format readable by PROTEUS.

The 11 group cross sections for different compositions were generated for each element type to reduce the number of cross section sets necessary for PROTEUS calculations (although they should be generated for each region with different material, location, and number density considering correct resonance self-shielding and neutron spectrum effects). Preliminary analysis indicated that generating composition-wise cross sections for each element type along with 11
groups did not affect solution accuracy significantly. Since an accuracy issue was found in the high-order scattering moments generated from Serpent, the transport approximation approach was used, in which total cross sections and within-group scattering cross sections were adjusted by the total P\textsubscript{1} scattering cross sections. It was observed that eigenvalue solutions were increased by ~100 pcm with P\textsubscript{2} scattering cross sections generated from Serpent and there were almost no differences between solutions with P\textsubscript{2} and P\textsubscript{3} scattering cross sections.

Figure 50. 2D Projected Core Mesh of M8CAL.

For deterministic codes, M8CAL is neutronically more challenging than MinCC because of its large hodoscope air channel. Based on the 2D mesh shown in Figure 50, a 3D full-core M8CAL model, simpler than the actual M8CAL experiment, was constructed excluding the experiment vehicle and simplifying geometry and composition axially. The control rods were inserted to make eigenvalue of the simplified model become 1.00497: 8 control/shutdown control rods were inserted deeply by 91.5 cm, 8 transient control rods were located to 48.7 cm from the top, and 4 compensation/shutdown control rods were out of the core. The hodoscope air channel in the permanent graphite block is 48.4 cm wide and 121 cm high, as shown in Figure 51.

The eigenvalue solution of MOCEX agrees very well with the Serpent solution within 66 pcm. The fast and thermal neutron flux distributions from MOCEX are illustrated in Figure 52, showing a significant neutron streaming through the hodoscope channel. This indicates that an accurate estimation of the neutron stream effect to the hodoscope channel is very important for this problem.
Figure 51. Core configuration of Full 3D M8CAL (Simplified Model).

Figure 52. 3D Fast (left) and Thermal (right) Flux Solutions of PROTEUS for Full 3D M8CAL.

For more detailed verification, axially integrated power distributions were compared between Serpent and MOCEX. MinCC is 1/8-th symmetric and has no control rods inserted. Power distributions are smoothly varying with depressions (relative power: 0.7) at the control rod locations. Therefore, MOCEX and Serpent solutions agree excellently within maximum difference of 0.44 % and RMS difference of 0.22 %. Since most of the large errors come from the outmost fuel elements, differences are down to maximum difference of 0.30 % and RMS difference of 0.14 % if excluding the outmost fuel elements.

Power distributions of M8CAL vary significantly with locations due to control rods inserted and hodoscope air channel from the core center all the way to the outside of the core. The relative powers at the control rod locations and around the hodoscope channel are down to 0.3. Nevertheless, MOCEX and Serpent solutions are in very good agreements within maximum difference of 1.25 % and RMS difference of 0.50 %. Excluding the outmost 3 fuel blocks, differences are down to maximum difference of 0.97 % and RMS difference of 0.32 %. Power and percent difference distributions for 3D MinCC and M8CAL are illustrated in Figure 53.
### Table 22. Summary of Performance Examinations for TREAT Experiments

<table>
<thead>
<tr>
<th>Case</th>
<th>Serpent</th>
<th>MOCEX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Angle</strong></td>
<td><strong>Δk, pcm</strong></td>
</tr>
<tr>
<td><strong>Serpent</strong></td>
<td><strong>L5T15</strong></td>
<td></td>
</tr>
<tr>
<td>2D Fuel Element</td>
<td>1.66673 ± 3 d)</td>
<td>-32</td>
</tr>
<tr>
<td>MinCC</td>
<td>1.29939 ± 15</td>
<td>15</td>
</tr>
<tr>
<td>2D Core w/o PGR a)</td>
<td>1.23041 ± 22</td>
<td>-213</td>
</tr>
<tr>
<td>3D Single FE b)</td>
<td>1.45473 ± 20</td>
<td>214</td>
</tr>
<tr>
<td>3D 2x2 (3FE, 1CE)</td>
<td>1.27926 ± 21</td>
<td>149</td>
</tr>
<tr>
<td>3D 2x2 (2FE, 2CE)</td>
<td>1.16952 ± 21</td>
<td>18</td>
</tr>
<tr>
<td>3D Full Core</td>
<td>1.00490 ± 19</td>
<td>-1</td>
</tr>
<tr>
<td>M8CAL</td>
<td>1.37609 ± 16</td>
<td>249</td>
</tr>
<tr>
<td>3D Partial Core</td>
<td>1.00497 ± 18</td>
<td>66</td>
</tr>
<tr>
<td>3D Full Core c)</td>
<td>1.00497 ± 18</td>
<td>66</td>
</tr>
</tbody>
</table>

Figure 53. Axially Integrated Fuel Element Power (top row) and Percent Difference (bottom row) Distributions for MinCC (left) and M8CAL (right)
5.3 Reactor Critical Facility (RCF) Cores

For systematic verification tests, PROTEUS calculations using the MOCEX solver were performed for different configurations of the RCF core: a single pin, an assembly, a 2D partial core with or without control rods in, a 2D full core with or without control rods in, and a 3D full core. Based on the angle refinement and scattering order studies, a Legendre-Tchebychev angular cubature of L5T15 and a scattering order of P3 were used for all cases. In fact small differences in eigenvalue were observed between P2 and P3, which indicated that a scattering order of P2 can capture most of the anisotropic scattering effect.

A mesh refinement study indicated that 289 vertices corresponding to 96 elements were sufficient for the pin case, where the fuel, clad, and moderator regions were divided into 3, 1, and 2 rings, respectively, along with 16 radial sectors. The assembly is a 21x21 lattice including 325 fuel pins. For the 3D case, the number of elements in the core periphery region was largely reduced to relieve the computational burden.

As aforementioned, 11 group multigroup cross sections for MOCEX were extracted from Serpent outputs of each case. Single pin, assembly, 2D partial core, 2D full core, and 3D full core problems used 4, 5, 11, 13, and 55 different cross sections, respectively. In each case, one set of 4 cross sections representing fuel, gap, cladding, and moderator (water) were used for all fuel pins. The neutron flux spectrum for a pin looks like a typical LWR pin, but those from the 2D cores include a larger amount of thermal neutrons due to a large water reflector pool, as shown in Figure 55.

For the 3D case with 171.77 cm high, the fuel region is 91.44 cm long with top and bottom reflectors of 34.01 cm and 46.32 cm, respectively. The control rods are almost fully inserted up to 114.43 cm from the top. Based on the extruded 2D mesh, the 3D full core case includes 19 axial
regions and total 32 axial meshes, having less than 10 cm for each axial mesh. The vacuum boundary condition was applied to the radial boundary of the 2D full core case and all boundaries of the 3D full core case. The 2D and 3D thermal and fast flux solutions are illustrated in Figure 54 and Figure 57, and the power comparison for the 2D cores with or without four control rods are shown in Figure 56, indicating that power differences between PROTEUS and MCNP were maximum 1.3% and RMS 0.6% and the large differences occurred at below-average powers.

Table 23. Eigenvalue Solutions from MOCEX and Serpent

<table>
<thead>
<tr>
<th>Case</th>
<th>Serpent</th>
<th>MOCEX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Element (pcm)</td>
<td>∆k pcm</td>
</tr>
<tr>
<td>Single Pin</td>
<td>1.41782 (19)</td>
<td>96</td>
</tr>
<tr>
<td>Assembly</td>
<td>1.26661 (9)</td>
<td>212,544</td>
</tr>
<tr>
<td>2D Partial Core w/o CR</td>
<td>1.03301 (13)</td>
<td>369,523</td>
</tr>
<tr>
<td>w/ CR</td>
<td>1.01919 (12)</td>
<td>203</td>
</tr>
<tr>
<td>CR Worth</td>
<td>1312</td>
<td></td>
</tr>
<tr>
<td>2D Full Core w/o CR</td>
<td>1.03398 (12)</td>
<td>426,344</td>
</tr>
<tr>
<td>w/ CR</td>
<td>1.01946 (12)</td>
<td>281</td>
</tr>
<tr>
<td>CR Worth</td>
<td>1377</td>
<td></td>
</tr>
<tr>
<td>3D Full Core</td>
<td>0.99327 (12)</td>
<td>51,162a</td>
</tr>
</tbody>
</table>

(Standard deviation of Serpent, pcm; a: 2D mesh; b: ∆ρ)

Figure 55. Fast (left) and Thermal (right) Flux Solutions from MOCEX for 2D RCF Cores with and without Control Rods
5.4 VHTR Assemblies

As an effort to verify the computational capabilities of the PROTEUS code, the MOCEX code was applied to the VHTR reactor problems. The 2D and 3D VHTR assembly problems derived from the control block configuration of NGNP core design [37] were solved using the MOCEX code. Figure 58 shows the 2D and 3D mesh structure of assembly problems. The 3D assembly problem is especially challenging in that the axial streaming effect is so pronounced in the large control rod hole as shown in the figure. Thus, the diffusion and 2D/1D solvers cannot capture this streaming effect properly. Using the Serpent code, the 11 group cross section sets were obtained.
for the 2D and 3D assembly problems along with the reference eigenvalues. The MOCEX calculations for these assembly problems were performed with the L7T15 angular cubature and the P₃ anisotropic scattering treatment, and the resulting flux distributions are visualized in Figure 59 and Figure 60. Table 24 compares the obtained eigenvalues with the reference Monte Carlo values. The eigenvalues obtained from the both codes agree well for 2D and 3D problems within the 100 pcm differences. These results indicate that the MOCEX can properly treat the strong streaming effect through the large control rod hole and the substantial neutron thermalization in the graphite block.

<table>
<thead>
<tr>
<th>Case</th>
<th>Code</th>
<th>SERPENT (1-σ)</th>
<th>MOCEX</th>
<th>Diff, pcm</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Assembly</td>
<td></td>
<td>1.37626 (21)</td>
<td>1.37533</td>
<td>-93</td>
</tr>
<tr>
<td>3D Assembly</td>
<td></td>
<td>1.36199 (18)</td>
<td>1.36115</td>
<td>-84</td>
</tr>
</tbody>
</table>

Table 24. Eigenvalue Results of VHTR Assembly Problems

![Mesh Structures of MOCEX Models for VHTR Assembly Problems](image)

(a) 2D Assembly  
(b) 3D Assembly

Figure 58. Mesh Structures of MOCEX Models for VHTR Assembly Problems
Figure 59. Flux Distributions of 2D VHTR Assembly Problem

(a) Fast Flux
(b) Thermal Flux

Figure 60. Flux Distributions of 3D VHTR Assembly Problem

(a) Fast Flux
(b) Thermal Flux
6. Conclusions

One of the major FY17 accomplishments for the NEAMS neutronics tools include the development of an advanced nodal solver in PROTEUS for use in routine nuclear reactor design and analysis. We also created a simplified sub-channel based thermal-hydraulic (T/H) module for use in PROTEUS-NODAL and PROTEUS-MOCEX to allow the thermal feedback to be computed in a time efficient manner compared with SHARP. Another major accomplishment are substantial performance improvements to PROTEUS-MOCEX using numerical acceleration and code optimization. Finally, we also improved the thermal cross section capability of MC\textsuperscript{2}-3 and demonstrated the accuracy of PROTEUS-MOCEX using several verification and validation tests.

The PROTEUS-MOCEX code was extensively updated to improve the performance and usability. The CMFD acceleration scheme was successfully implemented in MOCEX by incorporating a generic coarse mesh structure within the unstructured FEM framework of PROTEUS, tallying the surface currents on coarse mesh surfaces, and adding an under-relaxation scheme to the solution process to stabilize the convergence of the CMFD acceleration. Furthermore, the transport solver of MOCEX was optimized for use on large scale computing machines (e.g., BG/Q). Test results for the 3D C5G7 and TREAT problems indicated that the overall performance of MOCEX with CMFD acceleration and optimization was improved by about a factor of 10. A restart capability was implemented into MOCEX to enhance the code usability when running large problems on the parallel machine (very useful for perturbation theory calculations). The output processing tools were also developed to provide a convenient way to process and extract the data of interest from the PROTEUS output.

The PROTEUS-NODAL code was developed based upon the DIF3D-VARIANT methodology to provide a rapid and accurate calculation capability for users focusing on the scaling and design type analysis activities of nuclear reactors. The diffusion option now works for Cartesian, hexagonal and triangular-z geometries. In addition, the nodal simplified P\textsubscript{3} (SP\textsubscript{3}) solution option was used with a combination of radial triangle based polynomials and axial nodal expansion to expand the functionality past simple diffusion theory. Verification tests with the ABTR and MONJU fast reactor benchmark problems indicated that the NODAL SP\textsubscript{3} results agreed with the reference DIF3D-VARIANT P\textsubscript{8} transport solutions within 200 pcm for eigenvalues and 1% for fuel assembly powers.

Because Nek5000 requires substantial computational resources, a built-in T/H solver based on a simplified sub-channel model was added to PROTEUS. This capability was made available to both PROTEUS-NODAL and PROTEUS-MOCEX. To allow cross section variance with respect to state parameters including temperature, a tabulated cross section data set named ISOPAR was created for PROTEUS along with a utility code to generate ISOPAR data files from base ISOTXS files.
As part of work to continue improving the cross section generation capability, the thermal cross section capability in MC\(^2\)-3 was updated using the 1,700 thermal groups in the thermal energy range below 5 eV. Verification tests with various types of pin and assembly problems for light water reactors showed that MC\(^2\)-3 with the updated cross sections could predict eigenvalues and power distributions accurately.

For code verification and validation efforts, several thermal reactor problems were modeled and simulated especially using PROTEUS-MOCEX. From the 3D C5G7 problems, it was confirmed that the MOCEX solutions were converging to the Monte Carlo solutions as the angle and space discretization was refined. Further detailed calculations using MOCEX were performed for the M8CAL core of TREAT and the Reactor Criticality Facility core at RPI. Comparisons between MOCEX and MCNP solutions demonstrated that MOCEX could predict eigenvalues and powers very accurately, especially, even for the M8CAL core which is challenging to simulate due to the large neutron streaming through the hodoscope channel. Additionally, the 2D and 3D prismatic-type VHTR assembly problems based on the NGNP design were solved in order to demonstrate the accuracy of MOCEX for VHTR core analysis which is challenging for deterministic codes due to the neutron streaming in the large control rod holes.
REFERENCES


