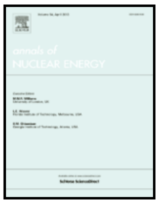
Annals of Nuclear Energy xxx (xxxx) 109365

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com



Code Verification and Solution Verification framework in pin-resolved neutron transport code MPACT☆

Jipu Wang[a, b, *], William R. Martin[b, *], Thomas J. Downar[b], Brendan Kochunas[b], Nathan C. Andrews[c, d], Lindsay Gilkey[d], Erik D. Walker[e], Benjamin S. Collins[e], Martin Pilch[f]

[a] *Shenzhen University, Shenzhen, Guangdong 518060, China*
[b] *University of Michigan, Ann Arbor, MI 48109, USA*
[c] *Sandia National Laboratories, Albuquerque, NM 87185, USA*
[d] *Sandia National Laboratories, Albuquerque, NM 87185, USA*
[e] *Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA*
[f] *MPilchConsulting, 26 Mustang Mesa Trail, Tijeras, NM 87059, USA*

ARTICLE INFO

Article history:
Received 16 February 2022
Received in revised form 22 July 2022
Accepted 1 August 2022

Keywords:
Verification
Code Verification
Solution Verification
Method of Manufactured Solutions
Convergence

ABSTRACT

Program verification in scientific computing encompasses the application of formal and mathematical techniques to a scientific computing code for its credibility, accuracy, and validity. Code Verification identifies bugs and performance issues in the software development stage. Solution Verification assesses the applicability of the code and the accuracy of the solution to problems of interest. Both activities utilize application cases and quantify the error against prescribed acceptance criteria. However, simply executing more application cases does not guarantee stronger or more comprehensive credibility. In this work, we establish a verification framework that involves Code Verification and Solution Verification, both of which work together such that the overarching goal of “converge to the correct answer for the intended application” can be reasonably inferred. The application of such a verification framework is demonstrated using the pin-resolved neutron transport code MPACT, where standard unit tests and regression tests are covered, and where the Method of Exact Solutions and the Method of Manufactured Solutions are successfully used. Additionally, the applicability of Method of Manufactured Solutions is extended to the OECD/NEA C5G7 benchmark problems of practical material and geometric configurations. Solution Verification activities are demonstrated on a practical hierarchy of application models of increasing complexity ranging from 2D pin cell problems to 3D assembly problems. The convergence behavior and rate of convergence with respect to each individual variable are studied and provided. This framework can be adapted broadly to other fields involving scientific computing codes.

1. Introduction

Program verification in scientific computing encompasses the application of formal and mathematical techniques to a scientific computational code for its credibility, accuracy, and validity. There is little to no doubt about the importance of Code Verification among the code development community, as Code Verification identifies bugs and performance issues in the software, which is generally considered the responsibility of code developer(s). On the other hand, Solution Verification

☆ Notice: This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (https://energy.gov/downloads/doe-public-access-plan).

* E-mail address: jipuwang@szu.edu.cn

* E-mail address: wrm@umich.edu

E-mail addresses: jipuwang@szu.edu.cn (J. Wang), wrm@umich.edu (W.R. Martin), downar@umich.edu (T.J. Downar), bkochuna@umich.edu (B. Kochunas), andrews@lanl.gov (N.C. Andrews), lngilke@sandia.gov (L. Gilkey), walkered@ornl.gov (E.D. Walker), collinsbs@ornl.gov (B.S. Collins), MPilchConsulting@gmail.com (M. Pilch).

https://doi.org/10.1016/j.anucene.2022.109365
0306-4549/© 20XX

assesses the applicability of the code and the accuracy of the solution to problems of interest. The responsibility of Solution Verification falls on the shoulders of either code developers or code users.

Myriad Code Verification application cases are discussed in numerous publications in any field involving scientific computing codes. There are even more code validation publications, consisting of a large body of application cases. However, the practice of merely growing the number of application cases for a code does not guarantee stronger or more comprehensive credibility. Furthermore, this practice reveals a common confusion on the correlation between Code Verification and Solution Verification. We believe that the value of Solution Verification extends beyond a verification tool in the hands of analysts. It is also a valuable tool in the code development process, which is not well appreciated and published.

In this work, we establish a verification framework that involves both Code Verification and Solution Verification. The values of each and the correlation between both are discussed and demonstrated successfully using our pin-resolved neutron transport code MPACT, or Michigan PArallel Characteristics Transport code (Kochunas et al., 2013). It is demonstrated that a well-designed suite of just enough representative categories of verification cases can provide a more convincing and comprehensive body of evidence to establish credibility for a code. This framework can be adapted broadly to other fields involving scientific computing codes.

2. Verification framework and its application in MPACT

Computational programs are expected to *converge* to the *correct answer* for *the intended application* (Pilch, 2019). Traditional code testing activities, such as unit testing, regression testing, and acceptance testing, have clear value in supporting robust code development, but there remains a substantial leap of faith that the code will converge to the correct answer for the intended application.

Code Verification tests the ability of the code to *converge* to the *correct answer* where the correct answer can be a benchmark with a known analytical or manufactured solution. Solution Verification, on the other hand, tests the ability of the code to *converge* for *the intended application*. Code Verification and Solution Verification each have gaps relative to the overarching goal; but this goal can be reasonably inferred from a verification framework that involves both.

In this framework, error is an essential metric that can be quantified. Code Verification *evaluates* computational error by comparing the computed solution with an analytical solution, from which rate of convergence can be obtained via a grid refinement process. Solution Verification *estimates* the computational error by analyzing the convergence behavior (e.g., Richardson extrapolation). Furthermore, the observed order of convergence in Solution Verification can be an indicator of code errors or algorithm deficiencies if it differs significantly from expectations established by theory or Code Verification activities. Note that for large application models, Solution Verification can be computationally challenging and often prohibitively impractical. In this situation, this framework recommends a building block approach where convergence behavior and numerical errors of large system models can be informed by a body of evidence established through Code Verification and then Solution Verification on a practical hierarchy of simpler application models of increasing complexity. This will be detailed and demonstrated using our neutron transport code MPACT.

The MPACT code (Kochunas et al., 2013) is part of the suite of codes developed under the Consortium for Advanced Simulation of Light Water Reactors (CASL) Program funded by the U.S. Department of Energy (DOE) to bring the benefits of high-performance computing to issues of importance to the U.S. nuclear power industry. To ensure that MPACT converges to the correct answer for the intended application, a verification framework supported by a suite of test problems is developed.

The framework emphasizes the Code and Solution Verification activities that *quantify numerical errors and convergence behavior* for a set of grids. The ensemble of Code and Solution Verification activities can be given context by considering the terms that are candidates for grid refinement. First, we do not consider energy as a candidate for grid refinement. The discretization of neutron energy into energy groups is not amenable to normal discretization methods. Further, the numerical errors associated with the discretization of energy into the default 51 group structure has already been studied by Palmtag (Palmtag, 2017) – by comparing the computational results from MPACT with those obtained from a continuous energy Monte Carlo code. Moreover, the above study showed that the 51-group library was adequate for the Pressurized Water Reactor (PWR) applications studied here.

Grid refinements for the neutron transport equation, as solved by the Method of Characteristics (MOC) code MPACT, can be particularly challenging because there are still six parameters that may be refined, and the convergence behavior for most of these terms is not well understood from theory or from Code Verification studies. The six terms that are candidates for refinement are.

(1). axial spatial division (z),
(2). radial spatial division (r),
(3). azimuthal spatial division (a)
(4). ray spacing (rs),
(5). polar angle (θ), and
(6). azimuthal angle (ϕ).

This paper is organized such that Section 3 focuses on Code Verification activities, including the common unit tests, regression tests etc., the Method of Exact Solutions (MES), and the state-of-the-art Method of Manufactured Solutions (MMS). Section 4 discusses Solution Verification activities in assembly geometry (global convergence) and in pin geometry (global and term-by-term convergence), respectively. Given that there were no expectations for convergence rates of all the six grid refinement terms listed above, a general approach to quantify numerical errors and global and term-by-term convergence rates is used. Last, Section 5 provides concluding remarks for this work.

3. Code Verification

3.1. Unit tests, regression tests, and testing suite

Source Code Verification during code development in MPACT has been directed toward identifying mistakes in the source code by establishing comprehensive software testing practices (Kochunas et al., 2013). The two principal components of source code testing in MPACT are unit testing and regression testing. Unit testing is a software testing method by which individual units of source code are tested in isolation to determine whether they are functioning as intended. In contrast, regression testing seeks to uncover new software bugs or *regressions* in existing functional and non-functional areas of the code after changes have been made. Regression tests also work as integration tests, in which individual units are tested as a group to show they work together. The following subsections will cover unit testing and regression testing practices in MPACT.

3.1.1. Unit testing

The overall goal of unit testing is to isolate each part of the program and show that the individual parts are correct. The unit testing in MPACT was designed to verify the smallest testable part of a program, and each test case was designed to be independent of the rest. This involves creating unit tests for all functions and methods. When the tests pass, the code development phase of the associated unit is considered complete. However, if a unit test fails, there is likely to be a bug in either the code unit or the tests, and the development phase of the unit and its associated tests are reexamined until all its associated tests pass. The unit tests accelerate the process of correcting the bug by allowing the location of the fault or failure to be easily traced. During MPACT development, unit testing has served the important role of finding problems early in the development cycle. All unit tests in MPACT are run repeatedly via an automated process. This has simplified the process of locating a fault or failure because the automated process alerts the development team of the problem before the code is handed off to testers or users.

Because unit testing tests only the functionality of the units themselves, unit testing will not catch every error in the program. Specifically, unit testing will not identify integration errors or broader system-level errors (such as functions performed across multiple units or non-functional areas such as performance). Therefore, unit testing is performed in conjunction with regression testing as described in the next subsection.

3.1.2. Regression testing

Regression testing serves an important role to ensure that changes in one part of the code do not introduce faults in other parts. The primary measure is to provide a series of functional tests that are repeatedly performed during code development so that the code output can be compared against previously recorded outputs of these tests to ensure that new features and enhancements do not alter the reproducibility of existing features. These regression tests are more comprehensive than unit tests and are designed to exercise significant sections of the program with various inputs. The MPACT regression testing targets key features that the user will need when applying the code to practical Light Water Reactor (LWR) problems. In addition, new regression tests in MPACT are sometimes added after software fixes. When a bug is located and fixed, a test is recorded that exposes the bug, and the test is rerun regularly after subsequent changes to the program. Previously completed unit and regression tests are rerun regularly to identify the re-emergence of previously fixed faults. During the code development process, the MPACT developer can also systematically select the appropriate minimum set of tests needed to adequately cover a particular change. After completion of the developer's change, all tests are then performed as part of the regression testing.

The regression tests in MPACT comprise a scripted series of program inputs with a driver layer that links to the code without altering the code being tested. An automated system is in place to rerun all regression tests nightly and generate a report of any failures. These tests are compared with previous solutions from MPACT to ensure consistent answers. The acceptance criteria for regression test problems in MPACT are currently set to be ± 10 pcm for the eigenvalue k_{eff}, 0.1 % for pin power absolute root-mean-square (RMS), and ± 1 ppm for the boron concentration.

3.1.3. MPACT regression test suite

As part of the c, a comprehensive regression test matrix was developed (Collins et al., 2016) to identify the key features of the code requiring coverage and to provide the roadmap for the development of regression testing in MPACT.

3.2. Code Verification using the method of exact solutions

3.2.1. Method of exact solutions

MES seeks problems with analytic solutions, which usually involves consulting the literature for published solutions. The exact solution is a closed-form mathematical expression that gives values of the solution at all locations in space and time. The problem with an exact solution can be modeled using the code, from which the numerical solutions can be compared against the exact solution. If the error is smaller than a prescribed acceptance criterion (e.g., round-off error), the code is verified. Furthermore, one can perform a grid refinement analysis using the same problem and observe the convergence rate, which can be compared against mathematical expectations if they exist or can be used to establish expectations otherwise.

3.2.2. Example: Ganapol benchmark

Benchmark problem 3.4 of Ganapol's analytical benchmark book (Ganapol, 2009) is an excellent Code Verification test problem that can be modeled using MPACT without special coding. Therefore, it can be included in the MPACT regression suite.

3.2.2.1. Ganapol benchmark description. The Ganapol benchmark problem is an analytic, or "semi-analytic," benchmark based on the exact solution of the singular integral equation that describes the single-group cylindrical transport problem. This solution methodology is a complex sequence of steps described in detail in (Ganapol, 2009).

The configuration is a homogeneous right circular cylinder that is infinite in height:

The homogeneous cylinder as depicted in Fig. 1 has radius $r = R$ and height $h \to \infty$ with a total cross section Σ_t and c secondary neutrons per collision, where $c = \left(\Sigma_s + \nu\Sigma_f\right)/\Sigma_t$; Σ_s and Σ_f represent scattering and fission cross section, respectively. The radius r is given in terms of mean free paths (mfp), so the physical radius is r/Σ_t. Benchmark results are given for several cases:

(a) Uniform isotropic source—the scalar flux $\phi(r)$ is tabulated for selected values of $c<1$
(b) Critical rod—the critical radius is tabulated as a function of $c>1$
(c) Critical rod—the scalar flux $\phi(r)$ is depicted for critical rods as a function of $c>1$

3.2.2.2. MPACT benchmark problem. For MPACT, critical rod problems (b) and (c) described in Section 3.2.2.1 are chosen as benchmark cases because they exercise both the 2D MOC solver and the eigenvalue solver.

Table 1 shows a table from Ganapol (Ganapol, 2009); it is the benchmark results for the critical rod problem and gives the critical rod radius as a function of c. The tabulated results are correct up to the last digit known to ± 1 unit, hence within 8 decimal points. This table shows the agreement with previously tabulated benchmark results (i.e., Ref. (Palmtag, 2017) and Ref. (Yamamoto et al., 2007)), where the shaded results indicate the digit that Ganapol (Ganapol, 2009) disagrees with.

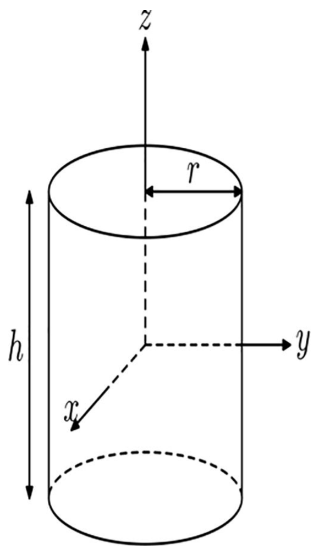


Fig. 1. Finite homogeneous cylinder.

Table 1
Critical radii comparisons from Table 3.4.3 (a) of Ganapol (2009).

c	R (Ref. (Palmtag, 2017))	R (Ref. (Yamamoto et al., 2007))
1.01E+00	1.312551649E+01	1.312551647E+01
1.02E+00	9.04325485E+00	------------
1.05E+00	5.41128829E+00	------------
1.10E+00	3.57739130E+00	3.57739129E+00
1.20E+00	2.28720926E+00	------------
1.30E+00	1.72500292E+00	1.72500292E+00
1.40E+00	1.39697859E+00	------------
1.50E+00	1.17834085E+00	1.17834085E+00
1.60E+00	1.02083901E+00	------------
1.80E+00	8.07426618E-01	------------
2.00E+00	6.68612867E-01	6.6861287E-01

3.2.2.3. MPACT results. Because MPACT is used for LWR lattices, special input processing would be needed to model the isolated cylinder. To avoid this, the cylinder is modeled as a fuel pin inside a non-scattering square bounding box (i.e., a void or a pure absorber) with vacuum boundaries. The angular flux along the rays starting at the bounding box boundaries will remain zero until the ray intersects the rod, where the scattering and fission sources within the rod begin to contribute to the solution.

The critical rod cases provide the critical rod size as a function of c. To avoid adding special coding in MPACT, which will involve an outer iteration to converge on the critical rod radius, MPACT solved for the eigenvalue k using the critical rod radii in Table 1 and cross sections that yield the tabulated c. For each of these cases, MPACT is expected to yield $k = 1$ to some precision.

All cases were run with the following phase space discretization:

- Bounding box side length = 30 cm
- Ray spacing = 0.0005 cm
- Number of radial rings = 160
- Number of azimuthal slices = 32
- Quadrature set = CHEBYSHEV-GAUSS 32 24
- Convergence criterion = 1e-7 for both k and ϕ

Table 2 gives the MPACT eigenvalues for all values of c in Table 1. The resulting k's are all within a few pcm from criticality, demonstrating excellent agreement with the benchmark results. This is a stringent Code Verification problem because an actual transport solution, not a manufactured solution, is being computed by MPACT. These cases, or a subset of them, make convenient and effective additions to the MPACT regression test suite.

The computed cell-averaged scalar fluxes are plotted against the relative radial location. The fluxes were normalized such that the innermost ring has a common cell-averaged scalar flux of 1. Both the normalized and unnormalized scalar flux shapes are shown in Fig. 2.

For $c = 1.05$, we plot the computed cell-averaged critical flux together with the reference solution in the same plot as in Fig. 3(a). The error is plotted separately in Fig. 3(b). The cell-averaged flux agrees well with the Ganapol benchmark results, with a relative error of less than 1 % at any radial location.

Table 2
MPACT calculated k_{eff} versus critical rod radius.

c	R(mfp)	$\nu\Sigma_f$(/mfp)	k	Error (pcm)
1.01	13.125516490	0.41	0.9999757	−2.43
1.02	9.043254850	0.42	0.9999783	−2.17
1.05	5.411288290	0.45	0.9999837	−1.63
1.1	3.577391300	0.5	0.9999895	−1.05
1.2	2.287209260	0.6	0.9999968	−0.32
1.3	1.725002920	0.7	1.0000006	0.06
1.4	1.396978590	0.8	1.000002	0.2
1.5	1.178340850	0.9	1.0000004	0.04
1.6	1.020839010	1	0.9999968	−0.32
1.8	0.807426618	1.2	0.9999864	−1.36
2	0.668612867	1.4	0.9999621	−3.79

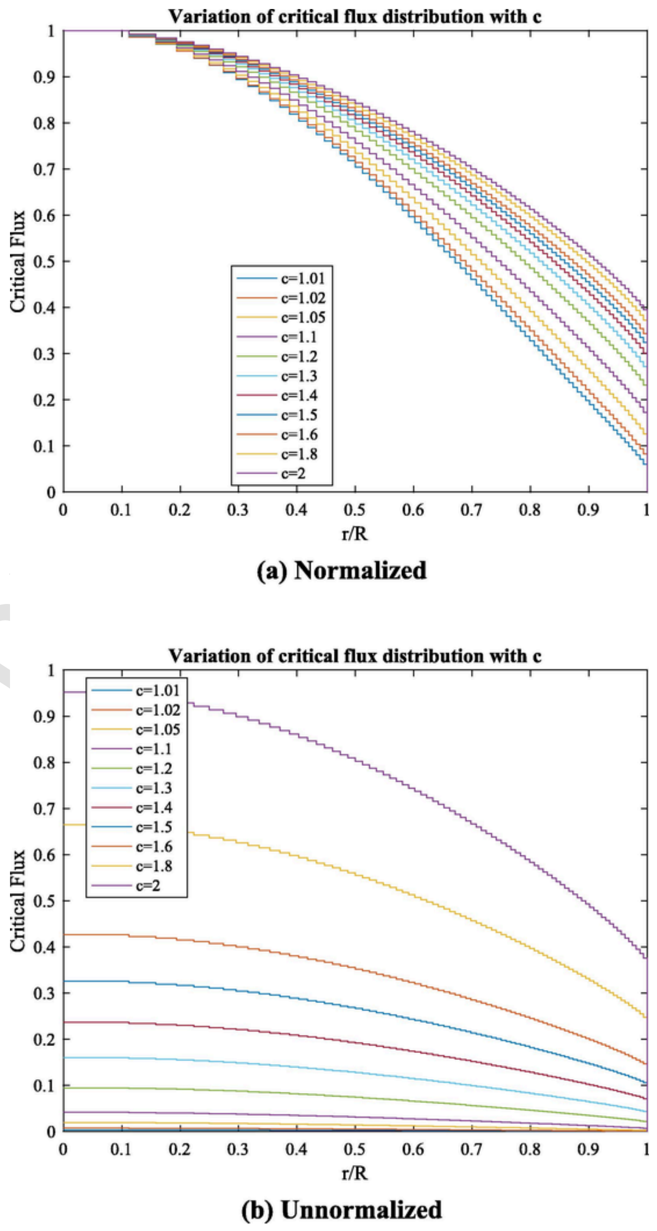


Fig. 2. Computed cell-averaged critical flux for variable c.

3.2.2.4. Mesh convergence analysis. Next, mesh convergence analysis is conducted for three discretization parameters: radial discretization, ray spacing, and polar angular discretization. Case $c = 1.01$ is selected as an example. The same set of phase space discretization parameters is used as the basis for the convergence analysis. The radial convergence curve is plotted in Fig. 4.

Fig. 4 shows good agreement with second order radial convergence (up to 160 rings) for $c = 1.01$, which is consistent with the spatial rate of convergence for flat-source MOC (Wang et al., 2017). The convergence curve for the eigenvalue k vs ray spacing and polar angular discretization are plotted in Figs. 5 and 6, respectively. The errors associated with both converge to zero. Fig. 5 indicates a linear convergence in error over a modest range of ray spacings, approaching a second-order convergence as the grid is refined, whereas Fig. 6 shows the error flat-

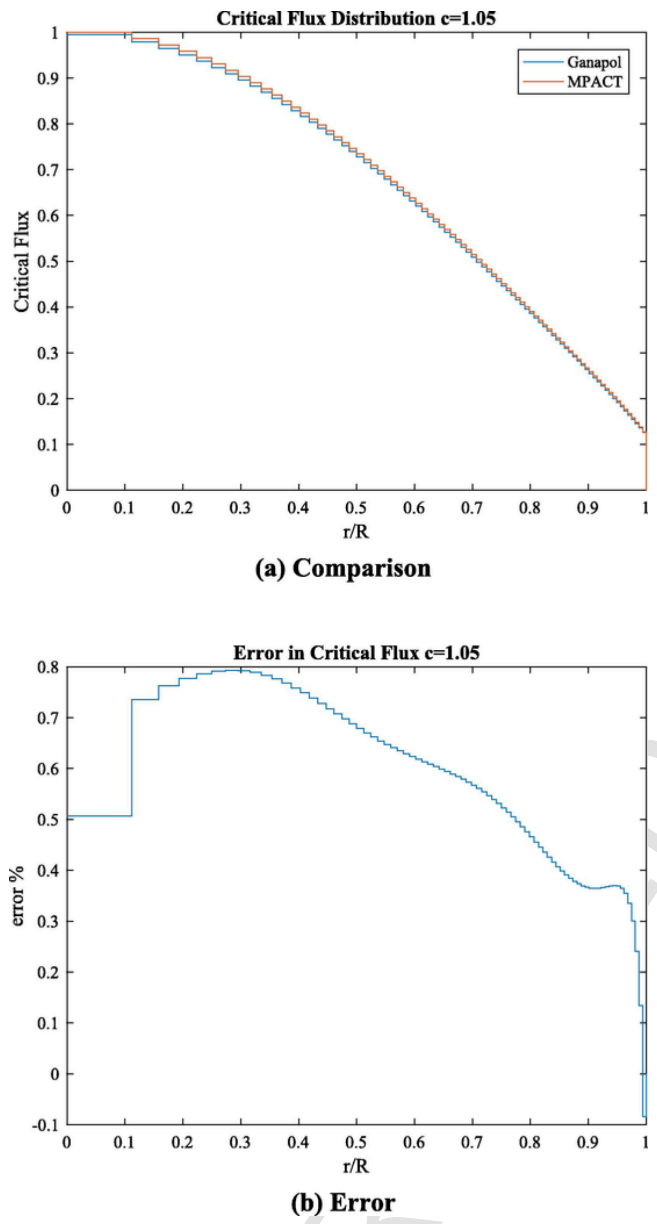


Fig. 3. Cell-averaged critical flux ($c = 1.05$).

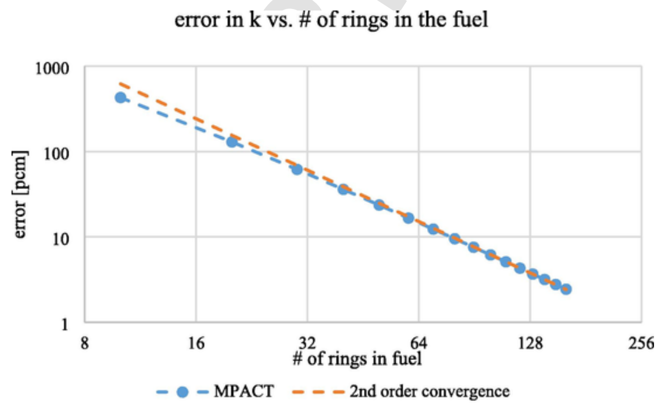


Fig. 4. Convergence with respect to radial discretization.

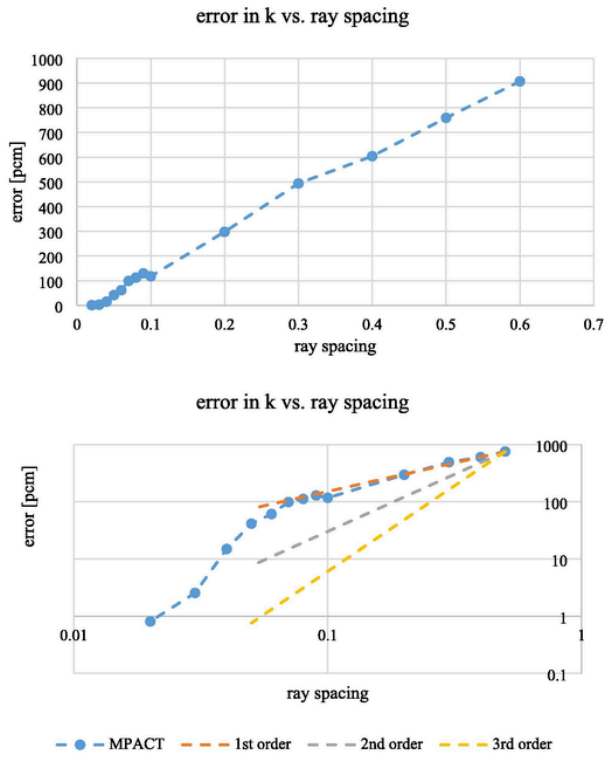


Fig. 5. Convergence with respect to ray spacing.

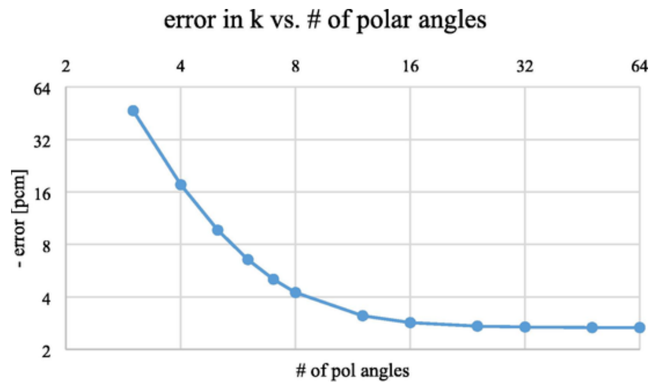


Fig. 6. Convergence with respect to the number of polar angles.

tens out at ~ 15 polar angles. The convergence rates for these grid refinement terms have not been studied theoretically.

3.2.3. *Summary*

Benchmark problem 3.4 in Ganapol (Ganapol, 2009) has been used as an example of MES for a Code Verification test for MPACT. The bare rod configuration of problem 3.4 was modified to mimic a square lattice by surrounding the rod with a bounding box with a non-scattering material. Vacuum boundary conditions were imposed on the surface of the bounding box, and several critical rod cases were analyzed with MPACT using the tabulated critical rod radii as a function of c, the mean number of secondaries per collision. MPACT k_{eff} eigenvalues agreed with the reference result ($k_{eff} = 1$) for all cases to within a few pcm. The cell-averaged flux agreed very well with the benchmark results, with a relative error of less than 1 % at any radial location. Additionally, Excellent convergence behavior has been observed using MPACT in this study. The radial rate of convergence is shown to be second order, which is consistent with the expected convergence rate for the flat-source approximation (Wang et al., 2017). The convergence curves with respect to ray spacing and polar angle quadrature set order were obtained — both of which converge to the analytic solution.

3.3. Code Verification using the Method of manufactured solutions

3.3.1. Method of manufactured solutions

MMS has been an effective Code Verification method for verifying the correctness of numerical algorithms and software implementation in a wide range of engineering applications. The essential idea of MMS is that instead of solving a problem with prescribed boundary and initial conditions, one can assume a solution beforehand and substitute it into the governing equation that the software intends to solve. The equation is then balanced by evaluating the resultant manufactured source. The boundary and initial conditions can be obtained by evaluating the manufactured solution at the boundary and at the initial time. The software is then used to solve the system with the manufactured source and boundary and initial conditions; the computed solution should equal the assumed solution to within some error associated with the discretization method used in the software.

Similar to MES, the code is considered verified if the error is smaller than a prescribed acceptance criterion (e.g., round-off error). Furthermore, one can perform grid refinements analysis using the same problem and observe the convergence rate, which can be compared against mathematical expectations. If mathematical expectations have not been established, the observed convergence behavior from MMS runs can help reveal the rate of convergence of the applied methods. MMS has more flexibility compared with MES in verifying computational functionalities of a computer code and has been used for a radiation transport code in planar geometry (Wang et al., 2018), but there have been limited applications of MMS to problems with realistic configurations (e.g., heterogeneous materials and complex geometry). The following section describes a technique developed to apply MMS to problems with realistic geometry and cross sections.

3.3.2. Example: Application of MMS using the C5G7 configuration

For demonstration purposes, the OECD/NEA C5G7 benchmark problem (Smith et al., 2003) was selected for its close resemblance to PWR mixed-oxide (MOX) fuel assemblies. The two-dimensional configuration is shown in Fig. 7(a) and 7(b). Each fuel assembly is made up of a 17 × 17 lattice of square fuel-pin cells with a pitch size of 1.26 cm. The MOX assembly contains fuel pins with three different plutonium enrichments (4.3 %, 7.0 %, and 8.7 %). The UO_2 rods are made of 3.7 % enriched U-235. For more details about the problem geometry, material, and the seven-group cross section library, refer to the C5G7 benchmark report (Smith et al., 2003).

The following section describes the application of MMS to verify a 2D MOC solver using the C5G7 problem. The test framework hereby developed involves two useful tests, the first of which is a consistency test where the converged solution of an eigenvalue problem is used to formulate a standalone fixed-source problem. The second is an MMS test where an analytical solution is assumed, and the corresponding MMS source is used to verify the fixed source solver. This could be extended to the more challenging eigenvalue problem, where one assumes both the neutron flux distribution and eigenvalue for verifying the eigenvalue solver.

3.3.2.1. Consistency test. For the consistency test, an eigenvalue problem is solved first. Once the eigensolver converges to a solution pairψ_{conv}, k_{conv}, the fission source $\frac{1}{k_{\mathrm{conv}}}F\psi_{\mathrm{conv}}$ is constructed and output to a data file, which is subsequently used in the fixed-source problem defined using the same geometric and material configurations. This fixed-source problem is modeled, and it is expected that the fixed source solver will, upon convergence, return the same solution as that from the eigensolver.

The consistency test is straightforward but important to perform. It is also a convenient way to verify the fixed source solver if MMS capabilities (e.g., general fixed source and general boundary condition capabilities) have not been established for a full-scale MMS test.

3.3.2.2. Application of MMS to a 2D multigroup fixed-source problem. For 2D, the multigroup (MG) Boltzmann transport equation with isotropic scattering and a fixed source takes the following form,

$$\eta\frac{\partial\psi_g(x,y,\eta,\xi)}{\partial x}+\xi\frac{\partial\psi_g(x,y,\eta,\xi)}{\partial y}+\Sigma_{t,g}(x,y)\cdot\psi_g(x,y,\eta,\xi)$$
$$=\frac{1}{2\pi}\sum_{g'=1}^{G}\Sigma_{s,g\leftarrow g'}(x,y)\cdot\phi_{g'}(x,y)+q_g(x,y,\eta,\xi),\quad g=1,\cdots,G \tag{1}$$

Given an MOC solver that is intended to solve the above equation with prescribed boundary conditions, the steps for applying MMS to verify the correctness of the fixed source solver are shown below.

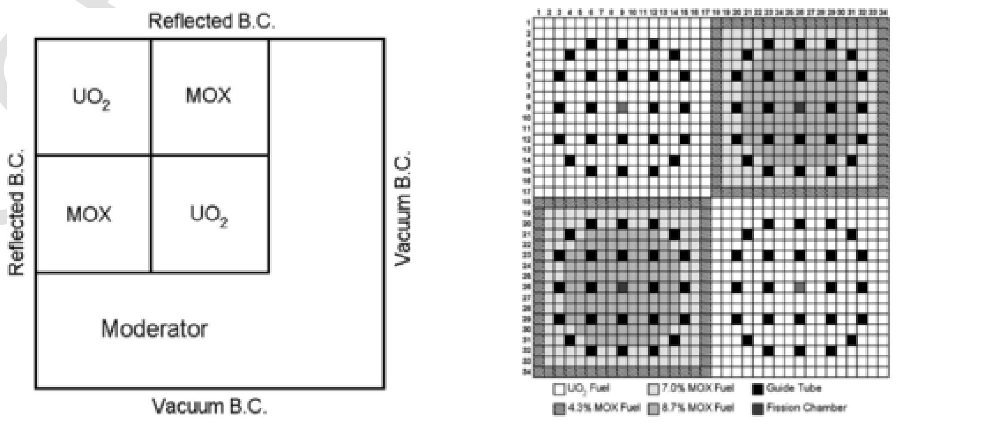
First, assume an analytical solution, e.g.,

$$\psi_g(x,y,\eta,\xi)=\psi_{\mathrm{MMS},g}(x,y,\eta,\xi),\quad g=1,\cdots,G$$

The corresponding scalar flux is.

$$\phi_g(x,y)=\int_{-1}^{1}d\eta'\int_{-\sqrt{1-\eta^2}}^{\sqrt{1-\eta^2}}\frac{\psi_{MMS,g}(x,y,\eta',\xi')}{\sqrt{1-\eta^2-\xi^2}}d\xi',\quad g=1,\cdots,G$$

Second, substitute the assumed solutions into the MG equation as defined in Equation. The equation will be balanced by an inhomogeneous MMS source, which can be expressed as in Equation (2),



(a) Core configuration **(b) Pin cell compositions and numbering scheme**

Fig. 7. Two-dimensional configuration for the C5G7 benchmark (Smith et al., 2003).

$$q_{\mathrm{MMS},g}(x,y,\eta,\xi) = \eta\frac{\partial\psi_{\mathrm{MMS},g}(x,y,\eta,\xi)}{\partial x} + \xi\frac{\partial\psi_{\mathrm{MMS},g}(x,y,\eta,\xi)}{\partial y} + \Sigma_{t,g}(x,y)\cdot\psi_{\mathrm{MMS},g}(x,y,\eta,\xi) - \frac{1}{2\pi}\sum_{g'=1}^{G}\Sigma_{s,g\leftarrow g'}(x,y)\cdot\phi_{\mathrm{MMS},g'}(x,y) \quad (2)$$

To get the discrete form of the manufactured source, the analytical form as defined in Equation is averaged over every fixed source region (FSR) indexed by region ID i and evaluated at each discrete angle (η_m, ξ_m) defined in the applied quadrature set,

$$q_{MMS,g,i,m} = \frac{1}{V_i}\int_{V_i} q_{MMS,g}(x,y,\eta_m,\xi_m)\,dxdy$$

For the first two terms (i.e., the leakage terms) of the manufactured source in Equation, the analytical form can be obtained using external software or a script, or one can calculate the derivative by hand. However, it would be a substantial challenge to average the analytical leakage terms over every FSR region, unless a simple solution was assumed, in particular, one with zero derivatives. The discrete form of the third term, the collision term, can be obtained by first having the code output the total cross section for all FSR regions as a vector and then performing an element-wise product of the cross-section vector with the discrete MMS solution vector. The scattering source term is even more complicated. However, one does not need to expend the effort to compute the scattering term because the MOC code already accumulates this term in its iterative process to solve the eigenvalue problem. In particular, the existing code infrastructure can be used to calculate the scattering source. To use this existing infrastructure, the discrete MMS solution should be used to initialize the first iteration of the fixed source solver. This is a “one-off” iteration because it will have to halt after one iteration, and once the in-scattering source and self-scattering source are accumulated, they are output to a data file, which can be processed to generate a discrete MMS source.

The third step is to obtain the boundary conditions for the manufactured problem. This is achieved by a straightforward evaluation of the assumed solution at the problem boundaries, which is carried out for all energy groups,

$$\psi_g(x,y,\eta,\xi)\Big|_{\mathrm{Boundary}} = \psi_{\mathrm{MMS},g}(x,y,\eta,\xi)\Big|_{(x,y)\in\partial V}, \quad \left(\eta\vec{i}+\xi\vec{j}\right)\cdot\vec{n}<0$$

where ∂V is the boundary of the system V, $\vec{n}$ is the surface normal, and angle (η,ξ) represents the direction of flight of neutrons.

In MOC, the boundary conditions are defined using the entering fluxes associated with the sets of characteristic rays that sweep through the geometry. Each set of those rays is characterized by an angle (η_m, ξ_m) defined in the applied quadrature set. If we use variable l_m to index each ray characterized by an angle(η_m, ξ_m), the discrete boundary condition can be expressed as.

$$\psi_{g,m,l_m} = \frac{1}{2}\psi_{\mathrm{MMS},g}\left(x_{l_m}, y_{l_m}, \eta_m, \xi_m\right)$$

where $\left(x_{l_m}, y_{l_m}\right)$ are the coordinates of the entry point for the characteristic ray l_m.

The discrete MMS source and boundary conditions, together with the geometry and materials configurations, are modeled using the fixed source solver under verification. If the solver converges to the assumed MMS solution, the fixed source solver is verified.

3.3.2.3. *Demonstration and numerical results.*

A. Numerical result for the consistency test.

As stated in the previous subsection, a consistency test starts with an eigenvalue calculation of the C5G7 problem. An MOC-based surrogate code MOCC (Young, 2016) is used for this demonstration. The converged eigenvalue was 1.18642. The converged scalar fluxes for the first energy group (fast) and the last energy group (thermal) are shown in Fig. 8 (a) and 8 (b).

The fission source distribution is shown in Fig. 9 in a pin-homogenized form for visualization, showing the strong spatial variation. This fission source is used for the subsequent fixed-source calculation.

The relative error of the scalar flux of the first energy group is plotted in Fig. 10, where the error has an order of magnitude of 1E-7. The relative error of the scalar flux of the thermal energy group can be close to 1E-5. The close-to-zero error indicates that the scalar flux from the fixed-source problem converges to the same solution as that from the eigenvalue calculation. Thus, the consistency of the inner workings between the eigensolver and the fixed source solver is verified.

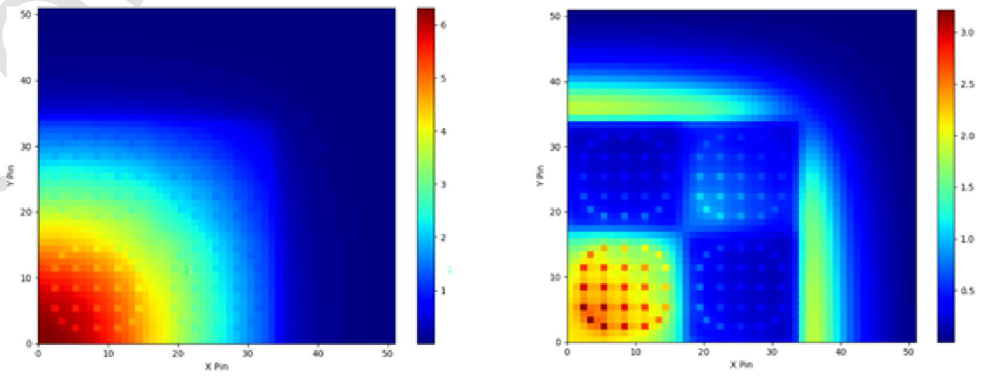
B. Calculation of the MMS source and boundary conditions for the C5G7 problem.

A constant manufactured solution is selected to demonstrate the feasibility of applying MMS to a fixed-source solver for realistic applications. As will be seen later, a constant solution is not as “simple” as it appears, because it requires a drastic spatially-varying manufactured source to yield a constant solution.

Assume a constant angular flux for each group,

$$\psi_g(x,y,\eta,\xi) = \psi_{\mathrm{MMS},g}(x,y,\eta,\xi) = \frac{1}{2\pi}, \quad g = 1,\cdots,7$$

The scalar flux can be obtained$\phi_{\mathrm{MMS},g}(x,y) = 1$.
With this, the MMS source can be found,



(a) The scalar flux of group 1 (b) The scalar flux of group 7

Fig. 8. Scalar flux of C5G7 eigenvalue calculation.

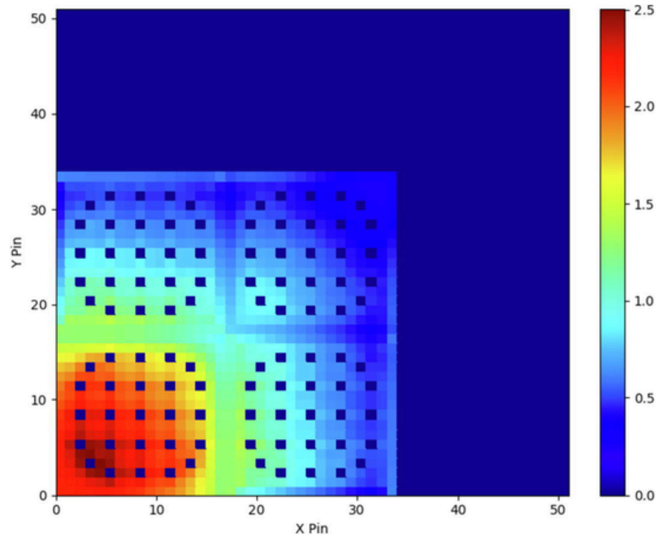


Fig. 9. Fission source distribution from eigenvalue calculation.

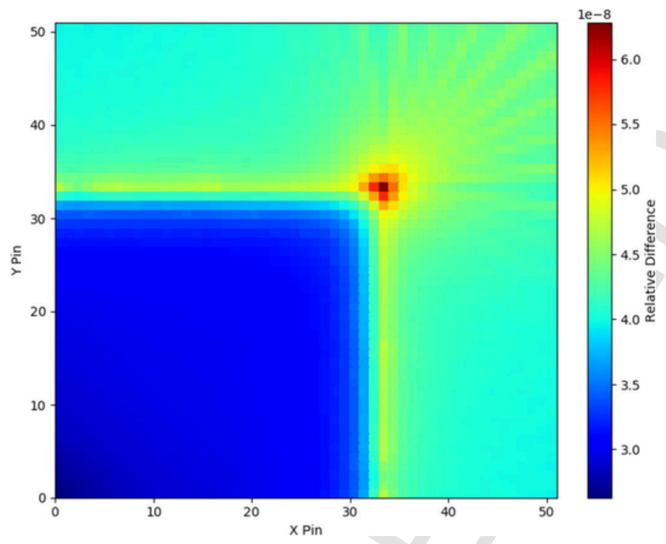


Fig. 10. Relative error of the converged scalar flux (group 1).

$$\begin{aligned}
q_{\mathrm{MMS},g}(x,y,\eta,\xi) &= \eta\frac{\partial\psi_{\mathrm{MMS},g}(x,y,\eta,\xi)}{\partial x}+\xi\frac{\partial\psi_{\mathrm{MMS},g}(x,y,\eta,\xi)}{\partial y} \\
+\Sigma_{t,g}(x,y)\cdot\psi_{\mathrm{MMS},g}(x,y,\eta,\xi) &- \tfrac{1}{2\pi}\textstyle\sum_{g'=1}^{G}\Sigma_{s,g\leftarrow g'}(x,y)\cdot\phi_{\mathrm{MMS},g'}(x,y) \\
&= \Sigma_{t,g}(x,y)\cdot\tfrac{1}{2\pi}-\tfrac{1}{2\pi}\textstyle\sum_{g'=1}^{G}\Sigma_{s,g\leftarrow g'}(x,y)\cdot 1 \\
&= \tfrac{1}{2\pi}\cdot\left(\Sigma_{t,g}(x,y)-\textstyle\sum_{g'=1}^{G}\Sigma_{s,g\leftarrow g'}(x,y)\right)
\end{aligned} \tag{3}$$

Because the MMS source is independent of angle, it can be treated as an isotropic external source when the manufactured problem is modeled.

$$Q_{\mathrm{MMS},g}(x,y)=\Sigma_{t,g}(x,y)-\sum_{g'=1}^{G}\Sigma_{s,g\leftarrow g'}(x,y) \tag{4}$$

The MMS source can be discretized in the following form,

$$Q_{\mathrm{MMS},g,i}=\Sigma_{t,g,i}-\sum_{g'=1}^{G}\Sigma_{s,g\leftarrow g',i}$$

where i is the FSR region index.

Note that the discontinuity in the MMS source from one FSR region to its neighbor is caused by the discontinuity in materials and is necessary for the solver to converge to a flat solution. The discrete boundary condition takes the following form,

$$\psi_{g,m,l_m}=\psi_{g,l_m}(\eta_m,\xi_m,\mu_m)=\frac{1}{4\pi},\quad g=1,\cdots,7$$

An easy way to implement this boundary condition is to modify the vacuum boundary condition code by replacing the zero incoming angular flux with constant $1/4\pi$.

The discrete MMS source and boundary conditions constitute a well-defined MMS problem. With the geometry and materials information from the C5G7 benchmark, the MMS problem can be modeled using the code under verification. The constant-in-space solution does not trivialize this MMS test case because the complexity in geometry and materials is retained and all routines involved are exercised. On the contrary, it is quite a challenge for the code to converge to a constant solution. This complicated manufactured source drives the solver to converge to a flat solution. This test helps to verify the correctness of the fixed source solver.

C. Numerical results for the MMS case.

The fuel pins are spatially discretized with five rings in the fuel region, three rings in the moderator region, and eight azimuthal slices for all rings. Non-fuel pins in the moderator assemblies are meshed with 3 × 3 square sub-regions.

A “dummy” input with the above spatial discretization is used for generating the scattering source and the collision term, which are output to a data file. The data file is processed with an external script for the calculation of the MMS source. The MMS source for the first energy group is shown in Fig. 11 as an example.

As shown in Fig. 11, with a flat MMS solution, the spatial variation of the MMS source coincides with material boundaries, which is expected. When the assumed solution is non-flat in space, the spatial variation of the MMS source will vary for each FSR. Note the extreme spatial dependence of the MMS source, because about half of the source regions are negative.

In modeling the manufactured problem, Chebyshev-Gauss quadrature is used with eight azimuthal angles and two polar angles in each quadrant. Ray spacing is taken as 0.05 cm. A maximum of 100 source iterations is performed, generating an RMS error between the scalar fluxes of two consecutive iterations being 2.37×10^{-8}. This error corresponding to the 100 source iterations is small enough to prove the capability of the code to reproduce the manufactured solution.

$$e_{100}=\sqrt{\frac{\sum_{i=1}^{\mathrm{nreg}}\sum_{g=1}^{G}\left(\phi_{i,g,\mathrm{new}}-\phi_{i,g,\mathrm{old}}\right)^2}{\mathrm{nreg}\cdot G}}\leqslant 2.37\times10^{-8}$$

Fig. 12 shows the error in pin flux for energy groups 1 and 7. The errors are close to zero, which is also true for energy groups 2 through 6. This indicates that the numerical solution converges to the MMS solutions, thus passing the MMS test and verifying the fixed source solver.

Note that the drastic spatial variation in the MMS source, including negative values in many regions, as seen in Fig. 11, does not affect the convergence of the scalar flux to the MMS solution. It is this strong spatial variation that drives the fixed source solver to converge to a solution that is flat in space.

4. Summary

This subsection described the application of MMS to the C5G7 benchmark problem which is a realistic reactor configuration with typical heterogeneity and complex geometry encountered in actual PWR nuclear fuel assemblies. The overall test framework included a consistency test and the application of MMS to the C5G7 problem with a fixed source. The consistency test is straightforward but important because it verifies that the fixed source solver is consistent with the eigenvalue solver. The successful application of MMS to the C5G7 problem with an

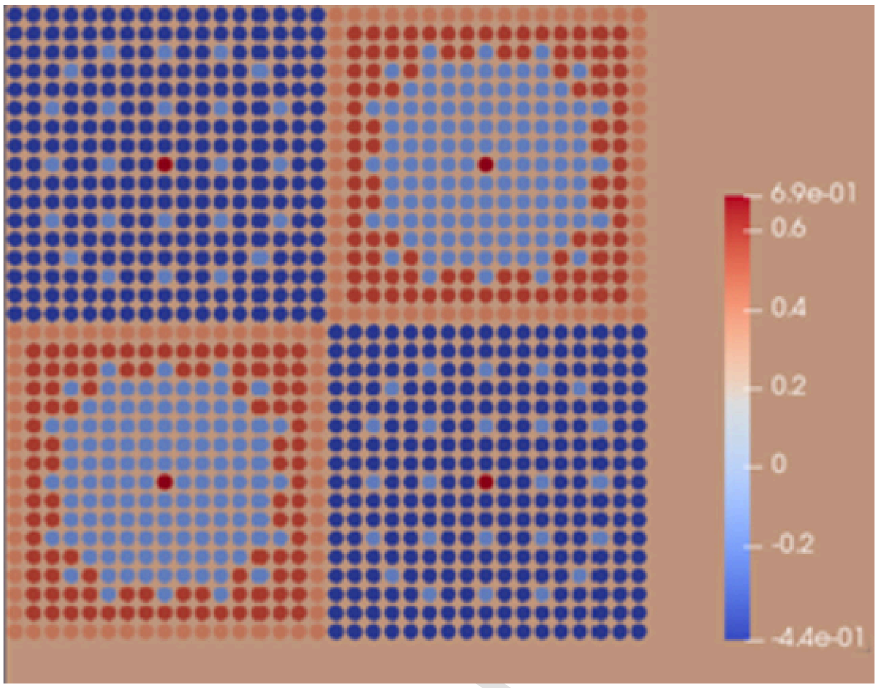


Fig. 11. Manufactured source due to a flat manufactured solution.

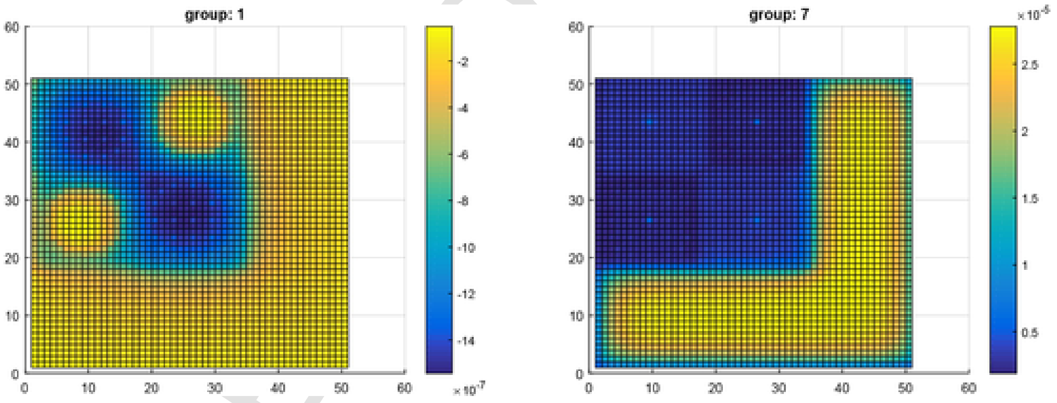


Fig. 12. Error in pin flux for energy groups 1 (left) and 7 (right).

assumed flat solution demonstrates that the fixed source solver for this realistic reactor configuration can be verified using MMS. The established test framework for a realistic application considerably extends the range of applicability of MMS.

5. Solution Verification

Solution Verification is used to assess the applicability of the code and the accuracy of the solution to problems of interest. This is normally supported by estimating the numerical errors in the simulation and revealing the convergence behavior of the computed solution.

5.1. Solution Verification with MPACT

To assess the convergence rates within MPACT, Solution Verification was performed with two separate analyses. The first was an exercise of a 3D assembly, where a uniform grid refinement was performed by simultaneously varying the six grid refinement terms given in Section 2. The second exercise was a convergence analysis with 2D pin geometry to better understand the convergence behavior of grid refinement terms for a problem of interest but with a manageable size. For these analyses, the quantity of interest is the multiplication constant (k_{eff}). Other metrics (e.g., scalar flux vector or fission sources) can be used if desired.

5.2. 3D assembly problem

5.2.1. Specifications for the 3D assembly

The problem consists of a single Westinghouse 17 × 17-type fuel assembly at beginning-of-life (BOL) and hot zero power (HZP) isothermal conditions. Each fuel assembly is composed of fuel rods, guide and instrument tubes, spacer grids, and top and bottom nozzles. Fig. 13 shows a radial model with quarter symmetry of a 2D array of fuel rods (a fuel lattice) typical of the central axial region of PWR fuel assemblies, and Fig. 14(a) shows the axial assembly geometry. The axial grid spacers were removed from the problem to ensure a uniform mesh. The core plates and axial end plug regions were also removed. Additionally, the thicknesses of each axial region were intentionally set to a multiple of 8 cm, leading to an easy mesh refinement. The geometry of the axial model is shown in Fig. 14(b).

5.2.2. Numerical results

A 3D uniform refinement assessment is performed by simultaneously varying the six grid refinement terms. Note that the default for

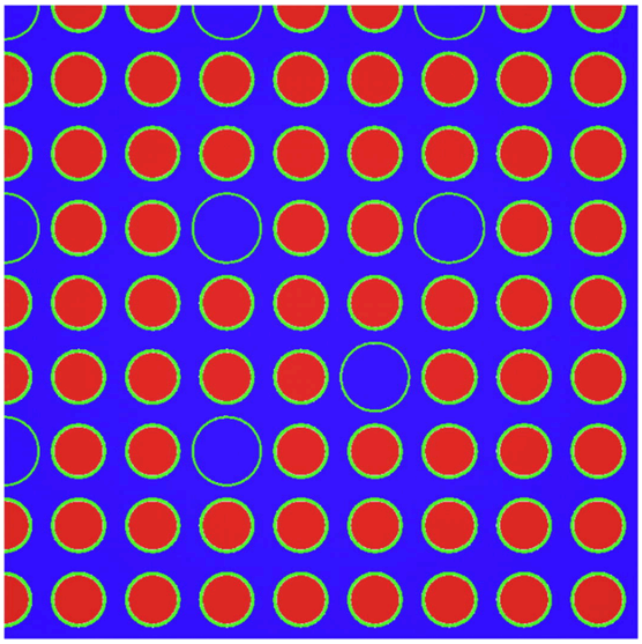


Fig. 13. Problem 2 KENO-VI geometry.

the polar angle uses the Yamamoto quadrature (Yamamoto et al., 2007), which is considered fully refined for N = 3 and is the maximum allowed in MPACT; consequently, the polar angle was not refined as part of this study. Five separate refinements are examined with the medium grid representing the defaults used by the code. These refinement cases can be seen in Table 3, with the total variation in refinement parameters spanning five orders of magnitude. The self-shielding calculation is performed for all of the grid refinement calculations (i.e., SS = on).

The results of the uniform grid refinement are shown in Fig. 15. The extrapolated converged solution, k-extrap, is taken to be 1.17694 by applying Richardson Extrapolation to the three finest grids. The error on any specified grid refers to k_{eff} on that grid compared with k-extrap. With self-shielding activated, the code is monotonically convergent on the four finest grids. After applying Richardson Extrapolation, the order of convergence was calculated to be 2.75.

If the “exact” solution is estimated from either super-refinement of the grid or from Richardson Extrapolation, the solution error on any other grid can be obtained by referencing the solution on the new grid to the exact solution. The solution error on the medium refinement grid above is 20 pcm, which is less than the accepted threshold of 50 pcm. This is significant because the medium grid corresponds closely to code defaults. This indicates that the default axial mesh has an acceptable amount of error. On the assembly scale, monotonic convergence can be seen with error decreasing from the medium grid case through the xxfine grid case. This is a significant result because the MPACT software is intended to be used for assembly-size problems or larger.

5.3. 2D pin-cell problem

The second Solution Verification exercise performed is a convergence analysis with 2D pin geometry (i.e., no grid refinements in the ax-

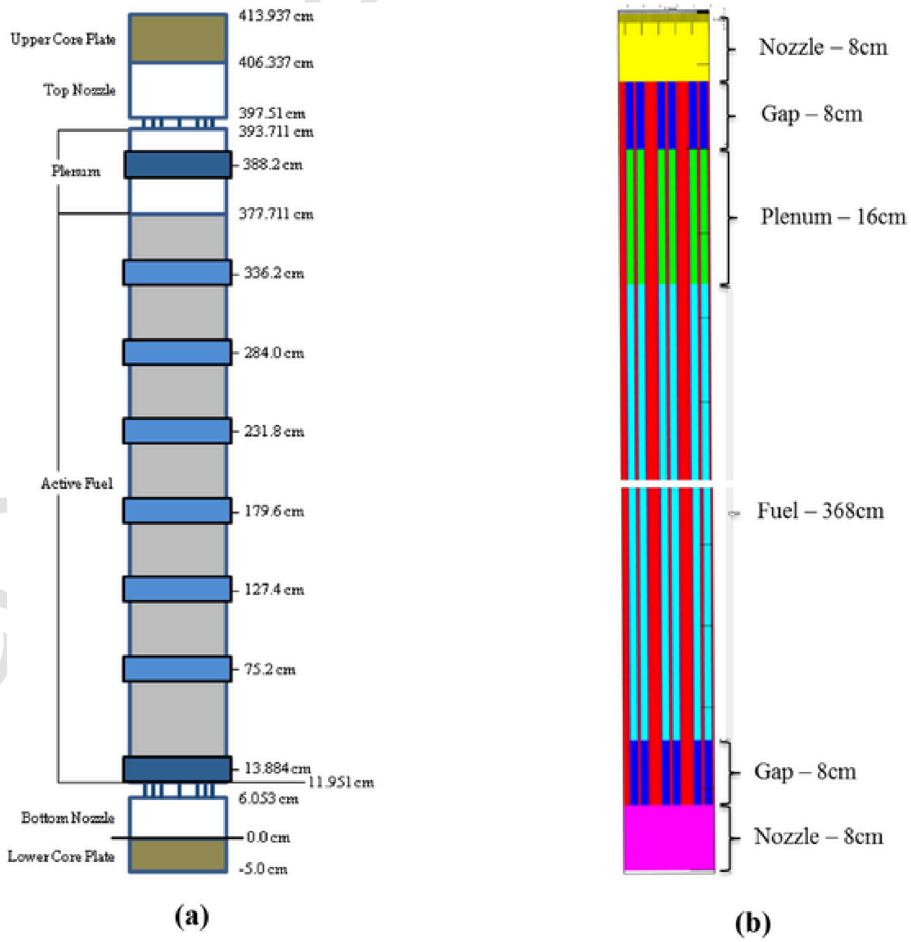


Fig. 14. Problem 3 axial geometry (without end plugs).

Table 3
Grids for uniform refinement in 3D assembly geometry.

	Number of Elements				
	coarse	medium	fine	xfine	xxfine
Reflector	4	16	64	256	
Reflector	2 × 2	4 × 4	8 × 8	16 × 16	
Axial Division (Fuel)	25	50	100	200	400
Axial Division (cm)	16	8	4	2	1
Radial Division (# rings)	3	6	12	24	48
Fuel	2	4	8	16	32
Gap	1	1	1	1	1
Clad	1	1	1	1	1
Moderator	1	2	4	8	16
Azimuthal Division	4	8	16	32	64
Rays	6	12	24	48	96
Ray Spacing (cm)	0.1	0.05	0.025	0.0125	0.00625
Polar Angle	3	3	3	3	3
Azimuthal Angle	8	16	32	64	128
Ntotal*	5.76E + 02	9.22E + 03	1.47E + 05	2.36E + 06	3.77E + 07

* Only the radial and azimuthal division, rays, and azimuthal angle are accounted for, because the rest are either fully converged or present no sensitivity to the results.

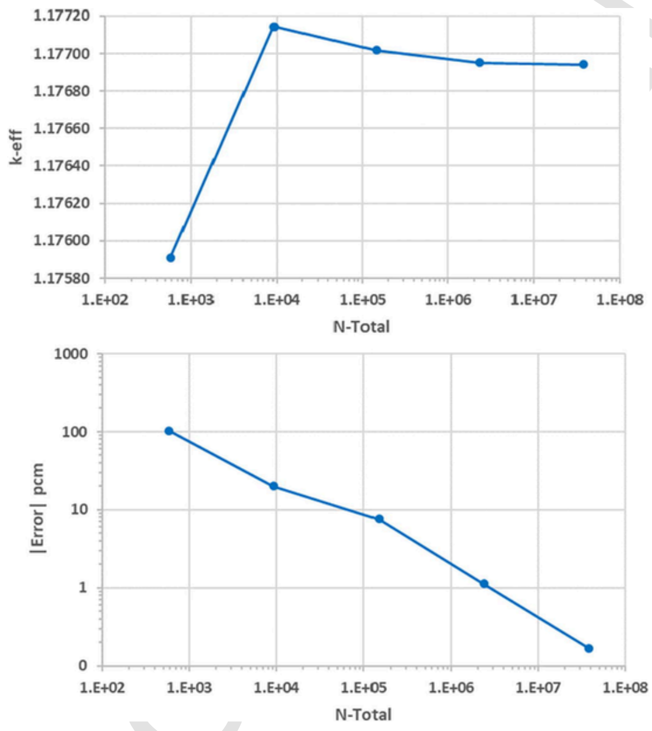


Fig. 15. The k_{eff} and error for uniform grid refinement in 3D assembly geometry.

ial direction). The purpose of this study is to better understand the convergence behavior of the grid refinement terms.

5.3.1. *Specifications for the 2D pin*

The 2D pin cell problem uses the first Virtual Environment for Reactor Applications (VERA) core physics benchmark problem that was intended to demonstrate VERA's capability to solve a simple 2D pin cell eigenvalue problem typical of PWR analyses, as shown in Fig. 16. The problem consists of a single Westinghouse 17 × 17-type fuel rod cell at BOL conditions. The materials are standard for this type of reactor: UO_2, zircaloy-4, and water. The moderator contains soluble boron as a chemical shim for maintaining criticality. The pellet-clad gap consists of helium gas, which may be neglected because of its insignificant neutron cross section. This problem represents typical zero power isothermal conditions, which are representative of power reactor startup physics testing.

5.3.2. *Numerical results*

The calculations include a re-coarsening analysis of five parameters of interest: radial division, azimuthal division, ray spacing, polar angle, and azimuthal angle. Additionally, the calculations include a uniform mesh refinement study akin to the 3D assembly study presented in Section 4.2. Besides calculations with self-shielding turned on, calculations with self-shielding turned off were also added to avoid the slightly updated cross section for each grid. Table 4 shows the number of elements used for each of the five parameters of interest. The total number of ele-

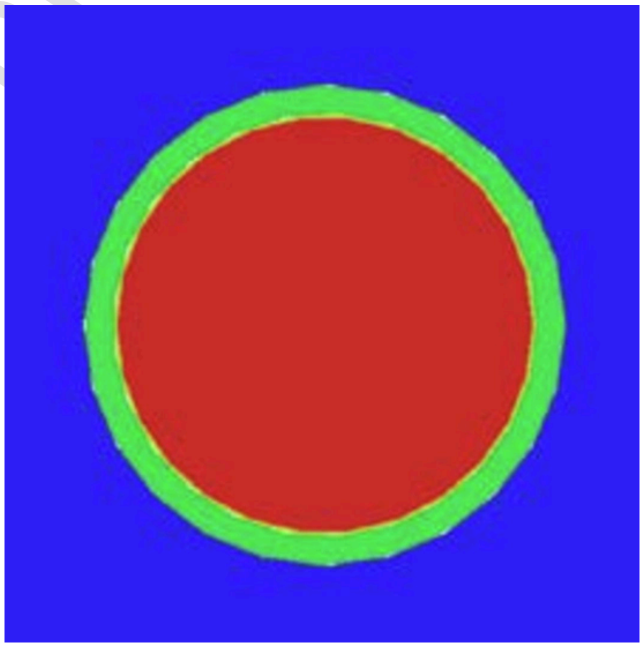


Fig. 16. Problem 1 KENO-VI geometry.

Table 4
Grids for uniform refinement in 2D pin geometry.

	Number of Elements				
	xcoarse	coarse	medium	fine	xfine
Radial Division (rings)	8	16	32	64	128
Fuel	4	8	16	32	64
Gap	1	1	1	1	1
Clad	1	1	1	1	1
Moderator	4	8	16	32	64
Azimuthal Division	4	8	16	32	64
Rays	63	125	250	500	1000
Ray Spacing (cm)	0.008	0.004	0.002	0.001	0.0005
Polar Angle	2	4	8	16	32
Azimuthal Angle	2	4	8	16	32
Ntotal	8.00E + 03	2.56E + 05	8.19E + 06	2.62E + 08	8.39E + 09

ments spans six orders of magnitude from the xcoarse case to the xfine case.

The results of the uniform mesh refinement study are shown in Fig. 17, with self-shielding both active (SS = On) and turned off (SS = Off). With self-shielding turned on, Richardson Extrapolation cannot be applied to the finest three grids because the solution is not monotonically convergent. This is a major difference from the 3D assembly calculation, where monotonic convergence is observed. It is believed to be due to a suppressed solution sensitivity to a localized self-shielding cross section update for a large system. On the other hand, when self-shielding is turned off, monotonic convergence is observed for the 2D pin problem.

The error convergence curves are also provided in Fig. 17. Due to the non-monotonicity, with SS = On, the finest grid result is used as the "exact" solution, so only four error data points are obtained. With SS = Off, the Richardson extrapolated result is used as the reference solution, so five data points are included. In this case, Richardson Extrapolation shows an order of convergence of 2.29. This can be compared with the 3D assembly order of converge of 2.75, although self-shielding was turned on for that case.

The impact of the self-shielding correction is again clearly indicated in Fig. 18, which shows the re-coarsening analysis of the radial division term in the fuel and moderator. When self-shielding is turned on, monotonic convergence is not observed. When self-shielding is turned off, MPACT is monotonically convergent, with an order of convergence of 1.51. This indicates that the number of radial rings has a significant effect on the multiplication constant. Additional sensitivity studies clearly demonstrated that the non-monotonicity inserted by self-shielding is confined entirely to the fuel with no sensitivity due to the moderator grid.

To understand the impact of self-shielding calculation on the convergence behavior, it is noted that generally, a grid refinement study solves the *same* problem, defined by geometry, material, sources, and

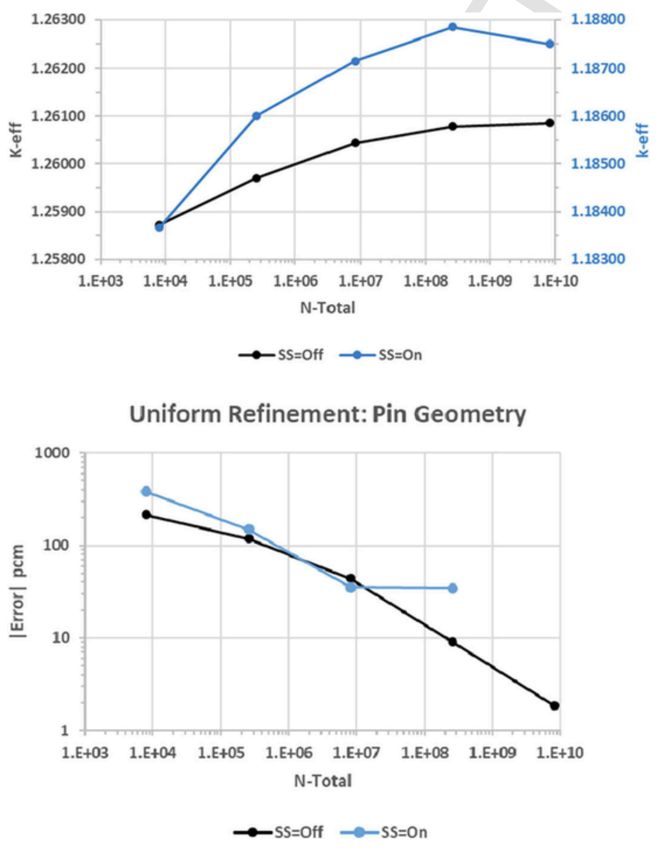


Fig. 17. The k_{eff} and error for uniform grid refinement 2D pin geometry.

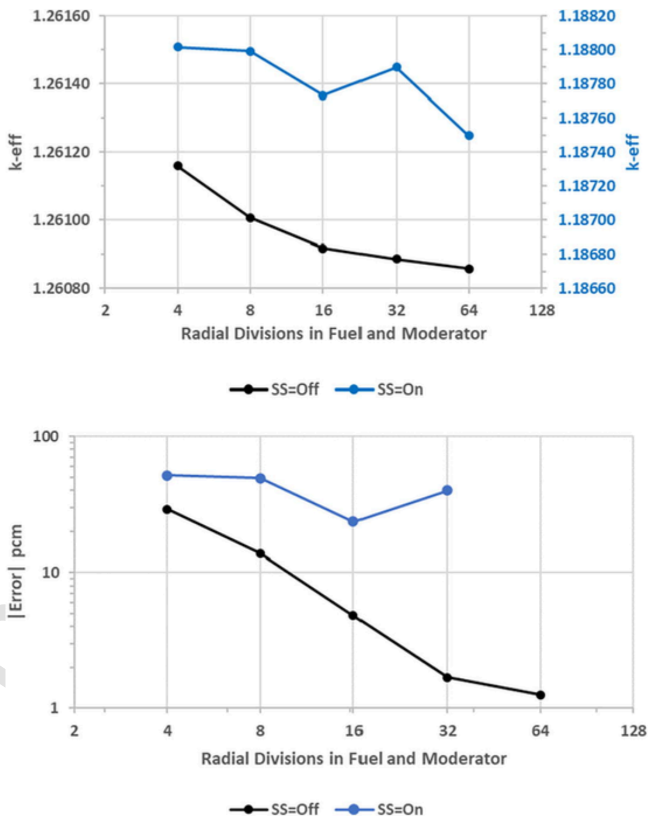


Fig. 18. The k_{eff} and error for isolation of radial division term, 2D pin geometry.

boundary condition, but *only* with different grid resolution. However, for each grid refinement, the self-shielding calculation produces an updated set of multigroup cross sections to account for the self-shielding effects. The inclusion of self-shielding calculation results in a slightly different problem to be solved for each grid. To reveal the rate of convergence concerning each independent discretization parameter via a consistent grid refinement, self-shielding needs to be turned off. The results showed that turning self-shielding off provides more monotonic convergence curves. Self-shielding has attributes analogous to mesh-dependent sub-grid models that sometimes appear in other disciplines (e.g., turbulence modeling). However, there has been a move away from such models in other disciplines because the results are typically calibrated to a specific grid, and the "model" is neither convergent nor predictive when you move away from the calibrated grid.

For reasons stated above, only results with self-shielding turned off are presented in the following. The convergence observed during the re-coarsening of the azimuthal division term is shown in Fig. 19, exhibiting an order of convergence of 1.26 with self-shielding turned off. For all cases, the total range in k_{eff} from the xfine to the coarse grid is less than 10 pcm, indicating that this term is not a major driver of error in results.

The re-coarsening study performed on the ray spacing term is shown in Fig. 20. The error observed in all cases compared with the most refined cases was below 10 pcm. Unfortunately, the default used by MPACT is 0.05 cm, which is not in the convergent region, but it results in a k_{eff} that is very close to the converged value and is computationally more affordable, especially for large problems, where k_{eff} is not as sensitive to ray spacing.

The azimuthal angle term shows the largest range of k_{eff} values for any of the parameters isolated in this analysis. For eight angles, the error is approximately 150 pcm, which exceeds the threshold for concern. It is not until 32 or more angles are used that the error associated with

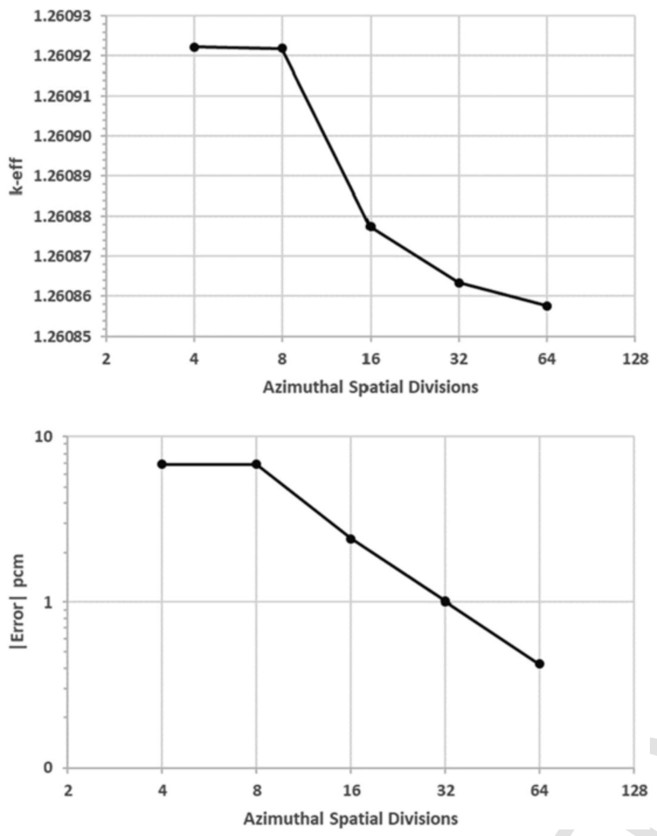


Fig. 19. The k_{eff} and error for isolation of the azimuthal spatial division term, 2D pin geometry.

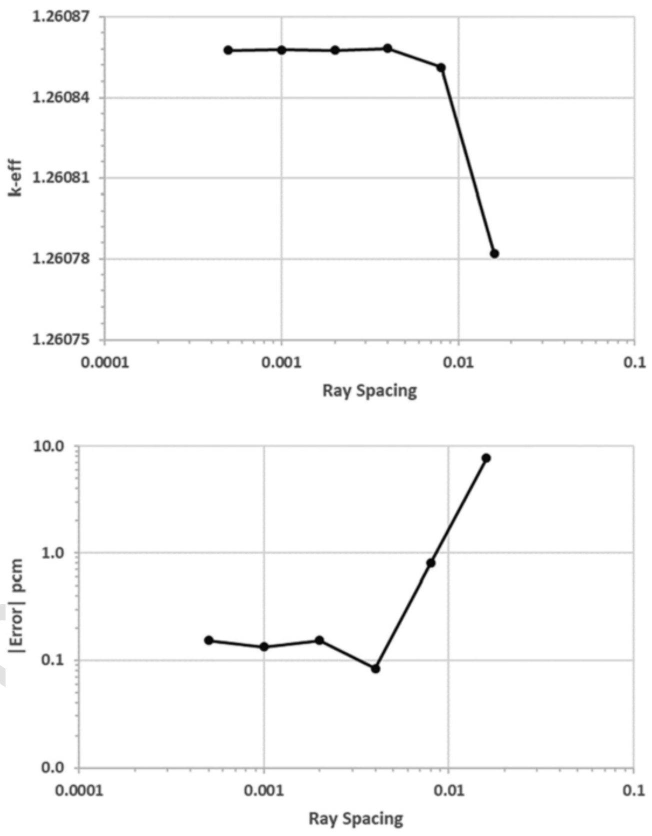


Fig. 20. The k_{eff} and error for isolation of ray spacing term, 2D pin geometry.

this term decreases to near 10 pcm. Monotonic convergence was observed for this term, and Richardson Extrapolation calculates the order of convergence to be 1.94 with self-shielding turned off. Of all the grid refinement terms isolated, the azimuthal angle term is the most dominant, having the largest impact on the overall convergence of the problem. The results of the MPACT cases for isolating the azimuthal angle are in Figs. 21 and 22. Fig. 21 shows the isolation when only an even number of angles are used, and Fig. 22 shows both odd and even numbers of angles. A clear difference in the results is observed: when both odd and even numbers of angles are used, saw-toothed behavior is observed. It is believed that the local oscillation has to do with the angle modularization and the change in ways that characteristic rays interact with the FSR mesh when the number of azimuthal angles is changed. This needs further study.

The final term isolated was the polar angle; results for this with self-shielding turned off are in Fig. 23. There seems to be a slight non-monotonic behavior around four polar angles. This could still be outside the convergence region. The rate of convergence is assessed with the last three data points using Richardson Extrapolation.

A summary of the results obtained from these calculations is provided in Table 5. Convergence rates can be quantified only when monotonic convergence is observed, which limits this discussion to the case with self-shielding turned off. The order of convergence (p) is the highest for ray spacing and the lowest for the number of azimuthal divisions. For radial convergence, 1.51 is reasonably close to the mathematical expectation and Ganapol Code Verification result of second order (Wang et al., 2017) for flat-source MOC. For other parameters, lacking guidance from theory or expectations established through Code Verification, the convergence rates are reasonable, suggesting that there is no large code implementation or algorithm concern.

The columns labeled “Error” are intended to estimate the error contribution of each term to the total error on or near the default grid. This

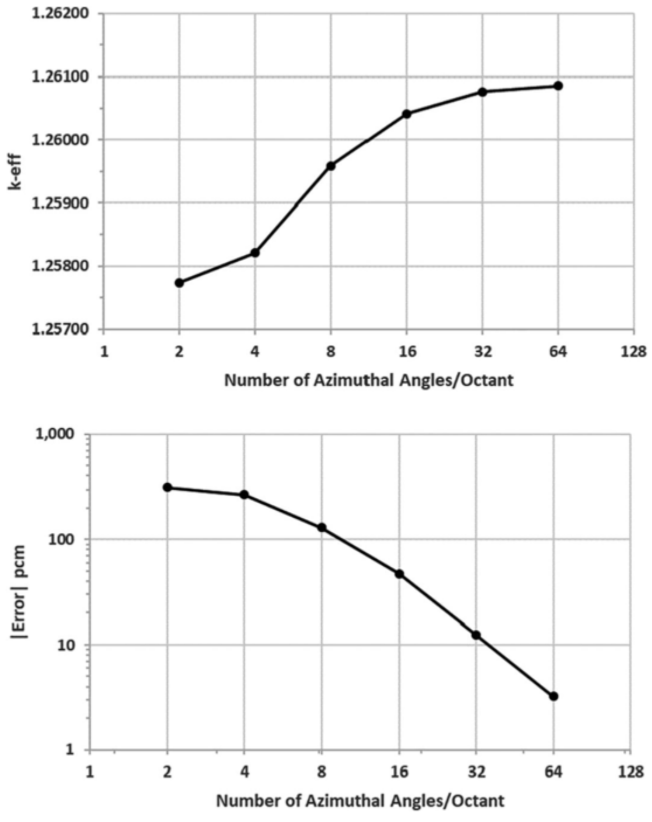


Fig. 21. The k_{eff} and error for isolation of azimuthal angle term with an even numbers of angles, 2D pin geometry.

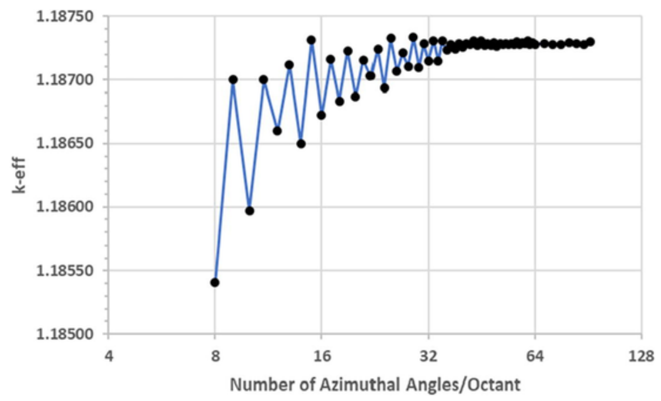


Fig. 22. The k_{eff} for isolation of azimuthal angle term with even and odd numbers of angles, 2D pin geometry.

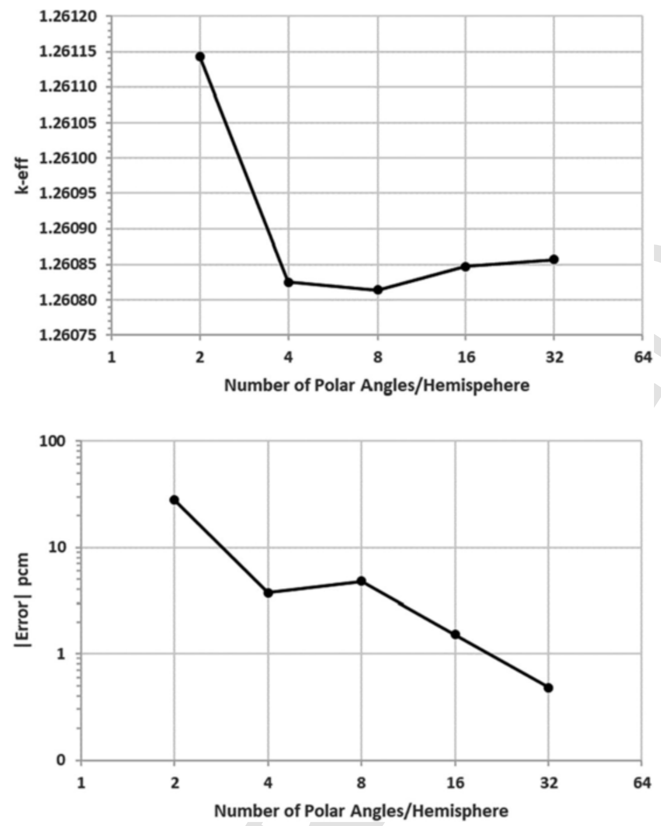


Fig. 23. The k_{eff} and error for isolation of polar angle term, 2D pin geometry.

Table 5
Summary of Solution Verification in 2D pin geometry.

	k-extrap	p	Error pcm
Resolved	1.26086		
Uniform Refinement	1.26088	2.29	
Radial Divisions	1.26087	1.51	30
Azimuthal Divisions	1.26085	1.26	6
Ray Spacing	1.26086	3.26	−8
Azimuthal Angle	1.26089	1.94	−128
Polar Angle	1.26086	1.66	−4

term is defined as k_{eff} on the near default grid minus k_{eff} on a fully refined grid. There are two differences from the default grid. First, the ray spacing for a default grid is 0.05 cm, whereas the coarsest grid in this study that did not crash the code was 0.016 cm. Second, Yamamoto quadrature with N = 2 is the default for polar angles, whereas traditional quadrature was used here.

Note that values for the error contributions can be either positive or negative depending on whether the term is converging from above or below, respectively; this is a consequence of compensating terms in the Richardson expansion. These compensating terms can be exploited to minimize the total error; however, a grid optimized in this manner might not be optimal for significantly different models.

The azimuthal angle is the dominant term. Of the five grid refinement terms isolated, it is the only one with an error range larger than 100 pcm. This suggests that a N = 8 azimuthal angles/octant should be increased to 16 angles/octant, at least for pin level calculations. This requirement can be relaxed for larger problems (e.g., assembly problems). Using 16 angles/octant would reduce the error associated with the azimuthal term to about 47 pcm and the total error to about 15 pcm, primarily because of the compensating effect of the radial division term.

6. Conclusions

A verification framework that involves both Code Verification and Solution Verification is established, in which two aspects work together such that the overarching goal of "converge to the correct answer for the intended application" can be reasonably inferred. The application of such a verification framework is demonstrated using the pin-resolved neutron transport code MPACT. Standard unit tests and regression tests prove effective tools in finding problems in the development cycle. The Ganapol 3.4 benchmark problem is a stringent and excellent analytical problem for verifying a transport solver. MPACT computed eigenvalues k_{eff} agree with references for all cases to within a few pcm. The cell-averaged flux agrees with the Ganapol benchmark results with a relative error of less than 1 % at any radial location. The applicability of MMS is extended to C5G7 problems with practical material and geometric configurations, including two useful tests, the first of which is a consistency test where the converged solution of an eigenvalue problem is used to formulate a fixed-source problem. The second is an MMS test where an analytical solution is assumed, and the corresponding MMS source is used to verify the fixed source solver. Solution Verification activities are demonstrated on a practical hierarchy of application models of increasing complexity ranging from 2D pin cell problems to 3D assembly problems. The convergence behavior and rate of convergence with respect to each individual variable are studied and provided. Under this framework, future work including building a theoretical set of benchmarks can be performed. This framework can be adapted broadly to other fields involving scientific computing codes.

CRediT authorship contribution statement

Jipu Wang : Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **William R. Martin :** Conceptualization, Methodology, Software, Validation, Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Thomas J. Downar :** Software, Validation, Resources, Supervision, Project administration, Funding acquisition. **Brendan Kochunas :** Software, Validation, Investigation, Resources, Writing – review & editing, Supervision. **Nathan C. Andrews :** Validation, Investigation. **Lindsay Gilkey :** Validation, Investigation, Visualization. **Erik D. Walker :** Software, Validation, Investigation, Writing – review & editing, Visualization. **Benjamin S. Collins :** Software, Validation, Supervision. **Martin Pilch :** Conceptualization, Methodology, Validation, Formal analysis, Resources, Data curation, Writing – review &

editing, Visualization, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (https://casl.gov/), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors, under U.S. Department of Energy Contract No. DE-AC05-00OR22725.

References

Collins, B.S., S. Stimpson, Wang, X., Regression Suite Improvements in MPACT, 2016. CASL-U-2016-1053-000, 2016.

Ganapol, B.D., *Analytical Benchmarks for Nuclear Engineering Applications*, Case Studies in Neutron Transport Theory, no. 6292. 2009. SBN 978-92-64-99056-2, NEA/DB/DOC (2008)1.

Kochunas, B., Collins, B., Jabaay, D., Downar, T.J., W. R. Martin, “Overview of development and design of MPACT: Michigan parallel characteristics transport code.”, American Nuclear Society, 555 North Kensington Avenue, La Grange Park, IL 60526 (United States); 2013 Jul 1.

Palmtag, S., 2017. MPACT Library Verification by Comparison of Assembly Calculations to Monte Carlo Results, CASL-U-2015-0281-001.

Pilch, M., Recommendations for Code Verification and Solution Verification for CASL Codes, CASL-U-2019-4108-000, 2019.

Smith, M.A., Lewis, E.E., Na, B.C., 2003. Benchmark on deterministic transport calculations without spatial homogenization: A 2-D/3-D MOX fuel assembly 3-D benchmark, *NEA/NSC/DOC (2003) 16*. Organisation for Economic Co-operation and Development, Nuclear Energy Agency.

Wang, J., Martin, W., Collins, B., 2017. Order of Accuracy of spatial discretization of method of characteristics. *M&C 2017 – International Conference on Mathematics & Computational Methods Applied to Nuclear Science & EngIneerIng.*

Wang, J., Martin, W., Collins, B., 2018. The application of method of manufactured solutions to method of characteristics in planar geometry. Ann. Nucl. Energy 121, 295–304.

Yamamoto, A., Tabuchi, M., Sugimura, N., Ushio, T., Mori, M., 2007. Derivation of optimum polar angle quadrature set for the method of characteristics based on approximation error for the Bickley function. J. Nucl. Sci. Technol. 44 (2), 129–136.

Young, M.T.H., 2016. MOCC: A Method of Characteristics based Nuclear Reactor Physics Simulator. University of Michigan, Ann Arbor. Ph.D. thesis.