COLLEGE OF ENGINEERING
NUCLEAR ENGINEERING & RADIOLOGICAL SCIENCES
UNIVERSITY OF MICHIGAN

Nuclear Reactor
Analysis & Methods
Group
NuRAM

NURAM-2021-002-00

MPACT Efficiency and Robustness Enhancements for Full-Core BWR Modeling

Sooyoung Choi, Qicang Shen, Daniel Jabaay, Yuxuan Liu, Brendan Kochunas, and Thomas Downar

University of Michigan

03/31/2021

This page is intentionally blank.

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0	03/31/2021	All	Initial Release

Document pages that are:

Export Controlled:	None
IP/Proprietary/NDA Controlled:	None
Sensitive Controlled:	None
Unlimited:	All

EXECUTIVE SUMMARY

Because of the Boiling Water Reactor (BWR)'s unique features such as the cruciform control blade, coolant void, two-phase flow, and heterogeneous geometry, the performance and robustness of MPACT for BWR applications is degraded compared to its performance for Pressurized Water Reactor (PWR) applications. In order to improve the efficiency and robustness for BWR simulations, several enhancements were considered and implemented in the work performed here, including:

- adoption of the linear-source MOC;
- improved iterative methods;
- memory reduction by using mixed single and double precision;
- optimization of problem initialization.

The Linear Source Approximation (LSA) and the optimized meshing were initially developed for PWR applications, and the LSA has not been used routinely for whole core calculations. In this work, issues with the linear source with respect to its robustness and optimal meshing for BWRs have been addressed. For the new iteration scheme, the multilevel in energy CMFD solver was combined with a sophisticated feedback-based partial convergence technique. The method was first shown to be very effective in reducing the number of outer iterations and MGCMFD iterations for multiphysics PWR applications, and was then adapted to BWR applications in this work. Although some considerations for robustness remain to be resolved. The mixed precision technique combined the use of different numerical precisions in the MPACT computational algorithm in order to reduce memory usage. Several variables with a large memory footprint that are not directly related to convergence checks (k_{eff}, fission source) are now stored as single-precision reals and converted back to double precision only in the calculation. All three enhancements improve the efficiency of MPACT for BWR simulations. The robustness of LSA and the new iteration scheme is improved in order to realize these efficiency gains. Finally, the problem initialization process has been optimized to speed up the geometry and meshing set-up at the beginning of a problem.

The new iterative methods have been shown to speed up the coupled MPACT simulation of the Peach Bottom 2 (PB2) cycle 1 problem by a factor of 2. The LSA and mixed precision reduce the total memory by 17% for the PB2 problem with a minimum runtime impact. The optimization of geometry and meshing setup results in a speedup of 30-40% in the problem initialization, depending on the number of unique assemblies and control cells. The remainder of the runtime speed up is attributed to the MEDPC algorithm.

This page is intentionally blank.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

BWR Boiling Water Reactor

CMFD Coarse Mesh Finite Difference

FS Flat Source

FSA Flat Source Approximation

HPC High Performance Computing

INL Idaho National Laboratory

LS Linear Source

LSA Linear Source Approximation

LLSA Limited Linear Source Approximation

MOC Method of Characteristics

MGCMFD multigroup CMFD

MED Multilevel-in-Energy Diffusion

NOPC Nearly-optimally Partially Converged

1GCMFD one-group CMFD

PB2 Peach Bottom 2

PWR Pressurized Water Reactor

SPH Super Homogenization

TH Thermal Hydraulics

TL Transverse Leakage

TCP$_0$ Transport-Corrected P_0

1. INTRODUCTION

The MPACT code was developed during the CASL project to analyze Pressurized Water Reactor (PWR) problems with the objective of efficiently solving the 3D neutron transport for PWR applications. The Boiling Water Reactor (BWR) core has similar material compositions compared to the PWR, but the neutron behavior, and therefore numerical behavior, are generally different because of the BWR's unique features. These features include the cruciform control blade, coolant void, two-phase flow, and highly heterogeneous geometry. These differences in physics, geometry and meshing were observed to degrade the performance of MPACT and in some cases introduce stability issues. For practical applications like the PB2 cycle depletion, longer computing time and convergence and stability issues were observed using the default options. Furthermore, the run time became longer since the BWR simulation normally requires more than 50 states for each burnup cycle (double that of a PWR simulation).

In order to speed up the calculation and reduce the memory usage for BWR simulations, in this milestone, several enhancements were considered and implemented, including:

- adoption of the linear source Method of Characteristics (MOC);
- improved iterative methods;
- memory reduction by using mixed single and double precision;
- optimization of problem initialization.

The Linear Source Approximation (LSA) and the optimized meshing were initially developed for PWR applications, and the LSA has not been used routinely for the whole core calculation. In this work, issues with the linear source with regard to its robustness and optimal meshing for BWRs have been addressed. For the new iteration scheme, the multilevel in energy Coarse Mesh Finite Difference (CMFD) solver was combined with a sophisticated feedback-based partial convergence technique [2, 3]. The method was first shown to be very effective in reducing the number of outer iterations and multigroup CMFD (MGCMFD) iterations for multiphysics PWR applications, and was then adapted to BWR applications in this work. The mixed precision technique combined the use of different numerical precisions in the computational algorithms in order to reduce memory usage. Several variables with large memory footprint that are not directly related to convergence checks (k_{eff}, fission source) are now stored as single-precision reals and converted back to double precision in the calculation. All three enhancements improve the efficiency of MPACT in BWR simulations. The robustness of LSA and the new iteration scheme is improved in order to realize these efficiency gains. Finally, the problem initialization process has been optimized to speed up the geometry and meshing set-up at the beginning of a problem. The remainder of the runtime speed up is attributed to the MEDPC algorithm.

Detailed descriptions of these enhancements are presented in Section 2, and the efficiency improvements for the PB2 cycle 1 depletion problem are discussed in Section 3.

2. METHOD AND CODE ENHANCEMENTS

2.1 Linear Source MOC

2.1.1 Brief Description for Linear Source MOC

The MOC based on the LSA is becoming a new standard in neutron transport calculations [4, 5, 6]. While the Flat Source Approximation (FSA) is convenient and used in many existing codes, the LSA can reduce the computational burden of the transport calculation. The LSA allows the use of a much coarser mesh discretization while maintaining solution accuracy [4]. Recently, the LSA has been developed and implemented in CASMO5 and OpenMOC [4, 5]. The method relies on trajectory-based numerical spatial moments to calculate the linear expansion coefficients of the scalar flux [4]. The method has also been implemented in MPACT with a slight reformulation in the LSA equations to facilitate application for the multiphysics calculations [6, 7].

The LSA in MPACT is based on the method implemented in CASMO5 [4]. In the LSA, MOC track-based spatial moments over source regions are calculated to obtain the LS expansion coefficients of the scalar flux. The differential form of the transport equation along a MOC segment is written as follows:

$$\frac{d\psi^g_{m,k,i}(s_m)}{ds_m} + \Sigma^g_{tr,i}\psi^g_{m,k,i}(s_m) = q^g_{m,k,i}(s_m) \ , \tag{1}$$

where m is the direction combining the azimuthal and polar directions; k is the segment index; g is the energy group index; i is the cell index; s_m is the distance along the MOC segment; $\psi^g_{m,k,i}$ is the angular flux; $\Sigma^g_{tr,i}$ is the transport cross section; and $q^g_{m,k,i}$ is the total source.

In the case of the eigenvalue problem with 2D MOC, the total source $q^g_{m,k,i}$ includes the fission source and scattering source. The LS along a MOC segment is approximated as:

$$\begin{aligned} q^g_{m,k,i}(s_m) &= \bar{q}^g_{m,k,i} + \hat{q}^g_{m,i}\left(s_m - \frac{s_{m,k,i}}{2}\right) \\ &= q^g_i + \mathbf{r}^c_{a,k,i}\cdot\hat{\mathbf{q}}^g_i + \frac{1}{\xi_i}\mathbf{u}_m\cdot\hat{\mathbf{q}}^g_i\left(s_m - \frac{s_{m,k,i}}{2}\right) \ , \end{aligned} \tag{2}$$

where $s_{m,k,i}$ is the total length of segment; $\bar{q}^g_{m,k,i}$ and $\hat{q}^g_{m,i}$ are the LS expansion coefficients; q^g_i is the cell-averaged total source; $\mathbf{r}^c_{a,k,i}$ is the segment midpoint coordinates; $\mathbf{u}_m$ is the direction vector; ξ_i is the scaling factor to conserve the analytic volume; and $\hat{\mathbf{q}}^g_i$ is the LS coefficient vector containing x and y components such that $\hat{\mathbf{q}}^g_i = \begin{bmatrix} q^g_{i,x} & q^g_{i,y} \end{bmatrix}^T$.

From the MOC sweep, the scalar flux ϕ^g_i and its spatial moments (e.g., $\phi^g_{i,x}$ and $\phi^g_{i,y}$) are calculated by accumulating the angular flux and product of the direction vector and angular flux. The scalar flux and its moments are used to calculate q^g_i and $\hat{\mathbf{q}}^g_i$ for the next MOC sweeping. The LSA in MPACT was formulated to eliminate the cross section dependence of the pre-computed coefficients for the multiphysics and 2D/1D simulations [7]. A complete derivation of the LSA is presented in [7].

2.1.2 Modified E_2 Function Calculation and Minor Code Error Fixes

Since the LSA has not been used routinely, there were some undiscovered code deficiencies related to the robustness of LSA. It was necessary to address these issues to use the LSA for the BWR applications. The first issue is related to the exponential function calculation. The LSA in MPACT uses the E_2 function to calculate the flux and its moments. The E_2 function is defined as a function of τ as follows:

$$E_2(\tau) = (1 - (1 - e^{-\tau})/\tau)/\tau \ , \tag{3}$$

where τ is the optical thickness. The E_2 function is tabulated and interpolated for efficient calculations of the exponential function which is one of the most expensive calculations in the MOC kernel. The E_2 function is used to calculate E_1 and T_2 in the following expression.

$$\frac{\Delta\psi^g_{m,k,i}}{\tau^g_{m,k,i}} = \left(\psi^{g,\text{in}}_{m,j,i} - \frac{\bar{q}^g_{m,k,i}}{\Sigma^g_{tr,i}}\right) E_1(\tau^g_{m,k,i}) - \frac{t_{m,k,i}}{2}\frac{\hat{q}^g_{m,i}}{\Sigma^g_{tr,i}} T_2(\tau^g_{m,k,i}) \ , \tag{4}$$

where $\Delta\psi^g_{m,k,i}$ is the change of angular flux; $\tau^g_{m,k,i}$ is the optical thickness; $\psi^{g,\text{in}}_{m,j,i}$ is the incoming angular flux; $t_{m,k,i}$ is the renormalized track segment length; and E_1 and T_2 are defined as

$$E_1(\tau^g_{m,k,i}) = 1 - \tau^g_{m,k,i} E_2(\tau^g_{m,k,i}) \ , \tag{5}$$

$$T_2(\tau^g_{m,k,i}) = 2E_2(\tau^g_{m,k,i}) - E_1(\tau^g_{m,k,i}) \ . \tag{6}$$

The E_2 is tabulated with Chebyshev points [8] between 0 to $\tau^{\max}$. The maximum value of τ (i.e. $\tau^{\max}$) is 40.0 which may be sufficiently large for the 2D calculation. However, the $\tau^g_{m,k,i}$ can exceed the maximum limit of the table in the case of 2D/1D calculation since the transport cross section becomes very large due to the Transverse Leakage (TL) splitting. When $\tau^g_{m,k,i}$ exceeds the range of table, $E_2(\tau^g_{m,k,i})$ is extrapolated. However, the extrapolation can give a negative value when $\tau^g_{m,k,i}$ exceeds the range as shown in Fig. 1. This is because the E_2 function is not sufficiently close to

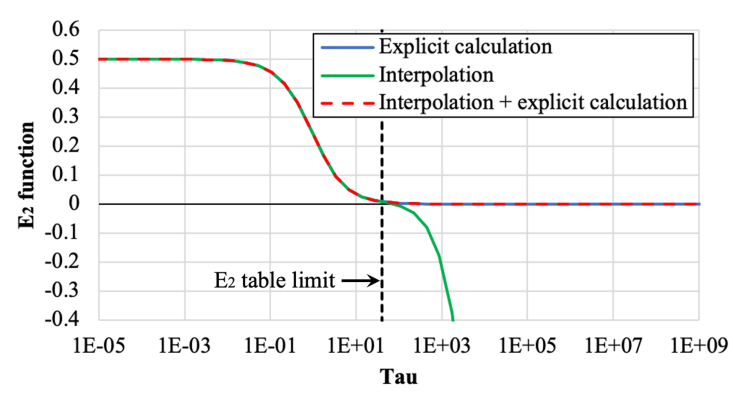


Figure 1. E_2 function calculations

0 at the end of table. The negative value from the E_2 extrapolation makes the solution diverge by causing the calculation of a negative flux during the iteration. This situation does not occur in the Flat Source (FS) MOC since the exponential function for the FS MOC (i.e. $1 - e^{-\tau}$) converges to 0 very quickly therefore the extrapolation gives 0 consistently. To resolve this issue, the E_2 calculation has been updated as follows:

$$E_2(\tau) = \begin{cases} \text{Interpolation of } E_2 \text{ table}, & \text{if } \tau \leq \tau^{\max}. \\ (1 - 1/\tau)/\tau, & \text{else if } \tau > \tau^{\max}. \end{cases} \tag{7}$$

Since $e^{-\tau}$ can be approximated to 0 if τ is larger than 40, Eq. (7) still calculates the E_2 function accurately as shown in Fig. 1. Moreover, the overhead may be negligible since there is no explicit calculation of the exponential function during the MOC sweep and τ rarely exceeds $\tau^{\max}$.

In addition to the E_2 table update, there were code errors related to the LS MOC solver. The angle decomposition algorithm was not completely implemented in the LS MOC solver, therefore the solution diverged when the angle decomposition with MPI was used. The domain decomposition in the LS MOC also had an issue with index out of bounds error that was not detected with the GCC-5.4.0 compiler. However, the issue caused a segmentation fault with the GCC-8.3.0 compiler. These code errors were also fixed.

2.1.3 Improvement in Robustness of Linear Source MOC

In MPACT and other transport codes adopting the 2D/1D method, many stabilization methods have been studied for the 2D/1D method based on the FSA to maintain and improve stability [9, 10, 11, 12, 13, 14]. However, none of the studies were performed for bigger BWR problems with LSA. In the LSA method, the linear component of the source always has some negative component when splitting the TL – this is a necessary result of the construction of the method. When testing the LSA, negative linear sources were encountered for some problems. The presence of a negative source is very detrimental to convergence stability. In the 2D MOC, the negative source can occur when the LS has a very steep source gradient compared with the actual source distribution. In the 2D/1D method, the TL splitting directly causes a negative source by letting half of the cell have a negative source to give a source that integrates to 0. Due to this lack of robustness, the LSA could not be used as a default option in MPACT unless the issue is resolved.

To address this negative source issue, and to ensure robust and stable convergence, a new method was developed in this work to reduce, or relax, the LS gradient of the problematic cell. We call this method the Limited Linear Source Approximation (LLSA). In the LLSA, the minimum local source $q_i^{g,\min}$ is computed by Eq. (8). This is the minimum value along a segment, and it is either at the entrance or exit of the segment.

$$q_i^{g,\min} = \min_{a,m,k} \left\{ \bar{q}^g_{m,k,i} \pm \hat{q}^g_{m,i} \frac{s_{m,k,i}}{2} \right\} . \tag{8}$$

To ensure positivity, we introduce the gradient reduction factor, γ_i^g, that is calculated by Eq. (9).

$$\gamma_i^g = \min\left(\frac{q_i^g}{q_i^g - q_i^{g,\min}},\ 1\right) . \tag{9}$$

When $q_i^{g,\min}$ is non-negative, γ_i^g is one because it is not necessary to reduce the LS gradient. Conversely, if $q_i^{g,\min}$ is negative in the first term, the denominator becomes larger than q_i^g . Therefore, the range of γ_i^g is between 0 and 1. When γ_i^g is equal to 0, the method will be the same as the FSA. γ_i^g is multiplied to $\hat{\mathbf{q}}_i^g$ to reduce the gradient of the LS leading to the following equation:

$$\begin{aligned} q'^g_{m,k,i}(s_m) &= \bar{q}'^g_{m,k,i} + \hat{q}'^g_{m,i}\left(s_m - \frac{s_{m,k,i}}{2}\right) \\ &= q_i^g + \gamma_i^g \mathbf{r}^c_{a,k,i}\cdot\hat{\mathbf{q}}_i^g + \frac{1}{\xi_i}\gamma_i^g \mathbf{u}_m\cdot\hat{\mathbf{q}}_i^g\left(s_m - \frac{s_{m,k,i}}{2}\right) , \end{aligned} \tag{10}$$

where $(\cdot)'$ indicates the parameter is modified by γ_i^g.

The LLSA changes the gradient of the linear component only, so that the average source and neutron population are still conserved. With this LLSA, the emergence of a negative flux or source is not detected if the cell-average source is non-negative.

When the LSA is used in the 2D/1D method, the only difference in the 2D/1D MOC equation is the existence of the TL from the axial nodal solver. There are many methods to construct the TL [15], but the limited flat and isotropic TL is used in this work. With the TL, L_i^g, the transport equation along a segment is written as:

$$\frac{d\psi^g_{m,k,i}(s_m)}{ds_m} + \Sigma^g_{tr,i}\psi^g_{m,k,i}(s_m) = \bar{q}^g_{m,k,i} + \hat{q}^g_{m,i}\left(s_m + \frac{s_{m,k,i}}{2}\right) + L_i^g . \tag{11}$$

In the 2D/1D method, a negative flux can be computed because the TL can have a negative value. If there is no constraint to prevent a negative TL, it is possible to encounter a negative source. Therefore, some TL splitting method [16] is used in most of the codes adopting the 2D/1D method. There are many variations in the TL splitting method [16], but the isotropic TL is the default option in MPACT. If a total source (i.e. $q_i^g + L_i^g$) is less than zero, then the source is moved to the left-hand side of the equation with an isotropic flux assumption as follows:

$$\begin{aligned} \frac{d\psi^g_{m,k,i}(s_m)}{ds_m} + \left(\Sigma^g_{tr,i} - \frac{q_i^g + L_i^g}{\phi_i^g}\right)\psi^g_{m,k,i}(s_m) &= \bar{q}^g_{m,k,i} + \hat{q}^g_{m,i}\left(s_m + \frac{s_{m,k,i}}{2}\right) - q_i^g \\ &= \mathbf{r}^c_{a,k,i}\cdot\hat{\mathbf{q}}_i^g + \frac{1}{\xi_i}\mathbf{u}_m\cdot\hat{\mathbf{q}}_i^g\left(s_m - \frac{s_{m,k,i}}{2}\right) . \end{aligned} \tag{12}$$

In both the FSA and LSA, the TL splitting is performed with the cell-averaged source q_i^g and TL, L_i^g. In the case of the FSA, the right-hand side of Eq. (12) is zero, so there is no need to consider

the negative source issue. In the LSA, however, the source term is not zero, and it is necessary to have a negative source when the integrated source over the cell is equal to zero.

As a part of the LLSA, a new TL splitting method is developed to prevent the negative source. This method is necessary to find the minimum source identical to Eq. (8).

$$q'^{g,\min}_i = \min_{a,m,k} \left\{ \bar{q}'^{g}_{m,k,i} \pm \hat{q}'^{g}_{m,i} \frac{s_{m,k,i}}{2} \right\} . \tag{13}$$

Note that the gradient reduction factor γ^g_i has been applied to the terms in Eq. (13). While $q'^{g,\min}_i$ can be obtained by Eq. (13), but this calculation can be more easily performed as follows if we already know $q^{g,\min}_i$ and γ^g_i:

$$q'^{g,\min}_i = \left(q^{g,\min}_i - q^g_i \right) \gamma^g_i + q^g_i . \tag{14}$$

In Eq. (14), $q^{g,\min}_i - q^g_i$ corresponds to the gradient of the LS, therefore γ^g_i is multiplied with that term. If $q'^{g,\min}_i + L^g_i < 0$, the TL splitting is performed with $q'^{g,\min}_i$ as follows:

$$\begin{aligned} &\frac{d\psi^g_{m,k,i}(s_m)}{ds_m} + \left(\Sigma^g_{tr,i} - \frac{q'^{g,\min}_i + L^g_i}{\phi^g_i} \right) \psi^g_{m,k,i}(s_m) \\ &\quad = \bar{q}'^{g}_{m,k,i} + \hat{q}'^{g}_{m,i} \left(s_m + \frac{s_{m,k,i}}{2} \right) - q'^{g,\min}_i \\ &\quad = q^g_i - q'^{g,\min}_i + \gamma^g_i \mathbf{r}^c_{a,k,i} \cdot \hat{\mathbf{q}}'^{g}_i + \frac{1}{\xi_i} \gamma^g_i \mathbf{u}_m \cdot \hat{\mathbf{q}}'^{g}_i \left(s_m - \frac{s_{m,k,i}}{2} \right) . \end{aligned} \tag{15}$$

The source-term of Eq. (15) is always non-negative if q^g_i is non-negative. The LSA is implemented for the P_0 and P_n MOC solvers in MPACT, but only the isotropic source is approximated as linear for a good compromise between calculation burden and accuracy [4, 17]. Therefore, implementing the LLSA for the P_n MOC solver can be done like what is described in this work.

The LSA and LLSA are compared for the PB2 cycle 1 single assembly problem [18, 19]. The radial view of problem geometry is shown in Fig. 2. The 2D/1D solver in MPACT is used for the simulations of both cases since they are 3D problems. The 2D MOC with P_2 scattering and P_3

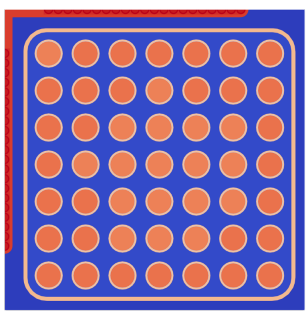


Figure 2. Radial view of PB2 3D single assembly problem

Table 1. Numerical results from PB2 assembly simulations

Method	k_{eff}	Difference (pcm)	Max difference in pin power (%)	Is negative LS detected?	Elapsed time (mm:ss)
LSA	1.03801	-	-	Yes	03:18
LLSA	1.03801	0.000	0.015	No	03:22

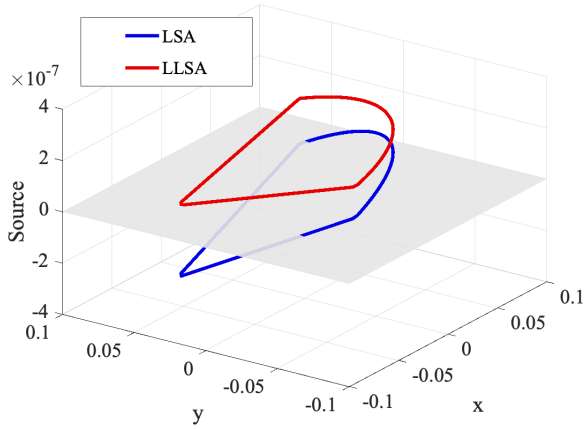


Figure 3. Source distribution in control blade region from PB2 single assembly simulation

one-node nodal method are used in the calculation. For the MOC ray and angle discretization, the Chebyshev-Yamamoto quadrature is used with a ray spacing of 0.05 cm, 16 azimuthal angles 2 polar angles per octant. The fuel pellet is divided into 2 regions radially, and the surrounding coolant region is divided into 4 regions in the azimuthal direction. The non-fuel rod, such as the guide tube or plenum, does not have subregions. The reflector pin-cell is divided to have 2x2 meshes. 40 MPI processes were used in the simulation.

The simulation results for both cases are summarized in Table 1. The LSA shows negative LSs for the PB2 assembly problem. In the case of the PB2 problem, the negative LS is observed from the 2nd outer iteration to convergence. Fortunately, the negative LS does not lead to instability in either problem, but the possibility remains.

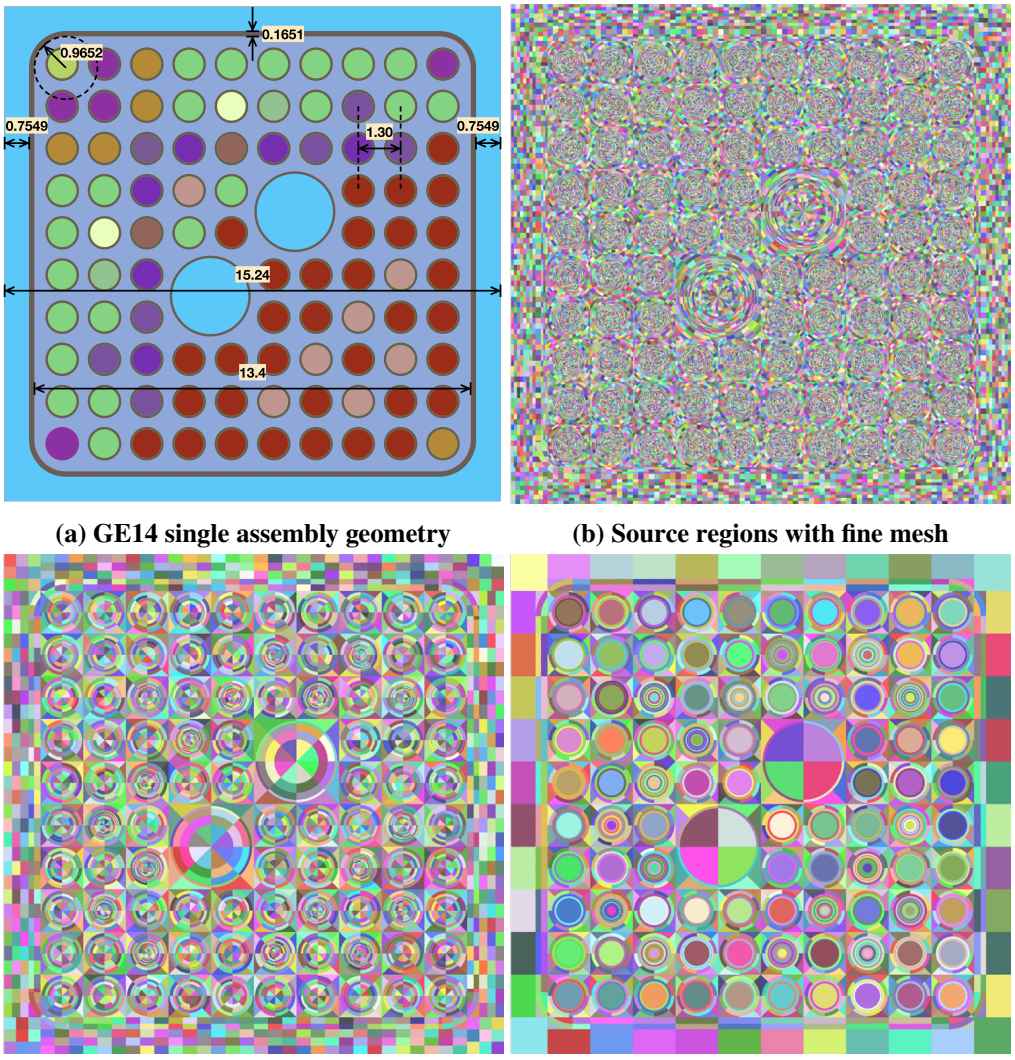
On the other hand, the LLSA does not show the negative LS. There are minimal differences between the results from the LSA and LLSA. The maximum difference in the pin power distribution is 0.015%. This is because the LLSA changes the LS coefficient and splits the TL by as little as possible to maintain a non-negative LS at the same time. It is concluded that the LLSA successfully resolves the negative LS issues caused by the LSA. There is also minimal overhead from the LLSA in the calculation time, so it is negligible.

The source distribution is plotted in Fig. 3 to illustrate the difference between the LSA and LLSA. Fig. 3 shows the source distributions at the end of the control blade mesh in the PB2 assembly problem. x and y in the figure indicate the relative position from the center of mesh. It is clearly observed that the LSA has negative values in Fig. 3 when the TL splitting is performed. The average source is 0, and half of the mesh has the negative LS. This is why the negative LS was observed in Table 1. On the other hand, the LLSA has positive sources in all locations. In the LLSA, both

LS gradient and the quantity of the split source are adjusted properly. This example shows why the LLSA is necessary to adjust the quantity of the split source and the gradient of LS simultaneously to avoid the negative LS that may cause instability.

2.1.4 Linear Source MOC Mesh Optimization

It is necessary to determine the default or recommended meshing options to use the LS MOC efficiently for BWR applications. The LS MOC mesh option was determined for PWR applications, but it has not been determined for BWRs. GE14 single assembly problems [20] are selected to perform the mesh sensitivity test. There are two configurations (i.e. unrodded and rodded) of the assembly depending on the existence of the control blade. Fig. 4 shows the geometry and mesh of a GE14 assembly. The *finemesh* was made to eliminate the mesh discretization error. The *finemesh* is also used as one of references in the comparison. The *default* option is the default mesh option with the FS. The LS optimum mesh is the mesh determined through mesh sensitivity tests.



(a) GE14 single assembly geometry

(b) Source regions with fine mesh

(c) Source regions with default mesh

(d) Source regions with LS optimum mesh

Figure 4. GE14 single assembly problem geometry and mesh discretizations

The k_{eff} and the pin power distribution results are presented in Table 3 using these mesh options. The solutions are compared to the Monte Carlo solution (Serpent) and also compared to the *finemesh* calculation with LS (i.e. LS-finemesh). The LS and FS give very similar results to each other for the GE14 unrodded case. The k_{eff} difference is 31 pcm and the RMS pin power difference is 0.04%. This means the LS and FS calculate an identical solution if the mesh discretization error is completely eliminated. The FS with default mesh option gives 130 ~ 193 pcm differences in the k_{eff} and about 0.5% RMS difference in the power distribution compared to the LS-finemesh calculations. This means the differences are caused by insufficient mesh discretization. However, these differences are within our acceptance criteria [21]. On the other hand, the LS with the optimum mesh shows 52 ~ 58 pcm difference in k_{eff} and less than 0.1% RMS difference in the pin power distribution. With using the LS, we can improve the accuracy for k_{eff} and the pin power distribution by 100 pcm and 0.4%, respectively. In this comparison, the very coarse assembly gap mesh (shown in Fig. 4d) is used in the LS MOC. However, it was found that the coarse mesh is not sufficient for the calculation with the Thermal Hydraulics (TH) feedback. The gap mesh discretization may become more important as the void fraction of the coolant inside the channel box increases. Therefore, two more meshing lines normal to the channel box and one more meshing line parallel to the channel box at the outer gap were added. The updated meshing option is used in the following calculations in this report.

Additional comparisons between the FS and LS are made with more progression problems in [20]. The results are presented in Table 2. On average, the LS is 0.4% more accurate than the FS in terms of the RMS pin power difference. The LS calculates 130 pcm larger differences in k_{eff} than the FS. This is probably because the Super Homogenization (SPH) factors embedded in the resonance data of MPACT's cross section library are based on the FS MOC with default mesh option. For the multigroup cross section calculations, MPACT uses the subgroup method by default. The subgroup method is also used in all calculations in this report. There are some error sources in the resonance self-shielding with the subgroup. The resonance interference effect is one of the most important sources of error. The resonance interference effect may not be considered correctly with the conventional subgroup method and equivalence theory [22, 23]. There are some other error sources such as the intermediate resonance source approximation, and heterogeneous resonance integral table [24, 25]. Even though the multigroup cross section is calculated accurately, there is still inconsistency between the multigroup calculation and the continuous energy Monte Carlo due to the angle dependency of the multigroup cross section [26]. To address these issues in the

Table 2. Mesh sensitivity test for GE14 single assembly problems

Case	Calculation options	Compared to Serpent			Compared to LS-finemesh		
		k_{eff} diff (pcm)	RMS power diff. (%)	MAX power diff. (%)	k_{eff} diff (pcm)	RMS power diff. (%)	MAX power diff. (%)
GE14 Unrodded	LS - finemesh	147	0.27	0.66	0 (Ref.)	0 (Ref.)	0 (Ref.)
	FS - finemesh	116	0.30	0.71	-31	0.04	0.10
	FS - default	17	0.65	1.65	-130	0.45	1.25
	LS - optimum	199	0.24	0.53	52	0.07	0.16
GE14 Rodded	LS - finemesh	182	0.26	0.49	0 (Ref.)	0 (Ref.)	0 (Ref.)
	FS - finemesh	108	0.33	0.78	-73	0.11	0.51
	FS - default	-11	0.71	1.61	-193	0.50	1.35
	LS - optimum	240	0.24	0.50	58	0.07	0.16

Table 3. Comparisons of FS and LS to BWR progression problems

Case	k_{eff} difference (pcm)		Pin power difference (%) 3D RMS		3D MAX	
	FS	LS	FS	LS	FS	LS
BWR-GE9-V00-C	-112	184	0.64	0.17	1.65	0.42
BWR-GE9-V00	-67	101	0.52	0.16	1.20	0.32
BWR-PEACH-6-00-C	-121	154	0.48	0.15	1.36	0.36
BWR-PEACH-6-00	-45	90	0.46	0.12	1.11	0.33
GE14-3D-00-HZP	52	241	1.59	1.23	7.49	7.29
GE14-3D-AX-HFP	49	185	2.58	1.99	8.67	5.77
GE14-4X4-00-C	19	217	0.55	0.22	1.50	0.58
GE14-4X4-00	18	214	0.54	0.22	1.30	0.65
PLR-00-C	59	273	0.71	0.22	1.70	0.66
PLR-00	60	235	0.58	0.20	1.50	0.43
Absolute Average	60.20	189.40	0.87	0.47	2.75	1.68

multigroup cross section calculations, we have used the SPH factor in MPACT [27]. However, one issue here is that the error from the mesh discretization is also corrected by the SPH factor as well as the other error sources in the multigroup cross section calculations since the FS MOC with the default mesh option is used in the SPH factor calculation. Therefore, we expect that the accuracy of k_{eff} with the LS MOC will be more accurate as well if the SPH factor is regenerated with the very fine mesh calculation to eliminate the error source from the mesh discretization. Recall that the LS gives more accurate results in terms of mesh convergence.

2.2 MEDPC

The efficiency of the BWR simulation can be improved by increasing the convergence rate of the fission source and decreasing the run time of the CMFD solver. MEDPC is the new CMFD solver used to improve the efficiency of the CMFD method. It is developed based on the Multilevel-in-Energy Diffusion (MED) solver [1], that is a more efficient CMFD solver compared to the default solver in MPACT for the problem without feedback. However, the MED is typically less stable than the default CMFD in problems with feedback. Therefore, the concept of the Nearly-optimally Partially Converged (NOPC)-CMFD [2] is introduced to improve the robustness of the MED solver and the new method is called MEDPC.

In the following three subsections, the theory of CMFD, MED and NOPC-CMFD are introduced.

2.2.1 CMFD Acceleration

The CMFD method utilizes the low-order diffusion equation to accelerate the convergence of the transport solution. In operator form, the diffusion equation is a generalized eigenvalue problem written as

$$\mathbf{M}\Phi = \lambda \mathbf{F}\Phi. \tag{16}$$

The matrix $M, \mathbf{F} \in \mathbb{R}^{NG \times NG}$, with N as the number of coarse meshes, and G as the number of group. The default CMFD system is formed with multigroup cross sections, i.e. $G \gg 1$.

To solve this eigenvalue problem, the Wielandt shifted power iteration is used. For the l_{th} power

iteration, the solution process is

$$\left[\mathbf{M} - \lambda_s^{(l)}\mathbf{F}\right]\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)} = \left[\lambda^{(l)} - \lambda_s^{(l)}\right]\mathbf{F}\mathbf{\Phi}^{(l)}\,, \tag{17}$$

$$\lambda^{(l+1)} = \lambda_s^{(l)} + \left[\lambda^{(l)} - \lambda_s^{(l)}\right]\frac{\left\|\mathbf{F}\mathbf{\Phi}^{(l)}\right\|}{\left\|\mathbf{F}\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)}\right\|}\,. \tag{18}$$

λ_s is the Wielandt shift parameter. The default CMFD linear system in MPACT is solved with the GMRES solver from PETSc [28]. In MPACT, the default λ_s is $2/3$.

Due to the large size of the linear system, the solution process of each power iteration is extremely slow. To balance the efficiency and accuracy, the eigenvalue problem is always partially converged with the number of power iterations smaller than or equal to L. The default value of L is 20. Nevertheless, for the full-core simulation, the run time of default CMFD solver is more than $1/3$ of the total run time [1].

2.2.2 MED

To improve the efficiency of the default CMFD solver in the solution of the diffusion linear system, the MED solver has been proposed and implemented in MPACT [1]. The solver is multilevel in that the power iteration is mainly performed on a one-group CMFD (1GCMFD) system. The 1GCMFD system Eq. (19) is obtained by collapsing the original MGCMFD system with the latest multigroup flux.

$$\mathbf{M}_{1G}\mathbf{\Phi}_{1G} = \lambda_{1G}\mathbf{F}_{1G}\mathbf{\Phi}_{1G} \tag{19}$$

The multigroup flux is obtained by solving the fixed source problem Eq. (20) with the fission source calculated for the one-group solution of:

$$\mathbf{M}_{MG}\mathbf{\Phi}_{MG} = \lambda_{1G}\chi_{MG}\mathbf{F}_{1G}\mathbf{\Phi}_{1G}\,. \tag{20}$$

Since the size of the one-group matrices M_{1G} and F_{1G} are much smaller, the computational cost for solving 1GCMFD system is cheaper compared to solving the MGCMFD system. To converge the eigenvalue and flux, multiple iterations are performed between the MGCMFD system and the 1GCMFD system.

The number of iterations between the 1GCMFD and MGCMFD is L_{MG}, and the 1GCMFD system is solved with L_{1G} power iterations per mgcmfd iteration. Therefore, the total number of power iterations performed is $L_{1G}L_{MG}$, which is much larger than 20. Consequently, the convergence rate of the MED case is faster for large-scale problem, compared to the case with the default CMFD solver.

Table 4. MED Solver vs Default CMFD Solver in GE14-3D-00-HZP Problem

	Default	MED
# of MOC inners	15	15
Total Run Time (s)	417	187
CMFD Run Time (s)	360	127

Table 5. MED Solver vs Default CMFD Solver in 2D Peach Bottom Problem [1]

	Default	MED
# of MOC inners	157	21
Total Run Time (s)	1319	350
CMFD Run Time (s)	275	65

In Table 4, we can observe that the CMFD run time is reduced to 1/3 when MED is used for the GE14-3D-00-HZP problem, that is a small-sized problem. In Table 5, we can observe that in the 2D Peach Bottom problem (i) the number of MOC iterations is much smaller, and (ii) the CMFD run time is much shorter.

2.2.3 NOPC-CMFD

The efficiency from the MED solver might not be observed in the problems with feedback [1]. On the contrary, it could make the solutions converge more slowly. The reason was that the CMFD problem is too tightly converged. In our previous research [2], it was theoretically shown that the tight convergence of the CMFD solutions will reduce the convergence rate in the problem without feedback. However, in the problems with feedback, the CMFD solution should be neither too tightly nor too loosely converged. As a result, a sophisticated method for the partial convergence of CMFD solutions was developed. Here the partial convergence parameters refer to the number of power iterations and the Wielandt shift value λ_s used to solve the diffusion eigenvalue problem.

We have proposed Eqs. (21) to (22) to determine the partial convergence in terms of the Wielandt shift parameter λ_s.

$$r = 1 - \frac{\left(1 - \frac{1-\gamma_g}{1+\frac{3\gamma}{\omega_p^2}}\right)^{\frac{1}{L}}}{1 - \left(1 - \frac{1-\gamma_g}{1+\frac{3\gamma}{\omega_p^2}}\right)^{\frac{1}{L}}} \frac{\omega_p^2}{3(1-c)}, \tag{21}$$

$$\lambda_s = r\lambda \tag{22}$$

Here, γ_g is a problem-dependent parameter characterizing the feedback intensity. ω_g is a parameter related to the size of the problem in terms of the mean free path and c is the scattering ratio. L is the number of power iterations we prefer to use to solve the eigenvalue problem. It should be noted that for the MED, L is $L_{MG}L_{1G}$, i.e. the total number of power iterations performed on the 1GCMFD system. CMFD with λ_s determined by Eqs. (21) to (22) is called NOPC-CMFD in [2, 3]. And MEDPC is the NOPC-CMFD solver that is implemented based on MED. Since the partial convergence is nearly-optimally determined, the use of the relaxation factor can be avoided. Moreover, the larger the feedback is, the looser the nearly-optimal partial convergence is. Therefore, with feedback intensity increasing, the linear system becomes easier to solve. This will potentially increase the efficiency.

We have proposed procedures to estimate the γ_g for PWRs in [3]. It has been observed that in PWR problems, MEDPC can let MPACT converge a problem with different power levels in a similar convergence rate and reduce the run time of a multi-state PWR depletion problem by more than

40%. The enhancement from using MEDPC in the BWR problem is shown in Section 3.

2.3 Mixed Precision

Mixed precision refers to the combined use of different numerical precisions in a computational method. There are a few benefits to using lower precision than 64-bit (double) floating point. First, it requires less memory for storing a variable. Second, it requires less memory bandwidth, thereby speeding up the data transfer operations. Third, math operations run faster in reduced precision. For example, mixed precision algorithms were developed to accelerate scientific computations for solving linear systems [29].

In the reactor neutronics calculation, since the convergence criteria for several quantities of interest, such as the effective multiplication factor and neutron flux, are at 10^{-6}, calculations performed with all single-precision variables may suffer from convergence issues. Therefore, a mixed precision scheme is proposed here to store selected quantities with a large memory footprint as single-precision reals, and convert them back to double precision when performing the actual calculations relevant to the numerical iteration and convergence.

To identify the variables with a large memory footprint, memory profiling was performed initially with the P5-2D problem with depletion. Fig. 5 shows the memory usage of the components in MPACT. The largest three of them are the MOC sweeper, the shielder (resonance self-shielding) sweeper, and the cross section mesh. We look further into each of the three components to understand the components of the memory usage, as shown in Tables 6-8.

In both MOC sweepers, the ray tracing data (ray segment length and region indices of a ray segment) and neutron fluxes (scalar flux and angular flux at the boundaries of a spatial domain) are

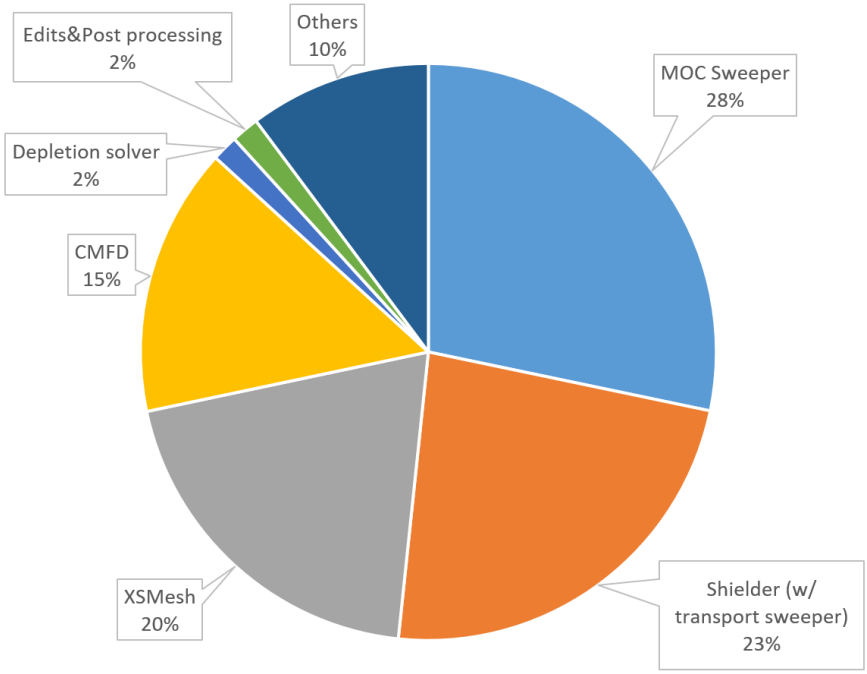


Figure 5. Memory profiling of the 5a-2d case with depletion

the dominating variables. In the cross section mesh data, the 1D and 2D cross sections account for more than 90% of the memory in the depletion calculation, and the isotope densities are a smaller fraction. Therefore, in the initial implementation of the mixed precision, three groups of variables with a large memory footprint are converted to single precision: 1) MOC ray segment lengths; 2) all cross sections on the macroscopic cross section mesh; 3) the angular flux at the boundaries.

Another reason for choosing the ray segments and the cross sections to be single precision is that in reality, these values are not known to double precision due to the manufacturing tolerance in geometry or the cross section measurements. For the angular flux at the boundaries, this is an iterative quantity so there could be some risk in making it single precision. However, it is not the primary iterative quantity used in the convergence check. Presently, we have not seen a convergence problem related to this change. The relative consumption of memory by the angular flux at the boundaries increases as the problem is further decomposed, so making this single precision has a bigger benefit on more spatially decomposed problems, and reduces the data in the communication, which can lead to reduced communication overhead. The scalar flux is kept as double precision since it directly relates to the fission source convergence. Also, we do not expect a runtime benefit from the current mixed-precision implementation since the single-precision variables are all converted back to double precision in the calculation.

Table 6. A Memory breakdown in the MOC Sweeper.

	Memory (GB)	Fraction
Fluxes (1/3 scalar flux + 2/3 angular flux BC)	2.135	35.8%
Ray tracing	3.038	51.0%
XS+Source	0.754	12.7%
Others	0.031	0.5%
Total	5.957	100%

Table 7. A Memory breakdown in the shielder MOC Sweeper.

	Memory (GB)	Fraction
Fluxes (1/3 scalar flux + 2/3 angular flux BC)	1.134	27.2%
Ray tracing	1.525	36.6%
XS+Source	0.841	20.2%
Coeff. for the fast sweeper	0.635	15.2%
Others	0.030	0.7%
Total	4.163	100%

Table 8. A Memory breakdown in the depleted XS Mesh.

	Memory (GB)	Fraction
Isotope densities	0.257	7.7%
1D XS	0.381	11.5%
Scattering matrix	2.102	63.5%
Isotopic fission XS	0.499	15.1%
Others	0.072	2.2%
Total	3.311	100%

An initial test for the same P5-2D problem showed that using single precision reals for segment length, boundary angular flux and cross sections will reduce the overall memory by 5%, 7% and 7%, respectively. We will expand the conversion to additional variables as the framework is established to facilitate the conversion.

2.4 Problem Initialization Speedup

The complexity of BWRs has shown that for full core problems the geometry processing in MPACT is significantly slow. A task done for this milestone was to increase the speed of the geometry and mesh setup to help the user.

To this end, there were several places where the code was optimized for speed. When setting up an assembly, it is sliced axially into 2-D lattices. In the problem initialization, all lattices were setting up the control blade geometry in the blade locations, regardless of whether that geometry and mesh had already been set up. A change was made to only place the blade in each unique lattice type, thus eliminating repeated operations by reducing the number of times a blade is added to the geometry.

Another place the code was sped up was in the rotation of all of the meshes. In MPACT, the geometry is first set up in an unrotated state. It is then rotated and the rotated geometry is meshed. There were several steps in this process where the same object was being created. Now, each unique pin cell mesh is setup once and rotated as needed, and similarly for lattices.

The effectiveness of the speedup is shown in Table 9. A full 3-D BWR core is initialized before and after the modifications mentioned above. The case name description refers to the number of unique assemblies or control cells in the case. The first case only has one unique assembly type used throughout the core map, and it is the same case as a single unique control cell. As the number of unique assemblies increases, so does the complexity of the setup of the problem geometry and mesh. Every unique assembly is loaded into the rotation map location that corresponds to no rotation. The control cell cases show how much speed up was gained specifically from the modifications to

Table 9. Core Geometry and Mesh Initialization Comparison

Number of Unique Geometries	Meshing Time (s)		Percent Speedup
	before	after	
1 Asy	50.83	36.26	28.66
2 Asy	79.59	54.31	31.76
4 Asy	138.32	94.09	31.98
8 Asy	283.77	184.25	35.07
16 Asy	487.94	314.14	35.62
32 Asy	956.97	695.69	27.30
2 Con. Cell	85.74	55.70	35.04
4 Con. Cell	157.18	97.70	37.84
8 Con. Cell	331.37	202.94	38.76
16 Con. Cell	576.95	352.90	38.83
32 Con. Cell	1152.25	641.35	44.34

the rotation routines, while the assembly cases show the speed up from the non-rotation portions of the code. The speedup is calculated as the difference of the before time and the after time, divided by the before time multiplied by 100.

3. RESULTS OF FULL-CORE CALCULATION

3.1 Description for Peach Bottom 2 Reactor Simulation

In this section, the impact of individual options (i.e. LS, mixed precision, and MEDPC) is analyzed against the PB2 core model [18, 19] to determine the improvement from each option. The radial view of the PB2 core geometry is shown in Fig. 6. Three types of fuel assemblies are loaded in the core. The control blade and the detectors are placed in the assembly gaps. To investigate improvements from the methods proposed in this work, we ran the PB2 model with a quarter code symmetry. As shown in Fig. 6, the actual core is not quarter symmetric. However, the primary purpose of this work is to determine how much efficiency and robustness can be improved by the proposed methods. Therefore, it is acceptable to use the quarter code model for this purpose. Fig. 7 presents the reactor conditions over the PB2 cycle 1 simulations. The reactor conditions in Fig. 7 are used to demonstrate the cycle 1 simulation as it is as much as possible.

The 2D/1D solver in MPACT is used for all simulations. The 2D MOC with the Transport-Corrected P_0 (TCP_0) scattering source option and the P_3 one-node nodal method are used in the calculation. For the MOC ray and angle discretization, the Chebyshev-Yamamoto quadrature is used with a ray spacing of 0.05 cm, 16 azimuthal angles 2 polar angles per octant. The TH feedback is calculated using the simplified TH module in MPACT. The quarter core models were run with 1200 cores on the Idaho National Laboratory (INL) High Performance Computing (HPC) machine, Sawtooth.

3.2 Impact of Linear Source MOC in the Full-core Calculation

The first comparison investigates the benefits of the LSA. The cycle 1 simulations were performed with the default options for the meshing and MOC solver and the LSA option. It should be noted that the high resolution vessel option was turned off in both simulations. The high resolution

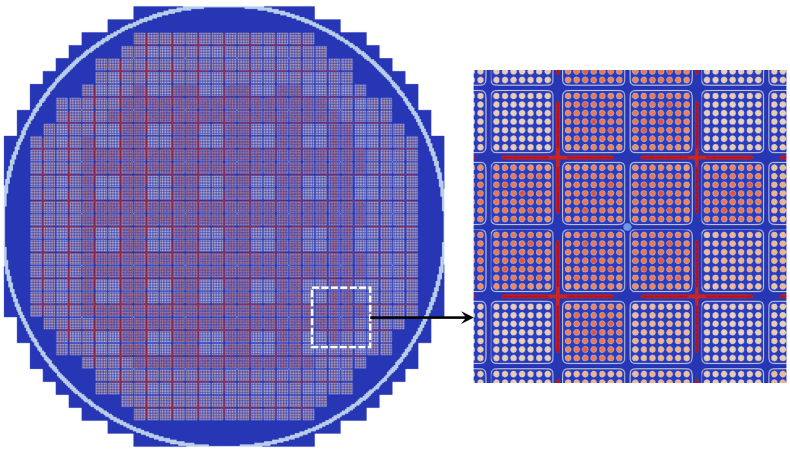
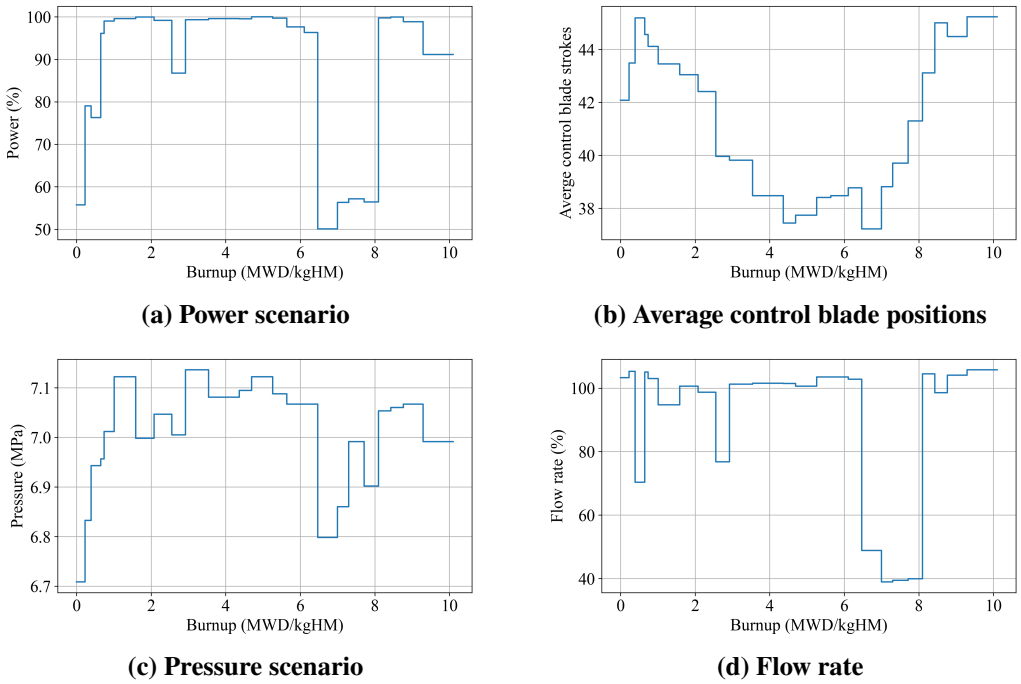


Figure 6. PB2 core radial view



(a) Power scenario **(b) Average control blade positions**

(c) Pressure scenario **(d) Flow rate**

Figure 7. PB2 cycle 1 conditions

Table 10. PB2 reactor cycle 1 simulation with LS

	Default	LS	Change (%)
# of states	25	25	0
# of predictors/correctors	48	48	0
# of XS calculations	1055	1052	0
# of subgroup FSP calls	435	425	-2
# of simplified TH calls	1054	1051	0
# of MGCMFD inners	1302500	1294400	-1
# of nodal inners	5295	5280	0
# of MOC inners	1059	1056	0
# of outer iterations	1059	1056	0
MOC (hours)	7.95	10.01	26
Nodal (hours)	2.09	3.05	46
CMFD (hours)	25.06	26.92	7
TH (hours)	14.77	12.69	-14
Depletion (hours)	0.47	0.32	-32
XS and etc. (hours)	5.00	2.62	-48
Total (hours)	55.34	55.61	0
Max memory per core (GB)	3.12	2.77	-11
Average memory per core (GB)	2.74	2.41	-12
Total memory (TB)	3.21	2.82	-12

vessel option is to model the vessel geometry more accurately. If this option is not used, the vessel geometry is composed of pin-cell size meshes. There is a minimal impact on the reactor parameters

by turning off the option, but the coarser mesh is desirable for the LS MOC. Fortunately, the LSA did not encounter a stability issue related to the negative LS, so the LLSA was not used in this simulation.

Table 10 shows the simulation results for the default and LS options. The number of outer iterations are similar (about 1060) for both cases. While the LSA does have more unknowns per cell to converge, relative to the FSA, it does not necessarily require more iterations since there are fewer cells. For the PB2 simulations, the effects from both points are not significant or they may have canceled each other, resulting the similar number of outer iterations. The total runtime from both simulations is similar at 55 hours. The LSA uses 2 hours more in the MOC calculations, but it can save the runtime in other places (such as the subgroup fixed source calculations) by decreasing the number of source regions. On the other hand, the LSA shows a 12% reduction in the total memory usage since coarser mesh can be utilized with the LSA.

3.3 Impact of MEDPC in the Full-core Calculation

The impact of the use of MEDPC is analyzed in Table 11. In contrast with the previous comparison, the high resolution vessel option was used for both the default and MEDPC cases since the mesh discretization is not a point of this comparison. From the use of MEDPC, it is expected to reduce the number of outer iterations and the number of MGCMFD iterations by improving the coupling scheme for feedback and the CMFD iteration. The improvement from MEDPC is clearly observed in Table 11. The number of outer iterations is reduced from 1032 to 633 by using the MEDPC. The nearly-optimal partial convergence shows a great improvement in reducing the outer iterations.

Table 11. PB2 reactor cycle 1 simulation with MEDPC

	Default	MEDPC	Change (%)
# of states	25	25	0
# of predictors/correctors	48	48	0
# of XS calculations	1029	680	-34
# of subgroup FSP calls	419	258	-38
# of simplified TH calls	1028	679	-34
# of MGCMFD inners	1269800	171465	-86
# of nodal inners	5160	3165	-39
# of MOC inners	1032	633	-39
# of outer iterations	1032	633	-39
MOC (hours)	7.59	4.83	-36
Nodal (hours)	1.81	1.26	-30
CMFD (hours)	24.47	4.53	-81
TH (hours)	14.20	9.53	-33
Depletion (hours)	0.45	0.46	2
XS and etc. (hours)	4.88	2.26	-54
Total (hours)	53.40	22.87	-57
Max memory per core (GB)	3.06	3.07	0
Average memory per core (GB)	2.69	2.70	0
Total memory (TB)	3.15	3.17	1

With the default options, the CMFD is the most time consuming part of the calculation. It took 24.47 hours with more than 1 million MGCMFD inner iterations for the cycle 1 calculation. The MEDPC reduces the MGCMFD iterations by 86% so that the elapsed time for CMFD is also reduced by 81%. Overall, the total runtime is reduced by 57%. There is a minimal overhead in the memory usage.

It should be noted that the `reset_sol` option has been used at states 2 and 3 for the MEDPC case shown in Table 11. The `reset_sol` option is to reinitialize the transport solution, and it is usually used if a stability issue occurs. MPACT has a feature that reinitializes the transport solution automatically if a high residual error (i.e. residual > 50.0) occurs during iterations. In the simulation with the default option, the solution was reinitialized 4 times due to the high residual error. The MEDPC case also showed the high residual error, and the residual became `NaN` in some states. The logic in MPACT does not reinitialize the solution automatically if the residual becomes `NaN`. Therefore, the `reset_sol` is used to reinitialize the solution manually. In the future, we will improve the current code logic so that the solutions can also be reset automatically for the MEDPC cases. We will also continue investigating the high residuals, and try to develop the method to predict whether the high residuals will happen, and then reset the solution at the beginning of the state.

3.4 Impact of Mixed Precision in the Full-core Calculation

Table 12. PB2 reactor two states simulation with mixed precision

	Default	Mixed Precision	Change (%)
# of states	2	2	0
# of predictors/correctors	2	2	0
# of XS calculations	58	58	0
# of subgroup FSP calls	29	29	0
# of simplified TH calls	57	57	0
# of MGCMFD inners	77100	75800	-2
# of nodal inners	290	290	0
# of MOC inners	58	58	0
# of outer iterations	58	58	0
MOC (hours)	0.52	0.52	0
Nodal (hours)	0.12	0.10	-17
CMFD (hours)	1.47	1.49	1
TH (hours)	0.83	0.77	-7
Depletion (hours)	0.02	0.02	0
XS and etc. (hours)	0.25	0.21	-16
Total (hours)	3.22	3.12	-3
Max memory per core (GB)	3.04	2.77	-9
Average memory per core (GB)	2.74	2.49	-9
Total memory (TB)	3.21	2.92	-9

The simulation result with the mixed precision is compared in Table 12. Results from simulations performing two states are compared in this section to investigate the impact of the mixed precision

efficiently. In this comparison, the `reset_sol` option (described in Section 3.3) is used for both the default and the mixed precision cases because one more reinitialization occurred with the mixed precision case. This situation can mislead the impact of using the mixed precision, so that the solution is reinitialized manually for both cases. The mixed precision does not lead to the high residual error directly. The calculations with feedback are not yet completely stable, so it is hypothesized that a small perturbation makes the high residual error occur, unfortunately. When using `reset_sol` for both cases, the number of outer iterations is the same. The use of the mixed precision does not change the runtime or iterations since the variables used in the actual calculations are still double precision. The total runtime is also similar with a minimal difference. The memory usage is reduced about 9% by using the mixed precision. With this initial implementation, the macroscopic cross section, the MOC segment length, and the boundary angular flux have been changed to the single precision. It may be possible to further reduce the memory by converting other variables to single precision.

3.5 Impact of Combining All Methods in the Full-core Calculation

The LS, MEDPC, and mixed precision are combined and tested for the PB2 cycle 1 calculation. Similar to the MEDPC case in Section 3.3, the `reset_sol` is used at states 2, 3, 4 and 12 to prevent the issue related to the `NaN` residual error. The `reset_sol` option was necessary for 2nd and 3rd states in the case of the MEDPC (see Section 3.3). However, two more states were necessary to reinitialize when all implements are used together since the path to the converged solution is changed due to combining the LS, the mixed precision, and MEDPC together.

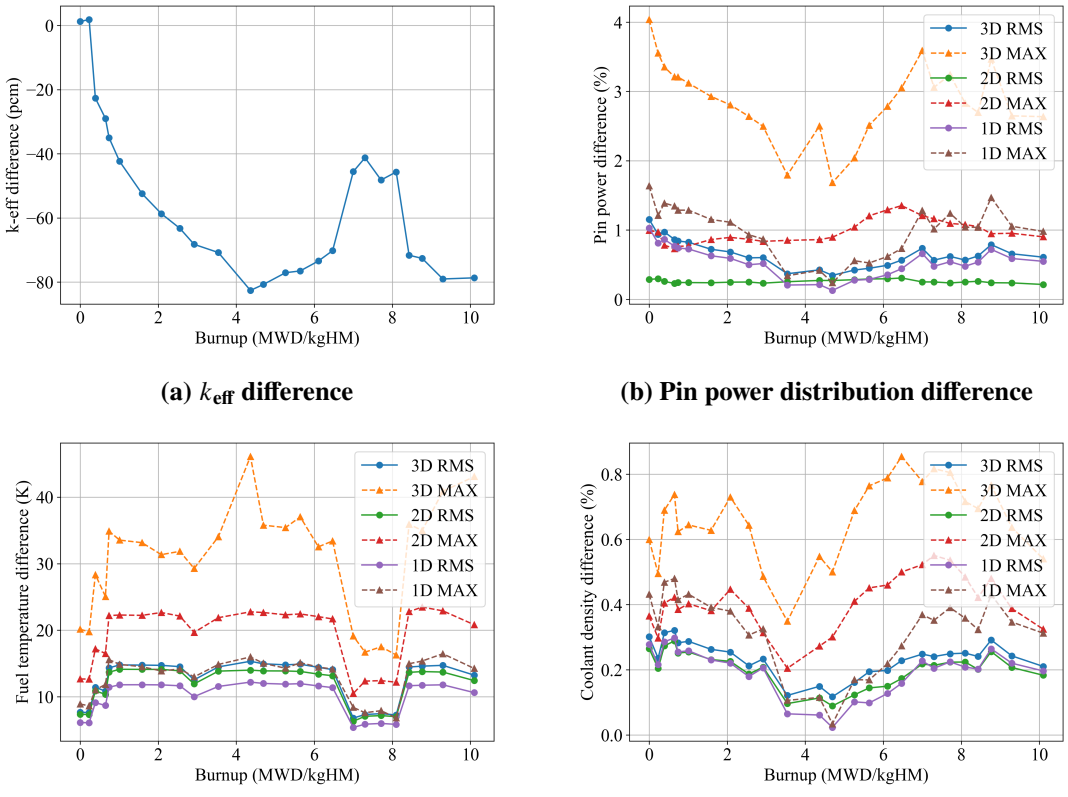
Table 13. PB2 reactor cycle 1 simulation with all options

	Default	All	Change (%)
# of states	25	25	0
# of predictors/correctors	48	48	0
# of XS calculations	1029	734	-29
# of subgroup FSP calls	419	255	-39
# of simplified TH calls	1028	733	-29
# of MGCMFD inners	1269800	182937	-86
# of nodal inners	5160	3425	-34
# of MOC inners	1032	685	-34
# of outer iterations	1032	685	-34
MOC (hours)	7.59	6.33	-17
Nodal (hours)	1.81	1.72	-5
CMFD (hours)	24.47	5.64	-77
TH (hours)	14.20	9.17	-35
Depletion (hours)	0.45	0.33	-27
XS and etc. (hours)	4.88	0.66	-86
Total (hours)	53.40	23.86	-55
Max memory per core (GB)	3.06	2.63	-14
Average memory per core (GB)	2.69	2.24	-17
Total memory (TB)	3.15	2.62	-17

With combining all options, we could reduce the runtime by 55% and the memory by 17% compared to the default case, where the runtime savings are due to the MEDPC, and the memory savings are due to the LS and mixed precision. The runtime of these calculations should be similar to the MEDPC in Table 11 since neither the LS or the mixed precision should increase the runtime. However, the case takes 1 hour more than MEDPC. By looking into the detailed breakdown of the runtime, it is the number of outer iterations of this case that takes 52 iterations more than MEDPC. The transport solution of this case has been reinitialized 2 more times manually, therefore it may be the reason for the more iterations. However, this case still shows 34% fewer iterations than the default case.

The solutions from the default and the new cases are compared in Fig. 8. The k_{eff} difference between the two simulations is about 60 pcm. The case with the new options calculates the underestimated k_{eff} compared to the default case. The average 2D and 3D RMS pin power differences are 0.26% and 0.66%, respectively. The 3D RMS difference for the fuel temperature difference is about 12.5 K.

The MEDPC and the mixed precision do not make a meaningful difference in the solution. The MEDPC and the default CMFD give the same solution within the convergence criteria. The main reason of the difference is due to the LS MOC. As discussed in Section 2.1.4, the default mesh option with the FS MOC does not calculate a sufficiently converged solution in terms of



(a) k_{eff} difference

(b) Pin power distribution difference

(c) Fuel temperature difference

(d) Coolant density difference

Figure 8. Comparison of reactor parameters between the default and the all new options

the mesh discretization. Another reason is that the TH solver assumes the fuel pellet is divided with equal-volume regions. In the LS MOC mesh, the gadolinia fuel is divided into 10 regions with equal-volume. However, the UO_2 fuel has two regions in the radial direction with an inner radius set to 0.875 of the pellet radius. This fractional ratio captures the ^{239}Pu accumulation more accurately with a small number of subregions [6]. This inconsistency between the neutronics and the TH calculations may cause the fuel temperature difference. The new case tends to overestimate the fuel temperature. This overestimated fuel temperature likely contributes to the underestimated k_{eff}. The LS MOC mesh has not been used routinely therefore there are still some issues to resolve in the spatial coupling. In the future, we will address this issue further to use the LS MOC in the production calculation.

4. CONCLUSION AND ONGOING WORK

To improve the efficiency and robustness, and reduce the memory of the MPACT BWR calculation, four enhancements are developed in this milestone, including: 1) adoption of the linear-source MOC; 2) new iterative methods; 3) mixed precision; and 4) problem initialization optimization. The optimization of the geometry and meshing initialization process shows a speedup of 30-40% during problem setup, depending on the number of unique assemblies and control cells. The new iterative methods have shown to speed up the MPACT and TH coupled calculation of the PB2 cycle 1 problem by a factor of 2. The LSA and mixed precision reduce the total memory by 17% for the PB2 problem with a minimum runtime impact.

The run time of the PB2 problem has been successfully reduced by the MEDPC method. However, the high residual (and `NaN`) issues occurred several times and the transport solution has to be reset in order to converge the MEDPC calculation. Therefore, further investigation and testing on more problems is needed to improve the robustness of the MEDPC for BWR applications. Also, we may extend the mixed precision to other variables for additional memory savings. Further gains for the input processing speed up could be found in optimizing the redundancy checking for the creation of each pin mesh. Currently, the check happens after a pin mesh is initialized. If the parameter lists for each pin mesh are stored during initialization and compared before the pin mesh is initialized, more input processing time could be saved.

ACKNOWLEDGEMENTS

We would like to thank Exelon Corporation for initiating this project with Oak Ridge National Laboratory, Idaho National Laboratory, University of Michigan, North Carolina State University, and the University of Illinois Urbana-Champaign under U.S. Department of Energy Contract No. DE-AC05-00OR22725.

This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

REFERENCES

[1] B. Yee, B. Kochunas, and E. Larsen. "A Multilevel in Space and Energy Solver for 3-D Multigroup Diffusion and Coarse-Mesh Finite Difference Eigenvalue Problems." *Nuclear Science and Engineering*, **volume 193**(7), pp. 722–745 (2019).

[2] Q. Shen and B. Kochunas. "A Robust Relaxation-Free Multiphysics Iteration Scheme for CMFD-Accelerated Neutron Transport k-Eigenvalue Calculations I — Theory." *Nuclear Science and Engineering* (2021).

[3] Q. Shen, S. Choi, and B. M. Kochunas. "A Robust Relaxation-Free Multiphysics Iteration Scheme for CMFD-Accelerated Neutron Transport k-Eigenvalue Calculations II — Numerical Results." *Nuclear Science and Engineering* (2021).

[4] R. M. Ferrer and J. D. Rhodes III. "A linear source approximation scheme for the method of characteristics." *Nuclear Science and Engineering*, **volume 182**(2), pp. 151–165 (2016).

[5] G. Gunow, B. Forget, and K. Smith. "Full core 3D simulation of the BEAVRS benchmark with OpenMOC." *Annals of Nuclear Energy*, **volume 134**, pp. 299–304 (2019).

[6] A. Fitzgerald. *Parallel 3-D Method of Characteristics with Linear Source and Advanced Transverse Integration*. Ph.D. thesis, University of Michigan (2020).

[7] A. Fitzgerald, B. Kochunas, and T. Downar. "Improved Formulation of the Method of Characteristics with Linear Source for 2D/1D and Multiphysics Calculations." In *M&C 2019*. Portland, Oregon, USA (2019).

[8] G. W. Stewart. *Afternotes on numerical analysis*. Society for Industrial and Applied Mathematics (1996).

[9] B. Collins, S. Stimpson, B. W. Kelley, M. T. Young, B. Kochunas, A. Graham, E. W. Larsen, T. Downar, and A. Godfrey. "Stability and accuracy of 3D neutron transport simulations using the 2D/1D method in MPACT." *Journal of Computational Physics*, **volume 326**, pp. 612–628 (2016).

[10] S. G. Stimpson. *An Azimuthal, Fourier Moment-Based Axial SN Solver for the 2D/1D Scheme*. Ph.D. thesis, University of Michigan (2015).

[11] M. Jarrett. *A 2D/1D Neutron Transport Method with Improved Angular Coupling*. Ph.D. thesis, University of Michigan (2018).

[12] B. W. Kelley and E. W. Larsen. "A consistent 2D/1D approximation to the 3D neutron transport equation." *Nuclear Engineering and Design*, **volume 295**, pp. 598–614 (2015). URL https://www.sciencedirect.com/science/article/pii/S0029549315002964.

[13] Z. Liu, C. Zhao, L. Cao, H. Wu, and L. Cao. "The material-region-based 2D/1D transport method." *Annals of Nuclear Energy*, **volume 128**, pp. 1–11 (2019).

[14] N. Choi and H. G. Joo. "Stability enhancement of planar transport solution based whole-core calculation employing augmented axial method of characteristics." *Annals of Nuclear Energy*, **volume 143**, p. 107440 (2020).

[15] M. Jarrett, B. Kochunas, E. Larsen, and T. Downar. "Improved Accuracy in the 2-D/1-D

Method with Anisotropic Transverse Leakage and Cross-Section Homogenization." *Nuclear Science and Engineering*, **volume 192**(3), pp. 219–239 (2018).

[16] M. Hursin. *Full core, heterogeneous, time dependent neutron transport calculations with the 3D code DeCART*. Ph.D. thesis, University of California, Berkeley (2010).

[17] N. F. Herring, A. Fitzgerald, B. Kochunas, and T. Downar. "Improved Anisotropic Linear Source Formulation for Multiphysics Problems." In *PHYSOR 2020 – Transition to a Scalable Nuclear Future*. Cambridge, UK (2020).

[18] N. Larsen. "Core Design and Operating Data for Cycles 1 and 2 of Peach Bottom 2." Technical report, General Electric Co., San Jose, CA (USA). Nuclear Energy Engineering Div. (1978).

[19] J. Solis, K. N. Ivanov, B. Sarikaya, A. M. Olson, and K. W. Hunt. "Boiling water reactor turbine trip (TT) benchmark. Volume I: Final Specifications." Technical report, Organisation for Economic Co-operation and Development (2001).

[20] C. Lawing, S. Palmtag, and M. Asgari. "BWR Progression Problems." Technical Report Draft, Oak Ridge National Laboratory (2020). URL https://github.com/cdlawing1/BWRProgressionProblemInputs.

[21] T. Downar, B. Kochunas, Y. Liu, B. Collins, and S. Stimpson. "MPACT Verification and Validation Manual (rev 4)." Technical Report CASL-U-2018-1641-000 (2018).

[22] S. Choi, A. Khassenov, and D. Lee. "Resonance self-shielding method using resonance interference factor library for practical lattice physics computations of LWRs." *Journal of Nuclear Science and Technology*, **volume 53**(8), pp. 1142–1154 (2016).

[23] Y. Liu, W. Martin, M. Williams, and K.-S. Kim. "A full-core resonance self-shielding method using a continuous-energy quasi–one-dimensional slowing-down solution that accounts for temperature-dependent fuel subregions and resonance interference." *Nuclear Science and Engineering*, **volume 180**(3), pp. 247–272 (2015).

[24] S. Choi. *Pin-based Pointwise Energy Slowing-down Method for Resonance Self-shielding Calculation*. Ph.D. thesis, Ulsan National Institute of Science and Technology (2017).

[25] Y. Liu and W. R. Martin. "Assessment of homogeneous and heterogeneous resonance integral tables and their applications to the embedded self-shielding method." *Annals of Nuclear Energy*, **volume 92**, pp. 186–197 (2016).

[26] H. Park and H. G. Joo. "Practical resolution of angle dependency of multigroup resonance cross sections using parametrized spectral superhomogenization factors." *Nuclear Engineering and Technology*, **volume 49**(6), pp. 1287–1300 (2017).

[27] K. S. Kim, M. L. Williams, D. Wiarda, and K. T. Clarno. "Development of the multigroup cross section library for the CASL neutronics simulator MPACT: Method and procedure." *Annals of Nuclear Energy*, **volume 133**, pp. 46–58 (2019).

[28] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, L. Curfman McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. "PETSc Users Manual." Technical Report ANL-95/11 - Revision 3.13, Argonne National Laboratory (2015). URL https:

//www.mcs.anl.gov/petsc.

[29] M. Baboulin, A. Buttari, J. Dongarra, et al. "Accelerating scientific computations with mixed precision algorithms." *Computer Physics Communications*, **volume 180**(12), pp. 2526–2533 (2009).