IMPROVEMENT OF TRANSPORT-CORRECTED SCATTERING STABILITY AND PERFORMANCE USING A JACOBI INSCATTER ALGORITHM FOR 2D-MOC

Shane Stimpson,^{1*} Benjamin Collins,¹ and Brendan Kochunas²

¹Oak Ridge National Laboratory

1 Bethel Valley Rd., Oak Ridge, Tennessee 37831, USA

stimpsonsg@ornl.gov; collinsbs@ornl.gov

²Department of Nuclear Engineering and Radiological Sciences

University of Michigan

1906 Cooley Bldg., 2355 Bonisteel Blvd., Ann Arbor, Michigan 48109, USA

bkochuna@umich.edu

***Corresponding Author**

Pages: 35

Tables: 5

Figures: 8

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the US Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States 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 the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

ABSTRACT

The MPACT code, being developed collaboratively by the University of Michigan and Oak Ridge National Laboratory, is the primary deterministic neutron transport solver being deployed within the Virtual Environment for Reactor Applications (VERA) as part of the Consortium for Advanced Simulation of Light Water Reactors (CASL). In many applications of the MPACT code, transport-corrected scattering has proven to be an obstacle in terms of stability, and considerable effort has been made to try to resolve the convergence issues that arise from it. Most of the convergence problems seem related to the transport-corrected cross sections, particularly when used in the 2D method of characteristics (MOC) solver, which is the focus of this work.

In this paper, the stability and performance of the 2-D MOC solver in MPACT is evaluated for two iteration schemes: Gauss-Seidel and Jacobi. With the Gauss-Seidel approach, as the MOC solver loops over groups, it uses the flux solution from the previous group to construct the inscatter source for the next group. Alternatively, the Jacobi approach uses only the fluxes from the previous outer iteration to determine the inscatter source for each group. Consequently for the Jacobi iteration, the loop over groups can be moved from the outermost loop—as is the case with the Gauss-Seidel sweeper—to the innermost loop, allowing for a substantial increase in efficiency by minimizing the overhead of retrieving segment, region, and surface index information from the ray tracing data.

Several test problems are assessed: (1) Babcock & Wilcox 1810 Core I, (2) Dimple S01A-Sq, (3) VERA Progression Problem 5a, and (4) VERA Problem 2a. The Jacobi iteration exhibits better stability than Gauss-Seidel, allowing for converged solutions to be obtained over a much wider range of iteration control parameters. Additionally, the MOC solve time with the Jacobi approach is roughly 2.0-2.5 $\times$ faster per sweep. While the performance and stability of the Jacobi iteration are substantially improved compared to the Gauss-Seidel iteration, it does yield a roughly 8–10% increase in the overall memory requirement.

Key Words: transport correction, MOC, jacobi, gauss-seidel, stability

I. INTRODUCTION

The MPACT code [1] being developed collaboratively by the University of Michigan and Oak Ridge National Laboratory (ORNL) is the primary deterministic neutron transport solver being deployed within the Virtual Environment for Reactor Applications (VERA) [2] as part of the Consortium for Advanced Simulation of Light Water Reactors (CASL) [3]. While there are several different solver options within MPACT, the most commonly used option to solve 3D problems is the 2D/1D method [4, 5, 6], which combines 2D method of characteristics (MOC) and 1D-nodal solutions, coupled through axial and radial transverse leakages. In many applications of the MPACT code, transport-corrected scattering has proven to be an obstacle in terms of stability, and considerable effort has been expended to try to resolve the convergence issues that arise from it. Most, if not all, of the convergence problems seem to be related to the 2D-MOC solver, which is the focus of this work.

Convergence issues with transport-corrected scattering are a long-standing issue and are not specific to MPACT [7, 8]. While it is possible that cross section library generation can lead to unphysical/negative solutions, particularly where there is a large variation in the energy group size, in many instances the convergence issues seem to be related to the path to the solution more than the final solution itself. One proposed solution for instabilities caused by the transport correction from Tabuchi and Yamamoto [8] has been to apply a relaxation factor during the MOC inner iterations, which has proven to be moderately successful, but it requires several inner iterations for MOC, which can be fairly costly. Other potential solutions, such as solving several iterations with positive-definite isotropic cross sections then incorporating the negative, transport-corrected scattering terms, have also been evaluated with mixed success. Accelerating the boundary conditions going into the 2D-MOC calculation [9] helps provide stability by

providing a better initial guess of the incoming angular fluxes. However, none of these efforts has sufficiently resolved the instabilities observed.

One of the more recent attempts to address the convergence issues has been focused on using a Jacobi inscatter iteration in which the inscatter source for each group is constructed based on the flux values from the previous outer iteration compared to a Gauss-Seidel approach, where the most recent flux solution data are used to determine the inscatter source as the groups are being swept. It has been observed that convergence is obtained much more reliably, particularly with fewer numbers of inner iterations. Additionally, using a Jacobi approach allows for the loop over groups to move from the outermost to close to the innermost, allowing for a notable improvement in efficiency by avoiding duplicate work to connect the modular ray data. This type of approach is more consistent with CASMO [10] and OpenMOC [11], but the impact on convergence and stability has not been evaluated in these codes. Furthermore, since the scattering source is lagged with the Jacobi approach, it indirectly relaxes the solution and likely falls somewhat in line with the relaxation approach proposed by Tabuchi and Yamamoto [8].

When transport-corrected scattering causes instabilities, it seems that the inscatter and fission source for a particular group is insufficiently large to accommodate a negative contribution from the self-scattering source, driving the total source negative. While not the focus of this work, in 2D/1D calculations, an axial transverse leakage contributes to the total source, and a negative leakage can further exacerbate any instabilities. If the incoming angular flux from the MOC solver is small enough, a negative segment angular flux (and by extension, scalar flux) can be encountered. As this is unphysical, it can lead to substantial problems in the iteration scheme. In the case of the Jacobi algorithm, the effective relaxation allows the inscatter source to develop in a more controlled manner. As it is sweeping through the groups, the Gauss-

Seidel approach may be using poor estimates of the source by using the most up-to-date values, and the errors in these sources could compound into convergence problems.

In Section II, the governing theory behind MOC is presented, setting up the discussion comparing the Gauss-Seidel and Jacobi sweeping algorithms. Then some general background on transport-correction will be given, highlighting the techniques being applied in the current 47-group library used in MPACT. To wrap up the theory section, the details behind the critical buckling implementation will be provided, as this comes into play in the results section, where two 2D critical benchmarks are assessed. In general, it has been observed that the convergence for cases with critical buckling specification have been particularly sensitive. In Section III, the convergence behavior of several cases is investigated, starting with the B&W 1810 Core 1 critical and the Dimple S01A-Sq critical. Then, a 2D quarter core model known as VERA Progression Problem 5a-2D will be assessed, analyzing both the convergence and performance. Lastly, a single quarter lattice (VERA Problem 2a) will be used to provide a more consistent performance comparison for other codes. Once all of the test cases have been explored, it will be seen that the Jacobi algorithm provides more robust convergence behavior, but it does generally incur additional outer iterations as a result of the inscatter information being lagged. However, the improved efficiency of the MOC kernel compensates for this. Given this added reliability and performance, it has been adopted as the default capability for 2D-MOC in MPACT.

II. GOVERNING THEORY

II.A The Method of Characteristics (MOC)

MOC is a commonly used solver for the multigroup Boltzmann neutron transport equation (Eq. 1) [4, 11, 12, 13, 14, 15],

$$\underline{\Omega} \cdot \nabla \varphi(\underline{r}, \underline{\Omega}) + \Sigma_{t,g}(\underline{r}) \varphi_g(\underline{r}, \underline{\Omega}) = q_g(\underline{r}),$$

$$q_g(\underline{r}) = \frac{\chi_g(\underline{r})}{4\pi k_{\text{eff}}} \sum_{g'=1}^{N_{grp}} \nu \Sigma_{f,g'}(\underline{r}) \int_0^{4\pi} \varphi_{g'}(\underline{r}, \underline{\Omega}') d\Omega' + \sum_{g'=1}^{N_{grp}} \int_0^{4\pi} \Sigma_{s,g' \rightarrow g}(\underline{r}, \underline{\Omega}' \cdot \underline{\Omega}) \varphi_{g'}(\underline{r}, \underline{\Omega}') d\Omega', \quad (1)$$

and there are a number of different approaches, such as flat source and linear source [16, 17]. In Equation (1), $\underline{r}$ represents the spatial vector, $\underline{\Omega}$ represents the angular vector that is comprised of both an azimuthal and polar angle, g denotes the neutron energy group index, and q_g is the source term which contains both the fission and scattering terms. The method can be applied in 1D [18], 2D, and 3D [19, 20, 21, 22] applications. The focus of the MOC equations in this paper is on flat source, 2D-radial applications. Above, the scattering term still preserves full angular dependence. In this work, however, focus is exclusively given to transport-corrected scattering which uses an isotropic scattering kernel (Eq. 2):

$$\Sigma_{s,g' \rightarrow g}(\underline{r}, \underline{\Omega}' \cdot \underline{\Omega}) \approx \frac{\Sigma_{s0,g' \rightarrow g}(\underline{r})}{4\pi}. \quad (2)$$

Substituting Eq. (2) into Eq. (1) yields a modified source term (Eq. 3) for isotropic scattering which now can be evaluated using the scalar flux (ϕ_g),

$$q_g(\underline{r}) = \frac{\chi_g(\underline{r})}{4\pi k_{\text{eff}}} \sum_{g'=1}^{N_{grp}} \nu \Sigma_{f,g'}(\underline{r}) \phi_{g'}(\underline{r}) + \frac{1}{4\pi} \left(\sum_{g'=1, g' \neq g}^{N_{grp}} \Sigma_{s0,g' \rightarrow g}(\underline{r}) \phi_{g'}(\underline{r}) \right) + \frac{1}{4\pi} \left(\Sigma_{s0,g \rightarrow g}(\underline{r}) \phi_g(\underline{r}) \right), \quad (3)$$

where the scalar flux is simply obtained by integrating the angular flux over all angles (Eq. 4):

$$\phi_{g'}(\underline{r}) = \int_0^{4\pi} \varphi_{g'}(\underline{r}, \underline{\Omega}') d\Omega'. \quad (4)$$

It is worth noting that the last two terms in Eq. 3 are the inscatter and self-scatter sources, respectively, where the self-scatter cross section ($\Sigma_{s0,g \rightarrow g}$) can become negative as a result of the transport-correction. This is discussed in more detail in Section II.C. The MOC can be applied to this problem by first introducing a characteristic direction ($\underline{r}_0 + s\underline{\Omega}$) (Eq. 5):

$$\underline{r} = \underline{r}_0 + s\underline{\Omega} = \begin{cases} x = x_0 + s\Omega_x \\ y = y_0 + s\Omega_y, \\ z = z_0 + s\Omega_z \end{cases} \quad (5)$$

and then casting a 1D version of Eq. (1) along it, as in Equation (6), which is a simple, ordinary differential equation:

$$\frac{d\varphi_g(\underline{r}_0 + s\underline{\Omega}, \underline{\Omega})}{ds} + \Sigma_{t,g}(\underline{r}_0 + s\underline{\Omega})\varphi_g(\underline{r}_0 + s\underline{\Omega}, \underline{\Omega}) = q_g(\underline{r}_0 + s\underline{\Omega}, \underline{\Omega}), \quad (6)$$

where s denotes distance along the ray from the point of origin. By removing the intraregional spatial dependence of the total cross section/source and solving Eq. (6), one can obtain the solution for the angular flux at any point s along the ray (Eq. 7):

$$\varphi_g(s, \underline{\Omega}) = \varphi_g^{in}(\underline{\Omega})e^{-\Sigma_{t,g}s} + \frac{q_g}{\Sigma_{t,g}}(1 - e^{-\Sigma_{t,g}s}). \quad (7)$$

Applying the discrete ordinates approximation and discretizing spatially, Eq. (7) now becomes Eq. (8), where a denotes the angle index, and r denotes the spatial region index:

$$\varphi_{g,a,r} = \varphi_{g,a,r}^{in}e^{-\Sigma_{t,g,r}s_{a,r}} + \frac{q_{g,r}}{\Sigma_{t,g,r}}(1 - e^{-\Sigma_{t,g,r}s_{a,r}}). \quad (8)$$

From this, the outgoing angular flux at the end of the ray (Eq. [9a]) and the average angular flux along the ray (Eq. [9b]) can be obtained.

$$\varphi_{g,a,r}^{out} = \varphi_{g,a,r}^{in}e^{-\Sigma_{t,g,r}s_{l,r}} + \frac{q_{g,r}}{\Sigma_{t,g,r}}(1 - e^{-\Sigma_{t,g,r}s_{a,r}}), \quad (9a)$$

$$\bar{\varphi}_{g,a,r} = \frac{\varphi_{g,a,r}^{out} - \varphi_{g,a,r}^{in}}{\Sigma_{t,g,r} S_{a,r}} + \frac{q_{g,r}}{\Sigma_{t,g,r}}. \quad (9b)$$

The outgoing angular flux ($\varphi_{g,a,r}^{out}$) is then used as the incoming angular flux ($\varphi_{g,a,r}^{in}$) for the next segment in the ray. Alternatively, if the end of the segment is along a system boundary, then the angular flux is determined by the boundary condition. Possible boundary conditions may include vacuum, reflective, rotational, periodic, etc. If the segment terminates along a parallel domain boundary, it will be sent to the neighboring process to be used in the next iteration. The average angular flux ($\bar{\varphi}_{g,a,r}$) is used to tally the scalar flux in each region ($\phi_{g,r}$) which is used to update the source term in Eq. (3).

II.B Gauss-Seidel vs. Jacobi Energy Sweeping Algorithms

In the previous subsection, the multigroup MOC equations were presented. However, in practice, these equations are generally solved one group at a time, looping over the groups as appropriate. There are several ways of doing this. This work focuses on the Gauss-Seidel and Jacobi group sweeping algorithms. In the Gauss-Seidel approach, as one group is swept and a new flux solution is obtained for group g , it is then used to determine the new inscatter source, represented by the second term in Eq. (3), for the next group. Figure 1 outlines a more detailed description of the looping structure of a transport sweep and the tasks performed by each one-group MOC kernel. As can be seen, the loop over energy groups is the outermost loop, and the source is updated as the loop progresses.

```

1: for each group ( $g$  from 1 to  $N_{groups}$ )
2:   Setup source for group  $g$  (using updated flux solution from previous groups)
3:   for each inner iteration ( $i$  from 1 to  $N_{inners}$ )
4:     for each azimuthal angle ( $a$  from 1 to  $N_{angles}$ )
5:       for each longray in angle  $a$  ( $l$  from 1 to  $N_{longrays}(a)$ )
6:         Connect modular rays and determine coarse mesh surfaces
7:         Calculate exponential values for each segment in longray  $l$  and group  $g$ 
8:         for each polar angle ( $p$  from 1 to  $N_{polar}$ )
9:           Gather incoming angular flux ( $\varphi_{l,p,g}^{in}$ ) at both ends of longray  $l$ 
10:          for each segment ( $s$  from 1 to  $N_{segments}(l)$ )
11:            Evaluate forward direction:
12:            Calculate outgoing angular flux ( $\varphi_{l,p,g}^{out}$ )
13:            Tally contribution to fine mesh scalar flux in region  $r1$  ( $\phi_{g,r1}$ )
14:            Evaluate backward direction:
15:            Calculate outgoing angular flux ( $\varphi_{l,p,g}^{out}$ )
16:            Tally contribution to fine mesh scalar flux in region  $r2$  ( $\phi_{g,r2}$ )
17:          end for
18:          Store outgoing angular flux ( $\varphi_{l,p,g}^{out}$ )
19:          for each coarse mesh surface intersection ( $c$  from 1 to  $N_{cm}$ )
20:            Tally surface flux and currents for coarse mesh surface  $c$ , group  $g$ 
21:          end for
22:        end for
23:      end for
24:    end for
25:  end for
26: end for

```

Fig 1. Pseudocode for Gauss-Seidel Sweeping with Group on Outermost Loop.

Alternatively, the multigroup flux used to construct the inscatter sources can be lagged, creating a Jacobi-like algorithm in energy. It would be possible to create this in a scheme similar to that outlined with Figure 1 by simply moving the source setup outside of the loops entirely. However, some performance can be gained by moving this to the innermost loop (as shown in Figure 2). With this scheme, the work performed to access the ray tracking data is only performed once, whereas it has to occur once for each group with the Gauss-Seidel approach. As can be seen, potential performance benefits of the Jacobi algorithm yielding a multigroup kernel could be worthwhile as long as the time to set up the modular ray connections is a non-negligible fraction of the total runtime.

```

1: Setup source for all groups
2: for each inner iteration ( $i$  from 1 to  $N_{inners}$ )
3:   for each azimuthal angle ( $a$  from 1 to  $N_{angles}$ )
4:     for each longray in angle  $a$  ( $l$  from 1 to  $N_{longrays}(a)$ )
5:       Connect modular rays and determine coarse mesh surfaces
6:       Calculate exponential values for each segment in longray  $l$  and all groups
7:       for each polar angle ( $p$  from 1 to  $N_{polar}$ )
8:         Gather incoming angular flux ( $\varphi_{l,p,g}^{in}$ ) at both ends of longray  $l$ 
9:         for each segment ( $s$  from 1 to  $N_{segments}(l)$ )
10:          for each group (1 to  $N_{groups}$ )
11:            Evaluate forward direction:
12:            Calculate outgoing angular flux ( $\varphi_{l,p,g}^{out}$ )
13:            Tally contribution to fine mesh scalar flux in region  $r1$  ( $\phi_{g,r1}$ )
14:            Evaluate backward direction:
15:            Calculate outgoing angular flux ( $\varphi_{l,p,g}^{out}$ )
16:            Tally contribution to fine mesh scalar flux in region  $r2$  ( $\phi_{g,r2}$ )
17:          end for
18:        end for
19:        Store outgoing angular flux ( $\varphi_{l,p,g}^{out}$ )
20:        for each coarse mesh surface intersection ( $c$  from 1 to  $N_{cm}$ )
21:          Tally surface flux and currents for coarse mesh surface  $c$ , group  $g$ 
22:        end for
23:      end for
24:    end for
25:  end for
26: end for

```

Fig 2. Pseudocode for Jacobi Sweeping with Group on Innermost Loop.

There is also the potential to sweep over the groups multiple times, particularly over the energy groups with upscatter cross section data. In this case there would be an additional loop in both figures to account for this. In this work, the Jacobi sweep still only uses the fluxes at the beginning of the outer iteration to update the sources, so subsequent “upscatter” iterations simply perform more inner iterations over the thermal groups. When 0 upscatter iterations are used in the results section, it means that only a single pass through the groups was executed.

II.C Transport-Corrected Scattering (TCP₀)

Transport-corrected scattering is a commonly used technique that aims to provide P₁ accuracy to an isotropic scattering kernel, which is computationally cheaper than explicit P_N scattering.

There are a variety of ways to develop a transport correction [7], but this section will only detail the ones incorporated into the 47-group library developed by ORNL [23] that MPACT uses: out-scatter, in-scatter, and a neutron leakage conservation (NLC) approximation.

The out-scatter approximation is one of the easiest to implement, as it does not require any *a priori* solution data to correct the cross sections. It simply subtracts the sum of the first moment out scatter cross sections (from group g to g') from the total and self-scattering cross sections (as in Eq. [10]):

$$\begin{aligned}\Sigma_{tr,g} &= \Sigma_{t,g} - \sum_{g'=1}^{N_{grp}} \Sigma_{s1,g \rightarrow g'}, \\ \Sigma_{s0,g \rightarrow g} &= \Sigma_{s0,g \rightarrow g} - \sum_{g'=1}^{N_{grp}} \Sigma_{s1,g \rightarrow g'}.\end{aligned}\tag{10}$$

While this generally leads to acceptable accuracy for heavier isotopes, lighter isotopes require an alternative formulation; it is common to use an inscatter approximation instead. There are also several different ways to formulate an inscatter correction; the MPACT library uses an approximation based on the P_1 moment equation as developed by Seoul National University [24], per Eq. (11):

$$\begin{aligned}\frac{1}{3}\phi_g - \Sigma_{t,g}D_g\phi_g &= - \sum_{g'=1}^{N_{grp}} \Sigma_{s1,g' \rightarrow g}D_{g'}\phi_{g'}, \\ \Sigma_{tr,g} &= \frac{1}{3D_g}.\end{aligned}\tag{11}$$

where a representative flux spectrum (ϕ_g) is provided, and the diffusion coefficient can be obtained. This type of correction is applied only to carbon and oxygen isotopes in the library.

An additional correction termed the *neutron leakage conservation* (NLC) approximation [25] is used only for hydrogen. The hydrogen treatment is of paramount concern for accuracy and stability, as it is the most dominate scattering isotope in a light water reactor. Many of the approximations presented can yield negative self-scattering cross sections, particularly as the energy group structure is refined, which can lead to numerical problems. With the NLC correction proposed by Herman, a uniform hydrogen slab buckling problem is solved per Eq. (12), where B^2 is the buckling, and W is the length of the slab (taken to be 100 cm):

$$J_{right,g} - J_{left,g} = D_g B^2 \phi_g W. \quad (12)$$

The diffusion coefficient and transport cross section (as in Eq. 13) can be solved by tallying the currents at the left and right tally regions of the slab, which are offset from the ends of the slab to avoid boundary effects:

$$\Sigma_{tr,g} = \frac{1}{3D_g} = \frac{B^2 \phi_g W}{3(J_{right,g} - J_{left,g})}. \quad (13)$$

While improving accuracy, the NLC correction also substantially improves the negative cross sections in hydrogen (as in Figure 3), but it does not eliminate them.

Fig 3. Cross Section Plot for Hydrogen with NLC Correction at 565 K.

As seen in Fig. 3, while the isotropic self-scattering cross sections are all positive, applying the transport correction yields negative cross section in many of the groups. It is worth noting that many of the fast groups have only slight corrections, which is typically not the case with other techniques (such as outscatter), which can have considerably more convergence issues. In theory, the negative cross sections in the lower energy groups should have a sufficient inscatter source to accommodate a negative self-scatter source. However, this is not necessarily guaranteed at every iteration in the calculation, which is why stability issues remain.

II.D Critical Buckling Treatment

Some of the cases to be analyzed in the next section deal with 2D critical experiments in which a buckling value is prescribed. Instead of explicitly modelling the experiments in 3D, the problems are simulated in 2D, assuming a uniform geometric configuration axially that is well approximated using buckling theory. With this, a $D_g B^2$ term is added to the transport cross

section. This can be applied a number of ways. For example, one option would be to calculate a diffusion coefficient (D_g) for each fine mesh region, applying a unique buckling term for each region. Alternatively, a core-averaged diffusion coefficient could be determined and applied such that each region receives the same adjustment. In the current implementation in MPACT, the critical buckling is applied using diffusion coefficients tabulated on the pin cell basis, which is the same mesh used for the coarse mesh finite difference (CMFD) acceleration.

III. RESULTS

To demonstrate the stability and performance of the new Jacobi algorithm, results are compared to Gauss-Seidel for several problems. The first two are Core I from the Babcock and Wilcox 1810 series of criticals [26] and Dimple S01A-Sq [27, 28], both of which are simulated in 2D using a specified critical buckling value. These are of interest because transport correction issues have been observed consistently and are likely attributable to the reasonably large reflector regions around the active core. The ability to predict these critical conditions accurately is also important for assessing the accuracy of the transport correction and validating the cross section library and transport methodology. Next, a 2D quarter core based on the Watts Bar Unit 1 reactor (VERA Progression Problem 5 [29]) is evaluated, taking note of the MOC performance. Lastly, a single quarter lattice (VERA Progression Problem 2 [29]) is shown to further assess the performance. No transport-correction convergence issues are observed on lattice cases, but analysis of the serial performance on a lattice case is useful for obtaining more consistent comparisons to other codes.

III.A Babcock & Wilcox (B&W) 1810 – Core I

The first case for analysis is the B&W 1810 Core I critical configuration [26] shown in Figure 4. It contains 2.46% fuel pins in a configuration with several vacant water-filled locations, with roughly 45 cm of borated water (1337.9 ppm) reflector surrounding the core. Not included in the legend, but also in the model, are the helium fuel-clad gap material and components of the rhodium detector in the central rod location (Inconel, stainless steel, and rhodium). For this analysis, a 0.05 cm ray spacing was used with 16 azimuthal and 3 polar angles in a Tabuchi-Yamamoto quadrature, and a buckling value of $4.1 \times 10^{-4} \text{ cm}^{-2}$ was specified.

Fig 4. Core Layout for B&W 1810 – Core I [26].

Table I shows the outer iterations and timing results to converge each problem, varying the number of upscatter and inner iterations. As can be observed, the Gauss-Seidel algorithm is very unforgiving in terms of convergence, with only one configuration (0 upscatter and 2 inner iterations) yielding a converged solution. However, the Jacobi algorithm is considerably more stable, particularly with fewer iterations. Eventually, though, a sufficient number of iterations will in fact cause divergence, though it uses significantly more inner iterations than is practically necessary. Recent work to incorporate an optimally diffusive CMFD (odCMFD) approximation has also improved stability with reduced numbers of inner iterations, which was a problem even with positive, non-transport-corrected cross sections [30].

TABLE I.
1810 Core I Results

MOC Iterations Upscatter-Inner	Gauss-Seidel			Jacobi		
	Iterations	MOC Time (min)	Total Time (min)	Iterations	MOC Time (min)	Total Time (min)
0-1	Div.	Div.	Div.	14	2.33	5.63
0-2	11	4.86	9.29	10	3.16	6.24
0-3	Div.	Div.	Div.	10	4.19	7.28
0-4	Div.	Div.	Div.	10	5.94	9.25
0-5	Div.	Div.	Div.	Div.	Div.	Div.

1-1	Div.	Div.	Div.	12	3.69	7.87
1-2	Div.	Div.	Div.	9	5.30	8.25
1-3	Div.	Div.	Div.	Div.	Div.	Div.
1-4	Div.	Div.	Div.	Div.	Div.	Div.
1-5	Div.	Div.	Div.	Div.	Div.	Div.

It is also important to note the trends in outer iteration counts and timing. Surprisingly, for the one case where both converge, Jacobi actually requires one less outer iteration. Additionally, as expected, increasing the number of inner iterations yields fewer overall outer iterations but at an increased cost in terms of runtime.

III.B Dimple S01A-Sq

The next problem being investigated is the Dimple S01A-Sq core (Figure 5) [27, 28]. As in B&W 1810 Core I, a 0.05 cm ray spacing was used with 16 azimuthal and 3 polar angles in a Tabuchi-Yamamoto quadrature. However, a considerably larger buckling value of 24.2×10^{-4} was specified.

Fig 5. Core Layout for Dimple S01A-Sq [28]

Table II shows the results for Dimple S01A-Sq, varying the number of upscatter and inner iterations. Fairly similar results are observed as in 1810 Core I, but with several more points of convergence (0–2, 0–4, and 1–2 for GS and all points for J).

TABLE II.
Dimple S01A-Sq Results

MOC Iterations Upscatter-Inner	Gauss-Seidel			Jacobi		
	Iterations	MOC Time (min)	Total Time (min)	Iterations	MOC Time (min)	Total Time (min)
0-1	Div.	Div.	Div.	24	0.58	2.05
0-2	14	0.89	2.89	18	0.71	2.94
0-3	Div.	Div.	Div.	16	1.00	3.06
0-4	12	1.21	3.10	16	1.32	3.62
0-5	Div	Div.	Div.	16	1.56	3.96
1-1	Div	Div.	Div.	14	0.49	2.59
1-2	8	0.89	2.58	11	0.75	2.71
1-3	Div	Div.	Div.	10	0.97	2.88
1-4	Div.	Div.	Div	10	1.32	3.33
1-5	Div.	Div.	Div.	10	1.60	3.64

Not surprisingly, though the Jacobi iteration approach provides more stability, it does incur more outer iterations and generally more total computation time. While the MOC performance is

improved, as will be detailed more in the coming results, the performance of the CMFD solver becomes more dominant, so the additional outer iterations yield notably more CMFD and overall time. Ongoing work to improve the CMFD runtime has proven to be highly effective [31], which will significantly alleviate any performance penalty incurred by the Jacobi approach.

III.C VERA Progression Problem 5a-2D

The last case to be evaluated for stability is the quarter core 2D slice problem known as VERA Progression Problem 5a-2D [29]. The core layout for this problem is shown in Figure 6, where it can be seen that there are three different fuel enrichments (2.1%, 2.6%, and 3.1%) in a ring-of-fire configuration. There are also several different configurations of Pyrex rods; the number in each assembly is indicated in the figure. Also included in the model but not shown in the figure are the 2.85 cm stainless steel baffle and the jagged reflector region that is approximately one assembly thick. As in the previous cases, the reflector region yields to most convergence problems. In fact, using a jagged reflector as in this model is very useful in mitigating these effects, but problems still arise.

Fig 6. Core Layout of VERA Problem 5a-2D

Table III shows the stability results for 5a-2D, varying the number of upscatter and inner iterations. These were obtained using a 0.05 cm ray spacing, 16 azimuthal angles, and 2 polar angles in a Tabuchi-Yamamoto quadrature and executed on 16 processors. As before, the Gauss-Seidel iteration scheme is much more sensitive to the number of inner iterations than the Jacobi iteration scheme. The performance improvement from the Jacobi kernels, plus complementary changes allowing the subgroup solver use the kernel [32, 33], allows the overall runtime to be

reduced despite the additional outer iterations. Lastly, there is one data point 2–4, in which Gauss-Seidel converges but Jacobi does not. While this is surprising, it is less concerning since it is not desirable to run that many iterations.

TABLE III.
P5-2D Results

MOC Iterations Upscatter-Inner	Gauss-Seidel			Jacobi		
	Iterations	MOC Time (min)	Total Time (min)	Iterations	MOC Time (min)	Total Time (min)
0–1	Div.	Div.	Div.	13	3.22	16.79
0–2	8	8.42	31.26	10	4.83	18.88
0–3	Div.	Div.	Div.	10	6.87	20.68
0–4	6	11.38	31.58	10	9.28	22.85
0–5	Div.	Div.	Div.	10	11.26	21.52
1–1	Div.	Div.	Div.	6	3.10	12.77
1–2	6	11.10	33.67	6	5.54	14.96
1–3	Div.	Div.	Div.	6	8.37	16.82
1–4	5	15.75	33.21	Div.	Div.	Div.
1–5	Div.	Div.	Div.	Div.	Div.	Div.

Table IV shows the timing results exclusively from the MOC solver from 5a-2D run on Titan [34]. The results are normalized per iteration to allow for a more equal comparison of the kernel efficiencies. These were obtained using 1 inner iteration with P_0 scattering to allow the Gauss-Seidel solver to converge. However, both P_0 and transport-corrected scattering use the same MOC kernel, so the performance is directly comparable. It is also important to note that when performing multiple inner iterations, two different MOC kernels are used. The last inner iteration uses a kernel that tallies to currents on the coarse mesh surfaces to be used in the subsequent CMFD iteration, whereas all previous inner iterations use a kernel without these tallies, since tallying the currents is not necessary until the end. Therefore, in Table IV, where one inner iteration is used, the performance comparison is for the kernel with the current tallies.

Table IV.

MOC Solver Timing Results (core-min/iteration) for Problem 5a-2D on Titan

73 Spatial Domains			257 Spatial Domains		
GS	J	Ratio	GS	J	Ratio
14.61	5.94	2.45	11.87	5.02	2.36

The Jacobi sweeper shows a considerable improvement in the MOC runtime. Much of this is attributable to the savings introduced in the multigroup kernel allowing the modular rays to be connected only once. However, additional improvements are gained by allowing the data communication between each spatial domain to be performed on a multigroup basis, whereas the Gauss-Seidel approach is passing the data one group at a time as it is sweeping through the

groups. This improvement allows the angular fluxes to be stored more optimally for the kernel. Overall, a roughly 2.3–2.5 $\times$ reduction is observed in the MOC solve time.

III.D Performance for Lattice Cases – VERA Progression Problem 2a

The last case evaluated is VERA Progression Problem 2a, which is a simple 2D, 17×17 quarter lattice case with 3.1% enriched fuel rods (Fig 7). Typically, transport-corrected convergence issues are not observed on lattice cases, so this case is primarily demonstrated for further comparison of the performance, focusing on serial execution, which can provide for a more consistent comparison to other lattice codes. As in previous cases, a 0.05 cm ray spacing, 16 azimuthal angles, and 3 polar angles (TY) were used.

Fig 7. Geometry Visualization of VERA Problem 2a [29].

Table IV shows the MOC solver timing results for this problem tested on two different machines, one with 2.3 GHz AMD processors (OpertonTM Processor 6376), and the other with 3.2 GHz Intel processors (Xeon® CPU E5-1650v3). Instead of showing the overall time of the solver as reported previously, Table IV shows the nanoseconds per integration (ns/integration), where an integration is considered to be the evaluation of the MOC equations along a single segment. Summing over all angles (azimuthal and polar), Problem 2a has approximately 715,000

segments. Additionally, the table includes a comparison of the kernels with and without current tallies, which— are used as appropriate with multiple inner iterations.

Table IV.

MOC Solver Timing Results (ns/integration) for Problem 2a

Kernel	AMD (2.3 GHz)			Intel (3.2 GHz)		
	GS	J	Ratio	GS	J	Ratio
Without Currents	7.39	5.38	1.37	3.52	2.92	1.21
With Currents	15.65	5.78	2.71	6.58	3.25	2.03

The difference between the Gauss-Seidel and Jacobi approaches is larger for the kernel with current tallies than without. In addition to connecting the modular ray data, the mapping between the rays and the coarse mesh surfaces also must be completed. With the multigroup Jacobi kernel, this can be performed once, whereas the Gauss-Seidel kernel will need to repeat the burdensome mapping for each group. Second, there is not a huge difference in efficiencies between the Jacobi kernels with and without current tallies because of the reduced work duplication, so more time is spent performing the actual integrations, which are consistent between the two. Lastly, the difference in ns/integration between the AMD and Intel cores relates fairly well to the clock speed differences between the two, but even accounting for this, the Intel cores notably outperform the AMD cores. On both cores, greater than $2\times$ improvement in the efficiency is observed.

III.E Assessment of the Rate of Convergence

From the overview of the Gauss-Seidel and Jacobi algorithms, it is expected that the convergence rate of Jacobi should be markedly lower than for Gauss-Seidel as the most recent scattering source information is lagged. However, with CMFD acceleration, this effect is diminished, and only a few extra iterations are necessary. To evaluate and compare these two algorithms in more detail, the Problem 2a case was run without CMFD, just using source iteration to converge the eigenvalue and flux distributions. Additionally, isotropic scattering was used to avoid stability issues and provide consistent comparisons of the two sweep algorithms. Figure 8 shows the results for this test, which includes both an eigenvalue and a fission source (FS) residual for each iteration.

Fig 8. Eigenvalue and Fission Source Residuals for Problem 2a with Source Iteration.

Here the eigenvalue residual is simply the difference between the eigenvalue of successive iterations, and the fission source residual is a 2-norm of the difference between successive fission source distributions. A convergence criteria of 10^{-6} was used for both of these

residuals. In both the cases, the eigenvalue residual was the most limiting, and both cases terminated once that criteria was met.

It can be seen that the Jacobi algorithm takes roughly 40% more iterations (177 vs. 125). The asymptotic rate of convergence (after roughly 40 iterations) is noticeably lower for Jacobi. Estimating the spectral radius from the fission source residual yields 0.946 with Jacobi vs 0.919 with Gauss Seidel. This is similar to what would be observed if a relaxation factor were applied, as in the approach proposed by Tabuchi and Yamamoto [8]. In a sense, because Jacobi lags the scattering source, the solution is effectively relaxed, providing additional stability.

III.F Memory Considerations

While what has been seen of the Jacobi approach has been positive in terms of stability and performance, it does come at a marginal cost in terms of memory. To move the group index of the MOC loop to create a multigroup kernel, additional storage is necessary. With the Gauss-Seidel kernel, only one group of source data is necessary at a time, and only the incoming angular flux must be stored for all groups. However, for the Jacobi kernel, the incoming and outgoing angular fluxes must be stored, as well as the multigroup source data. There are additional less sizable data (such as temporary variables), but the angular flux and source storage account for a vast majority of the requirement. Table V shows the total memory requirements for the VERA test problems presented in this work (2a and 5a-2D), including two different spatial decomposition schemes for 5a-2D.

Table V.

Total Memory Requirements (GB) for VERA Test Problems

Problem	GS	J	Ratio
2a	0.215	0.236	1.10
5a-2D (1 core)	12.434	13.459	1.08
5a-2D (73 cores)	23.527	25.929	1.10
5a-2D (257 cores)	37.221	40.737	1.09

In nearly all three cases, the Jacobi approach requires roughly 8-10% more overall memory, which is likely very easy to accommodate given the benefits observed in performance and stability. It is also worth noting the dependence of the memory requirement on the number of cores used to solve 5a-2D. The case with 257 cores requires substantially more total memory than 73 cores because of the boundary angular flux storage, though the memory per processor is decreased. Additionally, some consideration should be made when running larger problems, such as 5a-2D, on fewer cores. For example, many high performance clusters have around 2 GB of memory per core, which could be an issue if the quarter core case were run on ~12 or fewer cores. However, it is possible that future machines will have additional memory per core, making this less of an issue.

It is worth noting that the cross sections in MPACT are calculated up-front and the data for all groups is stored simultaneously. Other codes use a slightly different approach, recalculating the cross sections for each group as it is needed, only storing one group at a time.

Because it is using a multigroup kernel, the cross sections for all groups are needed, so if implemented into a code that usually only stores one group at a time for the Gauss-Seidel approach, the memory difference between Gauss-Seidel and Jacobi will be notably larger. However, the performance profile will also be different, likely reducing the time spent calculating cross sections.

IV. CONCLUSIONS

This paper evaluates the stability and performance of the Gauss-Seidel and Jacobi inscatter iterations for MOC. With the Gauss-Seidel approach, as the MOC solver loops over groups, it uses the flux solution from the previous group to construct the inscatter source for the next group. Alternatively, the Jacobi approach uses only the fluxes from the previous outer iteration to determine the inscatter source for each group. The focus of the stability assessment is with respect to transport-corrected scattering, which can demonstrate convergence issues because of negative self-scatter cross sections which arise from attempting to apply P_1 physics to the isotropic scattering kernel. The B&W Core I, Dimple S01A-Sq, and VERA Progression Problem 5a-2D cases were used to assess the stability by varying the number of upscatter and inner iterations used for each MOC outer iteration sweep. From these cases, it was observed that the Jacobi approach is considerably more stable than Gauss-Seidel, allowing for converged solutions to be obtained over a much wider range of iteration bounds.

An exploitable property of the Jacobi approach is that each group is effectively independent of all other groups since the inscatter source is constructed using previous iterate fluxes. As such, the loop over groups can be moved from the outermost loop, as is the case with the Gauss-Seidel sweeper, to the innermost loop, allowing for a substantial increase in efficiency by avoiding the

multiple accesses to the ray tracking data, which is independent of energy. The performance evaluation was focused on VERA Progression Problems 2a and 5a-2D, where it was found that the Jacobi approach is roughly $2.0\text{--}2.5\times$ faster on a per iteration basis. Not surprisingly, the Jacobi approach incurs more outer iterations, so the overall reduction in the MOC time is slightly less.

Lastly, the memory overhead between the two approaches was also assessed. While the performance and stability of the Jacobi approach are substantially improved, it does yield roughly an 8–10% increase in the overall memory requirements, at least based on VERA 2a and 5a-2D. It is the opinion of the authors that the benefits easily outweigh the costs, as this approach has been adopted as the default in MPACT. Some consideration must be made when deploying large problems on smaller machines in which the memory per core may pose an issue. This does not seem to be a hurdle for the current target application for MPACT, but it should be kept in mind.

It should also be noted that this approach does not resolve all transport-corrected scattering issues, but it does seem to be considerably more stable compared to Gauss-Seidel. Additionally, this work focused on its application to 2D problems, whereas 3D problems, if solved with the 2D/1D solver in MPACT, may demonstrate different behavior, as the axial transverse leakage terms for each planar MOC solver could also lead to negative sources and convergence problems. Ongoing and future work is being pursued to further address, understand and predict the instabilities encountered with transport-corrected P_0 scattering. Establishing a more theoretical foundation for the instabilities would be very worthwhile as a topic of subsequent publications.

V. ACKNOWLEDGMENTS

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

This research also made use of resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the US Department of Energy under Contract No. DE-AC05-00OR22725.

VI. REFERENCES

1. B. COLLINS et al., “Stability and Accuracy of Three-Dimensional Neutron Transport Simulations Using the 2D/1D Method in MPACT,” *Journal of Computational Physics*, 2016 (forthcoming).
2. S. PALMTAG, “Demonstration of Neutronics Coupled to Thermal-Hydraulics for a Full-Core Problem using VERA,” CASL-U-2013-0196-000 (December 30 2013): <http://www.casl.gov/docs/CASL-U-2013-0196-000.pdf> (accessed April 7, 2016).
3. “Consortium for Advanced Simulation of Light Water Reactors (CASL),” <http://www.casl.gov/> (accessed April 7, 2016).
4. H. G. JOO et al., “Methods and Performance of a Three-Dimensional Whole-Core Transport Code DeCART,” *Proc. PHYSOR 2004*, Chicago, IL (April 25–29, 2004).
5. B. COLLINS et al., “Assessment of 2D/1D Capability in MPACT,” *Proc. PHYSOR 2014*, Kyoto, Japan (September 28 – October 3, 2014).

6. Y. S. JUNG et al., “Practical numerical reactor employing direction whole core neutron transport and subchannel thermal/hydraulic solvers,” *Annals of Nuclear Energy*, **62**, pp. 357–374 (2013).
7. A. YAMAMOTO, Y. KITAMURA, and Y. YAMANE, “Simplified Treatments of Anisotropic Scattering in LWR Core Calculations,” *Journal of Nuclear Science and Technology*, **45**, No. 3, pp. 217–229 (2008).
8. M. TABUCHI, A. YAMAMOTO, T. ENDO, and N. SUGIMURA, “Convergence Analysis of MOC Inner Iterations with Large Negative Self-Scattering Cross-Section,” *Journal of Nuclear Science and Technology*, **50**, No. 5, pp. 493–502 (2013).
9. S. STIMPSON et al. “Boundary Acceleration Techniques for CMFD-Accelerated 2D-MOC,” *Proc. PHYSOR 2014*, Kyoto, Japan (September 28 – October 3, 2014).
10. K. SMITH, personal communication (2015).
11. W. BOYD et al., “The OpenMOC Method of Characteristics Neutral Particle Transport Code,” *Annals of Nuclear Energy*, **68**, pp. 43–52 (2014).
12. J. ASKEW, A characteristics formulation of the neutron transport equation in complicated geometries, AEEW-R-1108, U.K. Atomic Energy Authority (1972).
13. R. J. STAMM’LER et al., “HELIOS Methods & HELIOS Documentation Rev. No. 7,” Studsvik Scandpower (2005).
14. K. SMITH, J. RHODES, “CASMO-4 characteristics methods for two-dimensional PWR and BWR core calculations”, *Transactions of the American Nuclear Society*, 83, p. 294 (2000).
15. B. COLLINS, S. STIMPSON, B. W. KELLEY, M. T. H. YOUNG, B. KOCHUNAS, 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*, **326**, 612 (2016).
16. C. TANG and S. ZHANG, “Development and Verification of an MOC Code Employing Assembly Modular Ray Tracing and Efficient Acceleration Techniques,” *Annals of Nuclear Energy*, **36**, pp. 1013–1020 (2009).
 17. R. FERRER, J. RHODES, “A Linear Source Approximation Scheme for the Method of Characteristics,” *Nuclear Science and Engineering*, **182**, No. 2, pp. 151–165 (2016).
 18. S. G. STIMPSON, B. S. COLLINS, and T. J. DOWNAR, “Axial Transport Solvers for the 2D/1D Scheme in MPACT,” *Proc. PHYSOR 2014*, Kyoto, Japan (September 28 – October 3, 2014).
 19. Z. LIU et al., “A new three-dimensional method of characteristics for the neutron transport calculation,” *Annals of Nuclear Energy*, **38**, p. 447–454 (2011).
 20. S. SHANER et al., “Validation and Performance of the 3D Method of Characteristics Solver in OpenMOC,” *Proc. PHYSOR 2016*, Sun Valley, Idaho (May 1–5, 2016).
 21. J. B. TAYLOR, “The Development of a Three-Dimensional Nuclear Reactor Kinetics Methodology Based on the Method of Characteristics,” Ph.D. Dissertation, Pennsylvania State University (2007).
 22. B. M. KOCHUNAS, “A Hybrid Parallel Algorithm for the 3-D Method of Characteristics Solution of the Boltzmann Transport Equation on High Performance Compute Clusters,” Ph.D. Dissertation, University of Michigan (2013).
 23. K. S. KIM et al., “Development of a New 47-Group Library for the CASL Neutronics Simulators,” *Proc. M&C 2015*, Nashville, Tennessee, April 19–23 (2015).

24. M. RYU et al., “Incorporation of Anisotropic Scattering in nTRACER,” *Trans. of the Kor. Nucl. Soc.*, Autumn Meeting, Pyeongchang, Korea, October 30–31 (2014).
25. B. R. HERMAN et al., “Improved Diffusion Coefficients Generated from Monte Carlo Codes,” *Proc. M&C 2013*, Sun Valley, Idaho, May 5–9 (2013).
26. L. W. NEWMAN et al., “Urania Gadolinia: Nuclear Model Development and Critical Experiment Benchmark,” BAW-1810, The Babcock & Wilcox Company (1984).
27. D. HANLON et al., “Light Water Moderated and Reflected Low Enriched Uranium (3 wt% ^{235}U) Dioxide Rod Lattices DIMPLE S01,” **1**, NEA, NSC, DOC (2006).
28. B. C. YEE, B. KOCHUNAS, T. DOWNAR, *MPACT Modeling of DIMPLE Benchmark Problems*, CASL-U-2016-1045-000, Oak Ridge National Laboratory, Oak Ridge, TN (2016).
29. A. T. GODFREY, *VERA Core Physics Benchmark Progression Problem Specifications* (Rev. 4), Tech. Rep. CASL-U-2012-0131-004, Oak Ridge National Laboratory, Oak Ridge, TN. Available online. URL <http://www.casl.gov/publications.shtml> (2014).
30. A. ZHU et al., “An Optimized Artificial Diffusion Coarse Mesh Finite Difference Method to Accelerate the Neutron Transport Calculation,” *Nuclear Science and Engineering*, 2016 (forthcoming).
31. B. COLLINS et al., *Performance Improvements for Coarse Mesh Finite Difference Acceleration*, CASL-X-2016-1106-000, Oak Ridge National Laboratory (May 31, 2016).
32. S. STIMPSON et al., *MPACT Subgroup Self-Shielding Efficiency Improvements*, CASL-I-2016-1063-000, Oak Ridge National Laboratory (April 4, 2016).
33. Y. LIU et al., *Runtime Improvements to the Cross Sections Calculation in MPACT*, CASL-X-2016-1105-000, Oak Ridge National Laboratory (May 31, 2016).

34. OAK RIDGE LEADERSHIP COMPUTING FACILITY, “Introducing Titan – The World’s #1 Open Science Supercomputer,” available online. URL <http://www.olcf.ornl.gov/titan/> (2013).