M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, April 16-20, 2017, on USB (2017)

A Multigroup, Lumped Parameter MOC Approach for Subgroup Self-Shielding in MPACT

S. Stimpson,[1] Y. Liu,[2] B. Collins,[1] K. Clarno[1]

[1] *Oak Ridge National Laboratory: One Bethel Valley Road, P.O. Box 2008, Oak Ridge, TN 37831-6172*
stimpsonsg@ornl.gov; collinsbs@ornl.gov; clarnokt@ornl.gov

[2] *Department of Nuclear Science and Engineering, University of Michigan: 2355 Bonisteel Blvd. Ann Arbor, MI, 48105*
yuxuanl@umich.edu

Abstract - *The MPACT code being developed collaboratively by Oak Ridge National Laboratory and the University of Michigan is the primary deterministic neutron transport solver in the Virtual Environment for Reactor Applications (VERA). For most 3D problems, MPACT employs the 2D/1D method, using the 2D method of characteristics (MOC) to solve for each radial plane, while the 1D pin-wise nodal methods are used axially. An essential component of the neutron transport solver is the self-shielding calculation used to determine equivalence cross section. MPACT is currently using the subgroup self-shielding method, in which MOC is used to solve the purely absorbing fixed source problems defined by the subgroup approach. Recent efforts to speed up the MOC solvers in MPACT have reduced runtime by roughly 2× by incorporating multigroup kernels. By incorporating similar kernels for self-shielding and developing a novel lumped parameter approach to MOC, substantial improvements have also been made to the self-shielding computation efficiency without sacrificing any accuracy.*

These new capabilities have been demonstrated on two test cases: (1) a single lattice with quarter symmetry known as VERA Progression Problem 2a and (2) a 2D quarter-core slice known as Problem 5a-2D. From these cases, self-shielding computational time is reduced by roughly 3–4×, with a corresponding 15–20% increase in overall memory burden. Given these benefits, these approaches have been adopted as the default in MPACT.

I. INTRODUCTION

The primary goal of the Consortium for Advanced Simulation of Light Water Reactors (CASL) [1] is to provide high fidelity simulations of nuclear reactor core physics. To accomplish this, CASL is developing the Virtual Environment for Reactor Applications (VERA) [2], which consists of a collection of physics codes and multiphysics coupling drivers. The MPACT code [3] is the primary deterministic neutron transport solver in VERA, predominantly employing the 2D/1D method [4,5] to solve 3D transport problems. In this approach, the 2D method of characteristics (MOC) is used for each plane to solve the radial transport problem, and 1D pin-wise nodal methods are used axially [6]. In recent months, considerable attention has been focused on improving the computational performance of MPACT, and almost every sequence in the code has been improved. This paper presents some of the improvements made to the self-shielding calculation.

An essential component of the neutron transport solver is the self-shielding calculation, for which MPACT is currently solving with the subgroup method [7]. Recent efforts to improve the efficiency of the MOC solvers, which are the workhorse for radial transport in MPACT, have yielded efficient multigroup kernels that loop over several energy groups rather than looping over one group at a time [8]. This approach is consistent with the MOC kernels in CASMO [9] and OpenMOC [10]. These kernels have sped up the MOC sweeping time by roughly 2× during the eigenvalue calculation. The subgroup calculation typically requires a substantial amount of time, and it had not been reevaluated to take advantage of these new kernels. The first improvement addressed herein is the *integration of multigroup kernel concepts into the subgroup calculation,* which is then used as the basis for further extensions.

The next improvement discussed is the *lumped parameter MOC*. Because the subgroup calculation is solving purely fixed source problems (FSPs), and since multiple sweeps are performed only to update the boundary angular fluxes, the sweep procedure can be condensed to allow for instantaneous propagation of the flux across a spatial domain without the need to sweep along all segments in a ray. Once the boundary angular fluxes are considered to be converged, an additional sweep is completed to tally the scalar flux.

Once the theory and new algorithms are presented, these improvements will be demonstrated on a single lattice test case known as VERA Progression Problem 2a and on a 2D quarter-core pressurized water reactor (PWR) problem known as VERA Progression Problem 5a-2D [11].

II. THEORY

1. Method of Characteristics

MOC is a widely used deterministic method to solve the multigroup Boltzmann neutron transport equation (Eq. 1) [5, 12,13,14,15]. While some applications use a linear source representation spatially [16,17], most use a flat source approximation. The MOC equations in this paper focus on flat source 2D-radial applications with isotropic scattering kernels:

$$\underline{\boldsymbol{\Omega}} \cdot \boldsymbol{\nabla}\varphi\left(\underline{\boldsymbol{r}}, \underline{\boldsymbol{\Omega}}\right) + \Sigma_{t,g}(\boldsymbol{r})\varphi_g\left(\underline{\boldsymbol{r}}, \underline{\boldsymbol{\Omega}}\right) = q_g\left(\underline{\boldsymbol{r}}\right), \quad (1a)$$

where

$$\begin{aligned} & q_g\left(\underline{\boldsymbol{r}}\right) \\ & = \frac{\chi_g(\boldsymbol{r})}{4\pi k_{\text{eff}}} \sum_{g'=1}^{N_{grp}} \nu\Sigma_{f,g'}\left(\underline{\boldsymbol{r}}\right) \phi_{g'}\left(\underline{\boldsymbol{r}}\right) \\ & + \frac{1}{4\pi}\left(\sum_{g'=1,}^{N_{grp}} \Sigma_{s0,g'\to g}\left(\underline{\boldsymbol{r}}\right)\phi_{g'}\left(\underline{\boldsymbol{r}}\right)\right). \end{aligned} \quad (1b)$$

The scalar flux is simply obtained by integrating the angular flux over all angles (Eq. 1c):

$$\phi_{g'}\left(\underline{\boldsymbol{r}}\right) = \int_0^{4\pi} \varphi_{g'}\left(\underline{\boldsymbol{r}}, \underline{\boldsymbol{\Omega}}'\right) d\Omega'. \quad (1c)$$

In Equation 1, $\underline{\boldsymbol{r}}$ represents the spatial vector, $\underline{\boldsymbol{\Omega}}$ represents the angular vector comprised of both azimuthal and polar angles, g denotes the neutron energy group index, and q_g is the source term which contains both the fission and scattering terms. MOC can be applied to this problem by first introducing a characteristic direction and then casting a 1D version of Eq. 1 along the problem. By removing the spatial dependence of the total cross section/source, the solution is obtained for the angular flux at any point s along the ray:

$$\begin{aligned} \varphi_g\left(s, \underline{\boldsymbol{\Omega}}\right) = \varphi_g^{in}\left(\underline{\boldsymbol{\Omega}}\right)e^{-\Sigma_{t,g}s} \\ + \frac{q_g}{\Sigma_{t,g}}\left(1 - e^{-\Sigma_{t,g}s}\right). \end{aligned} \quad (2)$$

Applying the discrete ordinates approximation and discretizing spatially, Eq. 2 now becomes Eq. 3, where a denotes the angle index and r denotes the spatial region index:

$$\begin{aligned} \varphi_{g,a,r} = \varphi_{g,a,r}^{in}e^{-\Sigma_{t,g,r}s_{a,r}} \\ + \frac{q_{g,r}}{\Sigma_{t,g,r}}\left(1 - e^{-\Sigma_{t,g,r}s_{a,r}}\right). \end{aligned} \quad (3)$$

From this, the outgoing angular flux at the end of the ray (Eq. 4a) and the average angular flux along the ray (Eq. 4b) can be obtained.

$$\begin{aligned} \varphi_{g,a,r}^{out} = \varphi_{g,a,r}^{in}e^{-\Sigma_{t,g,r}s_{l,r}} \\ + \frac{q_{g,r}}{\Sigma_{t,g,r}}\left(1 - e^{-\Sigma_{t,g,r}s_{a,r}}\right), \end{aligned} \quad (4a)$$

$$\bar{\varphi}_{g,a,r} = \frac{\varphi_{g,a,r}^{out} - \varphi_{g,a,r}^{in}}{\Sigma_{t,g,r}s_{a,r}} + \frac{q_{g,r}}{\Sigma_{t,g,r}}. \quad (4b)$$

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.

2. Subgroup Method

MPACT is currently using the subgroup self-shielding method, in which the detailed cross section behavior of each coarse energy group is replaced by its probability density representation that preserves certain integrals. There are two groups of methodologies for determining the subgroup probability tables: the physical probability table and the mathematical probability table [18]. MPACT uses the physical probability table, in which the resonance integral tables are converted into a set of subgroup levels and weights by preserving effective cross sections over a range of background cross sections. The effective cross section is evaluated by these subgroup levels $\sigma_{x,g,l}$ and weights $w_{x,g,l}$ as:

$$\sigma_{x,g} \cong \frac{\sum_l \sigma_{x,g,l}\phi_{g,l}w_{x,g,l}}{\sum_l \phi_{g,l}w_{x,g,l}}, \quad (5)$$

where $\phi_{g,l}$ is the subgroup level–dependent flux. Assuming that subgroup parameters have been determined, application of the subgroup method involves obtaining the level dependent flux (or the equivalence cross section that can be converted from flux) from an FSP:

$$\boldsymbol{\Omega} \cdot \nabla \varphi_{g,c,l} + \left(\Sigma_{a,g,c,l} + \lambda \Sigma_{p,g}\right)\varphi_{g,c,l} = \frac{\lambda \Sigma_{p,g}}{4\pi} \Delta u_g. \qquad (6)$$

Eq. 6 must be solved for every resonance group g, resonance category c [7], and subgroup level l, resulting in significant computing time for subgroup calculation. Note that Eq. 6 is a purely absorbing problem that is achieved by a few approximations. The validity of these approximations is discussed by Liu and Martin [19]. The isotopes in the problem are generally divided into a set of resonant categories. For example, ^{238}U may be in one category, ^{235}U and other heavy metals in another, natural Zr isotopes in a third, and all others in a fourth category. This approach is commonly used in self-shielding methods to minimize the total number of transport calculations required to obtain accurate results.

3. Multigroup Kernels for Subgroup

The pre-existing subgroup calculation in MPACT consists of several loops: (1) over resonant groups, (2) over resonant categories, and (3) over sublevels [3]. These are typically solved one at a time, but the multigroup kernels allow for all (or some) to be solved concurrently. The calculation scheme for the three loop approach is shown in Figure 1. As can be seen, inside these loops, there is an iteration loop in which transport sweeps for each resonant group, category, and level are performed. Here, Σ_p is the potential cross section, Σ_t is the total cross section, and Σ_{eq} is the equivalence cross section.

```
1: for each resonant group (g from g_{res,beg} to g_{res,end})
2:     for each subgroup category (c from 1 to N_{cat}(g))
3:         for each subgroup level (l from 1 to N_{levels})
4:             Setup Σ_{t,g,c,l} based on Σ_{a,g,c,l} and λ_g Σ_p
5:             Setup source for this group/category/level based on λ_g Σ_p
6:             for each iteration (i from 1 to N_{iters})
7:                 Perform transport sweep for this group/category/level
8:                 Compare residual based on scalar flux (terminate if below criteria)
9:             end for
10:            Calculate equivalence cross section (Σ_{eq,g,c,l})
11:        end for
12:    end for
13: end for
```

Fig. 1. Pseudocode for pre-existing subgroup scheme.

To take advantage of the multigroup kernels that have been implemented into MPACT [8], the scheme must be slightly restructured. For the purposes of this work, a single combination of group/category/level is considered to be a *pseudogroup* [20]. The number of pseudogroups for the entire subgroup calculation will be the product of the number of resonant groups, the average number of subgroup categories per group, and the number of subgroup levels. In theory, the number of categories can vary from group to group, though this does not seem to be the case for the current libraries available to MPACT. In the 47-group library [21] used in this work, there are 17 resonant groups, 4 categories, and 4 levels yielding 272 pseudogroups. Based on this concept, a transport kernel could be constructed to sweep over all pseudogroups concurrently, effectively vectorizing the three loops of the original algorithm. However, the sources, cross sections, scalar fluxes, and angular fluxes must be stored for each pseudogroup up front, whereas in the previous scheme, only one group of storage at a time was necessary. Figure 2 shows the pseudocode for the refactored scheme, taking advantage of the multigroup kernel concept. When Fig. 2 is compared to Fig. 1, it can be seen that the three loops over resonant group, subgroup category, and subgroup level are condensed into a single loop over pseudogroups. What is not seen is the actual multigroup MOC kernel, where the loop over groups/pseudogroups is moved to the innermost loop, which is being called on line 6.

```
for each pseudogroup (pg from 1 to N_pseudogroups)
    Setup and store Σ_t,pg for this pseudogroup based on Σ_a,pg and λ_pg Σ_p
    Setup and store source for this pseudogroup based on λ_pg Σ_p
end for
for each iteration (i from 1 to N_iters)
    Perform transport sweep for all pseudogroups
    Compare residual based on scalar flux (terminate if below criteria)
end for
for each pseudogroup (pg from 1 to N_pseudogroups)
    Calculate equivalence cross section Σ_eq,pg
end for
```

Fig. 2. Pseudocode for subgroup scheme using the multigroup transport kernel.

As might be expected, the memory required to store source and flux data for 272 pseudogroups can be significant. One way to mitigate memory concerns while allowing the scheme to make use of the multigroup kernels is to divide the pseudogroups into batches. Here the 272 pseudogroups are decomposed into N_{batch} batches, dividing as evenly as possible. The main advantage of the multigroup kernels is that they eliminate the duplicate work of connecting the modular rays for the entire domain by moving the loop over groups to the innermost loop [8]. When using batches, some of this advantage is compromised, as the modular rays setup will still be performed N_{batch} times. However, this is still a significant reduction compared to performing it for each pseudogroup, as is the case for the original approach. Figure 3 shows the pseudocode for the batched approach, where each batch contains a starting and stopping pseudogroup index. The primary difference between Figures 2 and 3 is that Fig. 3 includes an additional outer loop over the number of batches, and the bounds for the pseudogroup loops is updating to reflect the lower and upper pseudogroup indexes for each batch.

```
for each batch (b from 1 to N_batch)
    for each pseudogroup (pg from pg_beg(b) to pg_end(b))
        Setup and store Σ_t,pg for this pseudogroup based on Σ_a,pg and λ_pg Σ_p
        Setup and store source for this pseudogroup based on λ_pg Σ_p
    end for
    for each iteration (i from 1 to N_iters)
        Perform transport sweep for all pseudogroups
        Compare residual based on scalar flux (terminate if below criteria)
    end for
    for each pseudogroup (pg from pg_beg(b) to pg_end(b))
        Calculate equivalence cross section Σ_eq,pg
    end for
end for
```

Fig. 3. Pseudocode for subgroup scheme using the multigroup transport kernel and batching.

4. Lumped Parameter MOC

In addition to using multigroup kernels, a *lumped parameter* MOC approach has been applied to the subgroup self-shielding problem [20]. Because the self-shielding calculation is a purely absorbing fixed source problem, and since multiple sweeps are performed only to update the boundary angular fluxes, the sweep procedure can be condensed to allow for the instantaneous propagation of the flux across a spatial domain without the need to sweep along all segments in a ray as is typically done. This requires an initial sweep to tabulate lumped parameter coefficients for the angular flux propagation. Subsequent sweeps use the lumped parameters to instantly update the

angular flux, bypassing all calculations along the ray. Once the boundary angular fluxes are considered to be converged, an additional sweep is completed to tally the scalar flux.

Because the MOC kernels in MPACT sweep over two angles travelling in opposite directions, forward and backward, at the same time, effectively two equations are needed (Eqs. 7a and 7b):

$$\varphi_{pg}^{out,for} = \varphi_{pg}^{in,for} A_{pg} + B_{pg}, \quad (7a)$$

$$\varphi_{pg}^{out,back} = \varphi_{pg}^{in,back} A_{pg} + C_{pg}. \quad (7b)$$

Here $\varphi_{pg}^{in,for}$ and $\varphi_{pg}^{in,back}$ are the incoming angular fluxes for pseudogroup pg at a system of parallel boundary at each end of an MOC ray, one forward along it and one backward. Similarly, $\varphi_{pg}^{out,for}$ and $\varphi_{pg}^{out,back}$ are the outgoing angular fluxes.

To visualize this, consider a ray in a simple pin cell problem (Fig. 4). On the left is the discretization showing 5 segments along the ray (blue) with the incoming and outgoing angular fluxes at the ends of the ray. On the right is the same problem but with all 5 segments condensed into one. The incoming and outgoing angular flux data are stored at both ends of the ray, as the same ray segment data are used to calculate both the forward and back directions simultaneously. To reiterate, the lumped parameter approach is only valid and effective because the source is not changing between iterations, as is the case during the eigenvalue calculation sweeps. Thus, the $A/B/C$ lumped parameters can be used in a fast intermediate kernel that only updates the outgoing angular flux.

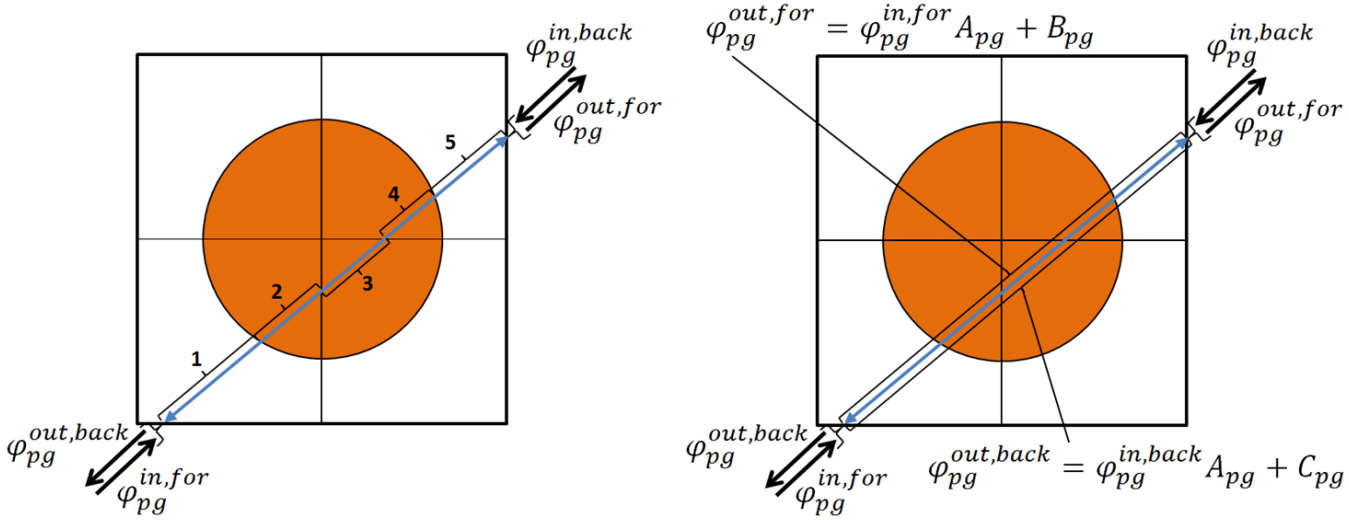

Fig. 4. Visualization of MOC ray tracing (left) and lumped parameter (right) on a pin cell.

Two approaches are available to derive equations for lumped parameters. The first would be to manually derive the parameters by solving for the flux solution as the MOC ray is traced. However, upon inspection, it can be concluded that A will be a product of the exponential terms for each segment (Eq. 8a). With A established, B and C can be easily calculated using the incoming and outgoing angular flux values (Eqs. 8b and 8c), which are known values. Here, N_{seg} is the number of segments along an MOC ray, $\Sigma_{t,i,pg}$ is the total cross section, and l_i is the segment length:

$$A_{pg} = \prod_{i=1}^{N_{seg}} e^{-\Sigma_{t,i,pg} l_i}, \quad (8a)$$

$$B_{pg} = \varphi_{pg}^{out,for} - \varphi_{pg}^{in,for} A_{pg}, \quad (8b)$$

$$C_{pg} = \varphi_{pg}^{out,back} - \varphi_{pg}^{in,back} A_{pg}. \quad (8c)$$

The lumped parameters must be calculated and saved for each angle and ray. Because there will only be three values over O(100) segments, the storage for this is not concerning. Figure 5 shows the pseudocode for lumped parameters, which is based on the multigroup kernel with batching. The key changes to note are (1) an initial sweep calculates the lumped parameters (line 6), (2) several "fast" sweeps simply apply the factors to update the angular flux (per Eq. 8 and line 8), and (3) a final standard sweep tallies the scalar flux (line 11), which is required for the equivalence cross section calculation (line 13).

```
for each batch (b from 1 to N_batch)
    for each pseudogroup (pg from pg_beg(b) to pg_end(b))
        Setup and store Σ_{t,pg} for this pseudogroup based on Σ_{a,pg} and λ_pg Σ_p
        Setup and store source for this pseudogroup based on λ_pg Σ_p
    end for
    Perform an initial sweep accumulating the A_pg/B_pg/C_pg lumped parameters
    for each iteration (i from 2 to N_iters)
        Perform a transport sweep applying A_pg/B_pg/C_pg parameters for this batch
        Compare residual based on boundary angular fluxes (terminate if below criteria)
    end for
    Perform a final, normal sweep accumulating scalar flux (φ_pg)
    for each pseudogroup (pg from pg_beg(b) to pg_end(b))
        Calculate equivalence cross section Σ_{eq,pg}
    end for
end for
```

Fig. 5. Pseudocode for subgroup scheme using the multigroup transport kernel, batching, and lumped parameter approach.

Since only the last iteration yields a scalar flux distribution, the convergence residual for this scheme is based on the angular flux updates instead of the scalar flux, which is used in the current scheme. Choosing the correct convergence criteria is important to ensure consistency between these two schemes. The current scheme imposes a maximum change of 1×10^{-6} for the scalar flux in any region for each pseudogroup. Since the new scheme will perform an additional sweep once the angular flux is considered to be converged, a similar maximum change is imposed on the angular flux, but with a criterion of 1×10^{-5}. In practice, this approach has been observed to be conservative, in most cases requiring one additional iteration, which is acceptable since it is only one additional fast iteration.

This approach would not be beneficial in problems with fully vacuum radial boundary conditions in serial. In this scenario, only one iteration would be necessary since the boundary conditions do not need to be converged, as a zero incoming angular flux is correct. This is not a likely scenario since most problems are executed with quarter symmetry and in parallel. It is expected that larger spatial domains will reap greater benefits. Pin cell problems would naturally have the least to gain since there are so few segments along a ray. The best case would likely be a full core problem without any radial decomposition, although this is likely impractical because of the substantial memory required. In general, at least 8 radial partitions are used on a quarter-core slice to allow for an acceptable amount of memory per core, where a domain consists of a few assemblies of data. In the results section below, 73 radial domains are used in the quarter-core problem, which amounts to a maximum of one assembly per domain, and good performance is observed there.

III. RESULTS

These improvements have been applied to a variety of problems, ranging from single lattices to full 3D quarter-core depletion cases. Two such test problems are highlighted here.

1. Single Lattice

The first is a single 17 × 17 lattice problem known as VERA Progression Problem 2a [11] (Figure 6). This problem uses a uniform enrichment of 2.1% and quarter symmetry.

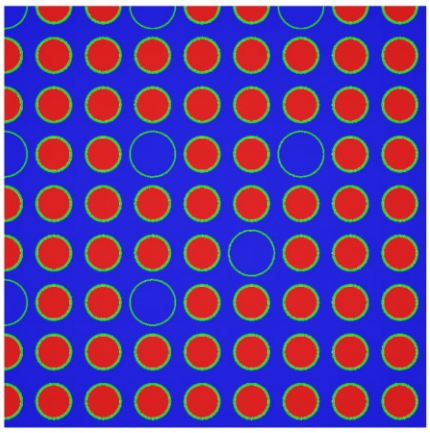

Fig. 6. Geometry visualization of VERA Problem 2a.

Table I contains results for the single lattice as the number of batches used in the new self-shielding approach is varied. The table includes (1) the average batch size, which is simply 272 divided by the number of batches, (2) the subgroup self-shielding computation time, and (3) the

overall memory burden of the problem. It is worth noting that the MOC kernels used in the eigenvalue solver use 47 groups, so using a number of batches that yields roughly as many pseudogroups would not be excessive.

Table I. Results for Problem 2a

Number of Batches	Average Batch Size (pseudogroups)	Subgroup Time (sec)	Total Memory (GB)
1	272.0	8.17	0.50
2	136.0	8.25	0.35
3	90.7	8.26	0.30
4	68.0	8.42	0.27
5	54.4	8.70	0.26
6	45.3	8.78	0.25
7	38.9	8.87	0.25
8	34.0	8.92	0.25
9	30.2	9.52	0.25
10	27.2	9.58	0.25
1G	---	24.36	0.20
MG	---	---	0.23

If only the multigroup subgroup kernels are used then the memory burden is also included, but without the lumped parameter approach. This results in a better indication of how much burden the lumped parameter value storage incurs. All of these cases were run on a small development cluster with AMD processors (Operton™ Processor 6376, 2.3 GHz). From these results, memory usage increases roughly 2.8×, or roughly 25%, when compared to the original 1-group methodology. Comparing to the multigroup memory burden, the lumped parameter values yield an increase of approximately 9% in overall memory used. One interesting trend to note is that the subgroup time continually increases with the number of batches, showcasing the advantage of using the multigroup kernels.

2. 2D Quarter-Core Slice

Figure 7 shows the VERA Problem 5a-2D quarter-core layout, [11], which consists of 17 × 17 assemblies with three different enrichments. A number of Pyrex rods are included in various configurations in some assemblies. Also included in the model is a radial reflector region with a core baffle and an assembly's width of moderator along the core's periphery.

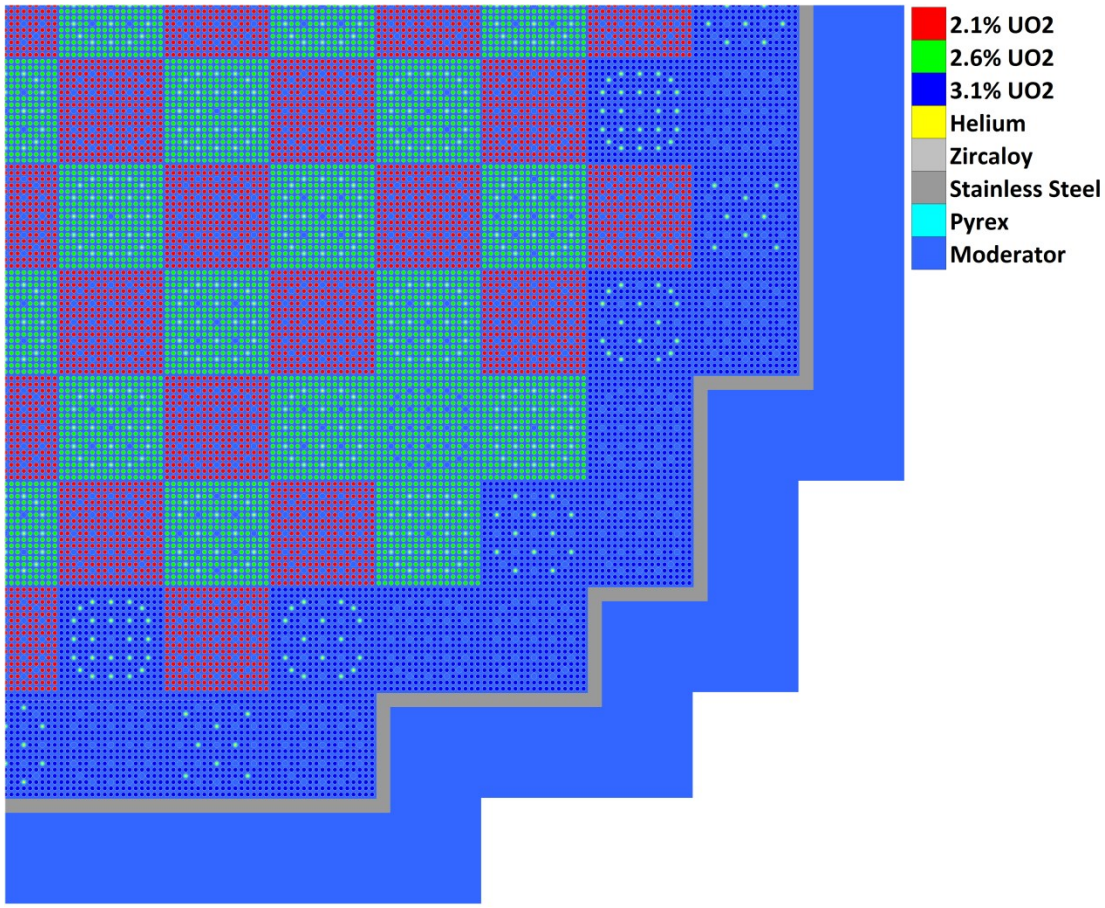

Fig. 7. Geometry visualization of VERA Problem 5a-2D.

Table II shows the timing results and aggregate memory requirements for the problem, as the number of batches in the new self-shielding approach is varied. These cases were run on the Titan supercomputer [22] with 73 radial domains. These results can be compared to the results with the one-group (1G) shielding approach.

Table II. Results for Problem 5a-2D

Number of Batches	Average Batch Size (pseudogroups)	Subgroup Time (sec)	Total Memory (GB)
1	272.0	76.32	67.53
2	136.0	56.14	44.98
3	90.7	52.10	37.98
4	68.0	48.97	34.41
5	54.4	48.82	33.03
6	45.3	49.58	32.89
7	38.9	50.97	32.87
8	34.0	49.38	32.87
9	30.2	52.31	32.87
10	27.2	53.96	32.87
1G	---	188.95	23.74
MG	---	---	28.40

These data show that the new self-shielding capability yields a 3.87× speedup (48.82 sec vs. 188.95 sec) with 5 batches, as well as a 16.3% increase in memory burden (33.03 GB vs. 28.40 GB). Interestingly, the subgroup time does not increase monotonically with the number of batches. Because this case was performed with a substantial number of radial domains, the parallel communication is a much greater factor. The trends observed here may relate to the large buffer sizes that can be encountered when passing angular flux data for a large number of pseudogroups. This can be supported by the total memory, which is substantially larger, particularly for the cases with only a few pseudogroups.

The benefits of these new approaches greatly outweigh the modest burden, so these methods have been adopted as the default within MPACT. These benefits have also been observed in 3D cases with 2D/1D, which is expected because 2D MOC is used for radial slices, and the results from the 2D quarter-core slice directly relate.

IV. CONCLUSIONS AND FUTURE WORK

Two improvements to the self-shielding calculation in MPACT have been presented. The use of multigroup kernels to vectorize the algorithm is shown, along with a batching approach to mitigate memory concerns. A lumped parameter MOC approach was presented to condense the solve along MOC rays; this approach is used as the basis for fast sweeps that immediately propagate the angular flux boundary conditions by taking advantage of the purely absorbing fixed source problems formed by the self-shielding method. The impact of these combined improvements was tested on a single lattice and a 2D quarter-core model, varying the number of batches. The results show an almost 4× reduction in runtime, with a modest 16.3% increase in the memory burden. Given its success, the method is now the current default in MPACT.

While these improvements have focused on the subgroup self-shielding method, the embedded self-shielding method (ESSM) [23] is also available in MPACT, and applying the same principles to ESSM, which aims to be more accurate than subgroup, could be useful. If ESSM were to become the default self-shielding method in MPACT, then priority may be given to increasing its efficiency, as well.

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.

REFERENCES

1. "Consortium for Advanced Simulation of Light Water Reactors (CASL)." Available online. URL http://www.casl.gov/ (2015).
2. S. PALMTAG, "Demonstration of Neutronics Coupled to Thermal-Hydraulics for a Full-Core Problem Using VERA," CASL-U-2013-0196-000, Oak Ridge National Laboratory (2013).
3. MPACT Team, "MPACT Theory Manual, Version 2.0.0," Tech. Rep. CASL-U-2015-0078-000, Oak Ridge National Laboratory and University of Michigan (2015).
4. H. JOO et al., "Methods and Performance of a Three-Dimensional Whole-Core Transport Code DeCART," *Proc. PHYSOR 2004*, American Nuclear Society, Chicago, IL (2004).
5. B. COLLINS, S. STIMPSON, B. KELLEY, M. YOUNG, B. KOCHUNAS, E. 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).
6. S. STIMPSON, B. COLLINS, and T. DOWNAR. "Axial Transport Solvers for the 2D/1D Scheme in MPACT," *Proc. PHYSOR 2014*, Kyoto, Japan (2014).
7. R. J. STAMM'LER et al., "HELIOS Methods," Studsvik Scandpower (1998).
8. S. STIMPSON, B. COLLINS, and B. KOCHUNAS, "MOC Efficiency Improvements Using a Jacobi

Inscatter Approximation," CASL-I-2016-1056-001, Oak Ridge National Laboratory (2016).
9. K. SMITH, personal communication (2015).
10. W. BOYD et al., "The OpenMOC Method of Characteristics Neutral Particle Transport Code," *Annals of Nuclear Energy*, **68**, pp. 43–52 (2014).
11. A. GODFREY, "VERA Core Physics Benchmark Progression Problem Specifications (Rev. 4)," Tech. Rep. CASL-U-2012-0131-004, Oak Ridge National Laboratory (2014).
12. 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).
13. J. ASKEW, "A characteristics formulation of the neutron transport equation in complicated geometries," AEEW-R-1108, U.K. Atomic Energy Authority (1972).
14. R. J. STAMM'LER et al., "HELIOS Methods & HELIOS Documentation Rev. No. 7," Studsvik Scandpower (2005).
15. 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).
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. D. Cacuci (Ed.), *Handbook of Nuclear Engineering*, Lattice Physics Computations, Chapter 9, Springer, New York (2010).
19. Y. LIU and W. R. MARTIN, "Assessment of homogeneous and heterogeneous resonance integral tables and their applications to the embedded self-shielding method," *Ann. Nucl. Eng*., **92**, 186 (2016).
20. S. STIMPSON, Y. LIU, B. COLLINS, K. CLARNO, "MPACT Subgroup Self Shielding Efficiency Improvements," CASL-U-2016-1063-001, Oak Ridge National Laboratory (2016).
21. K. KIM et al., "Development of a New 47-Group Library for the CASL Neutronics Simulators," *Proc. M&C 2015*, Nashville, Tennessee (2015).
22. Oak Ridge Leadership Computing Facility. "Introducing Titan - The World's #1 Open Science Supercomputer," Available online. URL http://www.olcf.ornl.gov/titan/ (2014).
23. M. WILLIAMS and K. KIM, "The Embedded Self-Shielding Method," *Proc. PHYSOR 2012*, Knoxville, Tennessee (2012).