CMFD for MOC Implementation in Lattice Physics Code Scarabée

Jason Gaiardelli, Dr. Hunter Belanger

Rensselaer Polytechnic Institute

CMFD for Scarabée

Developer's repository link: https://github.com/gaiarj/scarabee/tree/feature/new_cmfd

Licensing provisions: GPL-3.0

Programming Languages: C++, Python

Supplementary Material: None

Summary

Scarabée is a lattice physics code that implements multiple methods to solve the neutron transport equation in different situations. One of these solvers uses the MOC (method of characteristics) to solve the steady-state neutron transport equation in variable 2D geometry. CMFD (coarse mesh finite difference) acceleration is a method used to greatly accelerate the rate of convergence of MOC solvers. During an iteration of the fine mesh transport solve (MOC), the neutron currents on CMFD boundaries are tallied and used to link the physics of the fine mesh model to the coarse mesh model. The coarse mesh problem is then initialized after every MOC iteration. The coarse mesh problem is solved in a fashion similar to finite difference diffusion, using power iteration. Once the CMFD problem has converged, the results are used to update the flux in MOC flat source regions. The result is that the MOC problem converges with the same degree of accuracy, but in significantly faster time.

Statement of Need

One of Scarabée's purposes is to be lightweight enough to run on normal laptop and desktop computers. Therefore, performance is a major concern, and CMFD represents a large improvement to the performance of a MOC solver. CMFD exists in many different forms and there are variations in its implementation, which may be more appropriate for certain types of problems or systems. Therefore, it is our purpose to find the formulation of CMFD acceleration that will work best for general purpose lattice physics calculations without the use of large-scale computing systems. Additionally, another goal for the Scarabée project is to implement depletion calculations, which are very computationally demanding and may not run in a reasonable amount of time without CMFD acceleration.

State of the Field

CMFD is a common method for accelerating fine mesh neutron transport problems, including MOC. It is implanted in several other codes, including OpenMOC [1] and OpenMC [2]. Deterministic methods for neutron transport are the current standard in the nuclear power industry, due to their much faster speed than Monte Carlo methods and the repetition of lattice geometry in LWRs which does not require complex treatment of the geometry.

CMFD Implementation

CMFD can be thought of as a multigrid method for MOC neutron transport solvers. The MOC solver acts as the fine mesh problem while the CMFD coarse mesh problem is built on top of it, and the solution is used to update the MOC on every nth iteration. The result is a much faster

convergence of the MOC solution, especially in problems where the flux is low and therefore takes a long time to converge (i.e. large reflector regions). The steps of the CMFD solver can be summarized by Figure 1:

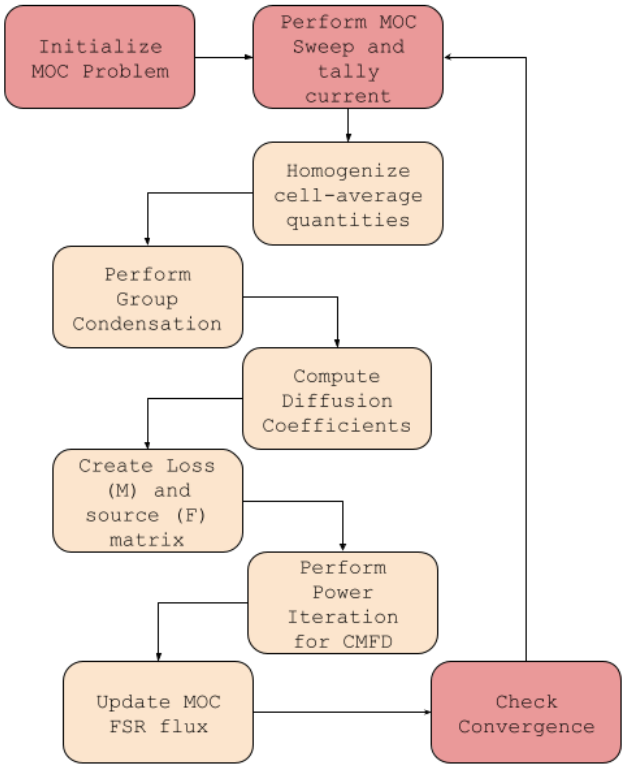

Figure 1: CMFD Flowchart

During the MOC sweep, the net current on each surface is tallied from the angular flux over all the tracks as such:

$$\tilde{J}_g^{i+\frac{1}{2},j} = \frac{4\pi}{\Delta y^{i,j}} \sum_{k\,[i+\frac{1}{2},j]} \sum_{g} \omega_{\phi(k)} \omega_\theta sin(\theta) \Delta s \Psi_{k,p,g} * \hat{n}$$

Special treatment is given to segments which pass through CMFD cell corners, where the flux is split by 0.5 and tallied on the four surfaces which lead to the diagonally adjacent cell, which preserves the neutron balance. The tallied net current is later used to compute the nonlinear diffusion coefficient for each surface ($\widetilde{D}$) which preserves the physics between the MOC and CMFD.

Once the MOC sweep is finished, the first step in the CMFD process is spatial homogenization and energy group condensation. The flat source regions (MOC mesh cells) are homogenized into

square CMFD cells by volume (Area in 2D) averaging the macroscopic cross section of each FSR material to get a single material cross section for the CMFD cell. The group condensation is performed by flux averaging the macroscopic cross section. The process can be generalized for each cross section for reaction type x as:

$$\Sigma^{\mathrm{x,i,j}}_{\mathrm{g_{cmfd}}} = \frac{\sum_{\mathrm{g_{moc}}\in\mathrm{g_{cmfd}}} \sum_{\mathrm{r}\in(\mathrm{i,j})} \Sigma^{\mathrm{x}}_{\mathrm{r,g_{moc}}} \Phi_{\mathrm{r,g_{moc}}} \mathrm{A_{r,g_{moc}}}}{\sum_{\mathrm{g_{moc}}\in\mathrm{g_{cmfd}}} \sum_{\mathrm{r}\in(\mathrm{i,j})} \Phi_{\mathrm{r,g_{moc}}} \mathrm{A_{r,g_{moc}}}}$$

The flux for CMFD cell (i,j) in group g is homogenized in a similar way, but the homogenized flux in a cell is simply summed to perform the group condensation.

The next step is to create the loss matrix and source matrices to set up the CMFD solve. The diffusion coefficients are calculated during the loss matrix creation. The diffusion equation for a single coarse mesh cell (i,j) in group g is:

$$\begin{aligned}
&\Delta y^{i,j}\left(\widehat{D}_g^{i-\frac{1}{2},j}\left[\phi_g^{i,j}-\phi_g^{i-1,j}\right]+\widetilde{D}_g^{i-\frac{1}{2},j}\left[\phi_g^{i,j}+\phi_g^{i-1,j}\right]\right)\\
&\quad-\Delta y^{i,j}\left(\widehat{D}_g^{i+\frac{1}{2},j}\left[\phi_g^{i+1,j}-\phi_g^{i,j}\right]+\widetilde{D}_g^{i+\frac{1}{2},j}\left[\phi_g^{i+1,j}+\phi_g^{i,j}\right]\right)\\
&\quad+\Delta x^{i,j}\left(\widehat{D}_g^{i,j-\frac{1}{2}}\left[\phi_g^{i,j}-\phi_g^{i,j-1}\right]+\widetilde{D}_g^{i,j-\frac{1}{2}}\left[\phi_g^{i,j}+\phi_g^{i,j-1}\right]\right)\\
&\quad-\Delta x^{i,j}\left(\widehat{D}_g^{i,j+\frac{1}{2}}\left[\phi_g^{i,j+1}-\phi_g^{i,j}\right]+\widetilde{D}_g^{i,j+\frac{1}{2}}\left[\phi_g^{i,j+1}+\phi_g^{i,j}\right]\right)+\Delta x^{i,j}\Delta y^{i,j}\Sigma_g^{R,i,j}\phi_g^{i,j}\\
&\quad=\Delta x^{i,j}\Delta y^{i,j}\frac{\chi_g^{i,j}}{k_{eff}}\sum_{g'=1}^{G}\nu\Sigma_{g'}^{F,i,j}\phi_{g'}^{i,j}+\Delta x^{i,j}\Delta y^{i,j}\sum_{\substack{g'=1\\ g'!=g}}^{G}\Sigma_{g'\to g}^{S,i,j}\phi_{g'}^{i,j}
\end{aligned}$$

Diffusion coefficients are not required by the MOC solver which is not based on diffusion theory, so they are calculated from the homogenized transport cross section and then group condensed. Even so, the coefficients obtained are for the cell (i,j), but we require the surface diffusion coefficient for each direction ($\widehat{\mathrm{D}}_{\mathrm{g}}^{\mathrm{i}\pm\frac{1}{2},\mathrm{j}\pm\frac{1}{2}}$). It is calculated simply as the average of the current and adjacent cell:

$$\widehat{D}_g^{i+\frac{1}{2},j} = \frac{2D_g^{i,j}D_g^{i+1,j}}{\Delta x^{i+1,j}D_g^{i,j}+\Delta x^{i,j}D_g^{i+1,j}}$$

This is straight forward to calculate for interior cells but requires a different treatment for boundary cells, as no (i+1) cell exists. The 3 boundary conditions implemented in the Scarabée MOC solver are vacuum, reflective, and periodic. At this time, only the reflective boundary condition is implemented for CMFD and simply treats the nonexistent (i+1) cell as the current cell (i) for obtaining the relevant quantities.

Once the surface diffusion coefficients for cell (i,j) are calculated, the nonlinear diffusion coefficients for each direction can also be calculated:

$$\widetilde{\mathrm{D}}_{\mathrm{g}}^{i+\frac{1}{2},j} = \frac{-\widehat{D}_g^{i+\frac{1}{2},j}(\phi_{\mathrm{g}}^{\mathrm{i+1,j}}-\phi_{\mathrm{g}}^{\mathrm{i,j}})-\tilde{J}_g^{i+\frac{1}{2},j}}{(\phi_{\mathrm{g}}^{\mathrm{i+1,j}}+\phi_{\mathrm{g}}^{\mathrm{i,j}})}$$

The loss matrix is cell-ordered so that the linear cell index changes the fastest, i.e. the row indexes are (C0G0, C1G0, C2G0… C0G1, C1,G1, C2,G1). The columns are indexed the same way but for the flux in each cell. As the equations are independent, the matrix can be constructed in parallel using OpenMP. Another caveat is that while the scattering source is technically part of the production matrix, we subtract the scattering terms from the loss matrix instead so that we can more easily compute the source vector Q during the power iteration. With those contributions in mind, we obtain the following sparsity pattern for the loss matrix for a 3 (x) by 3 (y) by 7 (groups) problem:

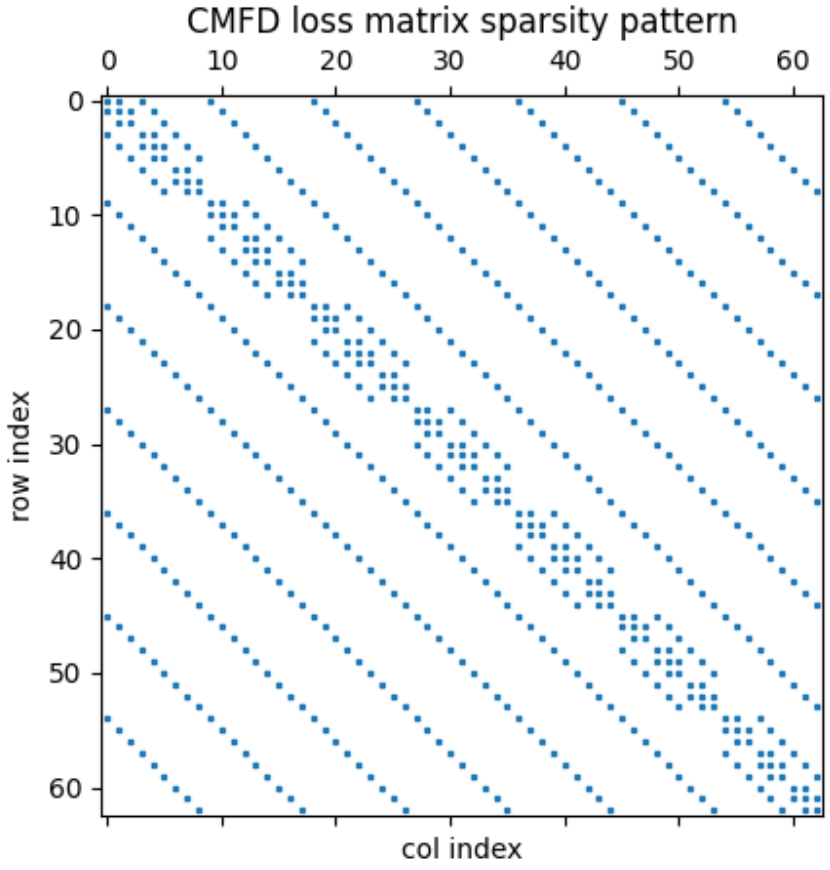

Figure 2: CMFD Loss Matrix Sparsity Pattern

The off-diagonals represent the scattering from other groups, while the main block diagonal contains the terms from the LHS of the cell balance equation.

Once the loss and production matrices are made, we have to solve the following system of equations:

$$\mathrm{M}\phi = \frac{1}{keff} F\phi$$

Where M is the loss matrix and F is the production matrix. Both sides depend upon the flux vector ϕ, which is our unknown, so we provide an initial guess for ϕ and keff, then solve iteratively until our convergence criteria (usually difference in keff and max flux difference) are met. Upon each iteration (n) the source vector Q is computed as such:

$$\mathrm{Q}^{\mathrm{n}} = \frac{1}{\mathrm{k}_{\mathrm{eff}}^{\mathrm{n-1}}} \mathrm{F}\phi^{\mathrm{n-1}}$$

And the system $\mathrm{M}\phi^{\mathrm{n}} = \mathrm{Q}^{\mathrm{n}}$ is solved for ϕ^{n}. Afterwards, our new keff is computed by multiplying the ratio of the new volume-weighed fission source to the old (n-1) fission source:

$$K_{eff}^{n} = K_{eff}^{n-1} \frac{\sum_{g=1}^{G} \sum_{l=0}^{nx*ny} V_l \nu\Sigma_{g,l}^{F} \phi_{g,l}^{n}}{\sum_{g=1}^{G} \sum_{l=0}^{nx*ny} V_l \nu\Sigma_{g,l}^{F} \phi_{g,l}^{n-1}}$$

The final step in the CMFD procedure is the MOC flux update (or prolongation) and has not yet been implemented in Scarabee, as currently the CMFD solver is not producing the correct result (and the latest update broke it so it doesn't converge anymore). The flux in each homogenized cell (i,j) is used to update the flux in each MOC flat source region r belonging to CMFD cell (i,j):

$$\phi_{\mathrm{r,g}} = \phi_{\mathrm{r,g}}{}^{*} \frac{\phi_{\mathrm{g,final}}^{\mathrm{i,j}}}{\phi_{\mathrm{g,initial}}^{\mathrm{i,j}}}$$

Afterwards the MOC driver checks for convergence, and begins another MOC iteration if the criteria are not met.

Author Contributions

Dr. Hunter Belanger wrote the lattice physics code Scarabée including the entirety of its MOC solver. He also wrote the initial framework for the CMFD implementation through the CMFD class. He has advised Jason Gaiardelli on how to implement CMFD.

Jason Gaiardelli has worked to complete the implementation of CMFD in Scarabée, including writing new functions, classes, and test cases for this purpose. He wrote this paper.

References

[1] W. Boyd, S. Shaner, L. Li, B. Forget, and K. Smith, “The OpenMOC Method of Characteristics Neutral Particle Transport Code,” Annals of Nuclear Energy, vol. 68, pp. 43–52, 2014.

[2] P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, B. Forget, and K. Smith, “OpenMC: A state-of-the-art Monte Carlo code for research and development,” Annals of Nuclear Energy, vol. 82, pp. 90–97, 2015.