TECHNIQUES FOR STABILIZING COARSE-MESH FINITE DIFFERENCE (CMFD) IN METHODS OF CHARACTERISTICS (MOC)

Lulu Li, Kord Smith, and Benoit Forget
Department of Nuclear Science & Engineering
Massachusetts Institute of Technology,
77 Massachusetts Avenue, Cambridge, MA 02139
lululi@mit.edu; kord@mit.edu; bforget@mit.edu

ABSTRACT

The Coarse-Mesh Finite Difference (CMFD) method has been widely used to effectively accelerate neutron transport calculations. It was however found to be at times unstable in the presence of strong heterogeneities. The common practice to improve stability is to employ a damping factor on the non-linear diffusion coefficient terms, but there is no method to determine the optimal damping factor for a practical reactor problem prior to the calculation. This paper investigates two problem-agnostic techniques that stabilize reactor calculations that would otherwise diverge with undamped CMFD. The first technique is to perform additional energy sweeps for the upscattering group region during the high-order MOC calculation to generate more accurate information to pass into the CMFD calculation. The second technique extends the traditional scalar flux prolongation to provide spatial variations inside each acceleration cell. This study uses the 2D C5G7 problem and the Babcock & Wilcox 1810 series critical experiment benchmark to evaluate these methods. Numerical simulations showed that both techniques stabilize CMFD, and that the linear prolongation technique did not incur additional computational cost compared to the optimally damped conventional method.

Key Words: Neutron transport theory, methods of characteristics (MOC), acceleration, low order, multigrid, Coarse-Mesh Finite Difference (CMFD), stability.

1 INTRODUCTION

The Coarse-Mesh Finite Difference (CMFD) method has for many years been a popular low-order acceleration method for diffusion and transport calculations. A number of variations have been derived and implemented as summarized in § 2.1. This paper focuses on the original form of CMFD commonly referred to as two-node CMFD (2-N CMFD) [1] [2] [3] or simply as the Smith formulation [4].

The issue of interest in this paper is the instability of this CMFD formulation in certain heterogeneous reactor problems. As summarized in § 2.2, some analysis has been performed to investigate how and why instability occurs, but little work has been published on how to mitigate such divergence behavior. One conventional approach is to under-relax the non-linear diffusion coefficient terms in the CMFD formulation. The drawback is that there is no way to determine the optimal damping factor for a heterogeneous practical problem.
To address the above issues, two techniques are presented and evaluated in this study for multi-group one-level two-node CMFD applied to 2D MOC method. The thermal energy sweeps technique performs additional sweeps over thermal energy, where upscattering occurs, as part of a transport iteration. These additional sweeps improve the accuracy of the data passed to the CMFD solver. The thermal sweeps technique has already been employed in existing codes, and this paper focuses on its ability to stabilize CMFD.

Additionally, this paper proposes a linear prolongation technique. It assumes that the high-order flat source regions (FSRs) are subdivided into sectors of multiples of four (which is commonly used in MOC for achieving reasonable accuracy). Instead of performing prolongation on every FSR in an acceleration cell by the same factor, this method uses a linear interpolation of the current cell’s and the neighboring cell’s prolongation factors.

These two techniques were implemented and evaluated in multi-group one-level two-node CMFD in a 2D MOC framework derived from OpenMOC [5]. The base case configuration for an iteration is to perform one high-order MOC sweep over all space and angle, followed by running the conventional two-node CMFD on a pin cell acceleration mesh using the same energy group structure as the high-order solver. Two sets of benchmark cases were chosen because they showed divergence behavior using the base case configuration. The first is the 2D C5G7 benchmark problem [6] and the second is the Babcock & Wilcox 1810 series critical experiment benchmark [7]. The kernel invocation count, that is, the number of executions of the inner solver loop, is reported for comparing the computational efficiency of the methods.

2 LITERATURE REVIEW

2.1 CMFD Overview

CMFD was first proposed by Smith for nodal diffusion calculations [8], and has been widely used in accelerating neutron diffusion and transport calculations ever since. The following terminologies are commonly used in discussing the various implementations and variations of CMFD:

1. One-level vs. two-level: since CMFD is applicable to general rectangular geometries, it is easy to generalized to multi-grid. One-level refers to one level of acceleration mesh (e.g., pin cell level, or assembly level), and two-level refers to having two layers of acceleration meshes (e.g., assembly level acceleration on top of pin cell level).

2. One-node (1-N) vs. two-node (2-N) CMFD formulation [1-3]. For brevity, the superscript \( I,I+1 \) designates the surface between the \( I \)-th and \( I+1 \)-th acceleration mesh cell, and similarly for \( I,I-1 \).

(a) 1-N CMFD formulation expresses a surface net current in term of the mesh cell scalar flux and the surface flux. For each mesh cell \( I \) and every energy group \( g \),

\[
J_{g}^{I,I\pm 1} = - \hat{D}_{g}^{I} (\phi_{g}^{I,I\pm 1} \mp \phi_{g}^{I}) - \tilde{D}_{g}^{I,I\pm 1} (\phi_{g} + \phi_{g}^{I,I\pm 1})
\]  

(1)
where $J_{g}^{I,I\pm1}$ is the net current crossing the surface between cell $I$ and $I \pm 1$, $\tilde{D}_{g}^{I,I\pm1}$ is the non-linear diffusion coefficient (or current correction factor CCF as in [3]) that couples the $I$-th and $I \pm 1$-th acceleration mesh cell, and $\hat{D}_{g}^{I}$ is the standard finite difference diffusion coupling coefficient that equals to $\frac{2D^{I}}{h^{I}}$ for uniform homogeneous meshes.

Multiple studies mentioned that the advantages of 1-N CMFD are its implementation simplicity and possibility for performing energy condensation during a low-order solve. But 1-N CMFD requires additional spatial sweeps and/or under relaxation to converge [2, 3].

(b) 2-N CMFD was the original version proposed in [8] for nodal diffusion calculations. It relates the net current across a surface with the scalar fluxes in the two adjacent cells:

$$J_{g}^{I,I\pm1} = -\tilde{D}_{g}^{I,I\pm1} (\bar{\phi}_{g}^{I\pm1} - \bar{\phi}_{g}^{I}) - \hat{D}_{g}^{I,I\pm1} (\bar{\phi}_{g}^{I\pm1} + \bar{\phi}_{g}^{I})$$

where $J_{g}^{I,I\pm1}$ is the net current of the surface between the $I$-th and $I + 1$-th acceleration mesh cells, $\tilde{D}_{g}^{I,I\pm1}$ is the finite difference form of diffusion coefficient coupling the two adjacent mesh cells, and $\hat{D}_{g}^{I,I\pm1}$ is the non-linear coupling coefficient.

The results and discussion in this paper are based on a multi-group one-level (pin level) two-node CMFD implemented in a 2D Methods of Characteristics (MOC) framework derived from OpenMOC [5].

2.2 CMFD Instability

CMFD’s potential divergence behavior has been reported in literature [9], [10], [11], [9], [12] and [13]. Most of these studies performed Fourier analysis on the linearized CMFD formulation using the method first proposed by Cefus and Larsen [14], and showed that CMFD can have a spectral radius exceeding unity for certain problems. Masiello investigated the reasons of the divergence behavior for heterogeneous lattice problems by using the local Lyapunov stability theory [12].

3 TECHNIQUES FOR STABILIZING CMFD

The conventional technique for stabilizing CMFD is to apply a damping or under-relaxation factor $\eta$ on the non-linear diffusion coefficient term:

$$\tilde{D}_{g}^{(h)} = \eta \tilde{D}_{g}^{(h)} + (1 - \eta) \tilde{D}_{g}^{(h-1)}$$

Page 3 of 12
where the superscript \((h)\) is the high-order MOC iteration index, and the damping factor \(\eta \in [0, 1]\). For the \((h)\)-th iteration, the actual non-linear coefficient used in the CMFD solver is a weighted sum of the one computed at this iteration and the one computed at the previous iteration \((h - 1)\). When the high-order calculation is far from the converged solution, \(\eta\) effectively under-relaxes the neutron transport term between neighboring cells. As the high-order solution approaches the converged solution, \(\eta\)’s effect is reduced and eventually the net current from the high-order calculation will be preserved at the low-order level.

The issue with the damping approach is that there is no analytical derivation of the optimal damping factor for a practical reactor problem. Hence the optimal damping factor for a problem has to be determined through numerical experiments by performing the same problem with different damping factors and comparing the convergence behavior. While this method is effective if one needs to perform the same or similar problems repeatedly, it is not the ideal stabilization technique, thus motivating us to investigate alternative methods.

This study employs a fixed damping factor scheme (that is, \(\eta\) is a constant independent of the iteration index \(h\)) as the base method. Two methods will be introduced and compared to the fixed damping factor scheme.

3.1 Technique 1: Thermal Energy Sweeps

Recall that the base method is to perform the high-order MOC sweep over all space (angles, tracks) and energy groups once as shown in Algorithm 1 [5]. One technique that has been employed in other MOC codes is to perform additional sweeps over the thermal energy groups where upscattering occurs. To do so, the loop over energy groups has to become the most outer loop. For brevity, we condense the sweep over all space variables into one for loop as shown in Algorithm 2, then the thermal energy sweeps consist of repeating the same calculations except only over the thermal energy groups.

Performing additional sweeps over the thermal energy groups during the high-order transport sweep has the effect of stabilizing CMFD as will be shown in §4.

3.2 Technique 2: Linear Prolongation

The conventional prolongation method for the \(i\)-th FSR scalar flux of the \(g\)-th energy group is to apply a multiplicative update using the \(I\)-th acceleration mesh cell that contains the \(i\)-th FSR:

\[
\phi_{g,i}^{(h+1)} = \frac{\phi_{g,i}^{(h+1/2)}}{\phi_{g,I}^{(h+1/2)}} \phi_{g,I}^{(h+1)}
\]

(4)

where \(\phi_{g,i}\) is the \(i\)-th FSR scalar flux and \(\phi_{g,I}\) is the \(I\)-th mesh cell cell-homogenized scalar.
Algorithm 1 A base transport iteration for OpenMOC

```plaintext
for all m ∈ M do
    for all k ∈ K(m) do
        for all s ∈ S(k) do
            for all g ∈ G do
                for all p ∈ P do
                    i ← I(s)
                    ∆Ψ_{k,i,g,p} ← (Ψ_{k,g,p} - \frac{Q_i}{\Sigma_{i,g}}) (1 - e^{-\tau_{k,i,g,p}})
                    Ψ_{1,g} ← Ψ_{1,g} + \frac{4\pi}{\Sigma_{k,g}} \omega_m \omega_p \omega_k \sin \theta_p \Delta Ψ_{k,i,g,p}
                    Ψ_{k,g,p} ← Ψ_{k,g,p} - ∆Ψ_{k,g,p}
            end for
        end for
    end for
end for
Ψ_{k,g,p}^{(0)} ← Ψ_{k,g,p} or 0
L ← L + Ψ_{k,g,p}
end for
Update k_{eff} and FSR sources Q_{i,g} ∀ i,
```

Algorithm 2 A transport iteration with the option of thermal energy sweeps

```plaintext
for all g ∈ G do
    for all m ∈ M, k ∈ K(m), s ∈ S(k), p ∈ P do
        Increment FSR scalar flux, update outgoing flux, tally leakage
    end for
end for
Update FSR sources Q_{i,g} ∀ i, g
if Perform thermal sweeps then
    for all g ∈ G_{thermal} do
        for all m ∈ M, k ∈ K(m), s ∈ S(k), p ∈ P do
            Increment FSR scalar flux, update outgoing flux, tally leakage
        end for
    end for
    Update FSR sources Q_{i,g} ∀ i, g
end if
```

flux. The superscript \((h + 1/2)\) designates results computed from the \(h\)-th higher order MOC calculation which is then used to construct the CMFD solve. The superscript \((h + 1)\) designates the converged CMFD results.

By monitoring these prolongation factors during a calculation where undamped CMFD diverges, one notices that at certain mesh cells, the mesh cell prolongation factor \(\phi_{\text{g},l}^{(h+1)}\) often times oscillates right around unity between successive iterations especially during earlier iterations. A natural extension is to provide the prolongation with some spatial variation inside of an acceleration cell.

The method proposed in this paper assumes that the high-order MOC discretizes the FSRs into multiples of four sectors which is a common practise in MOC solvers. For example, Fig. 1
illustrates two pin cells where the fuel region is discretized into four radial regions with four sectors, and the moderator region is discretized into three radial regions with eight sectors.

Define the left mesh cell in Fig. 1 as the *I*-th cell and the right one as the *I* + 1-th cell, then the conventional prolongation method would update all the FSRs in cell *I* with the same factor for each energy group.

The linear prolongation technique simply performs a linear interpolation using the prolongation factors of neighboring cells. For example, FSRs in the sector shaded in blue in Fig. 1 are now updated by a weighted sum of the prolongation factors of cell *I* and *I* + 1:

\[
\phi_{g,i}^{(h+1)} = \phi_{g,i}^{(h+1/2)} \left[ \frac{2}{3} \phi_{g,I}^{(h+1/2)} + \frac{1}{3} \phi_{g,I+1}^{(h+1/2)} \right]
\]

where the coefficients \(\frac{1}{3}, \frac{2}{3}\) are based on the geometric centroids.

![Figure 1. Illustration of applying linear prolongation](image)

This is a fairly simple extension to the conventional FSR scalar flux prolongation method by applying basic geometric information. It is worth mentioning that the linear prolongation technique is not very sensitive to the choice of the weighting factors (e.g., \((\frac{1}{3}, \frac{2}{3})\) vs. \((\frac{1}{2}, \frac{1}{2})\)). Furthermore a more complex scheme was evaluated where the true centroid of each FSR was used to calculate the weighting factors. No significant advantage over the simple linear prolongation was observed in numerical experiments, thus the simple linear prolongation method is presented in this paper.
4 RESULTS AND DISCUSSION

4.1 Problem descriptions

The base case configuration for an iteration is to perform one high-order MOC sweep over all space and angle, followed by running the conventional two-node CMFD on pin cell acceleration mesh until convergence using the same energy group structure as the high-order solver.

Two types of benchmark problems have been observed to have divergence issue:

1. 2D C5G7 benchmark problem [6] with two MOX and two UO₂ fuel assemblies surrounded by water as illustrated on the left in Fig. 2. Each assembly contains 17x17 pin cells, and each pin cell is further subdivided into radial regions and sectors as illustrated on the right in Fig. 2. All the results were generated using MOC representation of 64 azimuthal angles, three polar angles and 0.05cm track spacing, which generated 52,420 tracks and 25,879,848 segments. CMFD was performed on pin cell level with seven energy groups. The optimal damping factor for this problem was determined to be 0.7 through numerical experiments.

![Figure 2. 2D C5G7 problem: material distribution (left) and a single pin cell FSR discretization (right)](image)

2. Babcock & Wilcox 1810 series critical experiment benchmark (BAW1810) [7]. This benchmark consists of 20 cores representing realistic core configurations. They are mostly 5x5
array of 15x15 PWR assemblies containing a collection of different boron concentration, gadolinium fuel pins, Ag-In-Cd (AIC), B₄C rods or hollow rods depending on the core.

This benchmark set was performed using the same MOC representation as in the 2D C5G7 benchmark problem. The high-order MOC used eight-group cross-sections generated from CASMO-5 [15]. CMFD was performed on pin cell level with eight energy groups, and the optimal damping factor was 0.5.

In this paper, the stopping/convergence criteria is set to be $10^{-5}$ on RMS of pin cell fission source successive iteration fractional change (referred to as “RMS of fractional change” for brevity) defined as the following:

$$
\epsilon = \sqrt{\frac{\sum_{I} \left( \frac{F^h_I}{F^{h-1}_I} - 1 \right)^2}{\# \text{ non-zero } F^{h-1}_I}}
$$

(6)

where $F^h_I$ is pin cell I’s fission source summed over all energy groups at iteration $h$.

4.2 Results: thermal energy sweeps stabilizes CMFD

Recall that one transport iteration in the base case (“no thermal sweeps”) means sweeping over all space and energy once, whereas “with thermal sweeps” means the base case followed by additional sweeps over all space in the thermal group region where upscattering occurs.

Table I shows the number of transport iterations required to converge to $10^{-5}$ RMS of fractional change using different damping factor $\eta$. Without thermal sweeps, the optimal $\eta$ is 0.7, and CMFD diverges with $\eta >= 0.9$. With thermal sweeps, CMFD converges in less transport iterations, and does not require any damping to converge.

This paper reports kernel invocation count, the number of executions of the inner solver loop, as a better proxy for computation time than the iteration count, the number of executions of the outer solver loop. In this case, a kernel invocation is the execution of flux calculations at an azimuthal angle, a segment, an energy group and a polar angle. This is the basic unit of an MOC calculation.

Because all the scenarios sweep through the same number of spatial variables, the difference comes down to the number of energy sweeps performed. Without thermal sweeps, a transport iteration sweeps through 7 energy groups. With thermal sweeps, a transport iteration sweeps through 11 ($= 7 + 4$) energy groups, making its kernel invocation count 57% higher than the no thermal sweep case. Using the optimal damped no thermal sweeps case as the base, the relative kernel invocation counts are also reported in Table I.
With thermal sweeps, the undamped CMFD consumes 5% higher computational time than the optimally damped base case. But if a slight damping is applied ($\eta = 0.9$ for example), then the thermal sweeps technique reduces the computational time 13% further compared to the optimally damped base case.

<table>
<thead>
<tr>
<th>$\eta$</th>
<th>No thermal sweeps</th>
<th>With thermal sweeps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># transport iterations</td>
<td>Relative kernel invocation count</td>
</tr>
<tr>
<td>0.5</td>
<td>19</td>
<td>1.06</td>
</tr>
<tr>
<td>0.6</td>
<td>19</td>
<td>1.06</td>
</tr>
<tr>
<td>0.7</td>
<td>18</td>
<td>1 (base case)</td>
</tr>
<tr>
<td>0.8</td>
<td>27</td>
<td>1.5</td>
</tr>
<tr>
<td>0.9</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>1.0</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table I. Number of transport iterations and relative kernel invocation count for C5G7 benchmark problem

Fig. 3 plots the number of transport iterations to converge the BAW1810 problems. The base case (no thermal sweeps, optimal damping of $\eta = 0.5$) is compared with the undamped thermal sweeps case. Similar to the C5G7 problem, thermal energy sweeps stabilize CMFD and removes the need for damping. The number of transport iterations is lower with thermal sweeps, but each iteration’s kernel invocation count is 1.5 times that without thermal sweeps (8 total energy groups and 4 thermal groups). Thermal sweeps stabilize CMFD in these cases, and it could introduce an overhead of up to 20-30%.

4.3 Results: linear prolongation stabilizes CMFD

The effect of linear prolongation is summarized in Table II with the stopping criteria being $10^{-5}$ on RMS fractional change. For the cases tested, the linear prolongation method with no damping factor produced very similar, if not identical convergence behavior to the conventional prolongation method with optimal damping ($\eta = 0.7$ for 2D C5G7 and $\eta = 0.5$ for BAW1810). The computational overhead of the linear prolongation method is minimal because it uses existing information, like prolongation factor for each acceleration cell, and simply performs a weighted summation using factors from two neighboring cells. Depending on the implementation of the program, some additional work is needed to determine which sector of the acceleration cell that a FSR belongs to, though this has introduced negligible overhead in this study.

The linear prolongation method eliminates the dependency on any damping factor while keeping the computational cost at roughly the same as the optimally damped conventional method.

5 CONCLUSION

This paper reported the need for an under-relaxation or damping factor when applying CMFD in a basic configuration (2D MOC sweeping through space and energy once accelerated by a
Figure 3. Number of transport iterations for BAW1810 problems with and without thermal energy sweeps

multi-group one-level two-node CMFD) to certain heterogeneous reactor problems. The focus of this investigation was not to address why such divergence occurs as it has been discussed in previous literature summarized in \[\text{§2.2}\]. Rather this study is interested in ways to mitigate the need for damping factors, because there is no elegant way to determine the optimal damping factor given a practical reactor problem.

To this end, two sets of benchmark problems were selected because they showed divergence behavior when using undamped CMFD. Their optimal damping factors were determined through numerical experiment, and the optimally damped convergence behavior served as the baseline. Two techniques were investigated. The first one is to perform additional sweeps over the thermal energy groups where upscattering occurs during a high-order MOC iteration. The second technique, the linear prolongation method, is a simple extension of the conventional prolongation method by using a weighted summation of the current cell’s and its neighbors’ scalar flux prolongation factors.

Both methods presented in this work have been effective at removing the need for damping factors, making them more problem-agnostic than the conventional damping factor approach. The thermal energy sweep technique was able to reduce the number of transport iterations required for convergence, but after factoring in the extra energy sweeps performed at each transport iteration, it could require up to 20-30% overhead in computational requirements compared to the optimally damped base case. It is worth mentioning that applying a damping factor reduces this method’s computational cost to below that of the optimally damped base case. So this method presented two
CMFD Stabilization Techniques

<table>
<thead>
<tr>
<th></th>
<th>Conventional prolongation with optimal $\eta^*$</th>
<th>Linear prolongation with no damping</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D C5G7</td>
<td>18</td>
<td>16</td>
</tr>
<tr>
<td>1810 core 1</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 2</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 3</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 4</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 5</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 10</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 12</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>1810 core 13</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>1810 core 14</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>1810 core 15</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 16</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 17</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 18</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 19</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>1810 core 20</td>
<td>19</td>
<td>18</td>
</tr>
</tbody>
</table>

Table II. Number of transport iterations for convergence. Linear prolongation stabilizes CMFD, while regular prolongation needs a damping factor $\eta$ to stabilize CMFD. *C5G7 optimal $\eta = 0.7$, and BAW1810 optimal $\eta = 0.5$.

possibilities comparing to the optimally damped base case: either eliminating the need for under-relaxation entirely and still converging stably though at a higher cost, or applying a small under-relaxation and consuming less computational time than the optimally damped base conventional method.

The linear prolongation technique is an attractive alternative to the conventional prolongation method, as it retains the computational efficiency and stability of the optimally damped conventional method, while removing the need to determine an optimal damping factor.

6 REFERENCES


