212 Project Report
Apply CMFD on Random Ray MOC

Zhuoran Han

16 Dec

1 Introduction

The method of characteristics (MOC) method is one the mostly used technique for solving partial differential equation (PFE). In reactor physics, this method is applied to solve neutron transport equation, also known as Boltzmann equation. The essence of the method is to solve a PDE on a characteristic line. The notations of the following derivation is adapted from Samuel Shaner's master thesis [1].

We start with steady state multi-group neutron transport equation:

$$\Omega \cdot \nabla \psi(r, \Omega) + \Sigma_g^{tr}(r)\Psi(r, \Omega) = Q_g(r)$$

where g represents the group index, r is the spatial position, and Ω is the angular direction.

From some basic math derivation, we can obtain the characteristic form of the transport equation:

$$\frac{d\psi_g(s)}{ds} + \Sigma_g^{tr}(s)\psi_g(s) = Q_g(s)$$

where s the distance travelled along a characteristic line.

The right hand side of the equation is the source term. We use a flat source approximation. The geometry can be divided into many flat source region (FSR), which means the source is constant inside a region. In each FSR, the material properties are also constant. To simplify the problem, fission and scattering are all isotropic. The source term for group g in side an FSR can then be given as:

$$Q_{r,g} = \frac{1}{4\pi}(\frac{\chi_{r,g}}{k_{eff}} \sum_{g'=1}^{G} \nu\Sigma_{r,g'}^{F}\phi_{r,g`} + \sum_{g'=1}^{G} \Sigma_{r,g'\to g}^{S}\phi_{r,g'})$$

For a track k in region r, the characteristic equation of this segment is given by:

$$\frac{d\psi_{k,r,g}(s)}{ds} + \Sigma_g^{tr}(s)\psi_{k,r,g}(s) = Q_{r,g}(s)$$

If we integrate this equation from an entry point s' to an exiting point s'', we then have:

$$\psi_{k,g}(s'') = \psi_{k,g}(s')e^{-\tau_{k,r,g}} + \frac{Q_{r,g}}{\Sigma_{r,g}^{tr}}(1 - e^{-\tau_{k,r,g}})$$

The change of angular flux on this segment of ray is give by following.

$$\Delta\psi_{k,g} = (\psi_{k,g}(s')e - \frac{Q_{r,g}}{\Sigma^{tr}_{r,g}})(1 - e^{-\tau_{k,r,g}})$$

The average angular flux contribution to this segment of track k in region is :

$$\begin{aligned}\bar{\psi}_{k,r,g} &= \frac{1}{l_{k,r}} \int_{s'}^{s''} \psi_{k,r,g}(s)ds \\ &= \bar{\psi}_{k,r,g} = \frac{1}{l_{k,r}}[\frac{\psi_{k,g}(s')}{\Sigma^{tr}_{r,g}} e^{-\tau_{k,r,g}} + \frac{l_{k,r}Q_{r,g}}{\Sigma^{tr}_{r,g}}(1 - \frac{e^{-\tau_{k,r,g}}}{\tau_{k,r,g}})]\end{aligned}$$

where $l_{k,r} = s'' - s'$.

The FSR-averaged scalar flux in region r is given by:

$$\phi_{r,g} = \frac{1}{A_r} \int_A dA \int_{4\pi} d\Omega \psi_{k,r,g}$$

If we use quadrature rules to perform this integration, we can get:

$$\phi_{r,g} = \frac{4\pi}{A_r} \sum_{k \in A_r} \sum_{p=1}^{P} w_{m(k)} w_p w_k l_{k,r} \bar{\psi}_{k,r,g}$$

where $w_{m(k)} = \frac{\Delta m(k)}{2\pi}$ is the azimuthal quadrature weight, w_k is the ray spacing, and w_p is polar quadrature weight. $\Delta m(k)$ represents the angular space represented by tracks with azimuthal angle $m(k)$. For exmample, if there are 4 angles, each $\Delta m(k)$ takes a space of $\frac{\pi}{2}$.

If we only consider the problem in 2D with no polar quadrature, which means $w_p = 1$ and $p = 1$, we have:

$$\phi_{r,g} = \frac{4\pi}{A_r} \sum_{k \in A_r} w_{m(k)} w_k l_{k,r} \bar{\psi}_{k,r,g}$$

After some mathematical manipulation, we can express the scalar flux inside an FSR as a function of the sum of all changes of angular flux in this FSR.

$$\phi_{r,g} = \frac{4\pi}{\Sigma^t_{r,g} r}[Q_{r,g} + \frac{1}{A_r} \sum_{k \in A_r} w_{m(k)} w_k \Delta\psi_{k,r,g}]$$

As for current J crossing a surface, it can be calculate as:

$$\begin{aligned}J^{surf}_g &= \sum_{k \in surf} 2\pi w_{m(k)} \frac{w_k}{\cos\theta_k} \psi_{k,r,g} \cdot \hat{n} \\ &= \sum_{k \in surf} 2\pi w_{m(k)} w_k \psi_{k,r,g}\end{aligned}$$

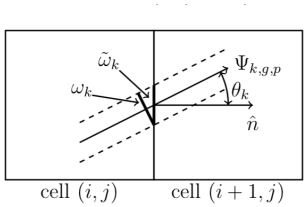

Figure 1: Current J crossing surface

The schematic is shown in Figure 1. The ray spacing term accounts for an integration of area. Therefore, this expression of current is the net current tallied over surface, not surface-averaged current.

The total leakage from the system is the current leaking out from all vacuum boundaries.

$$L = \sum_{\text{Vacuum B.C}} \sum_{g=1}^{G} J_g^{surf}$$

The k_{eff} is calculated as the total generation of neutrons divided by the total loss of neutrons.

$$k_{eff} = \frac{\sum_{r=1}^{R} \sum_{g=1}^{G} \nu \Sigma_{r,g}^{F} \phi_{r,g} A_r}{L + \sum_{r=1}^{R} \sum_{g=1}^{G} \Sigma_{r,g}^{A} \phi_{r,g} A_r}$$

The overall algorithm implemented in OpenMOC is given in Figure 2 and Figure 3.

Algorithm 1 Fixed source iteration for OpenMOC

$\Phi_{r,g} \leftarrow 0 \quad \forall\, r, g \in \{R, G\}$ # Initialize FSR scalar fluxes to zero
while $Q_{r,g}\ \forall\, r$ not converged **do**
  for all $m \in M$ **do** # Loop over azimuthal angles
    for all $k \in K(m)$ **do** # Loop over tracks
      for all $s \in S(k)$ **do** # Loop over segments
        for all $g \in G$ **do** # Loop over energy groups
          for all $p \in P$ **do** # Loop over polar angles
            $r \leftarrow R(s)$ # Get FSR for this segment
            $\Delta\Psi_{k,r,g,p} \leftarrow \left(\Psi_{k,g,p} - \frac{Q_{r,g}}{\Sigma_{r,g}^{tr}}\right)\left(1 - e^{-\tau_{k,r,g,p}}\right)$
            $\Phi_{r,g} \leftarrow \Phi_{r,g} + \frac{4\pi}{A_r}\omega_{m(k)}\omega_p\omega_k \sin\theta_p l_{k,r} \Delta\Psi_{k,r,g,p}$
            $\Psi_{k,g,p} \leftarrow \Psi_{k,g,p} - \Delta\Psi_{k,g,p}$
          end for
        end for
      end for
    end for
    if B.C. are reflective **then** # Set incoming flux for outgoing track
      $\Psi_{k',g,p}(0) \leftarrow \Psi_{k,g,p}$ # Reflective B.C.'s
    else
      $\Psi_{k',g,p}(0) \leftarrow 0$ # Vacuum B.C.'s
      $L \leftarrow L + 2\pi\omega_{m(k)}\omega_p\omega_k \sin\theta_p \Psi_{k,g,p}$
    end if
  end for
  Update k_{eff} and $Q_{r,g}\ \forall\, r$ # Equation 2.17 and Algorithm 2
end while

Figure 2: MOC main Algorithm

Algorithm 2 FSR source update for OpenMOC

for all $r \in R$ **do** # Loop over FSRs
  for all $g \in G$ **do** # Loop over energy groups
    $Q_{r,g}^{(n+1)} \leftarrow \frac{\chi_{r,g}}{4\pi k_{eff}} \nu \Sigma_{r,g}^{F} \Phi_{r,g}^{(n)}$ # Initialize new total source with fission
    for all $g' \in G$ **do** # Loop over energy groups
      $Q_{r,g}^{(n+1)} \leftarrow Q_{r,g}^{(n+1)} + \frac{1}{4\pi} \Sigma_{r,g' \rightarrow g}^{S} \Phi_{r,g'}^{(n)}$
    end for
  end for
end for

Figure 3: update FSR source algorithm

2 Methodology

2.1 Random Ray MOC

What Random Ray MOC differs from conventional MOC is how to evaluate the following integral:

$$\phi_{r,g} = \frac{1}{A_r} \int_A dA \int_{4\pi} d\Omega \psi_{k,r,g}$$

.

In conventional MOC, quadrature integration is used, where as Random Ray MOC utilize Monte Carlo Integration.

The algorithm for Random Ray MOC in John Tramm's dissertation is given in the following figures [2].

Algorithm 1 MOC Power Iteration

1: Initialize Scalar Fluxes to 1.0
2: **while** K-effective and Scalar Flux Unconverged **do**
3: Compute Source (Equation 2.2)
4: Set Scalar Flux to Zero
5: Transport Sweep (Algorithm 2)
6: Normalize Scalar Flux to Sum of Ray Distances
7: Add Source to Scalar Flux (Equation 2.3)
8: Calculate K-effective
9: **end while**

Figure 4: Random Ray Algorithm 1

$$V_r = \frac{d_r}{D_{\text{total}}} \tag{2.1}$$

$$Q_{r,e} = \frac{1}{4\pi\Sigma_{t,r,e}} \left[S_{r,e} + \frac{1}{k} F_{r,e} \right] \tag{2.2}$$

$$\phi_{r,e} = \frac{\phi_{r,e}}{\Sigma_{t,r,e} V_r} + 4\pi Q_{r,e} \tag{2.3}$$

Algorithm 2 Transport Sweep

```
for all Rays do
    Distance Travelled D = 0
    Generate Randomized Ray (r̂, Ω̂)
    Apply Ray Starting Flux Condition
    while D < Termination Distance do
        Set Nearest Neighbor distance s = ∞
        for all CSG Cell Neighbors do
            Ray Trace to Find Distance s_n to Neighbor Surface
            if s_n < s then
                s = s_n
            end if
        end for
        Attenuate Segment s (Algorithm 3)
        D = D + s
        Move Ray Forward or Reflect
    end while
end for
```

Figure 5: Random Ray Algorithm 2

Algorithm 3 Attenuate Segment

```
for all Energy Groups g ∈ G do
    Δψ_g = (ψ_g − Q_{r,g}) (1 − e^{−Σ_{t,r,g} s})
    φ_{r,g} = φ_{r,g} + 4πΔψ_g
    ψ_g = ψ_g − Δψ_g
end for
```

Figure 6: Random Ray Algorithm 3

However, this sweeping and attenuation process is more from an intuition rather than math derivation. A derivation for Random Ray MOC is needed. A preliminary attempt to show this process is given below. If we assume that angular flux is isotropic and rays in each direction have the same weight, we then have

$$\phi_{r,g} = 4\pi\bar{\psi}_{r,g}$$

where $\bar{\psi}_{r,g}$ represents the true average angular flux.

$$\begin{aligned}
\phi_{r,g} &= 4\pi\bar{\psi}_{r,g} \\
&= \frac{4\pi}{l_{k,r}}[\frac{\psi_{k,g}(s')}{\Sigma^{tr}_{r,g}}e^{-\tau_{k,r,g}} + \frac{l_{k,r}Q_{r,g}}{\Sigma^{tr}_{r,g}}(1 - \frac{e^{-\tau_{k,r,g}}}{\tau_{k,r,g}})] \\
&= \frac{4\pi}{l_{k,r}}[\frac{l_{k,r}Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{\Delta\bar{\psi}_{k,g}}{\Sigma^{tr}_{r,g}}]
\end{aligned}$$

We can rearrange the expression for $\phi_{r,g}$:

$$\begin{aligned}
\phi_{r,g}l_{k,r} &= \frac{l_{k,r}4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{4\pi\Delta\bar{\psi}_{k,g}}{\Sigma^{tr}_{r,g}} \\
l_{k,r}(\phi_{r,g} - \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}}) &= \frac{4\pi\Delta\bar{\psi}_{k,g}}{\Sigma^{tr}_{r,g}} \\
\phi_{r,g} &= \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{4\pi\Delta\bar{\psi}_{k,g}}{\Sigma^{tr}_{r,g}l_{k,r}}
\end{aligned}$$

In order to obtain a good estimation of $\phi_{r,g}$, we simply take the average value from N contributions in this FSR.

$$\begin{aligned}
\phi_{r,g} &= \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{4\pi\frac{\sum_{i=1}^{N}\Delta\psi^i_{k,g}}{N}}{\Sigma^{tr}_{r,g}\frac{\sum_{i=1}^{N}l^i_{k,r}}{N}} \\
&= \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + 4\pi\frac{\sum_{i=1}^{N}\Delta\psi^i_{k,g}}{\Sigma^{tr}_{r,g}\sum_{i=1}^{N}l^i_{k,r}} \\
&= \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{\sum_{i=1}^{N}4\pi\Delta\psi^i_{k,g}}{\Sigma^{tr}_{r,g}d_r}
\end{aligned}$$

where d_r is the total distance travelled by all rays in FSR region r. This process is the same as in John Tramm's thesis [2].

$$\begin{aligned}
\phi_{r,g} &= \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{(\sum_{i=1}^{N}4\pi\Delta\psi^i_{k,g})/D_{total}}{\Sigma^{tr}_{r,g}V_r} \\
&= \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{(\sum_{i=1}^{N}4\pi\Delta\psi^i_{k,g})/D_{total}}{\Sigma^{tr}_{r,g}d_r/D_{total}} \\
&= \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{\sum_{i=1}^{N}4\pi\Delta\psi^i_{k,g}}{\Sigma^{tr}_{r,g}d_r}
\end{aligned}$$

If we compare the expression for $\phi_{r,g}$ in conventional MOC and random MOC, we can conclude that area of FSR in random ray MOC is defined as the the total distance tallied in this FSR.

$$\phi_{r,g} = \frac{4\pi}{\Sigma^t_{r,g} r}[Q_{r,g} + \frac{1}{A_r}\sum_{k\in A_r} w_{m(k)} w_k \Delta\psi_{k,r,g}]$$

$$\phi_{r,g} = \frac{4\pi}{\Sigma^t_{r,g} r}[Q_{r,g} + \frac{1}{d_r}\sum_{i\in FSR} \Delta\psi^i_{k,g}]$$

From a simple dimensional analysis, we can see that this two expressions have the same unit for $\phi_{r,g}$ as $cm^{-2} \cdot s^{-1}$.

The next task is how to tally the current along the sweeping process. This part could be a little tricky. Let's first go back and see how we treat this problem in conventional MOC. The expressions for leakage and multiplication factor are given as follow.

$$J^{surf,MOC}_g = \sum_{k\in surf} 2\pi w_{m(k)} w_k \psi_{k,r,g}$$

$$L^{MOC} = \sum_{\text{Vacuum B.C}} \sum_{g=1}^{G} J^{surf}_g$$

$$k^{MOC}_{eff} = \frac{\sum_{r=1}^{R}\sum_{g=1}^{G} \nu\Sigma^F_{r,g}\phi_{r,g}A_r}{L + \sum_{r=1}^{R}\sum_{g=1}^{G} \Sigma^A_{r,g}\phi_{r,g}A_r}$$

The leakage term in the above expressions is net leakage through a surface. This can be shown by the unit of J and L. Since w_k has a unit of cm and $\psi_{k,r,g}$ is in $cm^{-2}s^{-1}$, the leakage term is then in a unit of $cm^{-1}s^{-1}$.

The macroscopic cross sections have a unit in cm^{-1}, so both fission term and absorption term have units of $cm^{-1}s^{-1}$, which match the leakage term.

For random ray, the current crossing a surface is tallied as:

$$J^{surf}_g = \frac{1}{N}\sum_{k\in surf} 2\pi\psi_{k,r,g} \cdot (\Omega \cdot \hat{n})$$

A mathematical explanation is given below. To tally currents on the right most surface paralleled with x-axis, we have the expression as:

$$J^+ = \int_{-1}^{1}\int_{0}^{\pi} (\Omega \cdot \hat{n})\psi_{k,r,g} d\mu d\theta$$

If we utilize Monte Carlo integration, we have:

$$J^+ = \frac{\sum_{i=1}^{N} 2\pi(\Omega^i \cdot \hat{n})\psi^i_{k,r,g}}{N}$$

where N means the number of crossings over the surface. It is easy to show J accumulated in this approach has the unit of ψ, $cm^{-2}s^{-1}$, which means J is a surface-averaged value.

To update the multiplication factor, we use:

$$k_{eff} = \frac{\sum_{r=1}^{R}\sum_{g=1}^{G}\nu\Sigma_{r,g}^{F}\phi_{r,g}d_r/D_{total}}{L+\sum_{r=1}^{R}\sum_{g=1}^{G}\Sigma_{r,g}^{A}\phi_{r,g}d_r/D_{total}}$$

Note that the fission term and absorption term is this expression has the same unit of reaction rate, $cm^{-3}s^{-1}$. The term $\frac{d_r}{D_total}$ represents the relative volume of a FSR when we have the overall volume is 1. This term is unitless. To balance the equation, we have to make L also in the unit of $cm^{-3}s^{-1}$. However, the tallied current J has unit $cm^{-2}s^{-1}$, a length term in cm should be divided on that. An intuitive guess would be D_{total}. However, if L is divided by a very long travel distance, this leakage term becomes extremely small comparing to the reaction rate terms while the real situation has a 70% leakage.

2.2 Coarse Mesh Finite Difference (CMFD)

Coarse Mesh Finite Difference (CMFD) is one of the most widely adopted acceleration scheme due to its simplicity and high acceleration performance. CMFD utilize a finite difference diffusion solver to reduce the number of iterations required for convergence. What differs it from regular finite difference method is the non-linear diffusion acceleration (NDA) scheme.

2.2.1 Condensation

CMFD starts with condensation process. For spatial condensation, cross sections and fluxes on finer grids need to be condensed on coarser grids. For energy condensation, multiple energy groups will be condensed into fewer energy groups. If there is no gap or extremely low density gas in the system, the condensation process is given in Figure 7. If there exists very low density area, the diffusion coefficients themselves will first be flux weighted. Then, the transport cross sections need to be flux weighted and the condensed diffusion coefficient is calculated by condensed transport cross sections.

$$\Sigma_{\mathbf{g}}^{A,i,j} = \frac{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\Sigma_{r,g}^{A}\Phi_{r,g}A_r}{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\Phi_{r,g}A_r} \qquad D_{\mathbf{g}}^{i,j} = \frac{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\frac{1}{3\Sigma_{r,g}^{tr}}\Phi_{r,g}A_r}{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\Phi_{r,g}A_r}$$

$$\Sigma_{\mathbf{g}}^{F,i,j} = \frac{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\Sigma_{r,g}^{F}\Phi_{r,g}A_r}{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\Phi_{r,g}A_r} \qquad \chi_{\mathbf{g}}^{i,j} = \frac{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\sum_{g\prime=1}^{G}\chi_{r,g}\nu\Sigma_{r,g\prime}^{F}\Phi_{r,g\prime}A_r}{\sum_{r\in(i,j)}\sum_{g\prime=1}^{G}\sum_{g\prime\prime=1}^{G}\chi_{r,g\prime\prime}\nu\Sigma_{r,g\prime}^{F}\Phi_{r,g\prime}A_r}$$

$$\nu\Sigma_{\mathbf{g}}^{F,i,j} = \frac{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\nu\Sigma_{r,g}^{F}\Phi_{r,g}A_r}{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\Phi_{r,g}A_r} \qquad \phi_{\mathbf{g}}^{i,j} = \frac{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\Phi_{r,g}A_r}{\sum_{r\in(i,j)}A_r}$$

$$\Sigma_{\mathbf{g}\to\mathbf{g}\prime}^{S,i,j} = \frac{\sum_{g\in\mathbf{g}}\sum_{g\prime\in\mathbf{g}\prime}\sum_{r\in(i,j)}\Sigma_{r,g\to g\prime}^{S}\Phi_{r,g}A_r}{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)}\Phi_{r,g}A_r}$$

Figure 7: Condensation in Energy and Space

For this project,the CMFD solver is essentially a 2D finite difference solver to solve the following transport equation.

$$
\begin{aligned}
&-\frac{\partial}{\partial x}\cdot D_g(x,y)\frac{\partial}{\partial x}\phi_g(x,y)-\frac{\partial}{\partial y}\cdot D_g(x,y)\frac{\partial}{\partial y}\phi_g(x,y)\\
&+(\Sigma_g^A(x,y)+\sum_{g'=1,g'\neq g}^{G}\Sigma_{g\to g'}^S(x,y))\phi_g(x,y)\\
&=\frac{\chi_g(x,y)}{k_eff}\sum_{g'=1}^{G}\nu\Sigma_{g'}^F(x,y)\phi_{g'}(x,y)+\sum_{g'=1,g'\neq g}^{G}\Sigma_{g'\to g}^S(x,y)\phi_{g'}(x,y)
\end{aligned}
$$

The discretized equation is given as:

$$
\begin{aligned}
&-\Delta(J_g^{i-1/2,j}-J_g^{i+1/2,j}))-\Delta(J_g^{i,j-1/2}-J_g^{i,j+1/2}))\\
&+\Delta^2(\Sigma_g^{A,i,j}+\sum_{g'=1,g'\neq g}^{G}\Sigma_{g\to g'}^{S,i,j})\phi_g^{i,j}-\Delta^2\sum_{g'=1,g'\neq g}^{G}\Sigma_{g'\to g}^{S,i,j}\phi_{g'}^{i,j}\\
&=\Delta^2\frac{\chi_g^{i,j}}{k_eff}\sum_{g'=1}^{G}\nu\Sigma_{g'}^{F,i,j}\phi_{g'}^{i,j}
\end{aligned}
$$

For Fick's Law, we can derive the linear diffusion coefficient as:

$$
\begin{aligned}
J^{i+1/2,j}&=-\hat{D}_g^{i+1/2,j}(\phi_g^{i+1,j}-\phi_g^{i,j})\\
\hat{D}_g^{i+1/2,j}&=\frac{2*D_g^{i,j}*D_g^{i+1,j}}{\Delta(D_g^{i,j}+D_g^{i+1,j})}
\end{aligned}
$$

In order to conserve neutron balance between the CMFD and Random Ray MOC, the net currents across the coarse mesh surfaces must be the same. However, there is no guarantee that these two values will be equal. A non-linear diffusion correction term is introduced to fix this problem.

$$
\frac{\hat{J}_g^{i+1/2,j}}{\Delta}=-\hat{D}_g^{i+1/2,j}(\phi_g^{i+1,j}-\phi_g^{i,j})-\tilde{D}_g^{i+1/2,j}(\phi_g^{i+1,j}+\phi_g^{i,j})
$$

where $\frac{\hat{J}_g^{i+1/2,j}}{\Delta}$ is the surface-averaged current on the corresponding surface, and $\tilde{D}$ is the non-linear correction. Rearrange the above equation, we have the non-linear diffusion term as:

$$
\tilde{D}_g^{i+1/2,j}=\frac{-\hat{D}_g^{i+1/2,j}(\phi_g^{i+1,j}-\phi_g^{i,j})-\frac{\hat{J}_g^{i+1/2,j}}{\Delta}}{\phi_g^{i+1,j}+\phi_g^{i,j}}
$$

The final finite difference form of the diffusion equation over a coarse mesh at position

(i,j) in energy group g is given as:

$$\begin{aligned}
&\Delta\Big(\hat{D}_g^{i-1/2,j}(\phi_g^{i-1,j}-\phi_g^{i,j})+\tilde{D}_g^{i-1/2,j}(\phi_g^{i-1,j}+\phi_g^{i,j})\Big)\\
&-\Delta\Big(\hat{D}_g^{i+1/2,j}(\phi_g^{i+1,j}-\phi_g^{i,j})+\tilde{D}_g^{i+1/2,j}(\phi_g^{i-1,j}+\phi_g^{i,j})\Big)\\
&+\Delta\Big(\hat{D}_g^{i,j-1/2}(\phi_g^{i,j-1}-\phi_g^{i,j})+\tilde{D}_g^{i,j-1/2}(\phi_g^{i,j-1}+\phi_g^{i,j})\Big)\\
&-\Delta\Big(\hat{D}_g^{i,j+1/2}(\phi_g^{i+1,j}-\phi_g^{i,j})+\tilde{D}_g^{i,j+1/2}(\phi_g^{i,j+1}+\phi_g^{i,j})\Big)\\
&+\Delta^2(\Sigma_g^{A,i,j}+\sum_{g'=1,g'\neq g}^{G}\Sigma_{g\to g'}^{S,i,j})\phi_g^{i,j}-\Delta^2\sum_{g'=1,g'\neq g}^{G}\Sigma_{g'\to g}^{S,i,j}\phi_{g'}^{i,j}\\
&=\Delta^2\frac{\chi_g^{i,j}}{k_eff}\sum_{g'=1}^{G}\nu\Sigma_{g'}^{F,i,j}\phi_{g'}^{i,j}
\end{aligned}$$

For reflective boundary conditions, we simply set $J=0$, which means both linear and non-linear diffusion coefficients are 0. For vacuum boundary conditions,we can use only non-linear terms to represent the true currents across coarse mesh surfaces.

$$\frac{\hat{J}_g^{i+1/2,j}}{\Delta}=-\tilde{D}_g^{i+1/2,j}(\phi_g^{i+1,j}+\phi_g^{i,j})$$

The algorithm for a basic CMFD module is given in the follow algorithm.

Algorithm 1 CMFD Process

1: Condensation on fluxes, cross sections, and diffusion coefficients
2: Compute $\hat{D}$ and $\tilde{D}$ based on the tallied current
3: Formulate finite difference equations into form $M\phi=\frac{1}{k}F$
4: Take an initial normalized flux vector $\phi^{(0)}$ and eigenvalue $k^{(0)}$
5: **while** ϕ, k $F\phi$ not converged **do**
6: $\quad b=\frac{1}{k^{(0)}}F\phi^0$
7: $\quad \phi^{(1)}=M^{-1}\phi^{(0)}$ ▷ Linear Solver
8: $\quad k^1=\frac{\|F\phi^1\|}{\|F\phi^0\|}k^0$
9: $\quad \phi^0=\phi^{(1)}$ and $\phi^0=\frac{\phi^0}{\|\phi^0\|}$ ▷ Normalization
10: **end while**

2.3 Coupling CMFD with Random Ray MOC

Overall Random Ray MOC with CMFD acceleration scheme is shown in the following algorithm.

Algorithm 2 CMFD on Random Ray MOC

1: Initialize Problem
2: **while** ϕ, k, fission source not converged **do**
3: Caculate source based on k and ϕ
4: Perform one cycle of Random Ray MOC
5: Tally surface currents
6: Input ϕ and XS into CMFD Solver
7: Run Algorithm 2 and return eigenvalue k and eigenvector ϕ
8: Update ϕ in new shape and normalize it
9: Check convergence
10: **end while**

To update scalar flux based on the CMFD results, we follow the equation below:

$$\phi_{r,g} = \phi_{r,g} \frac{\Phi_g^{i,j,new}}{\Phi_g^{i,j,old}}$$

$\Phi_g^{i,j,old}$ is the normalized and condensed flux before diffusion solver, $\Phi_g^{i,j,new}$ is the normalized flux obtained from diffusion solver,

3 Implementation and Problems

My random ray code uses pseudo random rays from the same random seed for each power iteration. Neutron tracks are always the same for every iterations. Since it is not true random, the result should be a little biased. However, it makes this problem converge faster. The implementation follows the procedure described in Figure 4 to 6. The CMFD module is inserted into the random ray code as described by Algorithm 1 in previous section. However, since there are still some problems in the process of tallying currents on the coarse mesh surfaces. The CMFD is not fully functional. An ad hoc approach is applied to solve a lattice of identical fuel pins with reflective boundaries. This essentially models infinite duplicates of a single fuel pin. The simple fix is to not use the non-linear diffusion terms, since the net currents are very small on the surfaces. The 2D diffusion solver will then help to converge the problem faster in energy, not for spatial variation. The 2D pincell model has a radius of 0.392cm with a pitch size of 1.26cm. The enrichment is 1.6%.

Another very important question is how to determine the total distance travelled for a ray, the number of rays needed, and the dead zone needed before actual attenuation in the MOC process. For my simulations, I choose to use a 15cm Dead Zone, because the effect of the initial guess of the angular flux will be negligible after sweeping for this amount. After some trial and error, I use 100 rays with 300cm each for the 3 by 3 lattice setup. The guideline I have been using at pre-testing is to check if the total lengths tallied in all FSRs maintain at a fixed ratio when I increase the number of rays and travel distance. If the ratios are the same and matches the physical volume ratio, I can conclude that the information provided by all rays are enough.

4 Results and Discussion

4.1 Reflective Boundary Conditions

Figure 8 and 9 shows how random rays sweep across the domain. Figure 8 is for only one ray travelling 300cm. Figure 9 shows the random sweep for 10 rays. The surface is already well covered at this point.

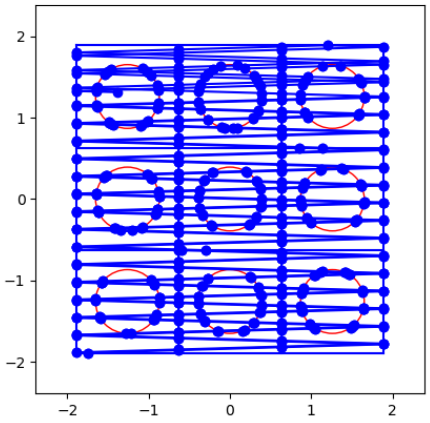

Figure 8: 1 ray, 300cm total length

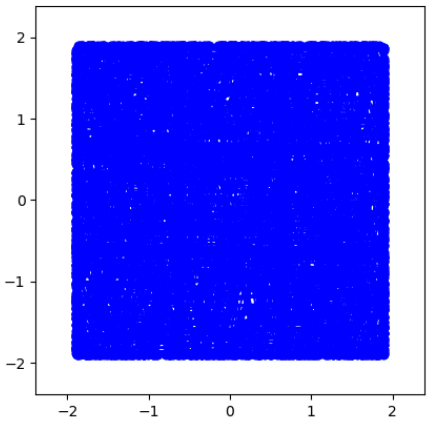

Figure 9: 10 rays, 300cm total length

Figure 10 shows the 10 group scalr flux solution in energy space. Flux is larger at high energy group and smaller at low energy group for both fuel and moderator. At thermal range, flux in moderator is larger than that in fuel.

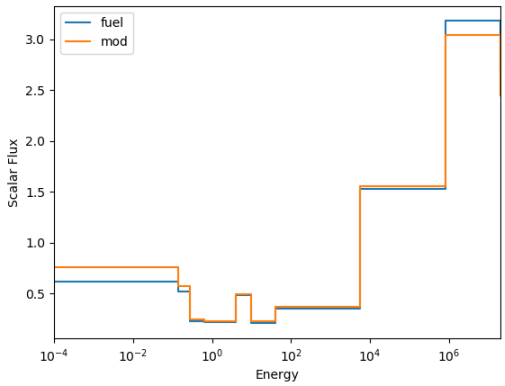

Figure 10: Flux in energy

The power iterations needed to converge both multiplication factor and scalar flux to 10^{-5} is given in the following table.

Iterations	RR alone	RR with CMFD
2G 3 by 3	31	20
2G 5 by 5	62	45
10G 3 by 3	28	19
10G 5 by 5	63	41

Table 1: Iterations

As we can see, the iterations required decreases when using CMFD. Although it doesn't help to converge the problem faster in spatial domain, it still contributes on converging flux shape faster in different energy groups. The iterations needed for 5 by 5 lattice is larger than 3 by 3. This is because we fixed the number of rays and travel distance. For larger problem, we need more iterations to reduce the stochastic error.

4.2 Vacuum Boundary Conditions

Since the current on mesh surface cannot be tallied correctly, nothing solid can be shown for now.

A homogeneous 1D problem is set up to verify if the leakage is tallied correctly. A schematic is shown in Figure 11. It is a 1 by 10 geometry, with only the left boundary as vacuum, and all other boundaries as reflective. This geometry models a 1D slab with 2 vacuum boundaries. By setting up one more reflective boundary, I can cut the size by half, and it is faster than setting ups a 1 by 20 geometry with two vacuum surfaces, because it take less time to search for all distances to all the planes in the sweeping process.

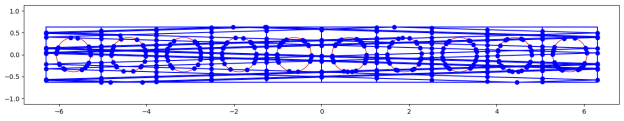

Figure 11: 1D Vacuum

5 Summary

5.1 Advantages of CMFD + Random Ray MOC

Even though there is no contribution for faster spatial convergence, CMFD still helps to converge faster for energy-wise flux shape. If the key problem, current tally, is solved in the future, we should expect to see a speed up just like regular MOC. Other than the speed up in convergence, using random can also have the following advantages.

- Do not need to store boundary flux and no need to store cyclical tracking info
- Resolve angular variation at very tiny meshes
- Avoid corner crossing issues when tally currents, because there is no tracking ray spacing associated with each ray.

5.2 Problems

There are definitely many problems to be solved.

- Current crossing mesh surfaces must be calculated correctly.
- This is a Monte Carlo process, and it takes a long time to run in serial.
- The convergence rate need to be studied with different parameters: Dead Zone, # particles, and total distance.

5.3 Future Work

Although this is a course project, I am very interested to turn this into a research project. I will keep this preliminary model as a side project for now and investigate how to correctly account for the leakage term. Then, I can make this problem into a true random code. When all the basic theoretical problems are solved, I would like to implement this problem in C++ and make it run in parallel.

References

[1] Samuel Christopher Shaner. *Transient method of characteristics via the Adiabatic, Theta, and Multigrid Amplitude Function methods*. PhD thesis, Massachusetts Institute of Technology, 2014.

[2] John R Tramm, Kord S Smith, Benoit Forget, and Andrew R Siegel. The random ray method for neutral particle transport. *Journal of Computational Physics*, 342:229–252, 2017.