CMFD on Random Ray MOC

22.212 Final Project

Zhuoran Han

Nov, 2018

Outline

Outline

1. MOC
2. Random Ray
3. CMFD
4. Failure Log
5. Results
6. Summary

MOC

Characteristic Form of Transport Equation

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

FSR

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

MOC

For track k in region r

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

Integrated from a entry point s' to an exit point s''

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

Contribution for this segment

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

MOC

Average angular flux contribution to FSR for track k ($l_{k,r} = s'' - s'$)

$$\bar{\psi}_{k,r,g} = \frac{1}{l_{k,r}} \int_{s'}^{s''} \psi_{k,r,g}(s) ds$$

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

MOC

FSR-averaged scalar flux in region r

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

Use quadrature to perform integration.
If no polar quadrature, a.k.a $w_p = 1$ and $p = 1$, we have:

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

where $w_{m(k)}$ is azimuthal weight, w_k is the ray spacing.

MOC

Scalar Flux Final Form

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

MOC

What about current J?

$$J_g^{surf} = \sum_{k \in surf} 2\pi w_{m(k)} \frac{w_k}{\cos\theta_k} \psi_{k,r,g} \cdot \hat{n}$$

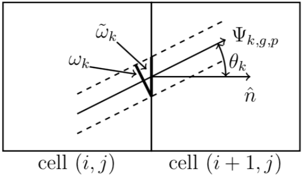

Figure 1: Current J crossing surface

MOC

Net Current tallied over surface, not averaged!!

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

This is the total leakage from the surface.

MOC

Update k_{eff}

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

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

MOC

Algorithm 1 Fixed source iteration for OpenMOC

```
Φ_{r,g} ← 0   ∀ r, g ∈ {R, G}                      # Initialize FSR scalar fluxes to zero
while Q_{r,g} ∀ r not converged do
  for all m ∈ M do                                 # Loop over azimuthal angles
    for all k ∈ K(m) do                            # Loop over tracks
      for all s ∈ S(k) do                          # Loop over segments
        for all g ∈ G do                           # Loop over energy groups
          for all p ∈ P do                         # Loop over polar angles
            r ← R(s)                               # Get FSR for this segment
            ΔΨ_{k,r,g,p} ← (Ψ_{k,g,p} − Q_{r,g}/Σ^{tr}_{r,g}) (1 − e^{−τ_{k,r,g,p}})
            Φ_{r,g} ← Φ_{r,g} + (4π/A_r) ω_{m(k)} ω_p ω_k sin θ_p l_{k,r} ΔΨ_{k,r,g,p}
            Ψ_{k,g,p} ← Ψ_{k,g,p} − ΔΨ_{k,g,p}
          end for
        end for
      end for
    end for
    if B.C. are reflective then                    # Set incoming flux for outgoing track
      Ψ_{k',g,p}(0) ← Ψ_{k,g,p}                    # Reflective B.C.'s
    else
      Ψ_{k',g,p}(0) ← 0                            # Vacuum B.C.'s
      L ← L + 2π ω_{m(k)} ω_p ω_k sin θ_p Ψ_{k,g,p}
    end if
  end for
  Update k_{eff} and Q_{r,g} ∀ r                   # Equation 2.17 and Algorithm 2
end while
```

Figure 2: MOC main Algorithm

MOC

update FSR source

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

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
    $Q^{(n+1)}_{r,g} \leftarrow \frac{\chi_{r,g}}{4\pi k_{eff}}\nu\Sigma^{F}_{r,g}\Phi^{(n)}_{r,g}$ # Initialize new total source with fission
    for all $g' \in G$ **do** # Loop over energy groups
      $Q^{(n+1)}_{r,g} \leftarrow Q^{(n+1)}_{r,g} + \frac{1}{4\pi}\Sigma^{S}_{r,g'\to g}\Phi^{(n)}_{r,g'}$
    end for
  end for
end for

Figure 3: update FSR source algorithm

Outline

Random Ray

Another Way to Evaluate the Integration?

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

Random Ray

Monte Carlo Integration

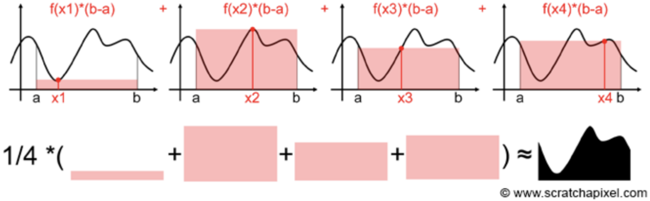

Figure 4: Monte Carlo Integration

Random Ray

Monte Carlo Integration

In a certain FSR, if we accumulate as many $\bar{\psi}_{i,r,g}$ as we can, and then take average, we shold get a good approximation of FSR-averaged scalar flux $\phi_{r,g}$.

$$\phi_{r,g} = 4\pi \frac{\sum_{i=1}^{N} \bar{\psi}_{r,g}^{i}}{N}$$

Random Ray

Derivation for Random Ray

For i^{th} trial in region r for energy g

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

Random Ray

Derivation for Random Ray

$$\phi^i_{r,g} l^i_{k,r} = \frac{l^i_{k,r} 4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{4\pi \Delta\psi^i_{k,g}}{\Sigma^{tr}_{r,g}}$$

$$l^i_{k,r}(\phi^i_{r,g} - \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}}) = \frac{4\pi \Delta\psi^i_{k,g}}{\Sigma^{tr}_{r,g}}$$

$$\phi^i_{r,g} - \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} = \frac{4\pi \Delta\psi^i_{k,g}}{\Sigma^{tr}_{r,g} l^i_{k,r}}$$

Random Ray

Derivation for Random Ray

If we take one guess of $\phi^i_{r,g}$, it would be

$$\phi^i_{r,g} = \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{4\pi \Delta\psi^i_{k,g}}{\Sigma^{tr}_{r,g} l^i_{k,r}}$$

However, this is not accurate.

Random Ray

Derivation for Random Ray

A good representation would be taking the average value from N contributions in this FSR

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

$$= \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + 4\pi \frac{\sum_{i=1}^{N} \Delta\psi^{i}_{k,g}}{\Sigma^{tr}_{r,g} \sum_{i=1}^{N} l^{i}_{k,r}}$$

$$= \frac{4\pi Q_{r,g}}{\Sigma^{tr}_{r,g}} + \frac{\sum_{i=1}^{N} 4\pi \Delta\psi^{i}_{k,g}}{\Sigma^{tr}_{r,g} d_r}$$

where d_r is the total distance travelled by all rays in FSR region r.

Random Ray

This is exactly the same process in John's dissertation.

Random Ray

Algorithm 1 MOC Power Iteration

```
Initialize Scalar Fluxes to 1.0
while K-effective and Scalar Flux Unconverged do
    Compute Source (Equation 2.2)
    Set Scalar Flux to Zero
    Transport Sweep (Algorithm 2)
    Normalize Scalar Flux to Sum of Ray Distances
    Add Source to Scalar Flux (Equation 2.3)
    Calculate K-effective
end while
```

Figure 5: Random Ray Algorithm 1

Random Ray

Random Ray

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

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

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

Random Ray

Random Ray

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 6: Random Ray Algorithm 2

Random Ray

Monte Carlo Integration

Algorithm 3 Attenuate Segment

1: **for all** Energy Groups $g \in G$ **do**
2: $\quad \Delta\psi_g = (\psi_g - Q_{r,g})\left(1 - e^{-\Sigma_{t,r,g}s}\right)$
3: $\quad \phi_{r,g} = \phi_{r,g} + 4\pi\Delta\psi_g$
4: $\quad \psi_g = \psi_g - \Delta\psi_g$
5: **end for**

Figure 7: Random Ray Algorithm 3

Random Ray

Some Explanation

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

Compare with conventional MOC

MOC Final Form

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

Random Ray Final Form

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

In random ray, "Area of FSR" is defined as the the total distance tallied in this FSR. Uniform weight, Spacing in unit.

What about current? How to update new k?

MOC Net

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

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

Random Ray

$$J_g^{surf} = \sum_{k \in surf} 2\pi \psi_{k,r,g}$$

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

There is always a bug!

Is J net current or averaged current?
What is the surface area for current to apply on?
Area = sum of length, what about length of an edge? Sum of collisions?
What I have tried on a 1D homogeneous slab:

- J is net current, use it directly. Leakage too small
- J is averaged current, length is just the pitch. L is still small.
- J is averaged current, length is total number of collisions on the surface. L is too large.

How to scale it?

Outline

Condensation

$$\Sigma_{\mathbf{g}}^{A,i,j} = \frac{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)} \Sigma^{A}_{r,g}\Phi_{r,g}A_r}{\sum_{g\in\mathbf{g}}\sum_{r\in(i,j)} \Phi_{r,g}A_r}$$

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

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

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

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

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

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

Figure 8: Condensation in Energy and Space

Note: For diffusion coefficient, since I don't have a gap in the model, I can do it this way. Otherwise, transport XS has to be flux weighted.

CMFD

2D Diffusion Equation

$$-\frac{\partial}{\partial x} \cdot D_g(x,y)\frac{\partial}{\partial x}\phi_g(x,y) - \frac{\partial}{\partial y} \cdot D_g(x,y)\frac{\partial}{\partial y}\phi_g(x,y)$$

$$+ (\Sigma_g^A(x,y) + \sum_{g'=1,g'\neq g}^{G} \Sigma_{g\to g'}^S(x,y))\phi_g(x,y)$$

$$= \frac{\chi_g(x,y)}{k_e ff} \sum_{g'=1}^{G} \nu\Sigma_{g'}^F(x,y)\phi_{g'}(x,y) + \sum_{g'=1,g'\neq g}^{G} \Sigma_{g'\to g}^S(x,y)\phi_{g'}(x,y)$$

CMFD

Discretization

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

Linear Diffusion Coefficient

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

$$
\hat{D}_g^{i+1/2,j} = \frac{2 * D_g^{i,j} * D_g^{i+1,j}}{\Delta(D_g^{i,j} + D_g^{i+1,j})}
$$

CMFD

Non Linear Diffusion Correction

Current across coarse mesh must match.

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

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

Here $\hat{J}$ is the overall net current on the surface

CMFD

Discretization

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

CMFD

Reflective B.C.

$$J = 0$$

Vacuum B.C.

Just use the non-linear term to balance the current

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

Algorithm for CMFD

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**

Overall Acceleration Scheme

Algorithm 2 CMFD on Random Ray MOC

```
Initialize Problem
while φ, k, fission source not converged do
    Caculate source based on k and φ
    Perform one cycle of Random Ray MOC
    Tally surface currents
    Input φ and XS into CMFD Solver
    Run Algorithm 2 and return eigenvalue k and eigenvector φ
    Update φ in new shape and normalize it
    Check convergence
end while
```

Should we pass $k - eff$ out of CMFD solver or not?

CMFD

Update ϕ

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

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

Outline

Yeah, my code is broken :(

Failure Log

- Started with a 3 by 3 identical pin cells, reflective B.C.It showed reasonable k-eff, but current tallied was too small. No need for non-linear correction, since flux shape is flat spatially.
- A center guide tube cell. Current tallied didn't look right.
- Hey, why not try vacuum B.C., homogenized, fake 1D, so that I can compare with my 2D diffusion solver.
- No cosine shape? Ahh, return ray should be set to 0.
- Had a curve shape, but it was still not cosine. Leakage was not right.
- Re-derived Random Ray, compared with conventional MOC.
- Hey! I can show the equivalence for Random Ray!
- Didn't help. Tried different methods to verify current. Failed.
- Tried more scaling methods. Still failed.
- This is my destiny. Ready to everyone I screwed up

Outline

Reflective

The Random Ray code is a pseudo random ray. By setting up the same random seed each in each iteration, neutron tracks are always the same. Since J cannot be tallied correctly, I only have some preliminary results for reflective B.C. cases. Instead of tallying currents and calculate $\tilde{D}$, we set them as 0 and only use linear diffusion terms. The 2D diffusion equations can only help to determine the flux shape in different energy, but not spatially.
2D Pincell Model: Pitch = 1.26cm and 1.6% enrichment fuel. k-eff is about 1.28. The statistic is on the 100 pcm order when having different dead zone, different number of particles, and different total length.

Reflective

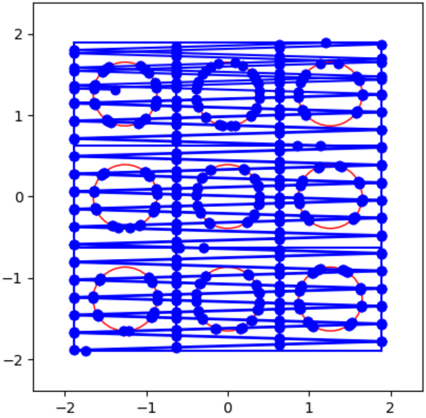

Figure 9: 1 ray, 300cm total length

Reflective

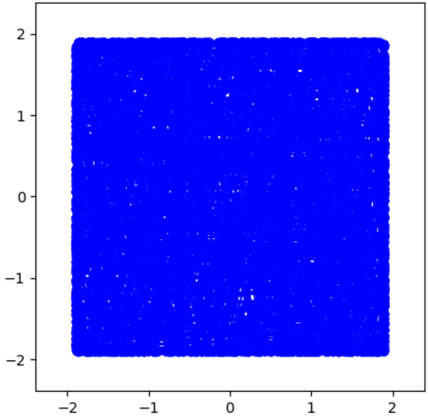

Figure 10: 10 rays, 300cm total length

Reflective

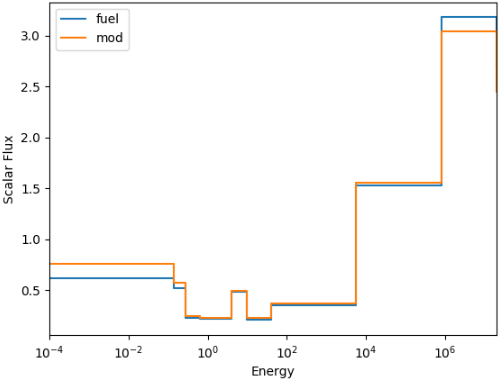

Figure 11: Flux in energy

Reflective

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

Vacuum

I set up a 1 by 10 geometry and set only the left boundary as vacuum to model a 1D slab with 2 vacuum boundaries. By setting up one more reflective boundary, I can cut the size by half, and it will be faster for Random Ray code to run. However, because J cannot be tallied, nothing solid to show.

Vacuum

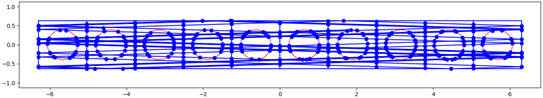

Figure 12: 1D Vacuum

Outline

Conclusion

Advantage

- No 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

Conclusion

Problems

- Current Tally
- MC, slow
- Stochastic Convergence
- Dead Zone, # particles, total distance. (For now, I have to perform pre-testing to determine the parameters: Check if the length tallied in FSR/total length is the same as really volume fraction)

CMFD

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.

Future Work

- Find the correct way to scale surface current.
- True Random.
- Python is slow. Rewrite in C++
- Ultimate goal: parallelization

Reference

- John Tramm's Ph.D. thesis
- Geoff Gunow's Ph.D. thesis
- Sam Shaner's M.S. thesis
- 212 Slides