22.211 Lecture 12
MOC

Benoit Forget

March 15, 2023

Massachusetts Institute of Technology

Outline

Massachusetts Institute of Technology

Outline

Massachusetts Institute of Technology

Method of Characteristic

- Define the emission density

$$Q_g(\mathbf{r},\mathbf{\Omega}) = \int_{4\pi} d\Omega' \sum_{g'=1}^{G} \Sigma_{sg'\rightarrow g}(\mathbf{r},\mathbf{\Omega}\cdot\mathbf{\Omega}')\psi_{g'}(\mathbf{r},\mathbf{\Omega}') + \frac{\chi_g}{4\pi}\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\mathbf{r})\phi_{g'}(\mathbf{r}) + S_g(\mathbf{r},\mathbf{\Omega})$$

- Introduce integrating factor

$$IF = e^{\int_{-\infty}^{s}\Sigma_{tg}(\mathbf{r}+s'\mathbf{\Omega})ds'}$$

Massachusetts Institute of Technology

Forward Characteristic

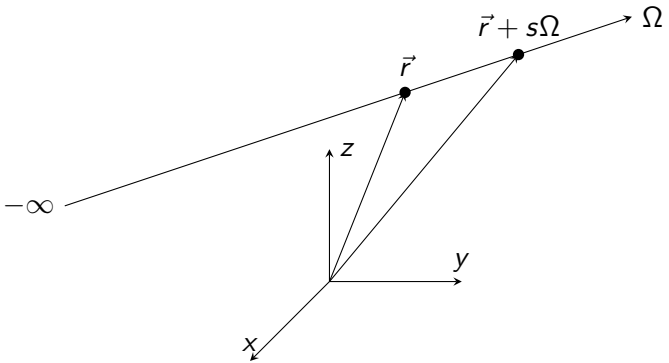
- Integrate between $-\infty$ and s

$$\psi_g(\mathbf{r}+s\mathbf{\Omega},\mathbf{\Omega}) = e^{-\int_{-\infty}^{s}\Sigma_{tg}(\mathbf{r}+s'\mathbf{\Omega})ds'} \int_{-\infty}^{s} e^{\int_{-\infty}^{s'}\Sigma_{tg}(\mathbf{r}+s''\mathbf{\Omega})ds''} Q_g(\mathbf{r}+s'\mathbf{\Omega},\mathbf{\Omega})ds'$$

- Simplify to yield

$$\psi_g(\mathbf{r}+s\mathbf{\Omega},\mathbf{\Omega}) = \int_{-\infty}^{s} e^{-\int_{s}^{s'}\Sigma_{tg}(\mathbf{r}+s''\mathbf{\Omega})ds''} Q_g(\mathbf{r}+s'\mathbf{\Omega},\mathbf{\Omega})ds'$$

Massachusetts Institute of Technology


$\vec{r}+s\Omega$
Ω
$\vec{r}$
z
$-\infty$
y
x

Backward Characteristic

If we reverse the direction of travel and set $s = 0$

$$\psi_g(\mathbf{r}, \mathbf{\Omega}) = \int_0^{\infty} e^{-\int_0^{s'} \Sigma_{tg}(\mathbf{r}-s''\mathbf{\Omega})ds''} Q_g(\mathbf{r} - s'\mathbf{\Omega}, \mathbf{\Omega})ds'$$

If we had a more meaningful boundary condition in a finite system

$$\psi_g(\mathbf{r}, \mathbf{\Omega}) = \int_0^{s_{BC}} e^{-\int_0^{s'} \Sigma_{tg}(\mathbf{r}-s''\mathbf{\Omega})ds''} Q_g(\mathbf{r} - s'\mathbf{\Omega}, \mathbf{\Omega})ds' + \psi_g(\mathbf{r} - s_{BC}\mathbf{\Omega}, \mathbf{\Omega}) e^{-\int_0^{s_{BC}} \Sigma_{tg}(\mathbf{r}-s'\mathbf{\Omega})ds'}$$

Massachusetts Institute of Technology

Optical Path Length

$$\tau_g(\mathbf{r}, \mathbf{r} - s'\mathbf{\Omega}) = \int_0^{s'} \Sigma_{tg}(\mathbf{r} - s''\mathbf{\Omega}) ds'' = \tau_g(\mathbf{r}, \mathbf{r}')$$

$e^{\tau_g(\mathbf{r},\mathbf{r}')}$ is the probability of not making a collision between $\mathbf{r}$ and $\mathbf{r}'$ in group g

Massachusetts Institute of Technology

Outline

Massachusetts Institute of Technology

Discrete Ordinates Approximation

Approximates the integral over all angles by a sum over discrete directions.

$$\phi_g(s) = \int_{4\pi} \mathrm{d}\mathbf{\Omega}' \psi_g(s, \mathbf{\Omega}') \approx \sum_{m=1}^{M} w_m \psi_g(s, \mathbf{\Omega_m})$$

We can thus re-write the source emission density as:

$$Q_{m,g}(s) = \sum_{g'=1}^{G} \sum_{m'=1}^{M} w_{m'} \Sigma_{sg' \to g}(s, \mathbf{\Omega_{m'}} \to \mathbf{\Omega_m}) \psi_{g'}(s, \mathbf{\Omega_{m'}}) + \frac{\chi_g(s)}{4\pi k_{eff}} \sum_{g'=1}^{G} \nu\Sigma_{fg'}(s)\phi_{g'}(s)$$

Massachusetts Institute of Technology

Quadrature

In numerical analysis, a quadrature rule is an approximation of the definite integral of a function, usually stated as a weighted sum of function values at specified points within the domain of integration. (Wikipedia)

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{I} w_i f(x_i)$$

The weights w_i and quadrature points x_i are selected to minimize integration errors (e.g. minimization problem or zeros of a given orthogonal set).

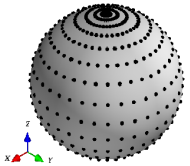
Massachusetts Institute of Technology

Product Quadrature

A product quadrature represents a two-dimensional system by independently selecting each variable.

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)dxdy = \sum_{i=1}^{I}\sum_{j=1}^{J} w_i w_j f(x_i, y_j)$$

Product quadratures are not ideal for integration on sphere since they create the same number of azimuthal angles for each polar angle, however they are simple and offer some advantages (e.g cyclical tracks, vectorization, ...).

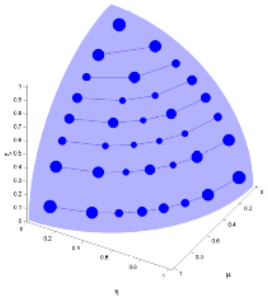


Massachusetts Institute of Technology

Levelset Quadrature

A common alternative (in SN methods) is the leveset quadrature that ties the variables.

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)dxdy = \sum_{n=1}^{N} w_n f(x_n, y_n)$$



Massachusetts Institute of Technology

Substituting in our characteristic equation

$$\psi_{m,g}(s) = \psi_{m,g}(\mathbf{r_0})e^{-\int_0^s \mathrm{d}s'\Sigma_{tg}(s')} + \int_0^{s_m} \mathrm{d}s'' Q_{m,g}(s'')e^{-\int_{s''}^s \mathrm{d}s'\Sigma_{tg}(s')}$$

The angular approximation can be further decomposed into azimuthal and polar angles

$$Q_{m,p,g}(s) = \sum_{g'=1}^{G}\sum_{m'=1}^{M}\sum_{p'=1}^{P} w_{m'}w_{p'}\Sigma_{sg'\to g}(s, \mathbf{\Omega_{m',p'}} \to \mathbf{\Omega_{m,p}})\psi_{g'}(s, \mathbf{\Omega_{m',p'}}) + \frac{\chi_g(s)}{4\pi k_{eff}}\sum_{g'=1}^{G}\nu\Sigma_{fg'}(s)\phi_{g'}(s)$$

$$\psi_{m,p,g}(s) = \psi_{m,p,g}(\mathbf{r_0})e^{-\int_0^s \mathrm{d}s'\Sigma_{tg}(s')} + \int_0^{s} \mathrm{d}s'' Q_{m,p,g}(s'')e^{-\int_{s''}^s \mathrm{d}s'\Sigma_{tg}(s')}$$

Isotropic Scattering

The approximation of isotropic scattering will also be made to simplify the source. This approximation is not necessary since flat source MOC can easily accommodate anisotropic scattering.

$$Q_g(s) = \frac{1}{4\pi}\left(\sum_{g'=1}^{G}\Sigma_{sg'\rightarrow g}(s)\phi_{g'}(s) + \frac{\chi_g(s)}{k_{eff}}\sum_{g'=1}^{G}\nu\Sigma_{fg'}(s)\phi_{g'}(s)\right)$$

Massachusetts Institute of Technology

Outline

Massachusetts Institute of Technology

Flat Source and Constant Cross-Section Approximation

The main approximation consists of assuming that the sources (i.e. scattering and fission) have a flat spatial distribution within a given region.

$$Q_{i,g} = Q_g(s') = Q_g(s'') = Q_g(s) \quad , \quad s \in [s', s'']$$

We also assume that the material properties are constant within that same region. The two approximations together yield an approximation of flat scalar flux. However, the angular flux is still allowed to vary within the region.

$$\Sigma_{tig} = \frac{\int_{\mathbf{r}\in A_i} \mathrm{d}\mathbf{r} \Sigma_{tg}(\mathbf{r})\phi_g(\mathbf{r})}{\int_{\mathbf{r}\in A_i} \mathrm{d}\mathbf{r}\phi_g(\mathbf{r})}$$

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

Massachusetts Institute of Technology

Solution on a Segment

On a segment, the source and cross-sections are assumed constant for which an analytical solution exists.

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

We can re-arrange the solution to reflect the change in angular flux along the segment.

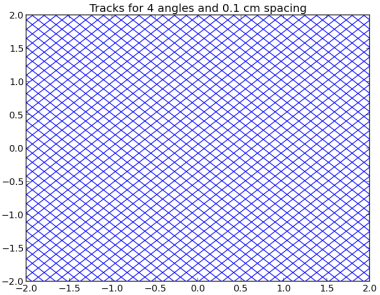
$$\Delta\psi_{k,g} = \psi_{k,g}(s') - \psi_{k,g}(s'') = \left(\psi_{k,g}(s') - \frac{Q_{i,g}}{\Sigma_{tig}}\right)(1 - e^{-\tau_{k,i,g}})$$

We can then relate $\Delta\psi_{k,g}$ to a contribution to the scalar flux.

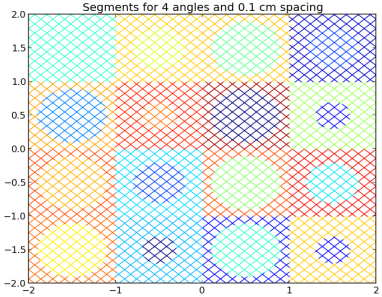
Massachusetts Institute of Technology

Tracks vs Segments

A track traverses the full geometry (often called long characteristic).

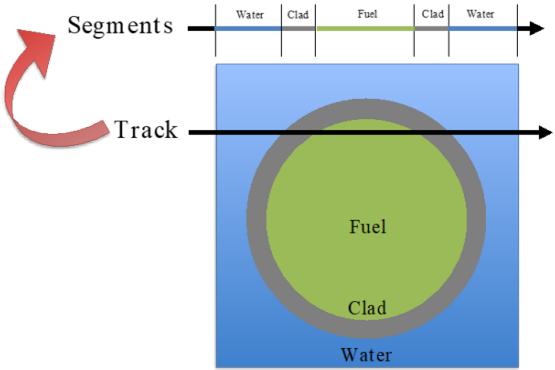


A segment is a portion of a track within an FSR assigned a given material.



Massachusetts Institute of Technology

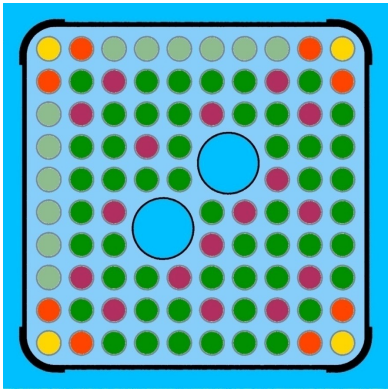
Tracks vs Segments

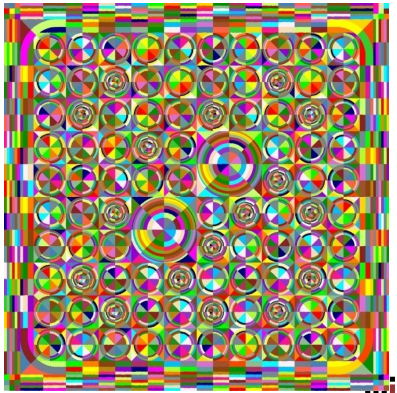


Massachusetts Institute of Technology

Lattice Example

To properly account for flux (and source) gradients, many FSRs are needed per material region.





Massachusetts Institute of Technology

Solution on a Segment

On a segment, the source and cross-sections are assumed constant for which an analytical solution exists.

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

We can re-arrange the solution to reflect the change in angular flux along the segment.

$$\Delta\psi_{k,g} = \psi_{k,g}(s') - \psi_{k,g}(s'') = \left(\psi_{k,g}(s') - \frac{Q_{i,g}}{\Sigma_{tig}}\right)(1 - e^{-\tau_{k,i,g}})$$

Massachusetts Institute of Technology

Track Area Approximation

The other quantity needed is the scalar flux in the FSR which we will obtain by integrating the average angular flux along each segment of a given FSR. We can define the average segment angular flux as:

$$\overline{\psi}_{k,i,g} = \frac{1}{l_{k,i}} \int_{s'}^{s''} \psi_{k,i,g}(s) \mathrm{d}s$$

which yields:

$$\overline{\psi}_{k,i,g} = \frac{1}{l_{k,i}} \left[\frac{\psi_{k,g}(s')}{\Sigma_{tig}} (1 - e^{-\tau_{k,i,g}}) + \frac{l_{k,i} Q_{i,g}}{\Sigma_{tig}} \left(1 - \frac{(1 - e^{-\tau_{k,i,g}})}{\tau_{k,i,g}} \right) \right]$$

Massachusetts Institute of Technology

Scalar Flux

Using the angular quadrature we can approximate the scalar flux in an FSR using the following relation:

$$\phi_{i,g} = \frac{\int_{\mathbf{r}\in A_i} \mathrm{d}\mathbf{r} \int_{4\pi} \mathrm{d}\mathbf{\Omega} \psi_g(\mathbf{r},\mathbf{\Omega})}{\int_{\mathbf{r}\in A_i} \mathrm{d}\mathbf{r}} \approx \frac{4\pi \sum_{k\in A_i} w_{m(k)} w_{p(k)} w_k l_{k,i} \sin\theta_{p(k)} \overline{\psi}_{k,i,g}}{\sum_{k\in A_i} w_k l_{k,i} \sin\theta_{p(k)}}$$

which we can express in terms of the angular flux change along a segment

$$\phi_{i,g} = \frac{4\pi}{\Sigma_{tig}} \left[Q_{i,g} + \frac{1}{A_i} \sum_{k\in A_i} \omega_{m(k)} \omega_{p(k)} \omega_k \sin\theta_{p(k)} \Delta\psi_{k,i,g} \right]$$

Massachusetts Institute of Technology

From an initial guess of k and ϕ, we can compute the source in each FSR for each group:

```
for all i ∈ I do                                                    # Loop over FSRs
    for all g ∈ G do                                                # Loop over energy groups
        Q_{i,g}^{(n+1)} ← (χ_{i,g} / k_{eff}) ν Σ_{i,g}^F Φ_{i,g}^{(n)}   # Initialize new total source with fission
        for all g' ∈ G do                                           # Loop over energy groups
            Q_{i,g}^{(n+1)} ← Q_{i,g}^{(n+1)} + Σ_{i,g'→g}^S Φ_{i,g'}^{(n)}   # Increment total source with scattering
        end for
    end for
end for
```

Once we have the source, we solve a fixed source problem to determine the angular fluxes at the boundary and scalar fluxes in each FSR. There is no need to store the angular fluxes at each FSR boundary, only at the domain boundary.

Algorithm 2 Transport Sweep Algorithm

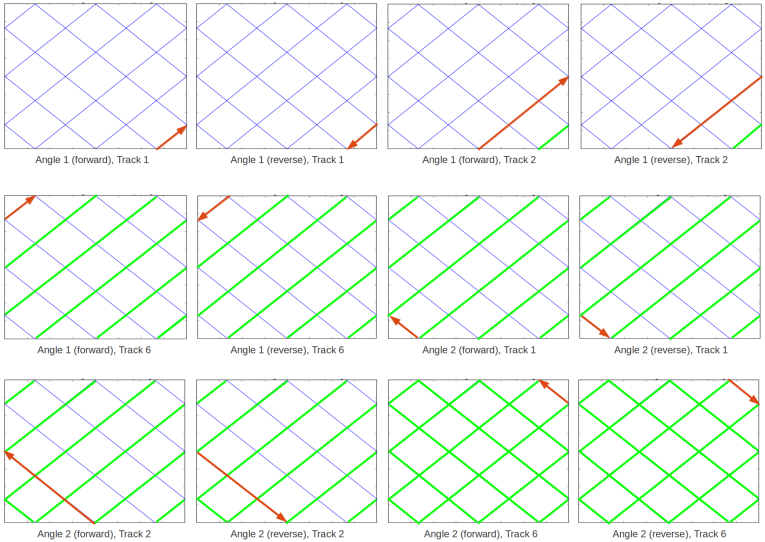
$\Phi_{i,g} \leftarrow 0 \quad \forall\, i, g \in \{I, G\}$ # Initialize FSR scalar fluxes to zero
while $\Phi_{i,g}\ \forall\ i$ not converged **do**
  for all $m \in M$ **do** # Loop over azimuthal angles
    for all $k \in K(m)$ **do** # Loop over tracks
      for all $s \in S(k)$ **do** # Loop over segments
        for all $g \in G$ **do** # Loop over energy groups
          for all $p \in P$ **do** # Loop over polar angles
            $i \leftarrow I(s)$ # Get FSR for this segment
            $\Delta\Psi_{k,i,g,p} \leftarrow \left(\Psi_{k,g,p} - \frac{Q_{i,g}}{\Sigma^t_{i,g}}\right)\left(1 - e^{-\tau_{k,i,g,p}}\right)$ # Compute angular flux change along segment
            $\Phi_{i,g} \leftarrow \Phi_{i,g} + \frac{4\pi}{\Lambda_i}\omega_m\omega_p\omega_k \sin\theta_p \Delta\Psi_{k,i,g,p}$ # Increment FSR scalar flux
            $\Psi_{k,g,p} \leftarrow \Psi_{k,g,p} - \Delta\Psi_{k,g,p}$ # Update track outgoing flux
          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
      $L \leftarrow L + \Psi_{k,g,p}$
    end if
  end for
  Update k_{eff} and FSR sources $Q_{i,g}\ \forall\ i$ # Algorithm 1
end while

Massachusetts Institute of Technology

Things to note

- Polar angles are the most inner loop since the same operation is performed on the same data P times.
- Group is the second most inner loop since for few group problems, the group cross-sections will fit nicely in cache.
- Azimuthal angles are on the outside and are the simplest loop to parallelize.
- No IF-ELSE statements in the core of the kernel
- Source contribution to scalar flux is added outside main loop
- Division by FSR volume/area is also done outside main loop
- Usually we do track in the forward direction and follow in backward direction to take advantage of any memory coalescence we can.

Massachusetts Institute of Technology



Massachusetts Institute of Technology

Power iteration

In reactor analysis we mainly care about the dominant eigenvalue (which yields a positive flux). The simplest way to obtain the dominant eigenvalue is the power iteration, on system $Ax = \lambda x$. Suppose A is diagonalizable with ordered eigenvalues:

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$$

then we have $A = V\Lambda V^{-1}$ where Λ is a diagonal matrix of eigenvalues. Note that

$$A^k = (V\Lambda V^{-1})^k = V\Lambda^k V^{-1} \Rightarrow A^k V = V\Lambda^k$$

For a random initial guess of x, that can be expressed as $x = V\tilde{x}$, we can show

$$A^{(k)}x = A^k V\tilde{x} = \sum_{j=1}^{n} v_j \lambda_j^k \tilde{x}_j$$

Massachusetts Institute of Technology

Power iteration

We can also pull out the dominant eigenvalue λ_1

$$A^k x = \lambda_1^k \left(\sum_{j=1}^{n} v_j \left(\frac{\lambda_j}{\lambda_1} \right)^k \tilde{x}_j \right)$$

Thus, as we increase the value of k (iterations), all higher modes will go to zero except for the dominant eigenvalue. In reactor physics, we actually use a slight variant called the inverse power iteration, because of the way we setup our problem. The ratio of λ_2/λ_1 is the slowest mode to go to zero, and is referred to as the dominance ratio.

Massachusetts Institute of Technology

Krylov methods

The power iteration method created a series of vectors that form a Krylov subspace

$$\kappa = span\left\{x^0, Ax^0, A^2x^0, ...A^kx^0\right\}$$

Krylov solvers will store multiple vectors in such a subspace (orthogonalized subspace) and project the matrix to a subspace on which accurate approximations of eigenvalues and eigenvectors can be obtained (Arnoldi's method).

Massachusetts Institute of Technology

How to solve - power iteration

1. Guess an initial normalized flux vector $\phi^{(0)}$ and eigenvalue $k^{(0)}$
2. Compute source $Q^{(0)}$, using $\phi^{(0)}$ and $k^{(0)}$
3. Sweep over all angles and cells to solve for $\psi^{(1)}$ which we can use to compute $\phi^{(1)}$
4. Compute $k^{(1)} = \frac{\langle F\phi^{(1)} \rangle}{\langle F\phi^{(0)} \rangle} k^{(0)}$ or by neutron balance
5. Normalize flux
6. Iterate until convergence (k, flux, fission source, ...)

Massachusetts Institute of Technology

Normalization

A normalized vector is a vector in the same direction but with a known length.

$$\hat{x} = \frac{x}{||x||}$$

where $||x||$ can take many forms

$$L_\infty \Rightarrow ||x||_\infty = max|x_j|$$

$$L_m \Rightarrow ||x||_m = \sqrt{\sum_{j=1}^{N} |x_j|^m}$$

Massachusetts Institute of Technology

Convergence

Convergence is obtained by comparing the change between successive iterations and comparing to a given threshold ϵ

$$\left|\left|\frac{x^i - x^{i-1}}{x^i}\right|\right| < \epsilon$$

Converging the eigenvalue is not sufficient, you must also verify convergence of the eigenvector. Common practice is to check convergence of fission source since it weighs the flux by the important regions of the problem.

Massachusetts Institute of Technology

Resources

- Chapter 9 of the handbook
- OpenMOC website (https://mit-crpg.github.io/OpenMOC/)

In 1D with 2 angles

1. Discretize the problem spatially
2. Guess an initial value of k and ϕ, and $\psi_{incoming}$ on the boundaries
3. Compute the source, Q, in each mesh
4. Start from left boundary, and solve for $\Delta\psi$ in each mesh when going left to right
5. When you reach the right boundary, apply BC and sweep from right to left computing $\Delta\psi$ for that direction
6. Compute scalar flux in each cell from the $\Delta\psi$ terms
7. Compute new k
8. Re-compute source with new ϕ and k, iterate

Massachusetts Institute of Technology