

NE 155, Topic 14, S21
Finite Difference and Volume Methods for the Eigenvalue form of the DE
April 1, 2021

Helmholtz Form

When we derived the Diffusion Equation, we skipped over the Helmholtz form. I am bringing it up now b/c it can be a convenient way to solve the DE. This approach can also be useful if you need to generate an analytical solution.

In steady state, we can write the diffusion equation this way:

$$-\nabla \cdot D(\vec{r}) \nabla \phi(\vec{r}) + \Sigma_a \phi(\vec{r}) = Q(\vec{r}) ,$$

where $Q(\vec{r}) = \nu \Sigma_f \phi(\vec{r}) + S(\vec{r}) ,$

which can be written as the Helmholtz equation of applied mathematics:

$$\nabla^2 \phi(\vec{r}) - \frac{1}{L^2} \phi(\vec{r}) = \frac{-Q(\vec{r})}{D} ,$$

where $L \equiv \sqrt{\frac{D}{\Sigma_a}} .$

L is called the neutron diffusion length. This is “how far a neutron diffuses from a source prior to absorption”.

In the Helmholtz formulation, ϕ is amplitude and $\frac{1}{L}$ is wave number.

This formulation is useful because we know how to solve it. We write

$$\phi(\vec{r}) = \phi_H(\vec{r}) + \phi_P(\vec{r}) ,$$

For example, we often have:

$$\phi_H(\vec{r}) = A \exp\left(-\frac{|\vec{r}|}{L}\right) + B \exp\left(\frac{|\vec{r}|}{L}\right) .$$

Going through how to solve this analytically in a variety of circumstances, geometries, etc. is

another class (NE 150/250).

Criticality Calculations

Recall:

We also want to think about the details of how to configure a reactor to get it to work the way we want it to. We can write our DE in steady state for a nuclear reactor core:

$$-\nabla \cdot D \nabla \phi(\vec{r}) + \Sigma_a \phi(\vec{r}) = \nu \Sigma_f \phi(\vec{r}) ,$$

with $\phi(\tilde{x}_s) = 0$.

Unless we have the proper combination of core composition (Σ_a , Σ_f , D) and geometry (\vec{r} , \vec{r}_s) details, there is no general solution.

To deal with this, we introduce a parameter k into the equation:

$$-\nabla \cdot D \nabla \phi(\vec{r}) + \Sigma_a \phi(\vec{r}) = \frac{1}{k} \nu \Sigma_f \phi(\vec{r}) .$$

Then, for any value of k we assert that there is always a solution. We use an iterative process to find the condition when $k = 1$, called “critical”.

Finite Difference Method, Eigenvalue Problem

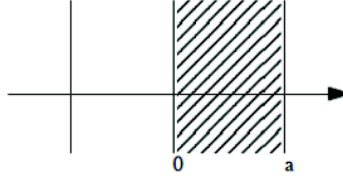
We can extend all of the finite difference and finite volume methods we just learned to the eigenvalue problem case, which is another layer of complication.

Now instead of a fixed source, we have an eigenvalue problem

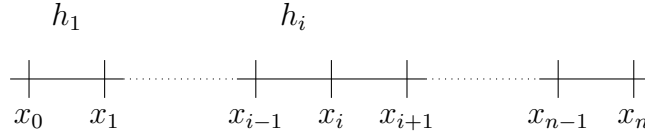
$$-\frac{d}{dx} D(x) \frac{d\phi(x)}{dx} + \Sigma_a(x) \phi(x) = \frac{1}{k} \nu \Sigma_f(x) \phi(x)$$

Let's again have a reflecting condition at the centerline ($x_0 = 0$) and vacuum on the right ($x_n = a$):

$$\begin{aligned} \frac{d}{dx} \phi(x) \Big|_{x=0} &= 0 && \text{zero net current,} \\ \phi(\tilde{a}) &= 0 && \tilde{a} = a + 2D . \end{aligned}$$



We again have a spatial mesh:



and in this configuration $x_0 = 0$, $x_n = a$, and h_i is the mesh spacing. There are $n + 1$ points and n mesh cells.

Material discontinuities will coincide with the cell *edges*, x_i . Thus, we can assume that the cross sections and the diffusion coefficient are constant in each cell. The unknown values are again defined at the mesh or cell *edges*, e.g. $\phi(x_i) = \phi_i$.

We derive the equations just like we did in the fixed source case, but now instead of S_i we have $\nu\Sigma_{f,i}$ on the rhs and a $1/k$ in multiplying the rhs vector.

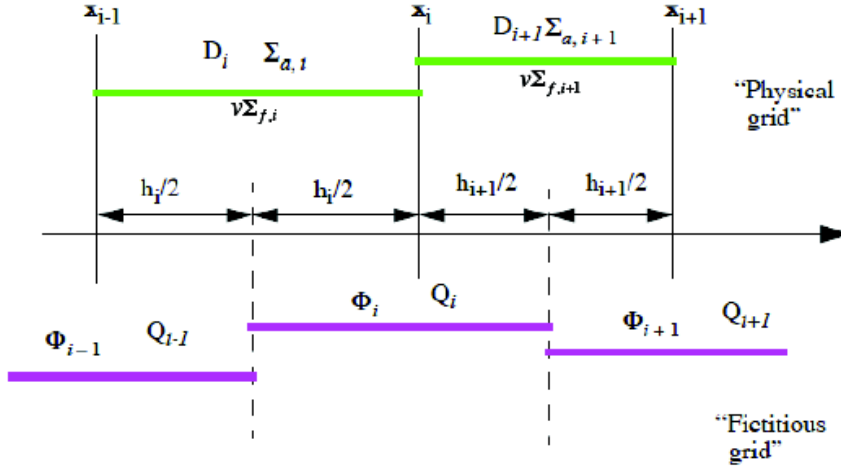
$$\begin{aligned} \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h_i^2} - \frac{1}{L_i^2}\phi_i &= -\frac{1}{k} \frac{\nu\Sigma_{f,i}}{D_i} \phi_i & i = 1, 2, \dots, n-1, \\ -\phi_{i-1} + \left(2 + \frac{h_i^2}{L_i^2}\right)\phi_i - \phi_{i+1} &= \frac{1}{k} h_i^2 \frac{\nu\Sigma_{f,i}}{D_i} \phi_i & i = 1, 2, \dots, n-1 \text{ or,} \end{aligned}$$

$$\frac{-D_i}{h_i^2}\phi_{i-1} + \left(\frac{2D_i}{h_i^2} + \Sigma_{a,i}\right)\phi_i - \frac{D_i}{h_i^2}\phi_{i+1} = \frac{1}{k}\nu\Sigma_{f,i}\phi_i \quad i = 1, \dots, n-1.$$

With this formulation we still have the problem that our unknown is defined the cell edges and if the properties in neighboring cells differ we will have discontinuities. To be able to handle *material discontinuities* we're going to do our volume integration again.

Finite Volume Method

Like last time, we will integrate the flux and source values across neighboring half-cells.



We again assume the cross section and diffusion coefficient are constant in each cell; for $i = 1, \dots, n$:

$$\begin{aligned} D(x) &= D_i, & x_{i-1} \leq x \leq x_i \\ \Sigma_a(x) &= \Sigma_{a,i}, & x_{i-1} \leq x \leq x_i \\ \nu\Sigma_f(x) &= \nu\Sigma_{f,i}, & x_{i-1} \leq x \leq x_i \\ h_i &\equiv x_i - x_{i-1}. \end{aligned}$$

We also still assume that the fluxes are constant over the interval centered around x_i ; for $i = 0, \dots, n$:

$$\phi(x) = \phi_i \quad \text{for } \left(x_i - \frac{h_i}{2}\right) \leq x \leq \left(x_i + \frac{h_{i+1}}{2}\right).$$

Now, we integrate the differential equation over each cell, $\left(x_i - \frac{h_i}{2}\right) \leq x \leq \left(x_i + \frac{h_{i+1}}{2}\right)$:

$$\int_{(x_i - \frac{h_i}{2})}^{(x_i + \frac{h_{i+1}}{2})} \left(-\frac{d}{dx} D(x) \frac{d\phi(x)}{dx} \right) dx + \int_{(x_i - \frac{h_i}{2})}^{(x_i + \frac{h_{i+1}}{2})} \Sigma_a(x) \phi(x) dx = \int_{(x_i - \frac{h_i}{2})}^{(x_i + \frac{h_{i+1}}{2})} \frac{1}{k} \nu\Sigma_f(x) \phi(x) dx$$

We'll only add the term we didn't do before:

$$\begin{aligned} \int_{(x_i - \frac{h_i}{2})}^{(x_i + \frac{h_{i+1}}{2})} \frac{1}{k} \nu\Sigma_f(x) \phi(x) dx &= \int_{(x_i - \frac{h_i}{2})}^{(x_i)} \frac{1}{k} \nu\Sigma_f(x) \phi(x) dx + \int_{(x_i)}^{(x_i + \frac{h_{i+1}}{2})} \frac{1}{k} \nu\Sigma_f(x) \phi(x) dx \\ &= \frac{1}{k} \left(\frac{\nu\Sigma_{f,i} h_i + \nu\Sigma_{f,i+1} h_{i+1}}{2} \right) \phi_i. \end{aligned}$$

Collecting all of the terms:

$$-D_{i+1}\left(\frac{\phi_{i+1} - \phi_i}{h_{i+1}}\right) + D_i\left(\frac{\phi_i - \phi_{i-1}}{h_i}\right) + \left(\frac{\Sigma_{a,i}h_i + \Sigma_{a,i+1}h_{i+1}}{2}\right)\phi_i = \frac{1}{k}\left(\frac{\nu\Sigma_{f,i}h_i + \nu\Sigma_{f,i+1}h_{i+1}}{2}\right)\phi_i.$$

To express this in matrix form we'll use the same abbreviations as last time, and add one for the fission term (where we've divided through by h_{ii} to get the x-sec terms):

$$\nu\Sigma_{f,ii} = \frac{\nu\Sigma_{f,i}h_i + \nu\Sigma_{f,i+1}h_{i+1}}{h_i + h_{i+1}}.$$

Then

$$a_{i,i-1}\phi_{i-1} + a_{i,i}\phi_i + a_{i,i+1}\phi_{i+1} = \frac{1}{k}\nu\Sigma_{f,ii}\phi_i \quad \text{for } i = 1, 2, \dots, n-1,$$

where the a s have the same value as last time. (Only the rhs is new.)

We again have a set of $n-1$ linear algebraic equations with $n+1$ unknowns. We will use the boundary conditions to get the rest of the information that we need.

Boundary Conditions

Again assume $x_n = \tilde{a}$, then the **vacuum condition** becomes

$$\phi_n = 0$$

and the last equation for $i = n-1$ becomes

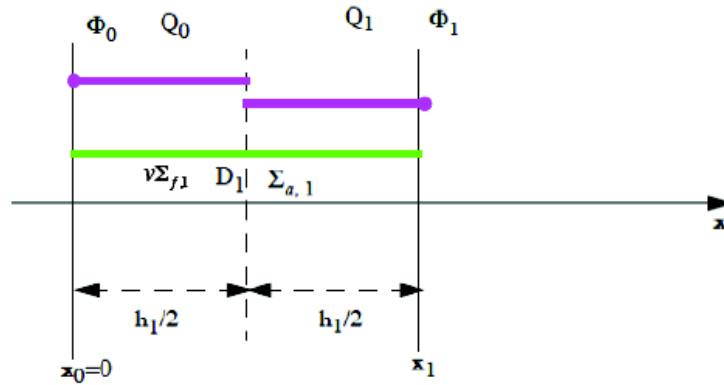
$$a_{n-1,n-2}\phi_{n-2} + a_{n-1,n-1}\phi_{n-1} + 0 = \frac{1}{k}\nu\Sigma_{f,(n-1,n-1)}\phi_{n-1}$$

To include the **reflecting** or zero current condition we integrate over $[0, h_1/2]$.

$$\int_0^{\frac{h_1}{2}} \left(-\frac{d}{dx}D(x)\frac{d\phi(x)}{dx} \right) dx + \int_0^{\frac{h_1}{2}} \Sigma_a(x)\phi(x)dx = \int_0^{\frac{h_1}{2}} \frac{1}{k}\nu\Sigma_f(x)\phi(x)dx.$$

We perform the integration and can apply the boundary condition $\frac{d\phi(x)}{dx}\big|_0 = 0$ just like last time, and the first equation ($i = 0$) becomes

$$a_{00}^*\phi_0 + a_{01}^*\phi_1 = \frac{1}{k}\nu\Sigma_{f,1}\phi_0,$$



where we redefine the a^* s to be the same as last time as well.

We now have n equations and n unknowns, but we formulate it a bit differently:

$$\mathbf{A}\vec{\phi} = \frac{1}{k}\mathbf{F}\vec{\phi},$$

where:

$$\mathbf{F} = \begin{pmatrix} \nu \Sigma_{f,1} & 0 & 0 & 0 & \cdots & 0 \\ 0 & \nu \Sigma_{f,11} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \nu \Sigma_{f,22} & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \nu \Sigma_{f,n-2,n-2} & 0 \\ 0 & \cdots & 0 & 0 & 0 & \nu \Sigma_{f,n-1,n-1} \end{pmatrix}.$$

Solution Methods

We are only going to talk about iterative solution methods for eigenvalue problems since no one uses direct methods in practice (dealing with directly solving an eigenvalue matrix problem rapidly becomes intractable). We will formulate the problem this way, where m is the eigenvalue or “outer” iteration index:

$$\mathbf{A}\vec{\phi}^{(m+1)} = \frac{1}{k^{(m)}}\mathbf{F}\vec{\phi}^{(m)}$$

There are a variety of ways you can choose to determine convergence. We will consider these

convergence criteria:

$$\left| \frac{k^{(m)} - k^{(m-1)}}{k^{(m)}} \right| < \epsilon_1 ,$$

$$\left| \frac{\phi_i^{(m)} - \phi_i^{(m-1)}}{\phi_i^{(m)}} \right| < \epsilon_2 \quad \forall i .$$

Where ϵ_1 is the eigenvalue convergence criterion (often 1×10^{-4} or smaller), and ϵ_2 is the flux error criterion (often 1×10^{-3} or smaller).

Finding k

But wait, that iterative method was only telling us how to update $\vec{\phi}$. How do we get new iterates for k ? To sort that out, we're going to think about the physical interpretation of k .

The multiplication factor, k , can be defined as

$$k = \frac{\text{total production rate}}{\text{total loss rate}}$$

We can define two **operators** (*not matrices*; to get the thing that we solve, we apply our specific discretization methods to turn the operators into matrices) to help us compute this:

$$A = -\frac{d}{dx}D(x)\frac{d}{dx} + \Sigma_a(x) \quad \text{is the loss operator,}$$

$$F = \nu\Sigma_f(x) \quad \text{is the production operator.}$$

This allows us to write an equation for k as

$$k^{(m+1)} = \frac{\int_0^{\tilde{a}} F\vec{\phi}^{(m+1)}(x)dx}{\int_0^{\tilde{a}} A\vec{\phi}^{(m+1)}(x)dx} .$$

We can express our iterative method with our operators,

$$A\vec{\phi}^{(m+1)} = \frac{1}{k^{(m)}}F\vec{\phi}^{(m)}$$

and substitute this into our k equation to get

$$k^{(m+1)} = \frac{\int_0^{\bar{a}} F \vec{\phi}^{(m+1)}(x) dx}{\frac{1}{k^{(m)}} \int_0^{\bar{a}} F \vec{\phi}^{(m)}(x) dx}.$$

This idea applies to any discretization strategy. To use the finite difference formulation we've developed, we discretize the operators and this can be expressed as:

$$k^{(m+1)} = k^{(m)} \left(\frac{\nu \sum_{f,1} \phi_0^{(m+1)} \frac{h_1}{2} + \sum_{i=1}^{n-1} \nu \sum_{f,ii} \phi_i^{(m+1)} \frac{h_{ii}}{2}}{\nu \sum_{f,1} \phi_0^{(m)} \frac{h_1}{2} + \sum_{i=1}^{n-1} \nu \sum_{f,ii} \phi_i^{(m)} \frac{h_{ii}}{2}} \right).$$

Power Method

Power Iteration (PI) is an old and straightforward algorithm for finding an eigenvalue/vector pair.

The basic idea is that any non-zero vector can be written as a linear combination of the eigenvectors of \mathbf{B} because the eigenvectors are linearly independent, namely $\vec{v}_0 = \gamma_1 \vec{x}_1 + \gamma_2 \vec{x}_2 + \cdots + \gamma_n \vec{x}_n$, where \vec{x}_j is the j th eigenvector and γ_j is some constant. This specific expression assumes a non-defective \mathbf{B} , though this assumption is not necessary for the method to work.

Another fact that is used to understand power iteration is that $\mathbf{B}^m \vec{x}_i = \lambda_i^m \vec{x}_i$. Thus

$$\mathbf{B}^m \vec{v}_0 = \gamma_1 \lambda_1^m \vec{x}_1 + \gamma_2 \lambda_2^m \vec{x}_2 + \cdots + \gamma_n \lambda_n^m \vec{x}_n.$$

Since $|\lambda_1| > |\lambda_i|, i \neq 1$, the first term in the expansion will dominate as $m \rightarrow \infty$ and $\mathbf{B}^m \vec{v}_0$ therefore becomes an increasingly accurate approximation to \vec{x}_1 .

In practice, it is desirable to avoid exponentiating a matrix, so we will use an algorithm that does something else. It is also helpful to normalize \vec{v}_0 to avoid possible over or underflow.

We are also quite interested in the *convergence behavior* of PI. After m steps, the iteration vector \vec{v} will be:

$$\vec{v}_m = \left(\frac{\lambda_1^m}{\vec{e}_1^T \mathbf{B}^m \vec{v}_0} \right) \left(\frac{1}{\lambda_1^m} \mathbf{B}^m \vec{v}_0 \right)'$$

where \vec{e}_1^T is a vector with 1 in the first entry and zeros elsewhere; it selects the first row of \mathbf{B} in the multiplication. If \mathbf{B} has eigenpairs $\{(\vec{x}_j, \lambda_j), 1 \leq j \leq n\}$ and \vec{v}_0 has the expansion $\vec{v}_0 = \sum_{j=1}^n \vec{x}_j \gamma_j$ then

$$\frac{1}{\lambda_1^m} \mathbf{B}^m \vec{v}_0 = \frac{1}{\lambda_1^m} \sum_{j=1}^n \mathbf{B}^m \vec{x}_j \gamma_j = \sum_{j=1}^n \left(\frac{\lambda_j}{\lambda_1} \right)^m \vec{x}_j \gamma_j. \quad (1)$$

From equation (1) it can be determined that the error is reduced in each iteration by a factor of $|\frac{\lambda_2}{\lambda_1}|$, which is called the dominance ratio. If λ_2 is close to λ_1 , then this ratio will be close to unity and the method will converge very slowly.

If λ_2 is far from λ_1 , then convergence will happen much more quickly. Put simply, PI is better suited for problems where \mathbf{B} has eigenvalues that are well separated.

Power iteration is very attractive because it only requires matrix-vector products and two vectors of storage space. Because of its simplicity and low storage cost, PI has been widely used in the transport community for criticality problems for quite some time.

Despite these beneficial characteristics, many current codes use an acceleration method with PI or have moved away from it altogether because of the slow convergence for many problems of interest. Nevertheless, it is still used in some codes, has historical relevance, and is used in many studies as a base comparison case.

Algorithm

The power method applies in our formulation when $\mathbf{B} \equiv \mathbf{A}^{-1}\mathbf{F}$, $\lambda \equiv k$ and $\vec{x} \equiv \vec{\phi}$. When we write it this way it is technically **inverse power iteration** because we're using the inverse of \mathbf{A} rather than \mathbf{A} (the theory is the same). Be careful when setting up these solvers about whether you are solving for k or $1/k$ to ensure you get the correct eigenvectors.

Here is an algorithm to implement the power method (note: this is not the most efficient way to do this, but it is likely the clearest).

1. get initial values for $k^{(0)}$ and $\phi^{(0)}$ for $i = 0, \dots, n - 1$; normalize $\phi_0 = \phi_0 / \|\phi_0\|$
2. compute the elements of \mathbf{A}
3. compute the initial fission source

$$\vec{Q}_f^{(0)} = \begin{pmatrix} Q_{f,0}^{(0)} \\ Q_{f,1}^{(0)} \\ \vdots \\ Q_{f,n-1}^{(0)} \end{pmatrix}$$

where $Q_{f,i}^{(0)} = \nu \Sigma_{f,ii} \phi_i^{(0)}$ for $i = 1, \dots, n - 1$ and $Q_{f,0}^{(0)} = \nu \Sigma_{f,1} \phi_0^{(0)}$

4. for $m = 1, \dots$, convergence:

(a) solve (using a method like Jacobi or GS)

$$\mathbf{A}\vec{\phi}^{(m)} = \frac{1}{k^{(m-1)}}\vec{Q}_f^{(m-1)}$$

(b) compute the next fission source $Q_{f,i}^{(m)} = \nu \Sigma_{f,ii} \phi_i^{(m)}$ for $i = 1, \dots, n-1$ and $Q_{f,0}^{(m)} = \nu \Sigma_{f,1} \phi_0^{(m)}$

(c) compute the next eigenvalue:

$$k^{(m)} = k^{(m-1)} \left(\frac{Q_{f,0}^{(m)} \frac{h_1}{2} + \sum_{i=1}^{n-1} Q_{f,i}^{(m)} \frac{h_{ii}}{2}}{Q_{f,0}^{(m-1)} \frac{h_1}{2} + \sum_{i=1}^{n-1} Q_{f,i}^{(m-1)} \frac{h_{ii}}{2}} \right)$$

(d) check for convergence

$$\left| \frac{k^{(m)} - k^{(m-1)}}{k^{(m)}} \right| < \epsilon_1$$

$$\left| \frac{\phi_i^{(m)} - \phi_i^{(m-1)}}{\phi_i^{(m)}} \right| < \epsilon_2$$