

VII. NUMERICAL SOLUTION OF ONE-GROUP DIFFUSION EQUATION IN SLAB GEOMETRY

So far we used analytical methods to solve the neutron diffusion equation. However, we were able to solve only a limited number of geometrically simple cases, such as, for example, one dimensional homogeneous slab geometry. We also had to reduce the number of energy groups to only one. For any more realistic case that has more complicated geometrical features, that is heterogeneous, multidimensional, or have more than one energy group, one must obtain numerical solution. In this Chapter we will explain how to numerically solve the one-group diffusion equation in slab geometry.

VII.1. Formulation of the Finite Difference Equations for the “Fixed-Source” Problem

Let us consider the one-dimensional one-group diffusion equation for a slab of width $2a$ surrounded by vacuum (Fig. VII.1):

$$-\frac{d}{dx}D(x)\frac{d}{dx}\Phi(x) + \Sigma_a(x)\Phi(x) = Q(x). \quad (\text{VII.1})$$

Our problem is symmetric. Thus, we will consider only one half of the slab, with the following boundary conditions:

(a) vacuum boundary conditions on the right side of the slab:

$$\Phi(\tilde{a}) = 0, \quad (\text{VII.2})$$

where $\tilde{a} = a + 2D$ is the extrapolated distance; and

(b) the reflecting boundary condition in the middle of the slab ($a = 0$):

$$\left. \frac{d}{dx}\Phi(x) \right|_{x=0} = 0. \quad (\text{VII.3})$$

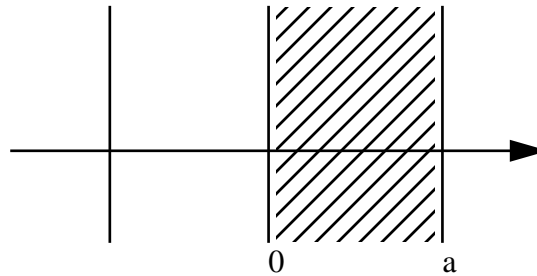


FIGURE VII.29. The one dimensional slab geometry

Note that the source on the RHS of Eq. VII.1 does not contain a fission source, and it is independent on the flux value. Thus, this problem is usually know as the “fixed-source” problem. Even this equation would be difficult to solve analytically. We now impose the following spatial mesh that divides the slab into “cells”:

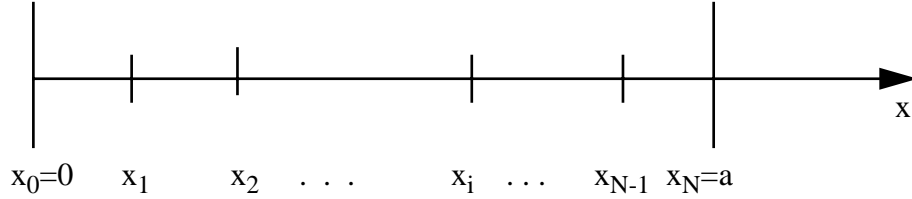


FIGURE VII.30. Cell-edged spatial mesh

Any 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:

$$D(x) = D_i, \text{ for } x_{i-1} \leq x \leq x_i \quad (\text{VII.4})$$

$$\Sigma_a(x) = \Sigma_{a,i}, \text{ for } x_{i-1} \leq x \leq x_i. \quad (\text{VII.5})$$

The unknown fluxes and known sources are defined at the mesh (or cell) edges:

$$\Phi(x_i) = \Phi_i \quad (\text{VII.6})$$

$$Q(x_i) = Q_i. \quad (\text{VII.7})$$

We also define the mesh spacing:

$$h_i \equiv x_i - x_{i-1}. \quad (\text{VII.8})$$

In addition, we assume that the fluxes and sources are constant over the interval centered around x_i :

$$\Phi(x) = \Phi(x_i) = \Phi_i, \text{ for } \left(x_i - \frac{h_i}{2}\right) \leq x \leq \left(x_i + \frac{h_{i+1}}{2}\right), \text{ and} \quad (\text{VII.9})$$

$$Q(x) = Q(x_i) = Q_i, \text{ for } \left(x_i - \frac{h_i}{2}\right) \leq x \leq \left(x_i + \frac{h_{i+1}}{2}\right). \quad (\text{VII.10})$$

Figure VII.3 describes “graphically” our assumptions.

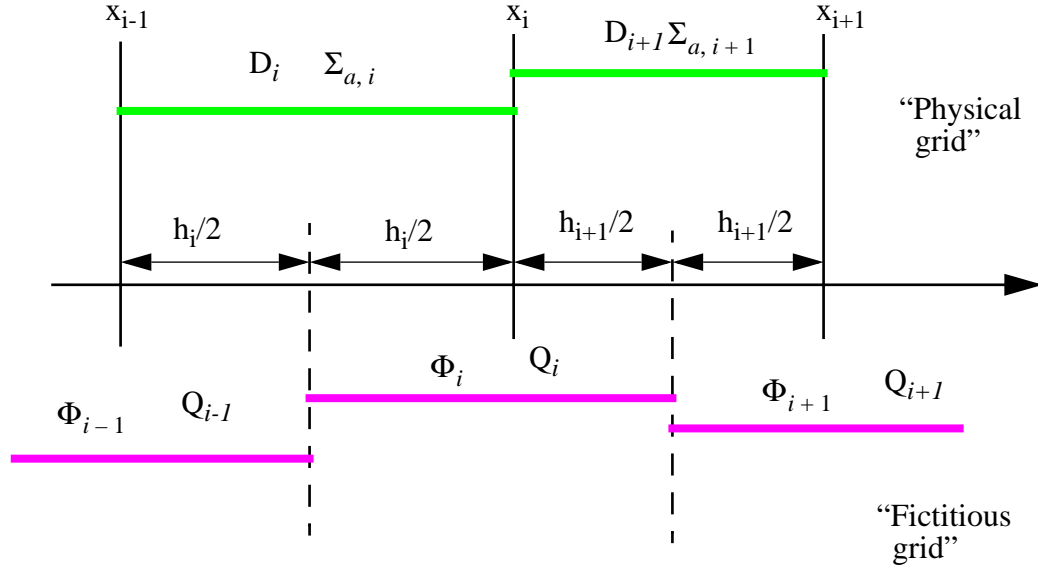


FIGURE VII.31. “Physical” and Fictitious” grid definition and assumptions

The next step is to integrate Eq. VII.1 over the “fictitious” cell

$$\left(x_i - \frac{h_i}{2}\right) \leq x \leq \left(x_i + \frac{h_{i+1}}{2}\right) :$$

$$\int_{(x_i - h_i/2)}^{(x_i + h_{i+1}/2)} \left(-\frac{d}{dx}D(x)\frac{d}{dx}\Phi(x)\right)dx + \int_{(x_i - h_i/2)}^{(x_i + h_{i+1}/2)} \Sigma_a(x)\Phi(x)dx = \int_{(x_i - h_i/2)}^{(x_i + h_{i+1}/2)} Q(x)dx. \quad (\text{VII.11})$$

Let us examine each term in Eq. VII.11.

$$\int_{(x_i - h_i/2)}^{(x_i + h_{i+1}/2)} Q(x)dx \approx Q_i \left(\frac{h_i + h_{i+1}}{2}\right) \quad (\text{VII.12})$$

In order to integrate the second term on the LHS of Eq. VII.11, we have to subdivide the integration interval into two, so that both, the flux and cross sections are constant over that interval:

$$\int_{(x_i - h_i/2)}^{(x_i)} \Sigma_a(x)\Phi(x)dx + \int_{(x_i)}^{(x_i + h_{i+1}/2)} \Sigma_a(x)\Phi(x)dx = \left(\frac{\Sigma_{a,i}h_i + \Sigma_{a,i+1}h_{i+1}}{2}\right)\Phi_i \quad (\text{VII.13})$$

The first term in Eq. VII.11 can be written as:

$$\int_{(x_i - h_i/2)}^{(x_i + h_{i+1}/2)} \left(-\frac{d}{dx} D(x) \frac{d}{dx} \Phi(x) \right) dx = - \left[D(x) \frac{d}{dx} \Phi(x) \right]_{(x_i - h_i/2)}^{x_i + h_{i+1}/2} \quad (\text{VII.14})$$

Now we will do the first numerical approximation - we will replace the derivative term by a "finite difference" which is simply an algebraic approximation to the derivative:

$$-D(x) \frac{d}{dx} \Phi(x) \Big|_{x_i + h_{i+1}/2} \cong -D_{i+1} \left(\frac{\Phi_{i+1} - \Phi_i}{h_{i+1}} \right), \quad (\text{VII.15})$$

and

$$D(x) \frac{d}{dx} \Phi(x) \Big|_{x_i - h_i/2} \cong D_i \left(\frac{\Phi_i - \Phi_{i-1}}{h_i} \right) \quad (\text{VII.16})$$

Collecting all of the terms, we obtain:

$$-D_{i+1} \left(\frac{\Phi_{i+1} - \Phi_i}{h_{i+1}} \right) + D_i \left(\frac{\Phi_i - \Phi_{i-1}}{h_i} \right) + \left(\frac{\Sigma_{a,i} h_i + \Sigma_{a,i+1} h_{i+1}}{2} \right) \Phi_i = Q_i \left(\frac{h_i + h_{i+1}}{2} \right) \quad (\text{VII.17})$$

Define:

$$h_{ii} \equiv \frac{h_i + h_{i+1}}{2} \quad (\text{VII.18})$$

$$\Sigma_{a,ii} \equiv \frac{\Sigma_{a,i} h_i + \Sigma_{a,i+1} h_{i+1}}{h_i + h_{i+1}} \quad (\text{VII.19})$$

By rearranging the terms in Eq. VII.17, and using the definitions in Eqs. VII.18 and VII.19, Eq. VII.17 can be written:

$$a_{i,i-1} \Phi_{i-1} + a_{i,i} \Phi_i + a_{i,i+1} \Phi_{i+1} = Q_i, \quad (\text{VII.20})$$

for $i = 1, 2, \dots, N-1$, where

$$a_{i,i-1} \equiv -\frac{D_i}{h_i h_{ii}} \quad (\text{VII.21})$$

$$a_{ii} = \frac{D_i}{h_i h_{ii}} + \frac{D_{i+1}}{h_{i+1} h_{ii}} + \Sigma_{a,ii} \quad (\text{VII.22})$$

$$a_{i,i+1} \equiv -\frac{D_{i+1}}{h_{i+1}h_{ii}} \quad (\text{VII.23})$$

Equation VII.20 represents a set of (N-1) algebraic equations with (N+1) unknowns: Φ_0 , Φ_1 , Φ_2 , ..., Φ_N . In order to assure that we have the same number of equations and unknowns, we have to use our boundary conditions.

(1) Vacuum boundary condition

If we assure that $x_N = \tilde{a}$, the vacuum boundary condition (Eq. VII.2) simply becomes:

$$\Phi_N = 0, \quad (\text{VII.24})$$

and the last equation of the set (Eq. VII.20) for $i = N-1$ becomes:

$$a_{N-1,N-2}\Phi_{N-2} + a_{N-1,N-1}\Phi_{N-1} = Q_{N-1} \quad (\text{VII.25})$$

(2) Reflecting (or zero-current) boundary condition

In order to incorporate the second boundary condition, we will integrate Eq. VII.1 over the region $(0, h_1/2)$. This cell is shown in Fig. VII. 4.

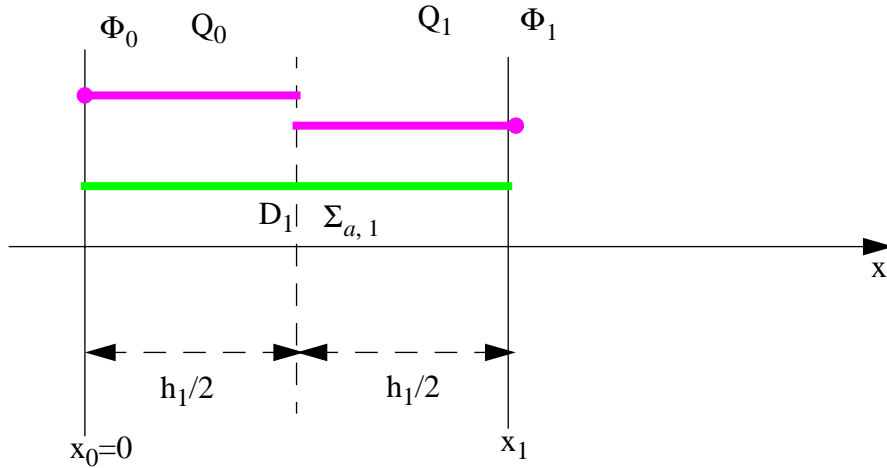


FIGURE VII.32. Boundary mesh cell

$$\int_0^{\left(\frac{h_1}{2}\right)} \left(-\frac{d}{dx}D(x)\frac{d}{dx}\Phi(x)\right)dx + \int_0^{\left(\frac{h_1}{2}\right)} \Sigma_a(x)\Phi(x)dx = \int_0^{\left(\frac{h_1}{2}\right)} Q(x)dx \quad (\text{VII.26})$$

After integration, we obtain:

$$-D(x)\frac{d\Phi}{dx}\Bigg|_{\left(\frac{h_1}{2}\right)} + D(x)\frac{d\Phi(x)}{dx}\Bigg|_0 + \Sigma_{a,1}\Phi_0\frac{h_1}{2} = Q_0\frac{h_1}{2} \quad (\text{VII.27})$$

Using our second boundary condition:

$$\frac{d\Phi(x)}{dx}\Bigg|_{x=0} = 0 \quad (\text{VII.28})$$

we obtain

$$-D_1\left(\frac{\Phi_1 - \Phi_0}{h_1}\right) + \Sigma_{a,1}\Phi_0\frac{h_1}{2} = Q_0\frac{h_1}{2}, \quad (\text{VII.29})$$

which can be written as

$$a_{00}\Phi_0 + a_{01}\Phi_1 = Q_0 \quad (\text{VII.30})$$

where

$$a_{00} = \frac{2D_1}{(h_1)^2} + \Sigma_{a,1}$$

$$a_{01} = -\frac{2D_1}{(h_1)^2}.$$

We can now combine all equations:

i = 0

$$a_{00}\Phi_0 + a_{01}\Phi_1 = Q_0 \quad (\text{VII.31})$$

i = 1

$$a_{10}\Phi_0 + a_{11}\Phi_1 + a_{12}\Phi_2 = Q_1 \quad (\text{VII.32})$$

i = 2

$$a_{21}\Phi_1 + a_{22}\Phi_2 + a_{23}\Phi_3 = Q_2 \quad (\text{VII.33})$$

Or, in general

$$a_{i,i-1}\Phi_{i-1} + a_{i,i}\Phi_i + a_{i,i}\Phi_i = Q_i \quad (\text{VII.34})$$

The last equation is obtained for i = N-1:

$$a_{N-1, N-2} \Phi_{N-2} + a_{N-1, N-1} \Phi_{N-1} = Q_{N-1} \quad (\text{VII.35})$$

We have a system of N linear algebraic equations with N unknowns that we can solve using either direct or iterative methods. This system can be also written in matrix form:

$$\underline{\mathbf{A}} \underline{\Phi} = \underline{\mathbf{Q}} \quad (\text{VII.36})$$

where

$$\begin{bmatrix} a_{00} & a_{01} & 0 & 0 & & 0 & 0 \\ a_{10} & a_{11} & a_{12} & 0 & \cdots & 0 & 0 \\ 0 & a_{21} & a_{22} & a_{23} & \cdots & 0 & 0 \\ & & & & \cdots & & \\ 0 & 0 & 0 & 0 & \cdots & a_{N-1, N-2} & a_{N-1, N-1} \end{bmatrix} = \underline{\mathbf{A}} \quad (\text{VII.37})$$

$$\begin{bmatrix} \Phi_0 \\ \Phi_1 \\ \cdot \\ \cdot \\ \Phi_{N-1} \end{bmatrix} = \underline{\Phi} \quad (\text{VII.38})$$

$$\begin{bmatrix} Q_0 \\ Q_1 \\ \cdot \\ \cdot \\ Q_{N-1} \end{bmatrix} = \underline{\mathbf{Q}}. \quad (\text{VII.39})$$

However, if we have a homogeneous medium and a uniform mesh, the system of equations will be simplified. In this case, for each i we have:

$$h_i = h,$$

$$D_i = D$$

$$\Sigma_{a, i} = \Sigma_a$$

and

$$-\frac{D}{h^2}\Phi_{i-1} + \left(\frac{2D}{h^2} + \Sigma_a\right)\Phi_i - \frac{D}{h^2}\Phi_{i+1} = Q_i \quad (\text{VII.40})$$

for $i = 1, 2, \dots, N-2$. For $i = 0$, we have

$$\left(\frac{2D}{h^2} + \Sigma_a\right)\Phi_0 - \frac{2D}{h^2}\Phi_1 = Q_0, \quad (\text{VII.41})$$

and for $i = N-1$, we have

$$-\frac{D}{h^2}\Phi_{N-2} + \left(\frac{2D}{h^2} + \Sigma_a\right)\Phi_{N-1} = Q_{N-1}. \quad (\text{VII.42})$$

VII.2. Solution of Tridiagonal System of Equations

The system of linear algebraic equations (Eq. VII.36) is called tridiagonal because only the main diagonal and the two diagonals immediately above and below the main diagonal have non-zero elements. The formal solution of this system requires inversion of matrix \underline{A} , i.e.

$$\underline{\Phi} = \underline{A}^{-1}\underline{Q} \quad (\text{VII.43})$$

However, we usually try to avoid direct inversion of a matrix, particularly if that matrix is large, because it requires considerable amount of computer memory and time. In that case, it is far more efficient to use an iterative method, since it does not require the inversion of a full matrix, and needs far less computer memory.

VII.2.1 Gaussian Elimination

The tridiagonal systems can be solved directly (explicitly) by using *Gaussian elimination* (“forward elimination - backward substitution”) method. The Gaussian elimination works well in our case, because matrix \underline{A} is *diagonally dominant*, i.e., each diagonal element is greater than the sums of absolute values of off-diagonal elements in the same row:

$$a_{ii} \geq |a_{i,i-1}| + |a_{i,i+1}|. \quad (\text{VII.44})$$

In order to come up with an algorithm that can be easily programmed on a computer, we will rewrite the system of equations (Eq. VII.36) in the following form:

$$\begin{bmatrix} B_1 & -C_1 & 0 & 0 & & 0 & 0 \\ -A_2 & B_2 & -C_2 & 0 & \cdots & 0 & 0 \\ 0 & -A_3 & B_3 & -C_3 & \cdots & 0 & 0 \\ & & & & \cdots & & \\ & & & & & B_{N-1} & -C_{N-1} \\ & & & & & -A_N & B_N \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \cdot \\ \cdot \\ \Phi_N \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ \cdot \\ \cdot \\ Q_N \end{bmatrix} \quad (\text{VII.45})$$

Note that we just changed the notation for the elements of matrix A, and shifted the index by 1. Thus $\Phi_{N+1} = 0$ in this notation. In order to describe the Gaussian elimination procedure, let us consider a small 3x3 system of equations:

$$B_1 \Phi_1 - C_1 \Phi_2 = Q_1 \quad (\text{VII.46})$$

$$-A_2 \Phi_1 + B_2 \Phi_2 - C_2 \Phi_3 = Q_2 \quad (\text{VII.47})$$

$$-A_3 \Phi_2 + B_3 \Phi_3 = Q_3 \quad (\text{VII.48})$$

The Gaussian elimination method has two steps: (1) the system is reduced to an “upper triangular” system, and (2) the upper triangular system is solved explicitly by a backward substitution,

STEP 1: Forward Elimination

1. Define

$$u_1 = B_1, \quad v_1 = Q_1 \quad (\text{VII.49})$$

and write Eq. VII.46 as

$$u_1 \Phi_1 - C_1 \Phi_2 = v_1. \quad (\text{VII.50})$$

2. Multiply Eq. VII.50 by A_2/u_1 :

$$A_2 \Phi_1 - \frac{A_2 C_1}{u_1} \Phi_2 = \frac{A_2 v_1}{u_1} \quad (\text{VII.51})$$

Add Eq. VII.51 to Eq. VII.47:

$$\left(B_2 - \frac{A_2 C_1}{u_1} \right) \Phi_2 - C_2 \Phi_3 = Q_2 + \frac{A_2 v_1}{u_1} \quad (\text{VII.52})$$

Define:

$$u_2 = B_2 - \frac{A_2 C_1}{u_1}; \text{ and } v_2 = Q_2 + \frac{A_2 v_1}{u_1}. \quad (\text{VII.53})$$

Then Eq. VII.52 can be written as

$$u_2 \Phi_2 - C_2 \Phi_3 = v_2 \quad (\text{VII.54})$$

3. Multiply Eq. VII.54 by A_3/u_2 :

$$A_3 \Phi_2 - \frac{A_3 C_2}{u_2} \Phi_3 = \frac{A_2 v_2}{u_2} \quad (\text{VII.55})$$

Add Eq. VII.55 to Eq. VII.48:

$$\left(B_3 - \frac{A_3 C_2}{u_2} \right) \Phi_3 = Q_3 + \frac{A_3 v_2}{u_2} \quad (\text{VII.56})$$

Define

$$u_3 = B_3 - \frac{A_3 C_2}{u_2}, \text{ and } v_3 = Q_3 + \frac{A_3 v_2}{u_2}, \quad (\text{VII.57})$$

and rewrite Eq. VII.56 as

$$u_3 \Phi_3 = v_3 \quad (\text{VII.58})$$

The system of equations (Eq. VII.46, VII.47, and VII.48) has now been converted in upper triangular system:

$$u_1 \Phi_1 - C_1 \Phi_2 = v_1 \quad (\text{VII.59})$$

$$u_2 \Phi_2 - C_2 \Phi_3 = v_2 \quad (\text{VII.60})$$

$$u_3 \Phi_3 = v_3 \quad (\text{VII.61})$$

STEP 2: Backward Substitution

4. From Eq. VII.61 we obtain: $\Phi_3 = v_3/u_3$.

5. Then, from Eq. VII.60 we obtain: $\Phi_2 = \frac{1}{u_2}(v_2 + C_2 \Phi_3)$.

6. Finally, from Eq. VII.59 we obtain: $\Phi_1 = \frac{1}{u_1}(v_1 + C_1\Phi_2)$.

The same procedure can be applied to a tridiagonally dominant system of any size. We can now generalize this procedure for our system given in Eq. VII.45:

STEP 1: Forward Elimination

1. Define for $i = 1$

$$u_1 = B_1, \quad v_1 = Q_1, \quad (\text{VII.62})$$

and then, for $i = 2, 3, \dots, N$, define recursively

$$u_i = B_i - \frac{A_i C_{i-1}}{u_{i-1}}, \text{ and } v_i = Q_i + \frac{A_i v_{i-1}}{u_{i-1}}. \quad (\text{VII.63})$$

STEP 2: Backward Substitution

2. Now, calculate

$$\Phi_N = \frac{1}{u_N} v_N \quad (\text{VII.64})$$

and then, for $i = N-1, N-2, \dots, 2, 1$, calculate recursively

$$\Phi_i = \frac{1}{u_i}(v_i + C_i\Phi_{i+1}). \quad (\text{VII.65})$$

This procedure explicitly solves the system of equations given by Eq. VII.45 for the fixed-source diffusion problem.

VII.2.2 Iterative Methods

In the case of large problems (more than 10,000 unknowns), Gaussian elimination can be very slow (or expensive), because it includes a large number of computations, and can require a large amount of computer memory. It is far more efficient to use an iterative method to invert large matrices. The idea of all iterative methods is to solve a simpler equation (or a set of equations) several times, hoping that the solutions will approach the true solution (i.e., converge to the true solution) with the increased number of iterations:

$$\lim_{n \rightarrow \infty} \Phi^{(n)} \rightarrow \Phi_{exact} \quad (\text{VII.66})$$

where n is the iteration index.

Jacobi Iterative Method

Suppose that we want to solve the following systems of equations:

$$\underline{\mathbf{A}}\Phi = \underline{\mathbf{Q}} \quad (\text{VII.67})$$

i.e., we need to invert matrix $\underline{\mathbf{A}}$. Let us write $\underline{\mathbf{A}}$ as a sum of two matrices:

$$\underline{\mathbf{A}} = \underline{\mathbf{D}} + \underline{\mathbf{B}}, \quad (\text{VII.68})$$

where $\underline{\mathbf{D}}$ consists of the diagonal elements of $\underline{\mathbf{A}}$ (off-diagonal elements are zero), and $\underline{\mathbf{B}}$ that consists of off-diagonal elements of $\underline{\mathbf{A}}$ (with the diagonal elements equal to zero). We can now rewrite Eq. VII.67 as

$$\underline{\mathbf{D}}\Phi = \underline{\mathbf{Q}} - \underline{\mathbf{B}}\Phi, \quad (\text{VII.69})$$

where we have the unknown flux vector on both sides of the equation. Because $\underline{\mathbf{D}}$ is a diagonal matrix, it can be easily inverted, and we can write Eq. VII.69 as

$$\Phi = \underline{\mathbf{D}}^{-1}\underline{\mathbf{Q}} - \underline{\mathbf{D}}^{-1}(\underline{\mathbf{B}}\Phi). \quad (\text{VII.70})$$

Now, we apply the iterative procedure. Suppose we guess Φ on the right-hand side of Eq. VII.70, and denote it as $\Phi^{(0)}$ or zeroth iterate. Then we use it to calculate a new value of flux, $\Phi^{(1)}$ from Eq. VII.70:

$$\Phi^{(1)} = \underline{\mathbf{D}}^{-1}\underline{\mathbf{Q}} - \underline{\mathbf{D}}^{-1}(\underline{\mathbf{B}}\Phi^{(0)}). \quad (\text{VII.71})$$

We continue with this process, and in the n -th iteration we have:

$$\Phi^{(n)} = \underline{\mathbf{D}}^{-1}\underline{\mathbf{Q}} - \underline{\mathbf{D}}^{-1}(\underline{\mathbf{B}}\Phi^{(n-1)}). \quad (\text{VII.72})$$

We continue such an iterative process until two successive iterates $\Phi^{(n-1)}$ and $\Phi^{(n)}$ agree within the given accuracy, and we say that the process has converged. Let us write Eq. VII.72 explicitly in terms of the algebraic system:

$$\begin{aligned}
 a_{11}\Phi_1^{(n)} + a_{12}\Phi_2^{(n-1)} + \dots + a_{1N}\Phi_N^{(n-1)} &= Q_1 \\
 a_{21}\Phi_1^{(n-1)} + a_{22}\Phi_2^{(n)} + \dots + a_{2N}\Phi_N^{(n-1)} &= Q_2 \\
 &\dots \\
 a_{N1}\Phi_1^{(n-1)} + a_{N2}\Phi_2^{(n-1)} + \dots + a_{NN}\Phi_N^{(n)} &= Q_N
 \end{aligned} \tag{VII.73}$$

Thus, we can solve for the (n-th) flux iterate immediately as:

$$\Phi_i^{(n)} = \frac{1}{a_{ii}} \left[Q_i - \sum_{j=1}^{i-1} a_{ij}\Phi_j^{(n-1)} - \sum_{j=i+1}^N a_{ij}\Phi_j^{(n-1)} \right], \quad i = 1, 2, \dots, N. \tag{VII.74}$$

Gauss-Seidel Iterative Method

The Gauss-Seidel iterative method is a simple modification of the Jacobi method, that speeds up the convergence process. If we analyze the Jacobi method more closely, we find out that it does not use all of the available information during each iteration. From the first equation in Eq. VII.73 we find $\Phi_1^{(n)}$. However, in order to find $\Phi_2^{(n)}$ from the second equation, $\Phi_1^{(n-1)}$ is used rather than the improved estimate $\Phi_1^{(n)}$. If the improved estimates are used as soon as they are generated, a more efficient iterative scheme (Gauss-Seidel scheme) is obtained:

$$\begin{aligned}
 a_{11}\Phi_1^{(n)} + a_{12}\Phi_2^{(n-1)} + \dots + a_{1N}\Phi_N^{(n-1)} &= Q_1 \\
 a_{21}\Phi_1^{(n)} + a_{22}\Phi_2^{(n)} + \dots + a_{2N}\Phi_N^{(n-1)} &= Q_2 \\
 &\dots \\
 a_{N1}\Phi_1^{(n)} + a_{N2}\Phi_2^{(n)} + \dots + a_{NN}\Phi_N^{(n)} &= Q_N
 \end{aligned} \tag{VII.75}$$

The solution can be written as:

$$\Phi_i^{(n)} = \frac{1}{a_{ii}} \left[Q_i - \sum_{j=1}^{i-1} a_{ij}\Phi_j^{(n)} - \sum_{j=i+1}^N a_{ij}\Phi_j^{(n-1)} \right], \quad i = 1, 2, \dots, N. \tag{VII.76}$$

The fact that the Gauss-Seidel method is using the latest iterates as soon as they become available yields a factor of two better error reduction per iteration than in the case of the Jacobi method.

Successive Overrelaxation (SOR) Method

We can accelerate the convergence even further by introducing an acceleration parameter to extrapolate the flux estimate. The iterative algorithm can then be written:

$$\Phi_i^{(n)} = \frac{\omega}{a_{ii}} \left[Q_i - \sum_{j=1}^{i-1} a_{ij} \Phi_j^{(n)} - \sum_{j=i+1}^N a_{ij} \Phi_j^{(n-1)} \right] + (1 - \omega) \Phi_i^{(n-1)}, \quad (\text{VII.77})$$

where the acceleration parameter ω ranges between 1 and 2. If the optimum value of ω is chosen, the SOR convergence rate can be as much as two orders of magnitude faster than the convergence rate of the Jacobi method. However, ω depends on the characteristics of matrix \underline{A} , and can strongly affect the convergence rate. Thus, the value for ω is usually determined experimentally.

VII.3. Formulation of the Finite Difference Equations for the “Eigenvalue (Criticality)” Problem

In the above, we described the numerical solution of the fixed source problem. Let us now formulate the finite difference equations of the k-eigenvalue problem:

$$-\frac{d}{dx} D(x) \frac{d}{dx} \Phi(x) + \Sigma_a(x) \Phi(x) = \frac{1}{k} \nu \Sigma_f(x) \Phi(x). \quad (\text{VII.78})$$

Our problem is symmetric. Thus, we will consider only one half of the slab, with the following boundary conditions:

(a) vacuum boundary conditions on the right side of the slab:

$$\Phi(\tilde{a}) = 0, \quad (\text{VII.79})$$

where $\tilde{a} = a + 2D$ is the extrapolated distance; and

(b) the reflecting boundary condition in the middle of the slab ($a = 0$):

$$\left. \frac{d}{dx} \Phi(x) \right|_{x=0} = 0. \quad (\text{VII.80})$$

The left-hand side of Eq. VII.78 is the same as the left-hand side of Eq. VII.1. Thus, the finite difference formulation is exactly the same. The term $\nu \Sigma_f(x) \Phi(x)$ is similar to the term $\Sigma_a(x) \Phi(x)$, and can be discretized similarly. Therefore, we obtain

$$a_{i,i-1} \Phi_{i-1} + a_{i,i} \Phi_i + a_{i,i+1} \Phi_{i+1} = \frac{1}{k} \nu \Sigma_{f,ii} \Phi_i, \quad i = 1, 2, \dots, N-1, \quad (\text{VII.81})$$

where the coefficients $a_{i,i-1}$, $a_{i,i}$ and $a_{i,i+1}$ are given by Eq. VII.21, VII.22, and VII.23 respectively, while the fission coefficient is given by

$$v\Sigma_{f,ii} \equiv \frac{v\Sigma_{f,i}h_i + v\Sigma_{f,i+1}h_{i+1}}{h_i + h_{i+1}}. \quad (\text{VII.82})$$

Equation VII.81 can be now expressed in matrix form as

$$\underline{A}\underline{\Phi} = \frac{1}{k}\underline{F}\underline{\Phi} \quad (\text{VII.83})$$

where \underline{A} is the same as in the fixed source problem (Eq. VII.37), and \underline{F} is a diagonal matrix

$$\begin{bmatrix} v\Sigma_{f,11} & 0 & & 0 \\ 0 & v\Sigma_{f,22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & & v\Sigma_{f,N-1,N-1} \end{bmatrix} = \underline{F} \quad (\text{VII.84})$$

This is an eigenvalue problem that can be solved using the *inverse power iteration* method. We start by guessing the initial values of flux vector, $\underline{\Phi}^{(0)}$, and eigenvalue $k^{(0)}$, and solve

$$\underline{A}\underline{\Phi}^{(1)} = \frac{1}{k^{(0)}}\underline{F}\underline{\Phi}^{(0)}. \quad (\text{VII.85})$$

we then use $\underline{\Phi}^{(1)}$ to calculate $k^{(1)}$, and continue with this process until the following criteria are satisfied:

$$\left| \frac{k^{(n)} - k^{(n-1)}}{k^{(n)}} \right| < \varepsilon_1 \quad (\text{VII.86})$$

$$\left| \frac{\Phi_i^{(n)} - \Phi_i^{(n-1)}}{\Phi_i^{(n)}} \right| < \varepsilon_2. \quad (\text{VII.87})$$

Typically, $\varepsilon_1 \cong 0.0001$ and $\varepsilon_2 \cong 0.001$. The first criterion is known as “the eigenvalue convergence criterion”, and the second as “the flux error criterion”.

We can now explain the inverse power iteration method in more detail.

First, we need to find a way to determine the new value for k. The multiplication factor k can be defined as:

$$k = \frac{\text{total production rate}}{\text{total loss rate}} \quad (\text{VII.88})$$

We can now introduce two operators (*not* matrices) A and F:

$$A = -\frac{d}{dx}D(x)\frac{d}{dx} + \Sigma_a(x), \text{ as a loss operator,} \quad (\text{VII.89})$$

$$F = \nu\Sigma_f(x) \text{ as a production operator.} \quad (\text{VII.90})$$

Equation VII.88 can then be written as

$$k^{(1)} = \frac{\int_0^{\tilde{a}} F\Phi^{(1)}(x)dx}{\int_0^{\tilde{a}} A\Phi^{(1)}(x)dx} \quad (\text{VII.91})$$

where we used the flux estimate in the first iteration to calculate new value of k. However, in order to compute the flux estimate in the first iteration we used:

$$A\Phi^{(1)}(x) = \frac{1}{k^{(0)}}F\Phi^{(0)}(x). \quad (\text{VII.92})$$

Thus, Eq. VII.91 can be written as

$$k^{(1)} = \frac{\int_0^{\tilde{a}} F\Phi^{(1)}(x)dx}{\left(\frac{1}{k^{(0)}}\right)\int_0^{\tilde{a}} F\Phi^{(0)}(x)dx}. \quad (\text{VII.93})$$

In the finite difference formulation, Eq. VII.93 can be expressed as

$$k^{(1)} = k^{(0)} \left(\frac{\nu\Sigma_{f,1}\Phi_0^{(1)}\frac{h_1}{2} + \sum_{i=1}^{N-1} \nu\Sigma_{f,ii}\Phi_i^{(1)}h_{ii}}{\nu\Sigma_{f,1}\Phi_0^{(0)}\frac{h_1}{2} + \sum_{i=1}^{N-1} \nu\Sigma_{f,ii}\Phi_i^{(0)}h_{ii}} \right). \quad (\text{VII.94})$$

The following are the steps in the inverse power iteration algorithm to solve the criticality problem:

1. Guess $k^{(0)}$ and $\Phi_i^{(0)}$ for $i = 0, 1, 2, \dots, N-1$.

2. Compute the elements of matrix \underline{A} .

3. Compute the fission source

$$\underline{Q}_f^{(0)} = \begin{bmatrix} Q_{f,0}^{(0)} \\ Q_{f,1}^{(0)} \\ \dots \\ Q_{f,N-1}^{(0)} \end{bmatrix} \quad (\text{VII.95})$$

where

$$Q_{f,i}^{(0)} = v \Sigma_{f,ii} \Phi_i^{(0)}, \quad i = 0, 1, 2, \dots, N-1. \quad (\text{VII.96})$$

4. Solve

$$\underline{A} \underline{\Phi}^{(1)} = \frac{1}{k^{(0)}} \underline{F} \underline{\Phi}^{(0)}. \quad (\text{VII.97})$$

5. Compute the next fission source

$$Q_{f,i}^{(1)} = v \Sigma_{f,ii} \Phi_i^{(1)}, \quad i = 0, 1, 2, \dots, N-1. \quad (\text{VII.98})$$

6. Compute the new eigenvalue $k^{(1)}$:

$$k^{(1)} = k^{(0)} \left(\frac{Q_{f,0}^{(1)} \frac{h_1}{2} + \sum_{i=1}^{N-1} Q_{f,i}^{(1)} h_{ii}}{Q_{f,0}^{(0)} \frac{h_1}{2} + \sum_{i=1}^{N-1} Q_{f,i}^{(0)} h_{ii}} \right). \quad (\text{VII.99})$$

7. Repeat for $n = 2, 3, \dots$

a) Solve

$$\underline{A} \underline{\Phi}^{(n)} = \frac{1}{k^{(n-1)}} \underline{F} \underline{\Phi}^{(n-1)}. \quad (\text{VII.100})$$

b) Compute

$$\dots \quad (\text{VII.101})$$

$$k^{(n)} = k^{(n-1)} \left(\frac{Q_{f,0}^{(n)} \frac{h_1}{2} + \sum_{i=1}^{N-1} Q_{f,i}^{(n)} h_{ii}}{Q_{f,0}^{(n-1)} \frac{h_1}{2} + \sum_{i=1}^{N-1} Q_{f,i}^{(n-1)} h_{ii}} \right). \quad (\text{VII.102})$$

c) For all $i = 1, 2, \dots, N-1$ check if the following criteria are satisfied:

$$\left| \frac{k^{(n)} - k^{(n-1)}}{k^{(n)}} \right| < \varepsilon_1 \quad (\text{VII.103})$$

$$\left| \frac{\Phi_i^{(n)} - \Phi_i^{(n-1)}}{\Phi_i^{(n)}} \right| < \varepsilon_2. \quad (\text{VII.104})$$

If YES, stop the iterative process; if NO continue with the iterative process.