1D2G - Numerical solution of the neutron diffusion equation

Y. Danon

Draft: 2/16/2019

1 Overview

A simple numerical solution of the neutron diffusion equation in one dimension and two energy groups was implemented. Both fast and thermal fluxes, and the multiplication factor of the system are solved for. The implementation allows to divide the geometry to several regions and use different meshing (number of grid points) in each region. One example is a core and reflector, where the thermal flux peaking in the reflector can be observed. The treatment of the diffusion term (Laplacian) allow geometry of slab, semi-cylinder and sphere.

1.1 History

The derivation and code were developed as a final project for senior undergraduate course in numerical computing that I took back in 1988. The original code was written in Pascal (specifically Turbo Pascal) which was an upcoming programming language at that time. Later I used it in Physics of Nuclear Reactor course at RPI as an example for a numerical solution of the diffusion equation. The at part of HW problem the solution was compared to simple analytic problems. for implementation in the course tt was rewritten in MathCad and JavaScript.
The document is provided as a documentation of the methods used in the implementation.

2 Solution method

We start by using the one dimension (1-D) two group (2-G) diffusion equations derived in the class notes:

$$\text{Fast:} \quad -\nabla D_1 \nabla \phi_1 + \Sigma_{R1}\phi_1 = \frac{1}{k}\left(\nu_1 \Sigma_{f1}\phi_1 + \nu_2 \Sigma_{f2}\phi_2\right) \tag{1}$$

$$\text{Thermal:} \quad -\nabla D_2 \nabla \phi_2 + \Sigma_{a2}\phi_2 = \Sigma_{s1\to 2}\phi_1 \tag{2}$$

The overall approach was described in the class notes and will be repeated here. For convenience we first write the equations using operators:

$$L_1 = \quad -\nabla D_1 \nabla + \Sigma_{R1} \tag{3}$$

$$L_2 = \quad -\nabla D_2 \nabla + \Sigma_{a2} \tag{4}$$

We also define the source term as:

$$S_1 = \nu_1 \Sigma_{f1}\phi_1 + \nu_2 \Sigma_{f2}\phi_2 \tag{5}$$

$$S_2 = \Sigma_{s1\to 2}\phi_1 \tag{6}$$

The diffusion equations can be rewritten as:

$$L_1\phi_1 = \frac{1}{k}S_1 \tag{7}$$
$$L_2\phi_2 = S_2 \tag{8}$$

2.1 Iterations

The method of solution is a variation of the simple source iteration (also known as power method). This is an iterative solution solves for the multiplication factor k and the flux shapes $\phi_1(r)$ and $\phi_2(r)$. Since this is a 1-D solution, we divide the spatial direction r to N nodes (or $N-1$ intervals) from r_0 to r_N. If needed the extrapolation length can easily be added by making the core size larger.

1. Guess an initial solution (iteration zero) k^0, $\phi_1(r)^0$ and $\phi_2(r)^0$
2. Set iteration index $i = 1$
3. Calculate S_1^{i-1} using $S_1^{i-1} = \nu_1\Sigma_{f1}\phi_1^{i-1} + \nu_2\Sigma_{f2}\phi_2^{i-1}$
4. Solve for $\phi_1(r)^i$ using $L_1\phi_1^i = \frac{1}{k^{i-1}}S_1^{i-1}$
5. Calculate S_2^{i-1} using $S_2^{i-1} = \Sigma_{s1\to 2}\phi_1^i$
6. Solve for $\phi_2(r)^i$ using $L_2\phi_2^{i-1} = S_2^{i-1}$
7. At this point we have new fluxes $\phi_1(r)^i$ and $\phi_2(r)^i$
8. To solve for k^i use $L_1\phi_1^i = \frac{1}{k^{i-1}}S_1^{i-1}$ and note that by integration we get:

$$k^i = \frac{\int_0^{r_N} S_1^i(r)dr}{\frac{1}{k^{i-1}}\int_0^{r_N} S_1^{i-1}(r)dr} \tag{9}$$

9. Using the values for iteration i the process can repeat from step 3 until some convergence criteria is met. in this implementation we use:

$$\Big|\frac{k^i - k^{i-1}}{k^i}\Big|100 < \epsilon \tag{10}$$

Where ϵ is a convergence criteria on k (for example 0.1 %)

2.2 Difference equation

Now that the iteration process is defined a method to solve for the spatial flux distribution is needed. This so called difference equation will solve the flux in equations 7 and 8. First write a generic 1-D diffusion equation:

$$-\nabla D_1\nabla\phi(r) + \Sigma_a(r)\phi(r) = S(r) \tag{11}$$

The leakage term in 1-D can be written as:

$$-\nabla D_1\nabla\phi(r) = -\frac{1}{r^\rho}\Big[\frac{d}{dr}\left(r^\rho D(r)\frac{d\phi(r)}{r}\right)\Big] \tag{12}$$

In this notation $\rho = 0$ is a slab , $\rho = 1$ resembles infinite cylinder (z-axis is missing), and $\rho = 2$ is for a sphere.
The problem is defined between $0 \le r \le a$ and the region is divided to N nodes ($N-1$ bins) such that the bin width is $\Delta r_k = r_{k+1} - r_k$ ($k = 0..N-1$). Note that the bin widths are not necessarily equal which provides some flexibility in the geometry description.
Equation 12 can thus be written:

$$\frac{d}{dr}\left(r^\rho D(r)\frac{d\phi(r)}{r}\right) = \left(\Sigma_a \phi(r) - S(r)\right) r^\rho \tag{13}$$

To get a difference equation we integrate from $r_{k-\frac{1}{2}}$ to $r_{k+\frac{1}{2}}$ In this case $k = 1, 2, ..., N-1$, we will deal with the boundary conditions of $k = 0$ and $k = N$ in the next section. In this notation we use $r_{k+\frac{1}{2}} = r_k + \frac{\Delta r_k}{2}$ and $r_{k-\frac{1}{2}} = r_k - \frac{\Delta r_{k-1}}{2}$

$$r^\rho D(r)\frac{d\phi(r)}{r}\Big|_{r_{k+\frac{1}{2}}} - r^\rho D(r)\frac{d\phi(r)}{r}\Big|_{r_{k-\frac{1}{2}}} = \int_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} \left(\Sigma_a \phi(r) - S(r)\right) r^\rho dr \tag{14}$$

The right term is approximated as constant in each bin:

$$\int_{r_{k-\frac{1}{2}}}^{r_{k+\frac{1}{2}}} \left(\Sigma_a \phi(r) - S(r)\right) r^\rho dr \approx \left(\Sigma_a(r_k)\phi(r_k) - S(r_k)\right) r_k^\rho \frac{\Delta r_k - \Delta r_{k-1}}{2} \tag{15}$$

for the derivatives in 14 we used the following approximations:

$$\frac{d\phi(r)}{dr}\Big|_{r_{k+\frac{1}{2}}} \approx \frac{\phi(r_{k+1}) - \phi(r_k)}{r_{k+1} - r_k} = \frac{\phi_{k+1} - \phi_k}{\Delta r_k} \tag{16}$$

$$\frac{d\phi(r)}{dr}\Big|_{r_{k-\frac{1}{2}}} \approx \frac{\phi(r_k) - \phi(r_{k-1})}{r_k - r_{k-1}} = \frac{\phi_k - \phi_{k-1}}{\Delta r_{k-1}} \tag{17}$$

The results can now be placed back into 14:

$$r^\rho_{k+\frac{1}{2}} D_{k+\frac{1}{2}} \frac{\phi_{k+1} - \phi_k}{\Delta r_k} - r^\rho_{k-\frac{1}{2}} D_{k-\frac{1}{2}} \frac{\phi_k - \phi_{k-1}}{\Delta r_{k-1}} = \left(\Sigma_{a_k}\phi_k - S_k\right) r_k^\rho \frac{\Delta r_k - \Delta r_{k-1}}{2} \tag{18}$$

This equation can be rearranged to:

$$a_k \phi_{k-1} + b_k \phi_k + c_k \phi_{k+1} = S_k \tag{19}$$

where:

$$a_k = \frac{r^\rho_{k-\frac{1}{2}} D_{k-\frac{1}{2}}}{\Delta r_{k-1}} \frac{2}{(\Delta r_k + \Delta r_{k-1}) r_k^\rho} \tag{20}$$

$$c_k = \frac{r^\rho_{k+\frac{1}{2}} D_{k+\frac{1}{2}}}{\Delta r_k} \frac{2}{(\Delta r_k + \Delta r_{k-1}) r_k^\rho} \tag{21}$$

$$b_k = a_k + c_k \Sigma_{a_k} \tag{22}$$

in this notation the diffusion coefficient as a function node number:

$$D_{k+\frac{1}{2}} = \frac{1}{2}(D_k + D_{k+1}) \tag{23}$$
$$D_{k-\frac{1}{2}} = \frac{1}{2}(D_k + D_{k-1}) \tag{24}$$

and similarly for r

$$r_{k+\frac{1}{2}} = \frac{1}{2}(r_k + r_{k+1}) \tag{25}$$
$$r_{k-\frac{1}{2}} = \frac{1}{2}(r_k + r_{k-1}) \tag{26}$$

2.3 Boundary conditions

We use two boundary conditions (BC) at the center of the core (implies symmetry) and at the dimension boundary. In this case the two boundary conditions are:

$$\phi(r_N) = 0 \tag{27}$$
$$\left.\frac{d\phi(r)}{dr}\right|_{r=0} = 0 \tag{28}$$

The first condition remove the last equation (N) and equation $N-1$ becomes:

$$a_{N-1}\phi_{N-2} + b_{N-1}\phi_{N-1} = S_{N-1} \tag{29}$$

To derive the derivative BC we repeat the integration in equation 14, but this time with the interval 0 to $r_{k+\frac{1}{2}}$. For the case of $\rho = 0$ ($r^\rho = 0$)we get:

$$\int_0^{r_{k+\frac{1}{2}}} (\Sigma_a\phi(r) - S(r))\, r^\rho \approx (\Sigma_{a_0}\phi_0 - S_0)\, \frac{\Delta r_0}{2} \tag{30}$$
$$r^\rho D(r)\left.\frac{d\phi(r)}{r}\right|_{r_{k+\frac{1}{2}}} - r^\rho D(r)\left.\frac{d\phi(r)}{dr}\right|_{r_0} = \left.\frac{d\phi(r)}{dr}\right|_{r_{k+\frac{1}{2}}} \approx D_0\frac{\phi_1 - \phi_0}{\Delta r_0} \tag{31}$$

Thus the diffusion equation reduces to:

$$D_0\frac{\phi_1 - \phi_0}{\Delta r_0} = (\Sigma_{a_0}\phi_0 - S_0)\, \frac{\Delta r_0}{2} \tag{32}$$

rearranging to the from $b_0\phi_0 + c_0\phi_1 = S_0$ results in:

$$b_0 = \frac{2D_0}{\Delta r_0^2} + \Sigma_{a_0} \tag{33}$$
$$c_0 = \frac{2D_0}{\Delta r_0^2} \tag{34}$$

for the case of $\rho > 0$ we can approximate $r = 0$ for this BC thus $r^\rho = 0$. which results in:

$$D_0\frac{\phi_1 - \phi_0}{\Delta r_0} = S_0 \tag{35}$$

and thus:

$$b_0 = 1 \tag{36}$$
$$c_0 = 1 \tag{37}$$

When placed in matrix notation we get:

$$\begin{bmatrix} b_0 & c_0 & 0 & .. & & & & \\ a_1 & b_1 & c_1 & 0 & .. & & & \\ 0 & a_2 & b_2 & c_2 & 0 & .. & & \\ \vdots & & & & & & & \\ \vdots & & & & & & & \\ & & & .. & 0 & a_{N-1} & b_{N-1} & c_{N-1} \end{bmatrix} \begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \vdots \\ \phi_{N-1} \end{bmatrix} = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ \vdots \\ \vdots \\ S_{N-1} \end{bmatrix} \tag{38}$$

In the implementation the a material matrix is created for the fast and thermal regions.

2.4 Matrix inversion

There are several ways to solve equation 38 to get the flux vector. In the implementation the Gaussian elimination method was selected because of its simplicity and efficiency. The derivation of the method can be found elsewhere.
The forward elimination:

$$h_0 = \frac{c_0}{b_0} \tag{39}$$
$$p_0 = \frac{S_0}{b_0} \tag{40}$$
$$h_k = \frac{c_k}{b_k - a_k h_{k-1}} \qquad k = 1, 2, ..., N-1 \tag{41}$$
$$p_k = \frac{S_k + a_k p_{k-1}}{b_k - a_k h_{k-1}} \qquad k = 1, 2, ..., N-1 \tag{42}$$

Notice that if not constructed correctly, the parameters in the matrix can cause division by zero. Finally back substitution give the solution for the neutron flux.

$$\phi_{N-1} = p_{N-1} \tag{43}$$
$$\phi_k = p_k + h_k \phi_{k+1} \qquad k = N-2, N-3, ..., 0 \tag{44}$$

2.5 Flux integration

In order to solve for the multiplication factor k, as described in equation 9, an integral of the flux is needed. For simplicity trapezoid integration can be used:

$$\int_0^{R_N} s_1(r)dr = \frac{1}{2} \sum_{i=0}^{N-1} (S_{1_i} + S_{1_{i+1}}) \Delta r_i \tag{45}$$