2D, One Group Neutron Transport Solver for a System with Vacuum Boundaries at Top and Left Faces and Reflective Boundaries at Bottom and Right Faces

Code: MATLAB R2018b

1. Introduction

The neutron transport equation is used to find the neutron angular flux at each given point in space. Neutron flux is fundamental in nuclear engineering as it determines neutron absorption and fission, aids in material design, etc. The transport equation is complex with a lot of unknown variables and requires approximation and analytical methods to solve the equation numerically. This project deals with developing a 2D one group neutron transport using deterministic approach where the space and direction vectors are discretized. The angles are discretized using discrete ordinate method and space is discretized using diamond difference method.

Furthermore, the geometry would be divided into small cells to use diamond difference method for approximations. In this project, the code is only trying to resolve simple geometries of square and rectangle only. Irregular and curved geometry needs very fine meshing and is computationally intensive and harder to code.

2. Derivations and Methods Used

The energy independent 3D Boltzmann neutron transport equation can be written as [1]:

$$\Omega \cdot \nabla\psi(r,\Omega) + \Sigma t(r)\psi(\vec{r}, \widehat{\Omega}) = q(\vec{r}, \widehat{\Omega}) \tag{2.1}$$

This equation is modified for 2-Dimensional transport equation in x, y coordinates as:

$$\left(\frac{d}{\partial x}\mu + \frac{d}{\partial y}\eta\right).\psi(x,y,\widehat{\Omega}) + \sigma(x,y).\psi(x,y,\widehat{\Omega}) = q(x,y,\widehat{\Omega}) \tag{2.2}$$

where $\sigma(x,y)$ is the macroscopic cross section and μ and η are the two angles of the direction vector $\widehat{\Omega}$. $q(x,y,\mu,\eta)$ is the scattering and independent source term defined as:

$$q(x,y,\mu,\eta) = \sum_{l=0}^{L}\sum_{m=0}^{l}(2-\delta_{m0})\, Y_{lm}^{e}(\widehat{\Omega})\sigma_l(x,y)\phi_l^m(x,y) + s(x,y,\widehat{\Omega}) \tag{2.3}$$

where $\sigma_l(x,y)$ is the l^{th} scattering moment and $\phi_l^m(x,y)$ is the l^{th} flux moment [2].

2.1 Discrete Ordinate method

The discrete ordinate's method is used to solve eq 2.1 and is given by:

$$\left(\frac{d}{\partial x}\mu_n + \frac{d}{\partial y}\eta_n\right).\psi(x,y,\widehat{\Omega}) + \sigma(x,y).\psi(\vec{r}, \widehat{\Omega}) = q(\vec{r},\widehat{\Omega}) \tag{2.4}$$

Where

$$q(\vec{r},\mu,\eta) = \sum_{l=0}^{L}\sum_{m=0}^{l}(2-\delta_{m0})\, Y_{lm}^{e}(\widehat{\Omega})\sigma_l(\vec{r})\phi_l^m(\vec{r}) + s(\vec{r},\widehat{\Omega}) \tag{2.5}$$

The discrete angles μ_n and η_n are selected using quadrature sets and spatial derivatives are discretized using either the finite difference or the finite volume method [2]. These selected angles which are relative to $\widehat{\Omega}$ are used to solve the neutron transport equation.

2.2 Level Symmetric Quadrature Set

For this project, the level symmetric quadrature set is used. It use the same set of N/2 positive values of direction cosines with respect to each of the two axes, i.e., for each level n $\mu_n = \eta_n$ is set. We describe a level *a* as the ordinate set that has cosine μ_n with respect to the x-axis. There are 4 octants for 2D geometries and have up and down symmetry. So, there are $N(N+2)/2$ levels for 2D geometries. The weights of the quadrature set are normalized as:

$$\sum_{n=1}^{\frac{N(N+2)}{2}} w_n = 1 \tag{2.6}$$

Hence, the flux moments can be approximated as:

$$\phi_l^m(\vec{r}) = \frac{1}{4} \sum_{n=1}^{\frac{N(N+2)}{2}} w_n Y_{lm}^e(\widehat{\Omega_n}) \psi(\vec{r}, \widehat{\Omega}_n) \tag{2.7}$$

2.3 Spatial Discretization

The diamond difference method is used to create a cell centered mesh where the values of the variables only change at the cell-edge boundaries and cross sections values remain constant. The spatial balance equation is thus obtained as:

$$\frac{\mu_n}{\triangle x_i}\left(\psi_{n,i+\frac{1}{2},j} - \psi_{n,i-\frac{1}{2},j}\right) + \frac{\eta_n}{\triangle y_j}\left(\psi_{n,i,j+\frac{1}{2}} - \psi_{n,i,j-\frac{1}{2}}\right) + \sigma_{ij}\psi_{nij} = q_{nij} \tag{2.8}$$

The outgoing fluxes (flux at the edge of each cell) are:

$$\psi_{n,i\mp\frac{1}{2},j} = 2\psi_{nij} - \psi_{n,i\mp\frac{1}{2},j} \tag{2.9}$$

$$\psi_{n,i,j\mp\frac{1}{2}} = 2\psi_{nij} - \psi_{n,i,j\mp\frac{1}{2}} \tag{2.10}$$

3. Algorithm

Since, cell centered mesh is used for spatial discretization, the angular flux is solved at each grid in the direction of the neutron travel. For the angles μ_n and η_n, there are four cases in two dimensions. We get the final flux equation by substituting equation 2.10 and 2.9 in equation 2.8 and rearranging to make the flux as subject. The four sweep cases with their respective flux equation is as follows:

I. $\boldsymbol{\mu_n > 0}$ & $\boldsymbol{\eta_n > 0}$ **left to right; top to bottom**

$$\psi_{nij} = \left(\frac{2\mu_n}{\Delta x_i} + \frac{2\eta_n}{\Delta y_j} + \sigma_{ij}\right)^{-1} \left[\frac{2\mu_n}{\Delta x_i} \cdot \psi_{n,i-\frac{1}{2},j} + \frac{2\eta_n}{\Delta y_j} \cdot \psi_{n,i,j-\frac{1}{2}} + q_{nij}\right] \tag{2.11}$$

II. $\boldsymbol{\mu_n < 0}$ & $\boldsymbol{\eta_n > 0}$ **right to left; top to bottom**

$$\psi_{nij} = \left(-\frac{2\mu_n}{\Delta x_i} - \frac{2\eta_n}{\Delta y_j} + \sigma_{ij}\right)^{-1} \left[-\frac{2\mu_n}{\Delta x_i} \cdot \psi_{n,i+\frac{1}{2},j} + \frac{2\eta_n}{\Delta y_j} \cdot \psi_{n,i,j-\frac{1}{2}} + q_{nij}\right] \tag{2.12}$$

III. $\boldsymbol{\mu_n > 0}$ & $\boldsymbol{\eta_n < 0}$ **left to right; bottom to top**

$$\psi_{nij} = \left(\frac{2\mu_n}{\Delta x_i} - \frac{2\eta_n}{\Delta y_j} + \sigma_{ij}\right)^{-1} \left[\frac{2\mu_n}{\Delta x_i} \cdot \psi_{n,i-\frac{1}{2},j} - \frac{2\eta_n}{\Delta y_j} \cdot \psi_{n,i,j+\frac{1}{2}} + q_{nij}\right] \tag{2.13}$$

IV. $\boldsymbol{\mu_n < 0}$ & $\boldsymbol{\eta_n < 0}$ **right to left; bottom to top**

$$\psi_{nij} = \left(-\frac{2\mu_n}{\Delta x_i} - \frac{2\eta_n}{\Delta y_j} + \sigma_{ij}\right)^{-1} \left[-\frac{2\mu_n}{\Delta x_i} \cdot \psi_{n,i+\frac{1}{2},j} - \frac{2\eta_n}{\Delta y_j} \cdot \psi_{n,i,j+\frac{1}{2}} + q_{nij}\right] \tag{2.14}$$

Substituting the cell centered flux into the angular quadrature equations (2.6 and 2.7), the scalar flux and Legendre moment at each grid point is given by:

$$\phi_{ij} = \frac{1}{4} \sum_{n=1}^{\frac{N(N+2)}{2}} w_n \psi_{nij} \tag{2.15}$$

$$\phi_{lmij} = \frac{1}{4} \sum_{n=1}^{\frac{N(N+2)}{2}} w_n Y_{lm}^{e}(\widehat{\Omega}) \psi_{nij} \tag{2.16}$$

4. Code Use

4.1 Understanding Geometry

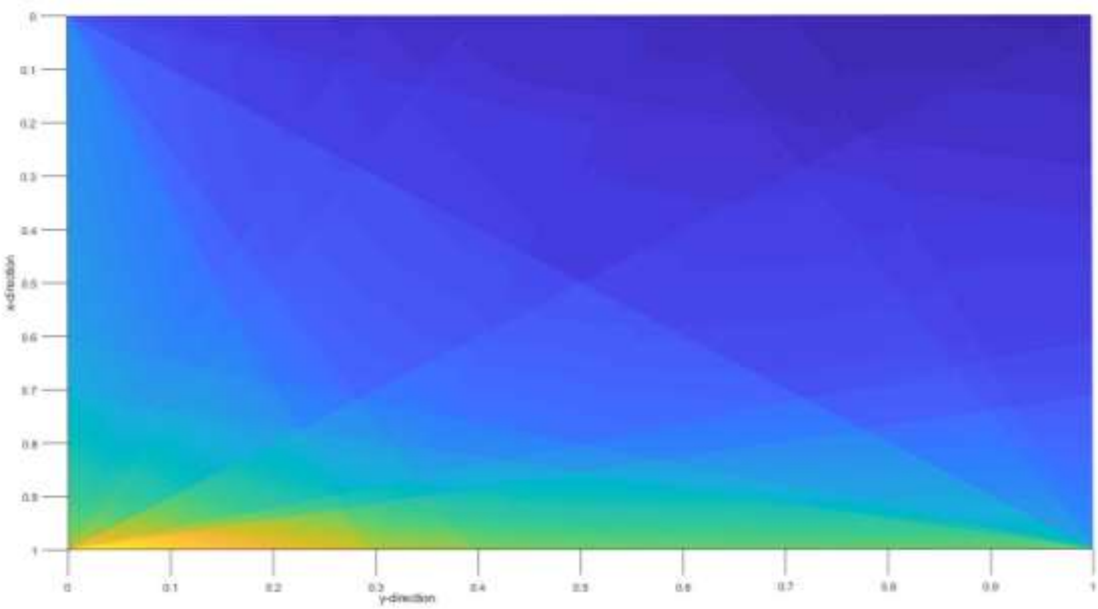

Figure 1: 2D geometry

Mesh is created with use of a matrix in MATLAB. In a matrix x is the row number and y is the column number. Hence for the result plot as shown above, x axis from 0 to 1 represents the vertical faces of the geometry (left and right) and vice versa for y axis from 0 to 1.

4.2 Input

Required inputs are at the very beginning of the code and is commented appropriately. The following are the inputs expected from the user:

- Vertical and horizontal dimension of the geometry (square or rectangle)
- Grid sizing for sweep
- Cross sections of the material
- Source
- Number of iterations to run
- Tolerance level for residual convergence

The quadrature set used is SN6 and can be changed if need be. Values must be entered manually for μ_n, η_n and w_n.

5. Test Input and results

For the test a unit geometry, 1x1, was used which is then split into 1000x1000 grids. The total cross section used was 1/cm and scattering cross section was 0.5/cm. Source is set as 10 coming from top and left faces (vacuum faces), illustrated in figure 2. Finally, 10 iterations are used to obtain results quickly rather than waiting for convergence.

More tests were run from different faces for verification of results and to check the application of boundary conditions.

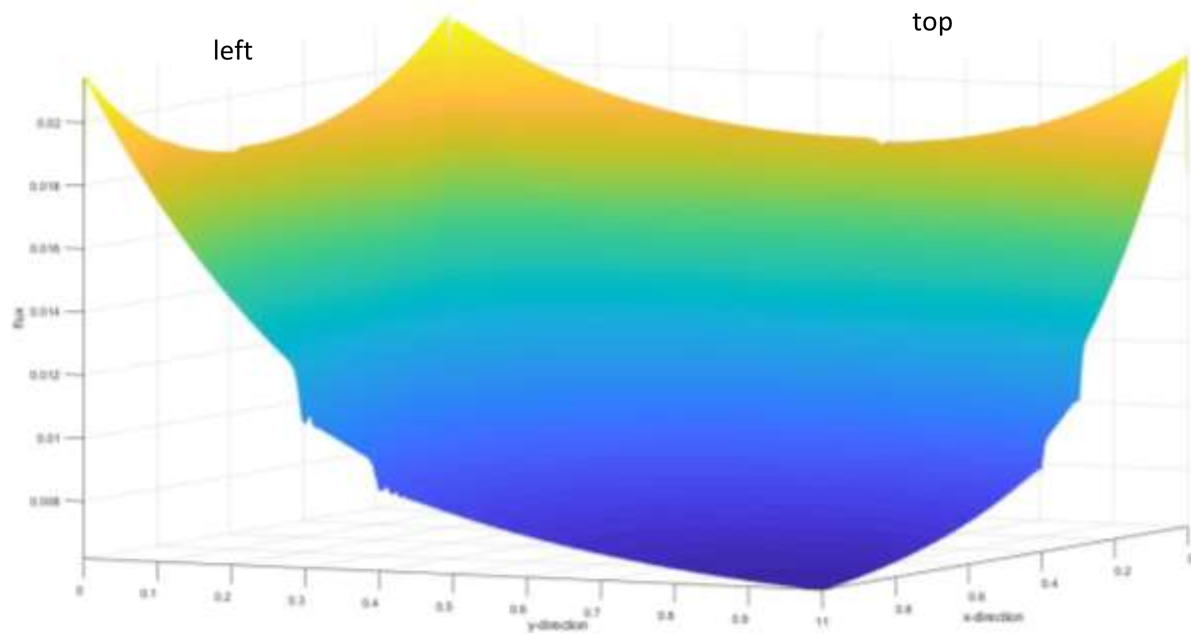

Figure 2: angular flux for neutron source from top and left faces

Results from this test verify two fundamental measure of success of the code, i.e., there is no negative angular flux and the angular flux decrease from the source faces towards the opposite faces. Also, note the maximum angular flux for vacuum boundary source was around 0.022/cm2.

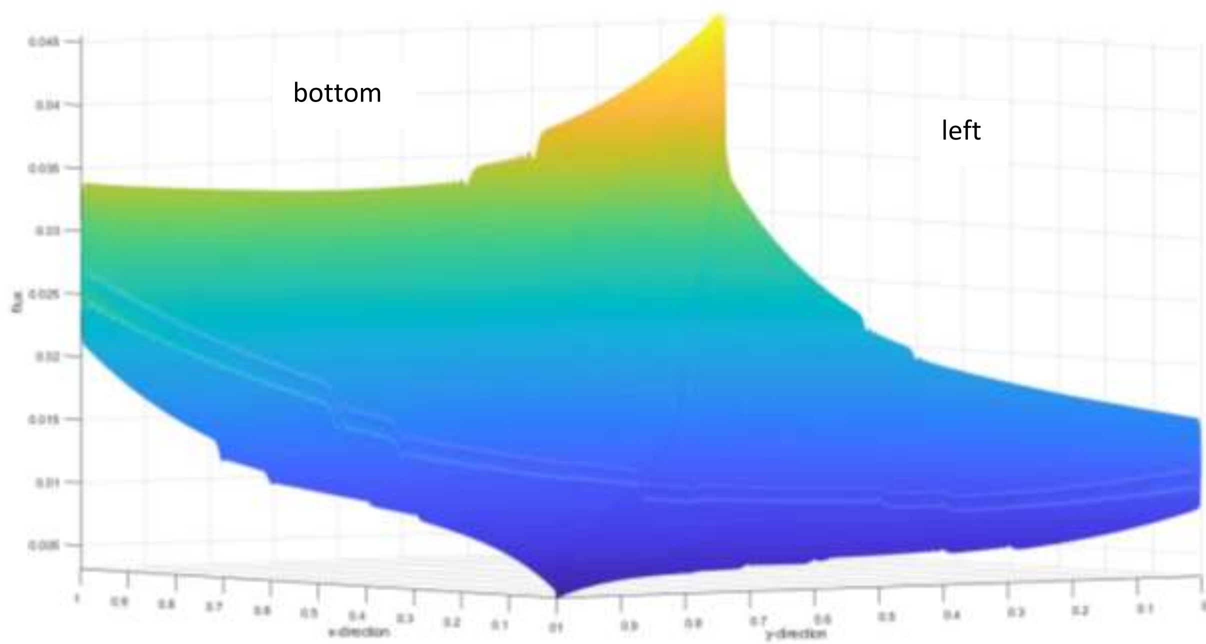

Figure 3: angular flux for neutron source from left and bottom faces

From figure 3, it was observed that the flux along the bottom reflective face is higher than the left vacuum face, confirming proper application of relective boundary condition along the bottom face. The maximum anglular flux at around 0.045/cm2 is alos higher than the previous test.

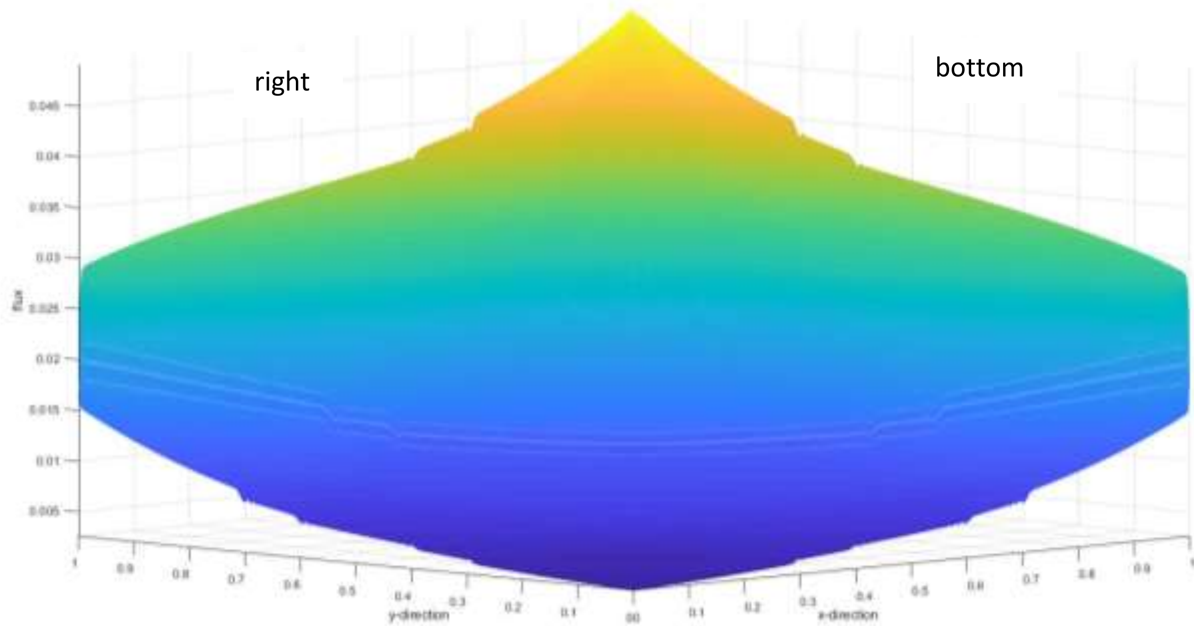

Figure 4: angular flux for neutron source from bottom and right faces

Finally, the symmetric pattern of angular flux from figure 4 confirms the proper application of boundary condition for right face as well.

6. Conclusion

The code was able to solve the intended transport equation with the correct implementation of boundary conditions. The code has been verified solely on expected pattern of angular flux for the presented conditions. However, the accuracy of values and convergence intervals has to be verified using an existing 2D neutron transport equation solvers. Furthermore, the optimum implementation of this project would have been to design an executable program in C++ for easier input collection from the user, avoid accidental edits to the code and prevent accessibility restrictions to any software.

7. References

[1] Chen, Q., Wu, H., & Cao, L. (2008). Auto MOC—A 2D neutron transport code for arbitrary geometry based on the method of characteristics and customization of AutoCAD. *Nuclear Engineering and Design*, *238*(10), 2828-2833.

[2] Ortega, M.I., *Two-Dimensional Neutron Transport Equation Solver with Diffusion-Synthetic and Transport-Synthetic Acceleration*, UC Berkeley

Indirect Sources

- Lecture notes
- M.G. Jarrett, *"A 2D/1D Neutron Transport Method with Improved Angular Coupling",* www.deepblue.lib.umich.edu/bitstream/handle/2027.42/147498/jarremic_1.pdf?sequence=1&isAllowed=y