# Two-Dimensional Neutron Transport Equation Solver with Diffusion-Synthetic and Transport-Synthetic Acceleration

## Information

Author: Mario I. Ortega
Organization: Department of Nuclear Engineering, University of California, Berkeley

## 1 Introduction

The neutron transport equation is an integro-differential equation for neutron angular flux. In the design of nuclear systems, neutron angular flux determines fission and absorption reaction rates, radiation damage to materials, and other quantities of interest to designers. The neutron transport equation is a function of space, angle, energy, and time, and analytical solutions exist only for the simplest of cases [1]. Therefore, it is usually necessary to solve the neutron transport equation numerically. One method to solve the neutron transport equation is the discrete ordinates method. In this report, we detail a two-dimensional discrete ordinates code on a Cartesian mesh with various acceleration (preconditioning) methods. The mathematics and algorithms of the code are explained and references provided. In this interim report, the plan for code completion is detailed along with proposed verification and validation studies.

## 2 Mathematics

The two-dimensional neutron transport equation in x-y coordinates is given by

$$\left[\mu\frac{\partial}{\partial x}+\eta\frac{\partial}{\partial y}+\sigma(x,y)\right]\psi(x,y,\hat{\mathbf{\Omega}})=q(\vec{r},\hat{\mathbf{\Omega}}), \tag{2.1}$$

where $\sigma(x,y)$ is the total macroscopic cross section and $\mu$ and $\eta$ are the angles necessary to specify the direction vector $\hat{\mathbf{\Omega}}$. $q(\vec{r},\mu,\eta)$ is the scattering and independent source term defined as:

$$q(x,y,\mu,\eta)=\sum_{l=0}^{L}\sum_{m=0}^{l}(2-\delta_{m0})Y_{lm}^{e}(\hat{\mathbf{\Omega}})\sigma_l(x,y)\phi_l^m(x,y)+s(x,y,\hat{\mathbf{\Omega}}), \tag{2.2}$$

where $\sigma_l(x,y)$ is the $l^{th}$ scattering moment and $\phi_l^m(x,y)$ is the $l^{th}$ flux moment. It is also necessary to specify boundary conditions. Usually these boundary conditions are vacuum, imposed angular flux, or reflective.

To numerically solve Equation 2.1, the discrete ordinates method is used [2].

### 2.1 The Discrete Ordinates Method

The two-dimensional neutron transport discrete ordinates equation in x-y geometry is given by:

$$\left[\mu_n\frac{\partial}{\partial x}+\eta_n\frac{\partial}{\partial y}+\sigma(x,y)\right]\psi(\vec{r},\hat{\mathbf{\Omega}})=q(\vec{r},\hat{\mathbf{\Omega}}). \tag{2.3}$$

where

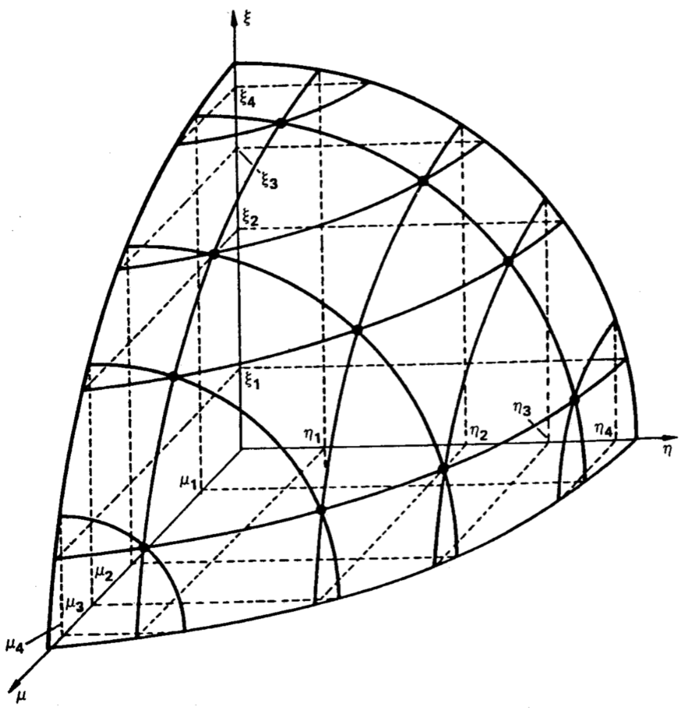
$$q(\vec{r},\mu_n,\eta_n)=\sum_{l=0}^{L}\sum_{m=0}^{l}(2-\delta_{m0})Y_{lm}^{e}(\hat{\mathbf{\Omega}})\sigma_l(\vec{r})\phi_l^m(\vec{r})+s(\vec{r},\hat{\mathbf{\Omega}}). \tag{2.4}$$

In the discrete ordinates method, discrete angles $\mu_n$ and $\eta_n$ are selected using some sort of quadrature and spatial derivatives are discretized using either the finite difference or the finite volume method. The discretized neutron transport equation is then solved along these selected angles. Using weights specified by the selected quadrature set, the scalar flux is constructed. In the case of no scattering, it can be shown that the angular flux converges in one iteration [3]. If there is scattering present, the angular flux is iterated on until it converges to some predetermined tolerance. In the next two sections we detail the angular quadrature selected and the discretization of the spatial derivatives.

### 2.1.1 Two-Dimensional Discrete Ordinates Quadrature Sets

In two dimensions, two angular coordinates are necessary to specify the direction $\hat{\mathbf{\Omega}}$ of neutron travel. Usually, these angles are defined relative to an orthogonal coordinate system, in our case, the x-y plane. We define $\mu$ and $\eta$ as the direction cosines of $\hat{\mathbf{\Omega}}$ relative to this coordinate system. In this code, a level symmetric quadrature set is used. The ordinates are arranged on a unit octant of the unit sphere and ordinates utilizes the same set of N/2 positive values for each of the two angles as shown in Figure 2.1.

**Figure 2.1:** Level Symmetric Quadrature in Octant. From [3].




There are N(N+2)/2 directions per octant of the unit sphere in the two-dimensional case. The weights of the quadrature points are normalized such that they sum to one in each of the octants:

$$\sum_{n=1}^{N(N+2)/2} w_n = 1. \tag{2.5}$$

Using the quadrature set, the flux moments can be approximated by

$$\phi_l^m(\vec{r}) = \frac{1}{4} \sum_{n=1}^{(N+2)/2} w_n Y_{lm}^e(\hat{\mathbf{\Omega}}_n) \psi(\vec{r}, \hat{\mathbf{\Omega}}_n). \tag{2.6}$$

### 2.1.2 Spatial Discretization of the Neutron Transport Equation

The x-y domain is divided into cells bounded by $x_{1/2}, x_{3/2}, ..., x_{I/2}$ and $y_{1/2}, y_{3/2}, ..., y_{I/2}$. Cross sections are taken to be be piecewise constant within cells and their value can only change at the half-integer (cell-edge) boundaries. Difference relations are derived by integrating cells between the half-integer boundaries. Averaged fluxes and source terms are defined over cell edges and cell areas:

$$\psi_{n,i+1/2,j} = \frac{1}{\Delta y_j} \int_j dy\ \psi_n(x_{i+1/2}, y), \tag{2.7}$$

$$\psi_{n,i,j+1/2} = \frac{1}{\Delta x_i} \int_i dx\ \psi_n(x, y_{i+1/2}), \tag{2.8}$$

$$\psi_{nij} = \frac{1}{\Delta x_i \Delta y_j} \int_i dx \int_j dy\ \psi_n(x, y), \tag{2.9}$$

and

$$q_{nij} = \frac{1}{\Delta x_i \Delta y_j} \int_i dx \int_j dy\ q_n(x, y). \tag{2.10}$$

Using these expressions, the spatial balance equation is obtained

$$\frac{\mu_n}{\Delta x_i}(\psi_{n,i+1/2,j} - \psi_{n,i-1/2,j}) + \frac{\eta_n}{\Delta y_j}(\psi_{n,i,j+1/2} - \psi_{n,i,j-1/2}) + \sigma_{ij}\psi_{nij} = q_{nij}. \tag{2.11}$$

Two auxiliary expressions (diamond difference approximation) relate cell average fluxes to cell edge fluxes:

$$\psi_{nij} = \frac{1}{2}(\psi_{n,i+1/2,j} + \psi_{n,i-1/2,j}), \tag{2.12}$$

$$\psi_{nij} = \frac{1}{2}(\psi_{n,i,j+1/2} + \psi_{n,i,j-1/2}). \tag{2.13}$$

The solution algorithm is detailed in the next section.

# 3 Algorithms

To solve for the cell-centered angular flux, we sweep through the grid in the direction of neutron travel. There are four cases in two dimensions [3]:

$\mu_n > 0,\ \eta_n > 0$ left to right; bottom to top

$\mu_n < 0,\ \eta_n > 0$ right to left; bottom to top

$\mu_n > 0,\ \eta_n < 0$ left to right; top to bottom

$\mu_n < 0,\ \eta_n < 0$ right to left; top to bottom.

The diamond difference relationships are used to relate cell-edge fluxes to cell-centered fluxes and are used to express the unknown fluxes in terms of known cell-edge fluxes imposed by boundary conditions. For each case, a corresponding relationship for the cell-centered flux is derived:

$$\psi_{nij} = \left(\frac{2\mu_n}{\Delta x_i} + \frac{2\eta_n}{\Delta y_j} + \sigma_{ij}\right)^{-1} \left[\frac{2\mu_n}{\Delta x_i}\psi_{n,i-1/2,j} + \frac{2\eta_n}{\Delta y_j}\psi_{n,i,j-1/2} + q_{nij}\right] \quad \mu_n > 0, \eta_n > 0 \tag{3.1}$$

$$\psi_{nij} = \left(-\frac{2\mu_n}{\Delta x_i} - \frac{2\eta_n}{\Delta y_j} + \sigma_{ij}\right)^{-1} \left[-\frac{2\mu_n}{\Delta x_i}\psi_{n,i+1/2,j} + \frac{2\eta_n}{\Delta y_j}\psi_{n,i,j-1/2} + q_{nij}\right] \quad \mu_n < 0, \eta_n > 0 \tag{3.2}$$

$$\psi_{nij} = \left(\frac{2\mu_n}{\Delta x_i} - \frac{2\eta_n}{\Delta y_j} + \sigma_{ij}\right)^{-1} \left[\frac{2\mu_n}{\Delta x_i}\psi_{n,i-1/2,j} - \frac{2\eta_n}{\Delta y_j}\psi_{n,i,j+1/2} + q_{nij}\right] \quad \mu_n > 0, \eta_n < 0 \tag{3.3}$$

$$\psi_{nij} = \left( -\frac{2\mu_n}{\Delta x_i} - \frac{2\eta_n}{\Delta y_j} + \sigma_{ij} \right)^{-1} \left[ -\frac{2\mu_n}{\Delta x_i}\psi_{n,i+1/2,j} - \frac{2\eta_n}{\Delta y_j}\psi_{n,i,j+1/2} + q_{nij} \right] \quad \mu_n < 0, \eta_n < 0 \tag{3.4}$$

Using these relationships, we obtain the angular flux as a function of the spatial grid points and angular direction pairs. Using the angular quadrature, we can determine the scalar flux and Legendre moments at each grid point using

$$\phi_{ij} = \frac{1}{4}\sum_{n=1}^{N(N+2)/2} w_n \psi_{nij} \tag{3.5}$$

and

$$\phi_{lmij} = \frac{1}{4}\sum_{n=1}^{N(N+2)/2} w_n Y^e_{lm}(\hat{\mathbf{\Omega}})\psi_{nij}. \tag{3.6}$$

The scalar flux is iterated upon until the norm is less than some predetermined tolerance. A more detailed algorithm chart will be included in the final report. Starting fluxes on cell-edges are determined by boundary conditions. Vacuum and reflective conditions will be implemented. Diffusion Synthetic- and Transport Synthetic Acceleration schemes will be implemented to accelerate the calculation of angular flux.

# 4 Verification and Validation

The code will be verified and validated using the method of manufactured solutions [4]. Solutions for the 2D NTE discrete ordinates equation will be assumed and the source term necessary to make such angular flux solution true determined. The code output angular fluxes will be compared to the analytical expressions to validate the code. One simple check on the code has already been conducted. For the case where there is no scattering or source, the angular flux solution is zero for vacuum boundary conditions. Using these parameters and boundary conditions, the code returns the zero solution as expected.

# 5 Plans for Completion

The current timeline for the project is shown below. Completed tasks are marked as such.

- Week 1 (3/28/2016 - 4/1/2016)
  - Initialization of Github repo for version control of code (Due date: 4/1/2016). **Completed**
    - Github repo link: `https://github.com/marort91/2DNeutronTransportDiscreteOrdinatesCode`
  - Begin work on MATLAB one-dimensional discrete ordinates code for testing and pedagogical purposes. **Completed**
  - Document writeup on 1D discrete ordinates method with finite differencing. **Completed**
  - **Deliverable:** Project abstract (Due date: 4/1/2016). **Completed**
- Week 2 (4/4/2016 - 4/8/2016)
  - 1D discrete ordinates writeup completed (Due date: 4/8/2016). **Completed**
  - 1D MATLAB discrete ordinate code complete (Due date: 4/8/2016). **Completed**
  - Begin work and research on two-dimensional discrete ordinates quadrature. **Completed**
- Week 3 (4/11/2016 - 4/15/2016)
  - Begin writeup on 2D discrete ordinates solution method (detail algorithm for final report). **Completed**
  - Angular and spatial discretization complete (Due date: 4/15/2016). **Completed**
  - **Deliverable:** Interim report (Due date: 4/15/2016). **Completed**
- Week 3 (4/18/2016 - 4/22/2016)

  - Implement transport sweep algorithm (Due date: 4/22/2016). **Completed**
  - Test different boundary conditions (vacuum, reflective, white(?), periodic(?))
  - Begin work on DSA (Due date: 4/29/2016).
  - Begin work on third acceleration method (Due date: 4/29/2016).

- Week 4 (4/25/2016 - 4/29/2016)
  - Complete work on acceleration methods
  - Compare effects of preconditioning on solution of neutron transport equation. Begin work on MATH221 poster presentation.

- Week 5 (5/2/2016 - 5/6/2016)
  - Begin work on final report and presentation (Due date: 5/10/2016).
  - Time permitting: implement criticality eigenvalue calculation for multiplying system.
  - Time permitting: extension to multigroup neutron transport equation solver.

- Week 6 (5/9/2016 - 5/10/2016)
  - **Deliverable:** Final report (Due date: 5/10/2016).
  - **Deliverable:** Final presentation (Due date: 5/10/2016).

# 6 References

## References

[1] G. I. Bell and S. Glasstone. Nuclear Reactor Theory. Division of Technical Information, US Atomic Energy Commission, 1970.

[2] B. G. Carlson, K. D. Lathrop, et al. Transport Theory: The Method of Discrete Ordinates. Los Alamos Scientific Laboratory, 1965.

[3] E. E. Lewis and W. F. Miller. Computational Methods of Neutron Transport. John Wiley and Sons, Inc., New York, NY, 1984.

[4] K. Salari and P. Knupp. Code Verification by the Method of Manufactured Solutions. Technical report, Sandia National Laboratories, Albuquerque, NM (US); Sandia National Laboratories, Livermore, CA (US), 2000.