Development of a One Dimensional Sn Method Neutron Transport Solver

Khaled Talaat

NE 610 Project Report

Abstract

The present report describes the development of a one dimensional, one group neutron transport solver. The code implements the discrete ordinates method for angular discretization. The weighted diamond difference closure is implemented to relate the angular flux at cell centers to that at the faces in spatial discretization. The code supports multiple regions with volume sources and fission sources. Both albedo and incident source boundary conditions are supported. The Reed cell benchmark and the two region criticality benchmark are used to assess the results. Excellent agreement is observed between the results obtained from the code and the benchmarks using diamond differencing and step differencing. Using step differencing, a 0.06% difference is observed in k compared to the criticality benchmark, while that of diamond differencing is within 0.003%.

Introduction

Accurate prediction of the neutron flux distribution in a medium is essential for reactor applications, shielding, and medical physics. The neutron transport equation governs the transport of neutrons in media under the assumption that the neutron density is large enough for the neutron flux to be described as a continuum but small enough for neutron-neutron collisions to be negligible. Under the transport equation, neutrons are treated as non-decaying, point classical particles with collisions defined by 2 body localized events. Neutrons are assumed to have no memory of past collisions. Additionally, in reactor applications, short-timescale neutron effects on the medium are ignored.

The neutron transport equation exists in two forms: integro-differential form and integral form. Analytical solutions of the neutron transport equation are limited to simple cases with very simple geometries. Practical cases of interest necessitate the use of numerical methods which rely on discretization in space, angle, and energy. For the results to be consistent and comparable to other work, independence of the discretization must be established within an acceptable tolerance. However, grid independence and convergence don’t guarantee accuracy of the results.

Accuracy of the results depends on the numerical schemes used in the calculations and the order of error associated with such schemes.

In addition to spatial discretization typical in most applications, the neutron transport equation requires careful treatment of the angle. Various methods have been developed such as the collision probability method, discrete ordinates method (Sn), spherical harmonics method (Pn), and the method of characteristics (MOC). The collision probability method solves the integral form of the transport equation and is useful when the scattering can be approximated as isotropic. It has the ability to handle complex geometry. It, however, requires the inversion of a large, dense matrix which can be computationally prohibiitive for large systems [1]. The discrete ordinates method, which is discussed in more technical detail later, is often used for the integro-differential form of the transport equation and can easily handle anisotropic scattering. It is, however, difficult to apply to irregular geometries [1]. The method of characteristics is similar to the Sn method but can handle complicated geometry through ray tracing to determine the intercepts [1]. Low-order Sn method also suffers ray effects due to the inability of low-order Sn quadrature to integrate accurately over the angular flux. The Pn method was developed to remedy this problem [2] . It expands the angular flux in terms of orthogonal polynomials.such as legendre polynomials and spherical harmonics as shown in Equations 1 and 2. S_n method based codes can easily be transformed into P_{n-1} codes with no fundamental changes to the main iteration or grid sweeping [2].

$$\psi_n(z) = \int_{-1}^{1} du \; P_n(\mu)\psi(z,\mu) \tag{1}$$

$$\frac{n+1}{2n+1}\frac{\partial \psi_{n+1}(z)}{\partial z} + \frac{n}{2n+1}\frac{\partial \psi_{n-1}(z)}{\partial z} + \left(\Sigma_t - \Sigma_{s,n}\right)\psi_n(z) = \nu\Sigma_f \psi_o(z)\delta_{n,0} + S_n(z) \tag{2}$$

The objective of the present project is to develop and benchmark a one dimensional, one group Sn method neutron transport solver. The solver should allow for multiple regions to be defined in a slab geometry with arbitrary mesh refinement in each region. Further, the solver should support volumetric sources, fission sources, and albedo and incident source boundary conditions.

Methodology

The neutron transport equation governs the transport of neutrons in the system. A particularly useful quantity for reactor design and shielding problems is the angular flux. The angular flux, $\psi(r, E, \Omega, t)$, is the product of the neutron density at a particular point in the phase space and the velocity of the neutrons. Unlike the scalar flux, it preserves information on the direction that the neutrons are traveling. The angular flux can be obtained by solving the neutron transport equation (Equation 3).

$$\frac{1}{v}\frac{\partial\psi(\mathrm{r,E,\Omega,t})}{\partial \mathrm{t}} + \Omega.\nabla\psi(r, E, \Omega, t) + \Sigma_t\psi(r, E, \Omega, t) = \int_0^\infty dE' \int_{4\pi} d\Omega' \, \Sigma_s(E' \to E;\ \Omega'.\Omega)\psi(r, E', \Omega', t) + \frac{x(E)}{4\pi}\int_0^\infty dE' \int_{4\pi} d\Omega' \, \nu(E')\Sigma_f(E')\psi(r, E', \Omega, t) + S(r, E, \Omega, t) \quad (3)$$

The direction dependent scattering kernel can be expanded in terms of Legendre polynomials and a direction independent scattering kernel (Equation 4). For most practical applications, a finite L can accurately characerize the original scattering kernel.

$$\Sigma_s(E' \to E;\ \Omega'.\Omega) = \Sigma_s(E' \to E; \mu_0) = \sum_{l=0}^{L=\infty} \frac{2l+1}{4\pi}\Sigma_{\mathrm{s,l}}P_l(\mu_0) \quad (4)$$

From the orthogonality of Legendre polynomials, we get that:

$$\Sigma_{s,l}(E' \to E;\ \mu_0) = \frac{1}{2}\Sigma_s(E' \to E)\int_{-1}^{1} \mathrm{d}\mu_0 \, P_l(\mu_0) \quad (5)$$

The relationship in Equation 5 can be used to obtain the angle independent scattering kernel. Using Equation 4 and reducing Equation 3 to a 1D slab case we get Equation 6.

$$\mu\frac{\partial\psi(\mathrm{z},\mu)}{\partial \mathrm{z}} + \Sigma_t(z)\psi(\mathrm{z},\mu) = \sum_{l=0}^{\infty}\frac{2l+1}{2}\Sigma_{\mathrm{s,l}}P_l(\mu)\int_{-1}^{1} du' p_l(\mu')\psi(\mathrm{z},\mu') + \frac{\nu\Sigma_f}{2k}\int_{-1}^{1} du'\psi(\mathrm{z},\mu') + \frac{Q(z)}{2} \quad (6)$$

The boundary conditions can be written in general (combined) form as in Equations 7 and 8. The combined form can be reduced to incident source or reflection forms by adjusting the parameters.

$$\psi(0,\mu) = \Gamma_{\mathrm{L}}(\mu) + \gamma\ \psi(0,-\mu) where\ 0 < \ \mu \le 1 \text{ and } 0 \le \ \gamma \le 1 \quad (7)$$

$$\psi(\mathrm{L},\mu) = \Gamma_{\mathrm{R}}(\mu) + \gamma\,\psi(\mathrm{L},-\mu) where -1 \leq \mu < 0 \text{ and } 0 \leq \gamma \leq 1 \tag{8}$$

Up to this point, the discussion has been restricted to transport theory. The general transport equation in Equation 3 was reduced to the one dimensional form with Legendre expansion of the scattering kernel in Equation 6. Now we need to reformulate Equation 6 into a form that can be implemented numerically. To achieve this, we need to discretize space, angle, and numerically evaluate the integrals using numerical integration quadrature. The continuous integral representing the scalar flux in Equation 6 could be written and evaluated as a discrete integral using Equation 9. By careful choice of the angles and weights, the scalar flux can be evaluated exactly to a polynomial fit order of 2N-1 using Gaussian quadrature. The integration points, x_i, are the roots of the Legendre polynomials. The weights of the Gauss-Legendre quadrature can be calculated from these roots as in Equation 10. The angular discretization is, therefore, not arbitrary but dictated by the integration points, $x_{i.}$

$$\phi(z) = \int_{-1}^{1} du'\,\psi(\mathrm{z},\mu') = \sum_{j=1}^{N} w_j \psi_j(z) \tag{9}$$

$$w_i = \frac{2}{(1-x_i^2)[P_n'(x_i)]^2} \tag{10}$$

Finally, the discretization in angle and the integration quadrature could be introduced into Equation 6 to obtain the discrete ordinate (Sn) transport equation for a one dimensional slab (Equation 11). This is the equation that is solved in the present project.

$$\mu_n \frac{\partial \psi_n}{\partial \mathrm{z}} + \Sigma_t(z)\psi_n(z) = \sum_{l=0}^{L} \frac{2l+1}{2} \Sigma_{\mathrm{s,l}}(z) p_l(\mu_n) \sum_{j=1}^{n} w_j p_l(\mu_j)\psi_j(z) + \frac{\nu\Sigma_f(z)}{2k} \sum_{j=1}^{N} w_j \psi_j(z) + \frac{S_n(z)}{2} \tag{11}$$

The outer iteration scheme implemented in the current project is described in Figure 1. Initially, the scalar flux is assumed to be uniform throughout the slab and of unity magnitude. The fission term is calculated and fed into the inner source iteration which is described in detail in Figure 2.

The k-eigenvalue is then evaluated from the ratio of the volume integral of the fission term in the current iteration, $\nu\Sigma_f(z)\phi(z)$, to that in the previous iteration. The scalar flux is then updated and a mass balance check is done. The root mean square difference in the scalar flux in the present iteration and that of the previous iteration is calculated to obtain the L2 error. Two tolerances are specified by the user: an inner convergence tolerance for the inner source iteration and an outer convergence tolerance for the outer iteration.

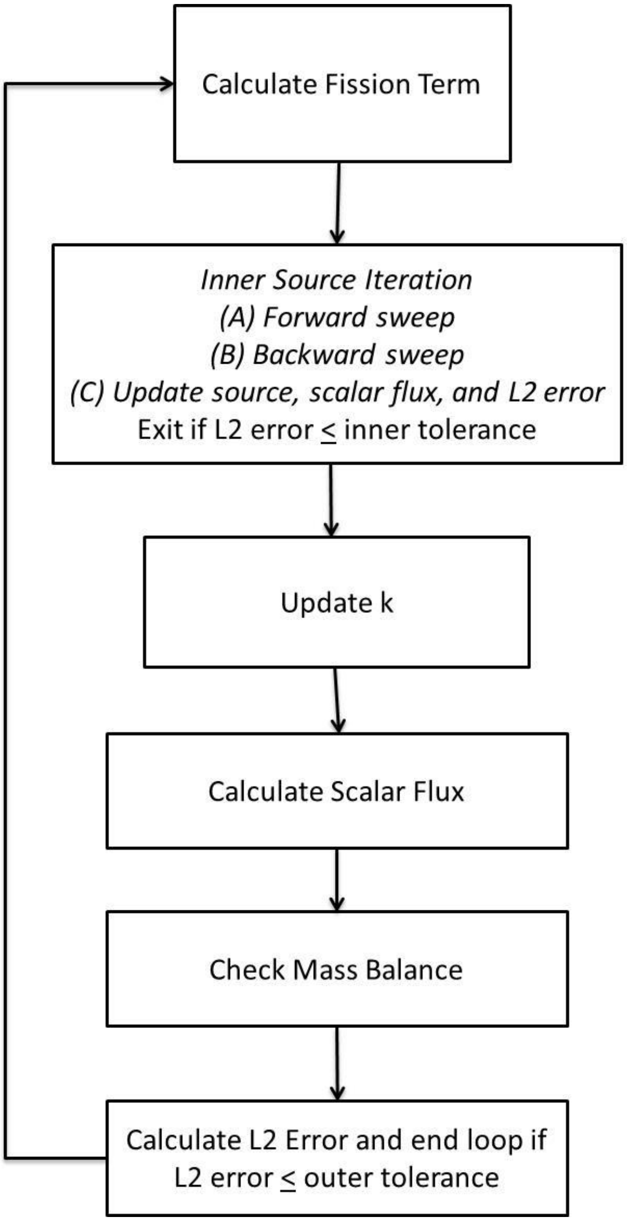
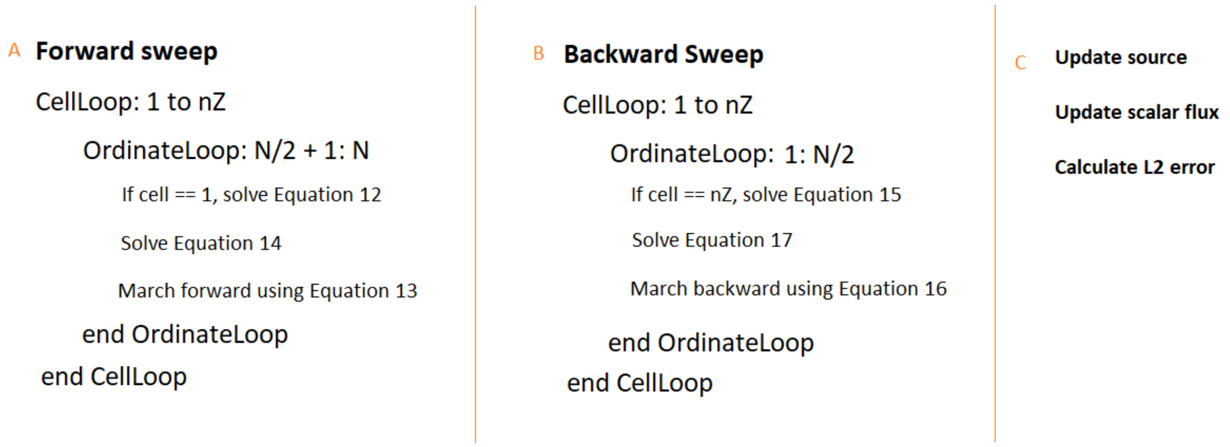

Calculate Fission Term
Inner Source Iteration
(A) Forward sweep
(B) Backward sweep
(C) Update source, scalar flux, and L2 error
Exit if L2 error ≤ inner tolerance
Update k
Calculate Scalar Flux
Check Mass Balance
Calculate L2 Error and end loop if
L2 error ≤ outer tolerance

Figure 1: Description of the outer iteration scheme.



```
A Forward sweep
    CellLoop: 1 to nZ
        OrdinateLoop: N/2 + 1: N
            If cell == 1, solve Equation 12
            Solve Equation 14
            March forward using Equation 13
        end OrdinateLoop
    end CellLoop

B Backward Sweep
    CellLoop: 1 to nZ
        OrdinateLoop: 1: N/2
            If cell == nZ, solve Equation 15
            Solve Equation 17
            March backward using Equation 16
        end OrdinateLoop
    end CellLoop

C   Update source
    Update scalar flux
    Calculate L2 error
```

Figure 2: Description of the inner iteration scheme. (A) Pseudocode for forward sweep, (B) Pseudocode for backward sweep, (C) Remainder of the steps in inner iteration

A pseudocode for the inner iteration scheme is shown in Figure 2. The inner iteration consists of three main parts. In the first part, a forward sweep is employed to calculate the angular flux for positive valued ordinates. The sweep starts at the left boundary where the boundary condition is specified and described in general form as in Equation 12. The general form can be reduced depending on the physical situation (e.g. $\gamma = 0$ if boundary is non-reflecting). Using the value at the left face and the weighted diamond difference closure in Equation 13, the angular flux can be evaluated at the cell center as in Equation 14. The weighted diamond difference closure can be used to obtain the angular flux value at the right face as in Equation 13 (forward march).

$$\psi(0,\mu_n) = \Gamma_{\mathrm{L}}(\mu_n) + \gamma\, \psi(0,\mu_{N-n+1}) \tag{12}$$
$$\mathit{where}\ 0 < \ \mu_n \leq 1, \text{and } 0 \leq \ \gamma \leq 1$$
$$\mathrm{n} = \frac{N}{2} + 1, \frac{\mathrm{N}}{2} + 2, \ldots, \mathrm{N}$$

$$\psi_{i+\frac{1}{2},n} = \frac{1}{1-\alpha}\psi_{i,n} - \ \frac{\alpha}{1-\alpha}\psi_{i-\frac{1}{2},n} \tag{13}$$

$$\psi_{i,n} = \left(1 \ + \left(\frac{(1-\mathrm{\alpha})\Sigma_{t,i}\Delta z_i}{|\mu_n|}\right)\right)^{-1} \left(\psi_{i-\frac{1}{2},n} \ + \frac{(1-\mathrm{\alpha})\Delta z_i q_{i,n}}{|\mu_n|}\right) \tag{14}$$

Similarly a backward sweep described in Eqs. 15-17 is conducted to obtain the angular flux at negative valued ordinates. After calculating the angular flux, the scattering term in the right hand side of Equation 11 could be updated. The scalar flux can be calculated from Equation 9 and the root mean square difference could be evaluated as discussed earlier and compared to the inner source iteration convergence tolerance.

$$\psi(\mathrm{L},\mu_n) = \Gamma_{\mathrm{R}}(\mu_n) + \gamma\,\psi(\mathrm{L},\mu_{N-n+1})\ where -1 \le \mu_n < 0 \text{ and } 0 \le \gamma \le 1 \qquad (15)$$
$$\mathrm{n} = 1, 2, .., \frac{N}{2}$$

$$\psi_{i-\frac{1}{2},n} = \frac{1}{1-\alpha}\psi_{i,n} - \frac{\alpha}{1-\alpha}\psi_{i+\frac{1}{2},n} \qquad (16)$$

$$\psi_{i,n} = \left(1 + \left(\frac{(1-\alpha)\Sigma_{t,i}\Delta z_i}{|\mu_n|}\right)\right)^{-1}\left(\psi_{i+\frac{1}{2},n} + \frac{(1-\alpha)\Delta z_i q_{i,n}}{|\mu_n|}\right) \qquad (17)$$

Global Mass Balance

To ensure proper implementation, a global mass balance check is conducted. The 1D slab transport equation shown in Equation 6 is first integrated over all angles to obtain Equation 18.

$$\frac{\partial J(z)}{\partial \mathrm{z}} + \Sigma_t(z)\phi(z) = \int_{-1}^{1} du \sum_{l=0}^{L} \frac{2l+1}{2}\Sigma_{\mathrm{s,l}}(z)p_l(\mu_n)\sum_{j=1}^{n} w_j p_l(\mu_j)\psi_j(z) + \frac{\nu\Sigma_f(z)}{k}\phi(z) + S(z) \qquad (18)$$

Second, Equation 18 is integrated over the length of the slab to obtain Equation 19 which can be expressed in symbolic form in Equation 20.

$$J(z_R) - J(z_L) + \int_{z_L}^{z_R} \Sigma_t(z)\phi(z)dz = \int_{z_L}^{z_R} dz \int_{-1}^{1} du \sum_{l=0}^{L} \frac{2l+1}{2}\Sigma_{\mathrm{s,l}}(z)p_l(\mu_n)\sum_{j=1}^{n} w_j p_l(\mu_j)\psi_j(z) + \int_{z_L}^{z_R} \frac{\nu\Sigma_f(z)}{k}\phi(z)dz + \int_{z_L}^{z_R} S(z)dz \qquad (19)$$

$$L_R + L_L + T = J^-(z_R) + J^+(z_L) + S + F + Sr \qquad (20)$$

The incident boundary source and the leakage terms in Equation 20 can be evaluated using Equations 21 and 22.

$$J^{+}(z_L) = \; abs\Big(\sum_{n=\frac{N}{2}+1}^{N} w_n \mu_n \psi_{0,n} \Big) \; and \; J^{-}(z_R) = \; abs\Big(\sum_{n=1}^{\frac{N}{2}} w_n \mu_n \psi_{I,n} \Big) \tag{21}$$

$$L_L = abs\Big(\sum_{n=1}^{\frac{N}{2}} w_n \mu_n \psi_{0,n} \Big) \; and \; L_R = \; abs\Big(\sum_{n=\frac{N}{2}+1}^{N} w_n \mu_n \psi_{I,n} \Big) \tag{22}$$

The boundary terms only involve angular integration. The transport, fission, and scattering terms involve integration over the slab. This integration is demonstrated in Equation 23 for the transport term.

$$T = \int_{z_L}^{z_R} \Sigma_t(z)\phi(z)dz = \sum_{i=1}^{I} \Sigma_{t,i}\, \phi_i \Delta z_i \tag{23}$$

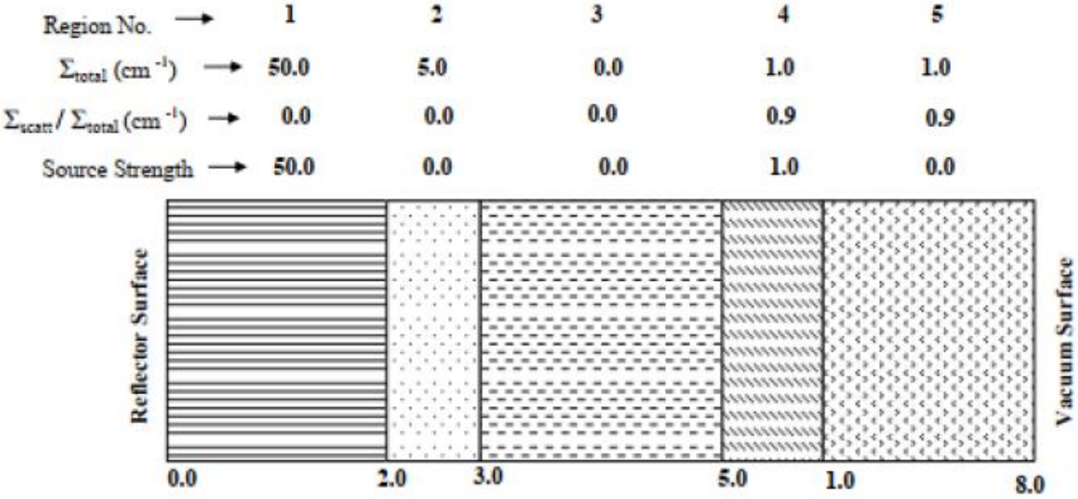
The scattering term involves multiple integrations over angle and space as shown in Equation 24. No simplifications were made to the scattering term to ensure applicability of the mass balance check in isotropic and anisotropic scattering conditions.

$$\begin{aligned} \int_{-1}^{1} du \sum_{l=0}^{L} \frac{2l+1}{2} \Sigma_{\mathrm{s,l}}(z) p_l(\mu_n) \sum_{j=1}^{n} w_j p_l\big(\mu_j\big) \psi_j(z) \\ = \sum_{n=1}^{n} w_n \sum_{l=0}^{L} \frac{2l+1}{2} \Sigma_{\mathrm{s,l}}(z) p_l(\mu_n) \sum_{j=1}^{n} w_j p_l\big(\mu_j\big) \psi_j(z) \end{aligned} \tag{24}$$

The present project has been implemented in C++ and has made use of the equations described in this section. Multi-region problems with fission and volumetric sources are supported. Both albedo and incident source boundary conditions can be used. The code has a flexible input file structure for geometry, nuclear data, and system configuration specification. The input structure is described in the software manual accompanying the code.

Results and Discussion

The five region Reed cell benchmark [3] was used to assess the implementation of the code. A description of the system in the Reed cell benchmark is shown in Figure 3. In the present simulation, an inner convergence tolerance of 1e-6 and an equal outer convergence tolerance of 1e-6 were used.



Region No. → 1 2 3 4 5
Σ_{total} (cm^{-1}) → 50.0 5.0 0.0 1.0 1.0
Σ_{scatt} / Σ_{total} (cm^{-1}) → 0.0 0.0 0.0 0.9 0.9
Source Strength → 50.0 0.0 0.0 1.0 0.0
Reflector Surface
Vacuum Surface
0.0
2.0
3.0
5.0
1.0
8.0

Figure 3: Description of the Reed cell problem [1]

First, mesh sensitivity analysis was conducted to ensure independence of the calculated results from the spatial mesh. The results were more sensitive to the mesh refinement in the first region than the other regions. This may be attributed to the much higher transport cross-section in the first region. Table 1 describes coarse and fine mesh cases that are compared in Figure 4.

Table 1: Number of cells in each region

Region:	1	2	3	4	5	Iterations
Fine Mesh	200	50	100	50	40	59
Coarse Mesh	20	10	20	10	20	57

As shown in Figure 4, oscillations are observed at the interface of the first and second cells in the coarse mesh case. When the number of cells in the first region was increased from 20 to 200 in the fine mesh case, these oscillations disappeared. Therefore, the fine mesh will be used for the remaining analysis for the Reed cell problem. It is interesting to note that the angular flux from the coarse mesh case exhibits similar trend as the fine mesh case. Using suitable ad-hoc filtering such oscillations in the coarse mesh solution may be eliminated and the solution from the coarse mesh case may be used as an initial flux distribution to the fine mesh case to accelerate convergence.

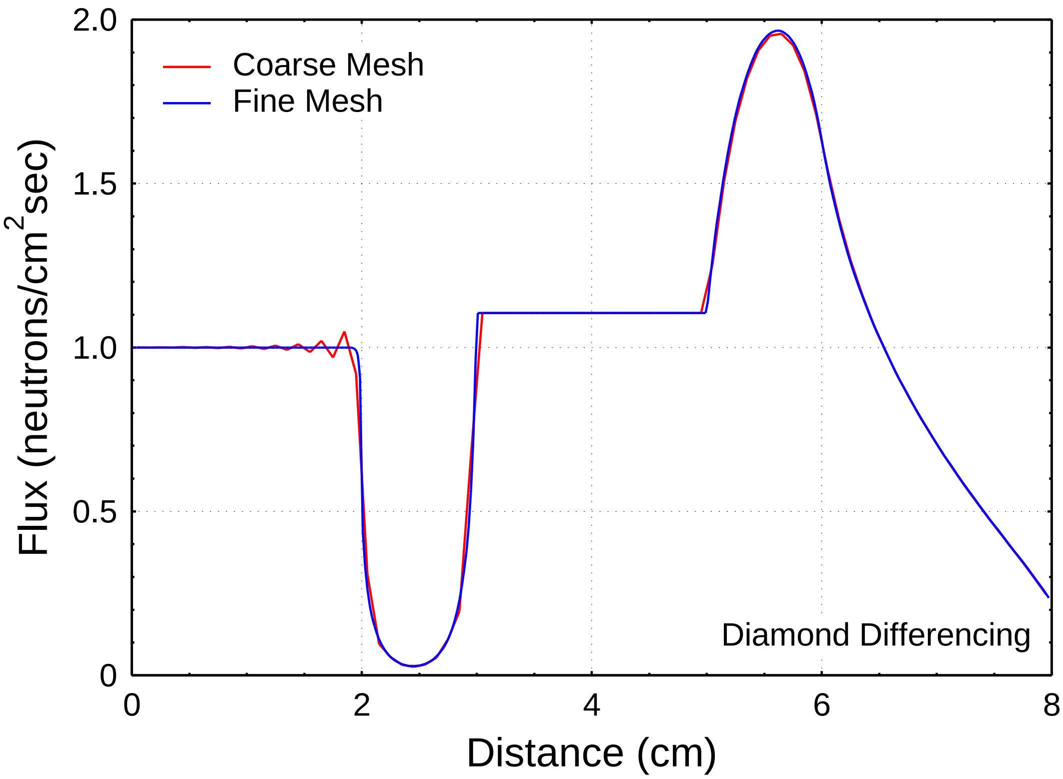


Figure 4: Comparison of the scalar flux obtained using a coarse mesh and a fine mesh

A comparison of the scalar flux obtained using diamond differencing to that obtained using step differencing is shown in Figure 5. The same spatial mesh and same number of ordinates of 8 was used in both cases. Overall, excellent agreement is observed. Minor differences in the fourth region which had the least mesh refinement (Table 1) should be noted. This minor difference decreases by increasing the mesh refinement. Further, a comparison of the angular flux in the diamond differencing case to that in the step differencing case is shown in Figure 6 using polar contours. The center of the contour corresponds to the left boundary while the outside of the contour corresponds to the right boundary. The plots show the angular flux at different ordinates. No visible difference between the diamond differencing case and step differencing case is observed.

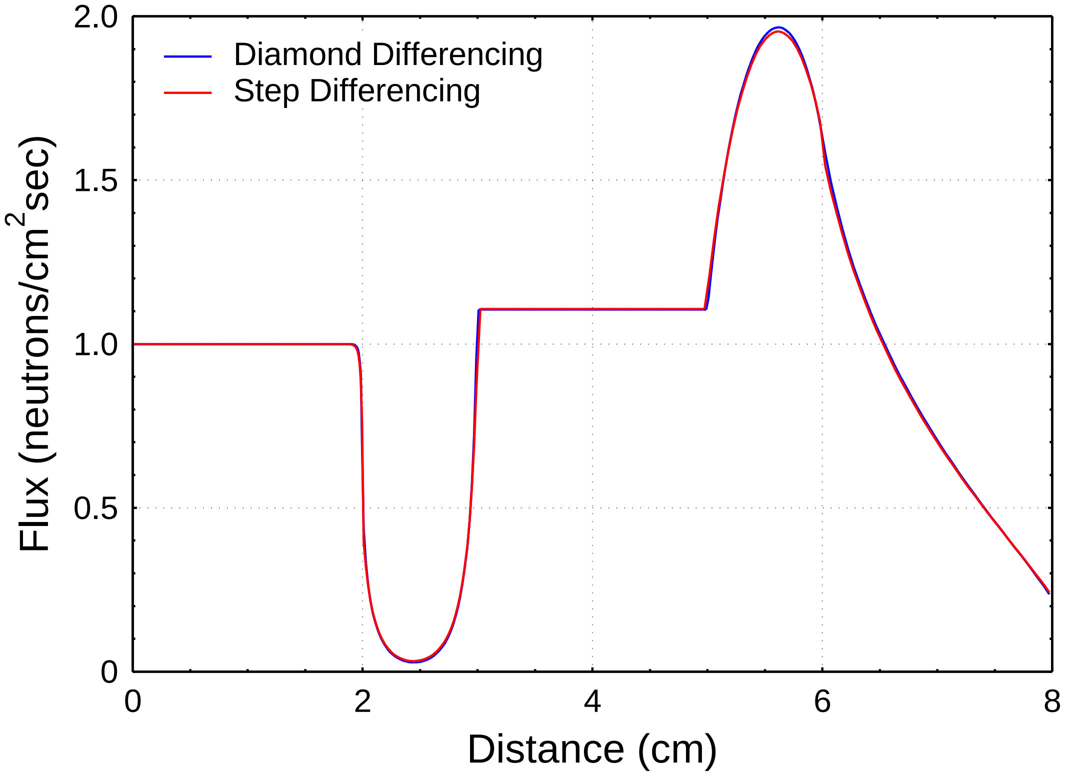


Figure 5: Comparison of the scalar flux obtained using diamond differencing and step differencing

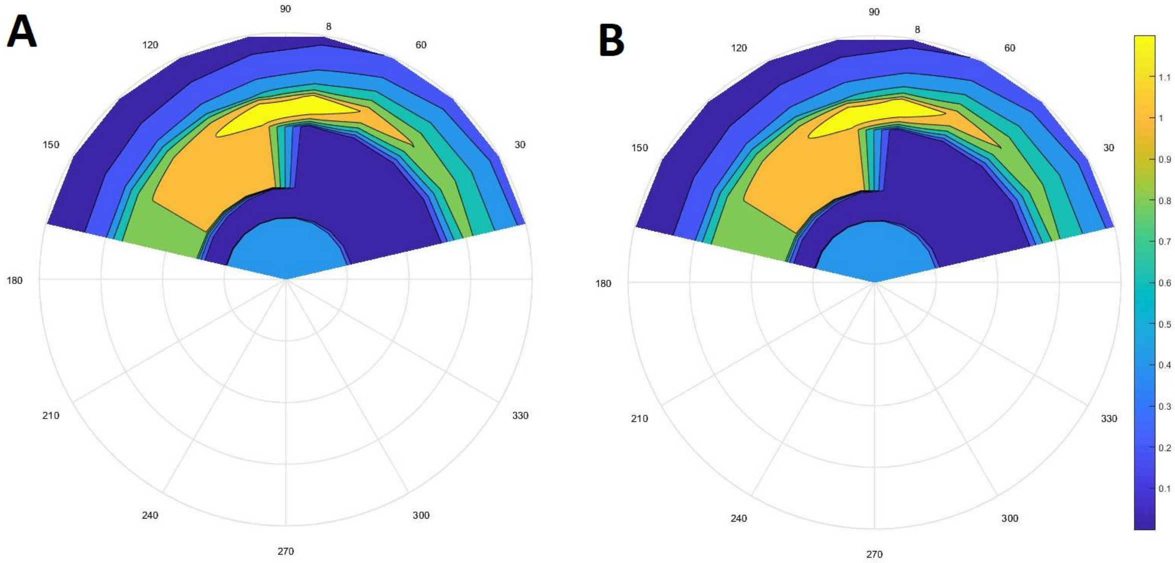


Figure 6: Spatial and angular dependence of the angular flux for (A) diamond differencing) (B) step differencing. The left boundary of the slab is represented by r = 0 and the right boundary is represented by the outer diameter.

The ISSA 2-Region criticality benchmark [4] was used to validate the k-eigenvalue calculation and the flux distribution in the presence of fission. A description of the geometry and material properties is shown in Figure 7. A comparison of the flux distribution and k-eigenvalue in diamond differencing and step differencing cases is shown in Figure 8. No visible differences in the flux could be observed. However, the k-eigenvalue obtained in the diamond differencing case was slightly different than that of the step differencing case. In particular, the k-eigenvalue in the diamond differencing case was 1.67834 and that in the step differencing case was 1.67733. Using the diamond differencing case as a reference, the relative difference between the two is ~0.06%. Further, the angular flux for each case is plotted in Figure 9. The center of the contour corresponds to the left boundary while the outside of the contour corresponds to the right boundary. The plots don’t show observable differences in angular flux between step and diamond differencing.

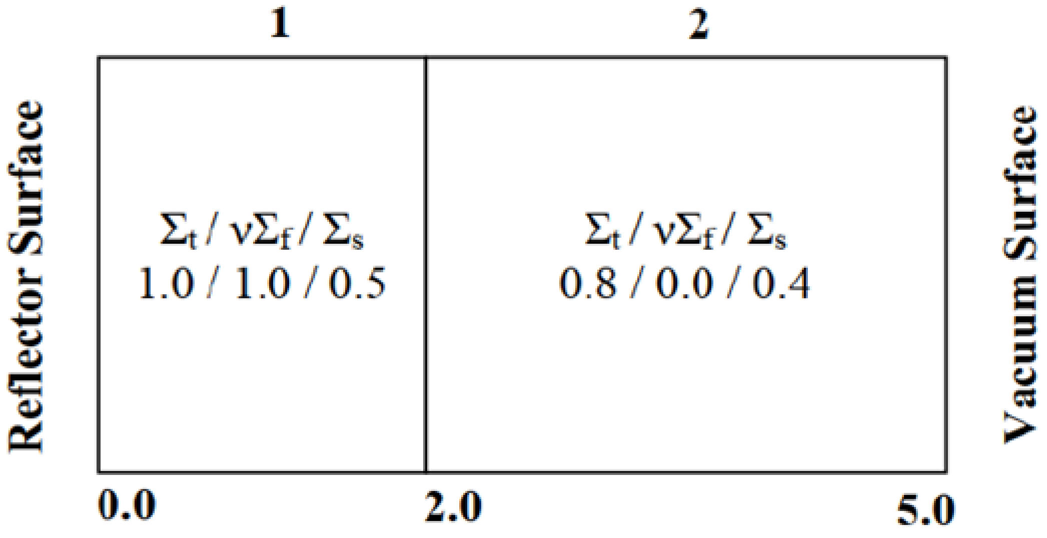


Figure 7: Description of the ISSA 2-Region Criticality Benchmark [1]

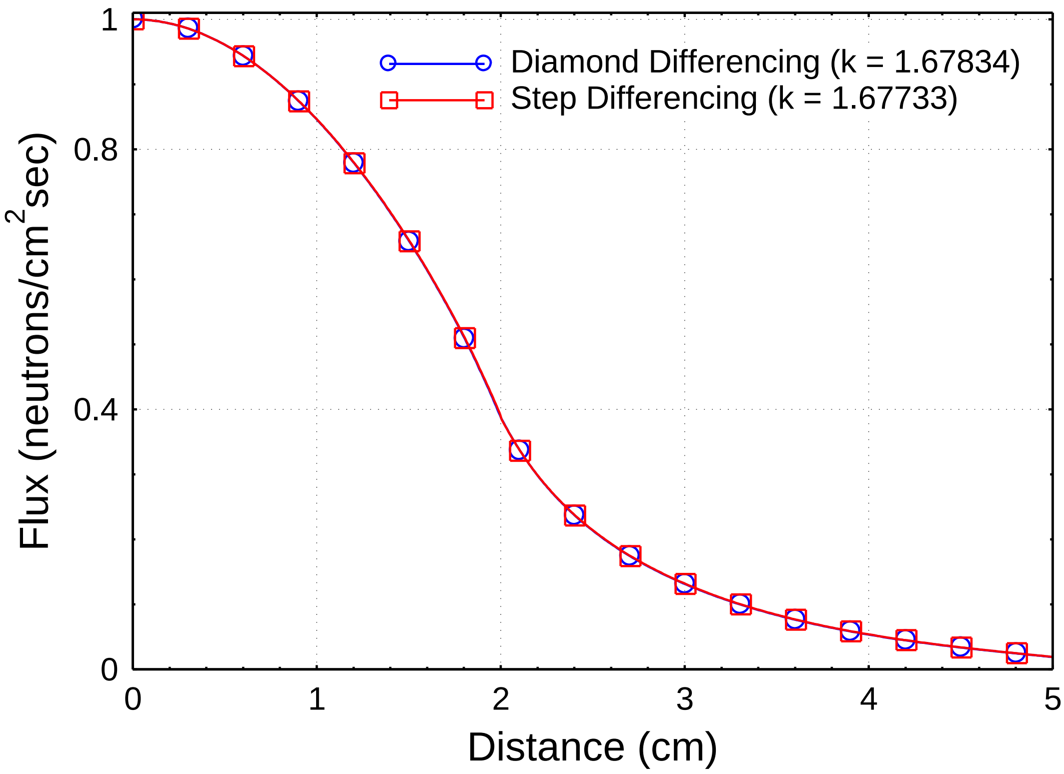


Figure 8: Comparison of the scalar flux obtained using diamond differencing and step differencing

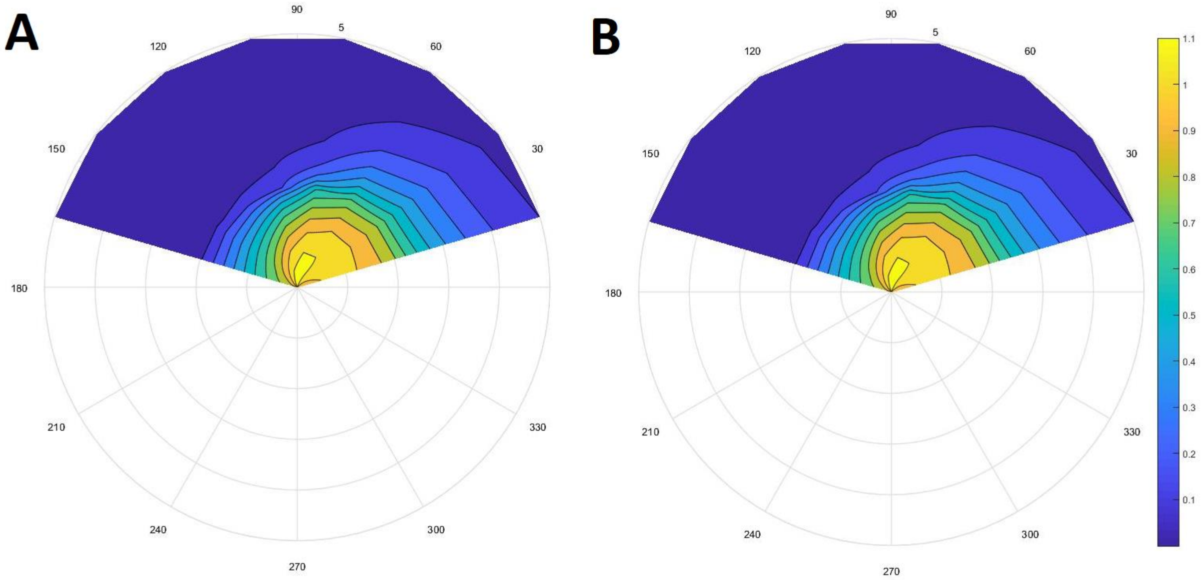


Figure 9: Spatial and angular dependence of the angular flux for (A) diamond differencing) (B) step differencing. The left boundary of the slab is represented by r = 0 and the right boundary is represented by the outer diameter.

Table 2 shows a comparison of the mass balance and iterations needed for scalar flux convergence within 1e-6 mean square difference for the two benchmark cases. The mass balance was observed to be sensitive to the convergence criterion used in the inner and outer iterations. For the same convergence criterion stated above, it was observed that the step differencing case had a slightly smaller mass balance than the diamond differencing case. However, the difference was very small and given that only two cases were considered these observations don't establish statistical significance. The Reed cell case required 59 iterations while that of the ISSA benchmark required a total of 257 inner iterations as a result of the outer fission iteration. The smaller mass balance in the ISSA benchmark case is a result of the higher number of total inner iterations due to the outer fission iterations.

Table 2: Comparison of mass balance and iterations needed for convergence in each cases

Case	Differencing Scheme	Mass Balance	Iterations
Reed Cell	DD	2.1918e-07	59
Reed Cell	SD	2.0547e-07	59
ISSA Benchmark	DD	3.0394e-09	257
ISSA Benchmark	SD	2.9985e-09	256

Finally, the benchmark results can be compared to that of Deo et al [1]. Excellent agreement with Deo et al. is observed for both the Reed Cell benchmark and the ISSA benchmark. The flux shapes for both cases are identical. Using step differencing, a 0.06% difference is observed in k compared to Deo et al., while that of diamond differencing is within 0.003%.

Acknowledgements

Corey Skinner (UNM) is acknowledged for providing ordinates function.

References

[1] Deo, K., Krishnani, P.D., and Modak, R.S., “Development of One-Dimensional Neutron Transport Theory Code Based on Method of Characteristics,” BARC/2014/E/015

[2] Lewis, E.E., Miller, W.F., “Computational Methods of Neutron Transport,” Wiley-Interscience, PP. 198-199

[3] Buchan, A. G. et. al., “Linear and Quadratic Octahedral Wavelets on the Sphere for Angular Discretisations of the Boltzmann Transport Equation”, ANE 32 (2005) 1224–1273

[4] Issa, J.G., Riyait, N.S., Goddard, A.J.H., et al., “Multigroup Application of the Anisotropic FEM code FELTRAN to One, Two, Three-dimensional and R-Z problems”, Prog. Nucl. Eng. 18 (1), 251–264 (1986)