An Analytical Nodal Discrete Ordinates Solution to the Transport Equation in Cartesian Geometry

Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Joshua Rocheleau

Graduate Program in Nuclear Engineering

The Ohio State University

2020

Thesis Committee

Dr. Dean Wang, Advisor

Dr. Richard Vasques

Abstract

A novel Analytical Nodal Discrete Ordinates (ANDO) method for the solution of the discrete ordinates (S_N) neutron transport equation in cartesian geometry is presented. A nodal method approximates the multi-dimensional transport equation as a system of coupled one-dimensional transport equations along each coordinate axis by transverse integration. The resulting transverse-integrated equations can then be discretized. The discretization utilized in the ANDO method is based on a recent closed-form analytical solution in slab geometry to give a truly closed-form solution on the computational cell. Further, the closed-form solution on any 2^n heterogenous domain is also readily obtained. The new ANDO method is free from spatial truncation error within the computational cell and is limited in accuracy only by the approximation used for the transverse leakage, as are all analytical nodal methods. Results for constant, linear, and quadratic transverse leakage approximations are presented. The ANDO method possesses a number of favorable properties such as high accuracy, rapid convergence, asymptotic preserving, positivity preserving, near linear computational complexity, and is local-hp adaptive. It is also shown that the ANDO method can easily be extended to higher order transverse leakage approximations, to 3-dimensional cartesian geometry, and to multi-group.

Acknowledgments

I would like to express my sincere gratitude to my research advisor Dr. Dean Wang. From day one when I met with Dr. Wang as an undergraduate junior, he remained committed to helping me achieve my goal of completing the Master's program with a research thesis in only one year, even if taking on a student for such a short period of time would be difficult and potentially of little benefit. Dr. Wang worked with me to explore several potential research projects that would be of an appropriate scope for a compressed timeline and I am thankful for the opportunity to build on his work of the one-dimensional analytical S_N solution. I am also thankful to Dr. Wang for offering me support in the form of a partial Graduate Research Associateship.

Vita

August 2016 - A.S. University Engineering Transfer, Sinclair Community College, with High Honors.

December 2019 - B.S. Mechanical Engineering, Ohio State University, Magna Cum Laude, with minor in Nuclear Engineering, with Honors in Engineering, with Honors Research Distinction in Engineering.

August 2018 to May 2019 - Honors Undergraduate Research, Thermal Properties of Materials for Extreme Environments Lab, Ohio State University.

May 2019 to August 2019 - Nuclear Engineer Intern, Framatome Inc.

Publications

J. Rocheleau, D. Wang, "A Novel Analytical Nodal Method for Solution of the SN Transport Equation," 2020 ANS Summer Meeting, Phoenix, AZ, June 7-11, 2020.

Fields of Study

Major Field: Nuclear Engineering

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction and Background

Introduction

Ever-expanding modeling and simulation capability promotes rapid technological development. Scientific advances allow for models of ever-increasing complexity and technological advances in computing power allow for simulation of ever-increasing sophistication. The optimization of any technology is dependent on a scientific model to describe the physical phenomena and the subsequent solution of the mathematical model to derive meaningful conclusions. Optimization of a technology is also dependent on the collection of meaningful measurements and subsequent analysis to inform design decisions. In the past, physical experimentation was the only means by which data could be collected and the complexity of mathematical models that could be evaluated was limited. Development was limited by measurements that were difficult or impossible to take, limited in scale, limited in the number of designs that could be tested, and limited by how quickly prototypes could be built and tested. Although simulation will never replace the need for physical experimentation to confirm a technology works as intended, simulation is replacing the need for physical experimentation to develop and optimize a technology. Simulation allows for vastly more data to be collected and allows for more designs to be tested than with physical prototyping and testing. However, at this point scientific theory and mathematical models exceed our ability to accurately simulate and predict the behavior of physical systems. Therefore, increased simulation capability further promotes technological development.

Principle among the modeling and simulation needs of nuclear engineering are fluid mechanics, heat transfer, and neutronics. The simulation of neutronics is a unique need of nuclear engineering. At the core of any model is an underlying governing equation to describe the physical phenomenon under study. The Boltzmann equation describes the physical behavior of neutrons and is the governing equation for neutronics. Many problems of practical interest to engineering cannot be solved analytically. If an analytical solution to the governing equation cannot be derived a numerical method can be employed. At the core of every simulation code is an underlying numerical method used to solve the governing equation. Development of accurate and efficient numerical methods can therefore improve simulation capability.

Background

Numerical methods can be classified into two broad categories: stochastic methods and deterministic methods. Stochastic methods approach the problem from the standpoint of statistical probability. A large number of particles are simulated with known interaction mechanisms of known probability and the behavior of the system is determined by the aggregate response of a sufficient number of particles. Stochastic methods are the most physically accurate model of real particles, whose behavior is statistical in nature. Stochastic methods can generally achieve excellent agreement with experimental measurements if a sufficient number of particle-lives are simulated. However, it often requires significant computational resources and time to simulate problems of appreciable size to an acceptable degree of certainty because a large number of particles must be traced.

Deterministic methods approach the problem from the solution of the underlying governing equation. Ideally, the governing equation can be solved directly to derive the analytical solution. The results of an analytical solution are completely free from uncertainty, unlike stochastic methods. The accuracy of an analytical solution is limited only by the adequacy of the governing equation used to model the physical system, which is generally sufficient for practical engineering work. However, the analytical solution of a PDE is often elusive for all but the simplest problems. Therefore, if an analytical solution cannot be derived the governing equation is discretized and subsequently solved using a numerical method. The resulting deterministic solution is free from uncertainty but limited in accuracy by the adequacy of the discretization. The solution of a deterministic numerical method will approach the analytical solution as the discretization becomes infinitely fine. However, it often requires significant computational resources and time to simulate problems of appreciable size to an acceptable degree of accuracy because a sufficient discretization is needed.

Thus, both stochastic and deterministic methods can require significant computational resources that ultimately inhibit the ability of modeling and simulation to advance engineering design. Therefore, more accurate, more efficient, numerical methods that can reduce the computational cost of modeling and simulation are clearly beneficial. This work will focus on deterministic methods, although the development of stochastic methods is just as important.

Significant work has been done since the 1950’s to develop computationally efficient numerical solutions to the Boltzmann equation. First, the neutronics can be modeled using either the diffusion equation or the transport equation. The diffusion equation is based on Fick’s Law and models the neutrons as diffusing through a medium, moving from regions of high concentration to regions of low concentration. The diffusion equation provides excellent analytical and numerical modeling of the neutronics of thermal reactors and has been used widely in the design of light water reactors. The transport equation models the neutron collisions as the neutrons traverse a region. The transport equation is more amenable to modeling radiation transport and other rarified phenomena. The transport equation also provides better modeling of the neutronics of fast reactors. This work used the transport equation as the governing equation.

The transport equation contains seven independent variables: three spatial coordinates, two angular directions, one temporal variable, and one energy (or speed) variable. The transport equation is thus typically simplified to reduce the number of independent variables. The time variable is typically discretized by employing a time stepping scheme for transient simulations or the problem is considered time independent to model steady state conditions. The energy variable is typically discretized into groups based on predetermined energy bands or the problem is considered mono-energetic. The direction can be discretized using either Spherical Harmonics (P_N) or Discrete Ordinates (S_N). The S_N approximation restricts the transport of neutrons into discrete directions spanning the dimension of the problem under consideration and assigns a weight to each discrete ordinate. The spatial variables can be any 1, 2, or 3-dimensional coordinate system. The dimensionality of the equation can be reduced to simplify the problem in an intuitive way. This work considers the time-independent, mono-energetic, discrete ordinates solution in 2D geometry and subsequently applies the methodology to 3D geometry.

In the discrete ordinates method, the transport equation is written for each respective ordinate to establish a system of partial differential equations (PDE’s). The system of PDE’s can then be simplified into a system of spatially one-dimensional ordinary differential equations (ODE’s) using either the Method of Characteristics (MOC) or the nodal method. In the MOC the streaming and collision operator is inverted exactly along each discrete ordinate and the resulting ODE can be solved to determine the angular flux value at any point along the characteristic. The spatial dependence of the angular neutron flux can be found by the translation of the characteristic line along the coordinate axes and the scalar flux can then be determined at the intersection of the characteristics. An alternative approach (the nodal method) is to integrate the equation along one dimension (or two dimensions for three-dimensional geometry) to obtain the transverse integrated equation along the cell. The leakage along the transverse cell faces is then approximated by a function and the resulting ODE can be solved to determine the angular flux along each cell. Using either approach, the resulting ODE is now amenable to any 1D discrete ordinates method. Both the MOC and the nodal method can be easily extended to 3D by following the approach described above.

A wide variety of 1D S_N numerical methods have been developed over the past few decades. The simplest approach is to solve the system of coupled ODE's using a finite difference method, such as is used in the Diamond Difference (DD) method. This work will focus the discussion on a class of methods that solve the system of equations analytically using Green's function method, Laplace transform method, and the decomposition method. Finally, the problem domain is spatially discretized and the flux at each node can be determined from the selected numerical method either iteratively or by matrix inversion techniques. The latter is generally not practical for all but the smallest problems due to the size of the solution matrix. The solution of large systems of equations can result in significant numerical roundoff error and can require significant computation time. Iterative solution schemes are thus preferred.

Previous Related Work

The method presented in this thesis will now be put into a historical context to highlight the contribution of previous work on similar methods. The group of methods are related, and differentiated from other methods, by the use of the transport equation to model the neutronics, the use of discrete ordinates to discretize the angular variable, the use of the nodal method to reduce the PDE to an ODE, and the use of analytical techniques to solve the system of ODE's. The methods differ only in the analytical technique applied to solve the system of ODE's. Each method developed in the 1990’s first as a one-dimensional solution and was later extended to higher dimensions using the nodal method. The nodal method was first developed for the solution of the diffusion equation and was later applied to the solution of the transport equation in the 1980’s [1-6].

In 1990 Barros and Larsen developed a method based on the Spectral Green’s Function [7-8] to derive the analytical solution. Notably, the method was free from spatial truncation error, a unique feature of analytical nodal methods. Barros and Larsen extended the method to multi-group [9] and extended the method to two-dimensional geometry using the nodal method [9]. Two aspects of the Spectral Green’s Function Discrete Ordinates (SGFSN) method are worth mentioning. First, the transverse leakage resulting from integration in the nodal method must be approximated. The SGF Constant Nodal (SGF-CN) approximated the transverse leakage as constant, although later work by Barros considered higher order approximations (2002) [10]. Second, SGF-CN used the so-called Cell Block Inversion (CBI) iteration scheme instead of the more typical source iteration scheme. The CBI scheme solves both sets of transverse integrated equations in each cell simultaneously and updates the angular flux on the cell edge. CBI significantly improves the convergence of analytical nodal methods over methods that require source iteration. This work independently employed the CBI scheme. The SGF-CN method was extended to multi-group in 2018 [11].

In 1991 Barichello and Vilhena publish a new analytical approach based on the Laplace Transform (LTSN) to obtain the eigen values and eigen vectors of the problem [12]. LTSN was generalized in 1993 [13] and extended to multi-group in 1995 [14]. Based on the LTSN method, Zabadal, Vilhena, and Barichello extended the method to two-dimensional geometry using the nodal method (1993) [15] and to three-dimensional geometry (1994) [16]. The utility of the LTSN method was later applied to curvilinear coordinates [17] and to criticality problems [18] in 1997.

The LTSN method is free from spatial truncation error within each node, as are all analytical nodal methods. The eigen values and eigen vectors are determined by the inversion of a large coefficient matrix without iteration.

Alternatively, in 1997 Vargas and Vilhena found that the eigen values and eigen vectors of the solution matrix could be determined by the decomposition method [19]. The decomposition method was based on a 1988 publication by mathematician Adomian [20]. The SN decomposition was based on the nodal LTSN but was novel in two important ways. First, the average flux and the cell edge flux were combined whereas LTSN assigned the average and cell edge flux separate coefficients. Second, the SN decomposition method solves for the eigen values and eigen vectors using eigen decomposition rather than inversion of a large coefficient matrix. Both of these aspects reduce the number of unknowns and allow the final expression to be closed form.

The LTSN method was adapted to radiative transfer problems by Barichello in the late 1990's [21-22]. In 1997 Barichello and Siewert developed a method that also employed eigen value decomposition, rather than the Laplace Transform, to determine the eigen values and eigen vectors [23-24] and termed the approach the Analytical Discrete Ordinates (ADO) method. The decomposition method was based on a 1991 publication by Golub and Van Loan [25] and, given the publication date, was likely not based on the work of Vargas using the decomposition method. Further, the ADO method treats the average flux and the cell edge flux separately and was first applied to radiative transfer problems, unlike the SN decomposition of Vargas.

The ADO method was then extended to two-dimensional problems in 2009 by Barichello, Cabrera, and Prolo Filho using the nodal method to develop the Analytical Nodal Discrete Ordinates (ANDO) method [26-27]. ANDO was further developed by Picoloto et. al. for reflected boundaries (2013) [28], heterogenous domain (2015) [29], and anisotropic scattering (2017) [30]. Linear and exponential transverse leakage approximations were also considered (2019) [31]. Like the SGFSN method, the ANDO method is free from spatial truncation error within a cell and approximates the transverse leakage as constant. The ANDO method also solves the problem non-iteratively by inversion of a large coefficient matrix.

In 2017 Wang and Byambaakhuu developed a new analytical SN solution [32] based on the work of Vargas using the eigen value decomposition [19]. The novelty of the approach is that the homogenous and particular solution of the ODE are solved directly whereas the typical approach, as in the ADO method, is to solve the homogenous and particular solution separately. Further, the scattering and streaming terms are treated together in the eigen value decomposition, as in the SN decomposition method, whereas the ADO method treats the average flux and cell face flux separately. Both aspects of the method allow all the unknown coefficients to be determined analytically resulting in a final expression that is truly closed form. Like all analytical methods, the new analytical SN method is free from spatial truncation error but unlike previous analytical methods contains no coefficients to be solved for numerically. The latter point is especially significant for the computational efficiency of the method when extended to larger problems. This work is the extension of the new ADO method to higher dimensions using the nodal method.

Thus, both analytical discrete ordinates methods have their origin in the Laplace Transform method of 30 years ago [12]. 1997 was a fortuitous year leading to the development of this work; the LTSN method was well established by extension to curvilinear and criticality problems [18], the SN Decomposition method that would later lead to this work was published [19], and the ANDO method based on eigen decomposition was developed [23-24]. Twenty years later in 2017 the latest development in the ANDO method was published [30] and the new analytical S_N solution that would lead to this work was published [32].

Chapter 2. Theory and Formulation

Derivation of the ANDO Method

The mono-energetic neutron transport equation in two-dimensional cartesian geometry with isotropic scattering and constant neutron source is written in discrete ordinates from as

$$\mu_m \frac{\partial}{\partial x}\psi(x,y,\Omega_\mathrm{m}) + \eta_m \frac{\partial}{\partial y}\psi(x,y,\Omega_\mathrm{m}) + \Sigma_t \psi(x,y,\Omega_\mathrm{m}) =$$

$$\Gamma^{-1}\Sigma_s \sum_{k=1}^{M} w_k \psi(x,y,\Omega_k) + \Gamma^{-1}Q(x,y), \qquad (1)$$

where μ_m and η_m are the sine and cosine, respectively, of the angle $\Omega_m = (\mu_m, \eta_m), m = 1, \dots, M$, defined by the angular quadrature set, ψ is the angular flux, Σ_t and Σ_s are the total and scattering macroscopic cross section, respectively, $\Gamma = 4$ for 2D plane geometry, and Q is the external neutron source term. For homogenous medium the spatial dependence of the material terms can be dropped for notational convenience.

Now consider a spatially discretized region bound on the x-axis by $\left(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right)$ and on the y-axis by $\left(y_{i-\frac{1}{2}}, y_{i+\frac{1}{2}}\right)$. The partial differential Equation (1) can be integrated over y from $y_{i-\frac{1}{2}}$ to $y_{i+\frac{1}{2}}$ to obtain an easily invertible ordinary differential equation along the x direction. The transverse integrated equation along the x direction is

$$\mu_m \frac{\partial}{\partial x} h_y^{-1} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} dy\, \psi_{i,j}(x,y,\Omega_m) + \eta_m h_y^{-1}\left[\psi_{i,j}\left(x, y_{j+\frac{1}{2}}, \Omega_m\right) - \psi_{i,j}\left(x, y_{j-\frac{1}{2}}, \Omega_m\right)\right] +$$

$$\Sigma_t h_y^{-1} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} dy\, \psi_{i,j}(x,y,\Omega_m) = \Gamma^{-1}\Sigma_s \sum_{k=1}^{M} w_k h_y^{-1} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} dy\, \psi_{i,j}(x,y,\Omega_k) + \Gamma^{-1}Q. \qquad (2)$$

The derivation presented here will follow the solution of the transverse integrated equation along the x. An equivalent expression can be obtained for the solution along the y by integrating Equation (1) over x from $x_{i-\frac{1}{2}}$ to $x_{i+\frac{1}{2}}$. An explicit distinction between the resulting expressions for the solution along the y will be made where such a distinction is informative.

For notational convenience, let

$$\psi_{i,j}(x,\Omega_m) = h_y^{-1}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} dy\,\psi_{i,j}(x,y,\Omega_m), \tag{3}$$

and

$$L_{i,j}(x,\Omega_m) = h_y^{-1}\left[\psi_{i,j}\left(x,y_{j+\frac{1}{2}},\Omega_m\right) - \psi_{i,j}\left(x,y_{j-\frac{1}{2}},\Omega_m\right)\right]. \tag{4}$$

The transverse leakage term $L_{i,j}$ is an approximation to the transverse integration performed on Equation (1). The accuracy of the ANDO method is limited only by the accuracy of this approximation. This work considers approximating the transverse leakage term as constant along the computational cell face and as a linear and quadratic distribution along the cell face. The derivation will follow the linear distribution approximation as it differs from the derivation for the constant transverse leakage only in the inclusion of higher order terms needed to establish the slope of the linear distribution. An explicit distinction between the resulting expressions will be made where such a distinction is informative. Equation (4) is now posed as

$$L_{i,j}(x,\Omega_m) = \mathbb{L}_{x,i,j}(\Omega_m) + L_{rx,i,j}(\Omega_m) + \mathrm{r}_{x,i,j}(\Omega_m)x, \tag{5}$$

where

$$\mathbb{L}_{x,i,j}(\Omega_m) = \left(\bar{\psi}_{i,j+\frac{1}{2}}(\Omega_m) - \bar{\psi}_{i,j-\frac{1}{2}}(\Omega_m)\right)h_y^{-1}, \tag{5a}$$

$$\mathrm{r}_{x,i,j}(\Omega_m) = h_x^{-1}h_y^{-1}\alpha_{x,i,j}(\Omega_m), \tag{5b}$$

$$L_{rx,i,j}(\Omega_m) = -h_y^{-1}2^{-1}\alpha_{x,i,j}(\Omega_m) = -\mathrm{r}_{x,i,j}(\Omega_m)h_x 2^{-1}. \tag{5c}$$

The $\mathbb{L}_x$ term recovers the constant transverse leakage approximation if the r_x and L_{rx} terms are neglected, r_x accounts for the linear distribution on the cell face, and L_{rx} accounts for the intercept of the linear distribution with consideration to the $\mathbb{L}_x$ term. The slope of the line, α_x, can be calculated using various approaches.

Equations (3) and (5) are substituted into Equation (2) and rearranged to give

$$\mu_m \frac{\partial}{\partial x} \psi_{i,j}(x, \Omega_m) + \Sigma_t\, \psi_{i,j}(x, \Omega_m) - \Gamma^{-1} \Sigma_s \sum_{k=1}^{M} w_k \psi_{i,j}(x, \Omega_\mathrm{k}) =$$

$$\Gamma^{-1} Q - \eta_m (\mathbb{L}_{x,i,j}(\Omega_m) + L_{x,i,j}(\Omega_m) + \mathrm{r}_{x,i,j}(\Omega_m) x), \tag{6}$$

which is different from the 1D analytical solution [32] only in the inclusion of the second term on the right-hand side of the equation. It is convenient at this point to write Equation (6) in a matrix form

$$\frac{\partial}{\partial x} \boldsymbol{\psi}_x + \boldsymbol{\mu}_\mathbf{x}^{-1} \Sigma_t \boldsymbol{\psi}_x - \boldsymbol{\mu}_\mathbf{x}^{-1} \Gamma^{-1} \Sigma_s \boldsymbol{W}_x \boldsymbol{\psi}_x = \boldsymbol{\mu}_\mathbf{x}^{-1} \Gamma^{-1} Q \mathbf{1} - \boldsymbol{\mu}_\mathbf{x}^{-1} \boldsymbol{\eta}_x (\mathbb{L}_x + \boldsymbol{L}_{rx} + \mathbf{r}_\mathbf{x} x), \tag{7}$$

where

$$\boldsymbol{\psi}_x = \begin{bmatrix} \psi_{i,j}(x, \Omega_m) \\ \vdots \\ \psi_{i,j}(x, \Omega_\mathrm{M}) \end{bmatrix}, \text{ angular flux vector;} \tag{7a}$$

$$\mathbb{L}_x = \begin{bmatrix} \mathbb{L}_{\boldsymbol{i,j}}(\boldsymbol{\Omega_m}) \\ \vdots \\ \mathbb{L}_{\boldsymbol{i,j}}(\boldsymbol{\Omega_M}) \end{bmatrix}, \text{ transverse flux leakage constant vector;} \tag{7b}$$

$$\boldsymbol{L}_x = \begin{bmatrix} L_{x,i,j}(\Omega_m) \\ \vdots \\ L_{x,i,j}(\Omega_M) \end{bmatrix}, \text{ transverse flux leakage intercept vector;} \tag{7c}$$

$$\mathbf{r}_x = \begin{bmatrix} \mathrm{r}_{x,i,j}(\Omega_m)) \\ \vdots \\ \mathrm{r}_{x,i,j}(\Omega_M) \end{bmatrix}, \text{ transverse flux leakage linear vector;} \tag{7d}$$

$$\boldsymbol{\mu}_\mathbf{x} = \begin{bmatrix} \boldsymbol{\mu}_x & \\ & -\boldsymbol{\mu}_x \end{bmatrix}, \text{ with } \boldsymbol{\mu}_x = \begin{bmatrix} \mu_m & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mu_{M/2} \end{bmatrix}; \text{ and} \tag{7e}$$

$$\boldsymbol{\eta}_{\boldsymbol{x}} = \begin{bmatrix} \eta_m & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \eta_M \end{bmatrix}, \boldsymbol{W}_{\boldsymbol{x}} = \begin{bmatrix} w_m & \cdots & w_M \\ \vdots & \ddots & \vdots \\ w_m & \cdots & w_M \end{bmatrix}. \tag{7f}$$

The quadrature points $\Omega_m = (\mu_m, \eta_m)$ and weights w_m are ordered such that $\mu_m > \mu_{m+1}$ for $m = 1,2,\dots,\frac{M}{2}$ and $\mu_{m+\frac{M}{2}} = -\mu_m$, where M is the number of quadrature points, but that η_m is ordered to match μ_m and is not ordered sequentially.

For scattering ratio $c = \Sigma_s/\Sigma_t$ and defining $\boldsymbol{q}_{\boldsymbol{x}}$ as

$$\boldsymbol{q}_{\boldsymbol{x}} = \boldsymbol{\mu}_{\mathbf{x}}^{-1}(\Gamma^{-1}Q\mathbf{1} - \boldsymbol{\eta}_{\boldsymbol{x}}\mathbb{L}_{\boldsymbol{x}} - \boldsymbol{\eta}_{\boldsymbol{x}}\boldsymbol{L}_{\boldsymbol{rx}}), \tag{8}$$

Equation (7) can be rearranged to give

$$\frac{\partial}{\partial x}\boldsymbol{\psi}_x + \Sigma_t\boldsymbol{\mu}_{\mathbf{x}}^{-1}(\boldsymbol{I} - c\Gamma^{-1}\boldsymbol{W}_x)\boldsymbol{\psi}_x = \boldsymbol{q}_{\boldsymbol{x}} - \boldsymbol{\mu}_{\mathbf{x}}^{-1}\boldsymbol{\eta}_x\mathbf{r}_{\mathbf{x}}x. \tag{9}$$

An equivalent expression can be obtained for the y direction as

$$\frac{\partial}{\partial y}\boldsymbol{\psi}_y + \Sigma_t\boldsymbol{\eta}_{\boldsymbol{y}}^{-1}\left(\boldsymbol{I} - c\Gamma^{-1}\boldsymbol{W}_y\right)\boldsymbol{\psi}_y = \boldsymbol{q}_y - \boldsymbol{\eta}_y^{-1}\boldsymbol{\mu}_y\boldsymbol{r}_y x, \tag{10}$$

where the matrices $\boldsymbol{\mu}_{\mathbf{y}}$ and $\boldsymbol{\eta}_{\mathbf{y}}$ are defined with $\Omega_n = (\mu_n, \eta_n)$ and w_n ordered such that $\eta_n > \eta_{n+1}$ for $n = 1,2,\dots,\frac{N}{2}$ and $\eta_{n+\frac{N}{2}} = -\eta_n$, where N is the number of quadrature points. Note that the ordering of Ω_m is different from the ordering of Ω_n so that the angle of each ordinate decreases along each respective direction.

The matrix $\Sigma_t\boldsymbol{\mu}_{\boldsymbol{x}}^{-1}(\boldsymbol{I} - c\Gamma^{-1}\boldsymbol{W}_{\boldsymbol{x}})$ appearing in Equation (9) is diagonalizable and the following eigen value problem can be solved

$$\Sigma_t\boldsymbol{\mu}_x^{-1}(\boldsymbol{I} - c\Gamma^{-1}\boldsymbol{W}_x) = \boldsymbol{R\Lambda R^{-1}}. \tag{11}$$

For a given quadrature set, $\boldsymbol{\Lambda}$ could contain repeated eigen values but $\boldsymbol{R}$ is non-singular, so the matrix is diagonalizable. The eigen values $\boldsymbol{\Lambda}$ and eigen vectors $\boldsymbol{R}$ are not subscripted because, for the level symmetric quadrature set ordered as described above, the matrix $\boldsymbol{\mu}_{\mathbf{x}}$ is identical to the matrix $\boldsymbol{\eta}_{\mathbf{y}}$.

Substituting Equation (11) into Equation (9) and defining

$$\mathbb{X} = \boldsymbol{R}^{-1}\boldsymbol{\psi}_x, \tag{12a}$$

$$\boldsymbol{b_x} = \boldsymbol{R}^{-1}\boldsymbol{q_x}, \tag{12b}$$

$$\boldsymbol{\rho_x} = -\boldsymbol{R}^{-1}\boldsymbol{\upmu}_{\mathbf{x}}^{-1}\boldsymbol{\eta_x}\mathbf{r_x}, \tag{12c}$$

Equation (9) can be written as

$$\frac{\partial}{\partial x}\mathbb{X} + \boldsymbol{\Lambda}\mathbb{X} = \boldsymbol{b_x} + \boldsymbol{\rho_x}x. \tag{13}$$

Note that the additional term $\boldsymbol{\rho_x}$ and all subsequent terms containing $\boldsymbol{\rho_x}$ are not present in the formulation for constant transverse leakage. Equation (13) can be easily integrated to give the analytical solution by assuming linear transverse leakage as

$$\mathbb{X} = \boldsymbol{\Lambda}^{-1}\boldsymbol{b_x} - \boldsymbol{\Lambda}^{-2}\boldsymbol{\rho_x} + \boldsymbol{\Lambda}^{-1}\boldsymbol{\rho_x}x - e^{-x\boldsymbol{\Lambda}}\boldsymbol{a_x}, \tag{14}$$

where $\boldsymbol{a_x}$ is a vector of unknown constants $a_{x,m}$ resulting from the solution of the system of ordinary differential equations (13). The vectors and matrices in Equation (14) can be split into forward and backward directions notated by + and – subscripts, respectively, for notational convenience and the x subscript can be dropped for clarity to give

$$\begin{bmatrix}\mathbb{X}_+\\ \mathbb{X}_-\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Lambda}_+^{-1}\boldsymbol{b}_+ - \left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_+ + \left(\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho}\right)_+ x - e^{-x\boldsymbol{\Lambda}_+}\boldsymbol{a}_+\\ \boldsymbol{\Lambda}_-^{-1}\boldsymbol{b}_- - \left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_- + \left(\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho}\right)_- x - e^{-x\boldsymbol{\Lambda}_-}\boldsymbol{a}_-\end{bmatrix}. \tag{15}$$

The constants $\boldsymbol{a}$ can be determined by the boundary conditions at $x = 0$ and $x = h_x$:

$$\boldsymbol{a}_+ = \boldsymbol{\Lambda}_+^{-1}\boldsymbol{b}_+ - \left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_+ - \mathbb{X}_+^0, \tag{16a}$$

$$\boldsymbol{a}_- = \mathrm{e}^{h_x\boldsymbol{\Lambda}_-}\boldsymbol{\Lambda}_-^{-1}\boldsymbol{b}_- - \mathrm{e}^{h_x\boldsymbol{\Lambda}_-}\left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_- + \mathrm{e}^{h_x\boldsymbol{\Lambda}_-}\left(\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho}\right)_- h_x - \mathrm{e}^{h_x\boldsymbol{\Lambda}_-}\mathbb{X}_-^h, \tag{16b}$$

where $\begin{bmatrix}\mathbb{X}_+^0\\ \mathbb{X}_-^{\mathrm{h}}\end{bmatrix}$ can be determined by the following equation:

$$\begin{bmatrix}\boldsymbol{\psi}_+^0\\ \boldsymbol{\psi}_-^h\end{bmatrix} = \begin{bmatrix}\boldsymbol{I} & \\ & \mathbf{0}\end{bmatrix}\boldsymbol{R}\begin{bmatrix}\mathbb{X}_+^0\\ \mathbb{X}_-^0\end{bmatrix} + \begin{bmatrix}\mathbf{0} & \\ & \boldsymbol{I}\end{bmatrix}\boldsymbol{R}\begin{bmatrix}\mathbb{X}_+^h\\ \mathbb{X}_-^h\end{bmatrix}. \tag{17}$$

with $\boldsymbol{\psi}_{+}^{\mathbf{0}}$ and $\boldsymbol{\psi}_{-}^{\boldsymbol{h}}$ as the incoming angular flux on the left and right cell faces, respectively, for a cell of length h_x. The vectors $\mathbb{X}^0$ and $\mathbb{X}^h$ can be found by solving Equation (15) at $x = 0$ and $x = h_x$ to give

$$\begin{bmatrix} \mathbb{X}_{+}^{0} \\ \mathbb{X}_{-}^{0} \end{bmatrix} = \begin{bmatrix} \mathbb{X}_{+}^{0} \\ \boldsymbol{\Lambda}_{-}^{-1}\boldsymbol{b}_{-} - \left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_{-} - \boldsymbol{a}_{-} \end{bmatrix} = \begin{bmatrix} \mathbb{X}_{+}^{0} \\ \mathbf{A}_{-}^{0} + \mathrm{e}^{h_x\boldsymbol{\Lambda}_{-}}\mathbb{X}_{-}^{h} \end{bmatrix}, \tag{18b}$$

$$\begin{bmatrix} \mathbb{X}_{+}^{h} \\ \mathbb{X}_{-}^{h} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_{+}^{-1}\boldsymbol{b}_{+} - \left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_{+} + \left(\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho}\right)_{+} h_x - \mathrm{e}^{-h_x\boldsymbol{\Lambda}_{+}}\boldsymbol{a}_{+} \\ \mathbb{X}_{-}^{h} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{+}^{\mathrm{h}} + \mathrm{e}^{-h_x\boldsymbol{\Lambda}_{+}}\mathbb{X}_{+}^{0} \\ \mathbb{X}_{-}^{h} \end{bmatrix}, \tag{18b}$$

with

$$\mathbf{A}_{-}^{0} = \boldsymbol{\Lambda}_{-}^{-1}\boldsymbol{b}_{-} - \left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_{-} - \mathrm{e}^{h_x\boldsymbol{\Lambda}_{-}}\boldsymbol{\Lambda}_{-}^{-1}\boldsymbol{b}_{-} + \mathrm{e}^{h_x\boldsymbol{\Lambda}_{-}}\left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_{-} - \mathrm{e}^{-h_x\boldsymbol{\Lambda}_{-}}\left(\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho}\right)\_h_x, \tag{19a}$$

$$\mathbf{A}_{+}^{\mathrm{h}} = \boldsymbol{\Lambda}_{+}^{-1}\boldsymbol{b}_{+} - \left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_{+} + \left(\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho}\right)_{+} h_x - \mathrm{e}^{-h_x\boldsymbol{\Lambda}_{+}}\boldsymbol{\Lambda}_{+}^{-1}\boldsymbol{b}_{+} + \mathrm{e}^{-h_x\boldsymbol{\Lambda}_{+}}\left(\boldsymbol{\Lambda}^{-2}\boldsymbol{\rho}\right)_{+}. \tag{19b}$$

From Equations (17-19) the following expression can be obtained

$$\begin{bmatrix} \mathbb{X}_{+}^{0} \\ \mathbb{X}_{-}^{h} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12}\mathrm{e}^{h_x\boldsymbol{\Lambda}_{-}} \\ \boldsymbol{R}_{21}\mathrm{e}^{-h_x\boldsymbol{\Lambda}_{+}} & \boldsymbol{R}_{22} \end{bmatrix}^{-1} \left(\begin{bmatrix} \boldsymbol{\psi}_{+}^{0} \\ \boldsymbol{\psi}_{-}^{h} \end{bmatrix} - \begin{bmatrix} \boldsymbol{R}_{12}\mathbf{A}_{-}^{0} \\ \boldsymbol{R}_{21}\mathbf{A}_{+}^{h} \end{bmatrix} \right). \tag{20}$$

Recombing the equations and defining the following matrices:

$$\boldsymbol{M}_{1} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12}\mathrm{e}^{h_x\boldsymbol{\Lambda}_{-}} \\ \boldsymbol{R}_{21}\mathrm{e}^{-h_x\boldsymbol{\Lambda}_{+}} & \boldsymbol{R}_{22} \end{bmatrix}, \tag{21a}$$

$$\boldsymbol{M}_{2} = \begin{bmatrix} \boldsymbol{I} - \mathrm{e}^{-h_x\boldsymbol{\Lambda}_{+}} & \\ & \boldsymbol{I} - \mathrm{e}^{h_x\boldsymbol{\Lambda}_{-}} \end{bmatrix}, \tag{21b}$$

$$\boldsymbol{M}_{3-} = \begin{bmatrix} \boldsymbol{I} & \\ & -\mathrm{e}^{h_x\boldsymbol{\Lambda}_{-}} \end{bmatrix}, \; \boldsymbol{M}_{3+} = \begin{bmatrix} \boldsymbol{I} & \\ & \mathrm{e}^{h_x\boldsymbol{\Lambda}_{-}} \end{bmatrix}, \tag{21c}$$

$$\boldsymbol{M}_{4} = \begin{bmatrix} \boldsymbol{R}_{11}\mathrm{e}^{-h_x\boldsymbol{\Lambda}_{+}} & \boldsymbol{R}_{12}\mathrm{e}^{-h_x\boldsymbol{\Lambda}_{-}} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22} \end{bmatrix}, \tag{21d}$$

the system of equations becomes

$$\begin{bmatrix} \mathbf{A}_{+}^{\mathrm{h}} \\ \mathbf{A}_{-}^{0} \end{bmatrix} = \boldsymbol{M}_{2}\boldsymbol{\Lambda}^{-1}\boldsymbol{b} + \left(h_x\boldsymbol{M}_{3-} - \boldsymbol{M}_{2}\boldsymbol{\Lambda}^{-1}\right)\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho}, \tag{22a}$$

$$\begin{bmatrix} \mathbb{X}_{+}^{0} \\ \mathbb{X}_{-}^{\mathrm{h}} \end{bmatrix} = \boldsymbol{M}_{1}^{-1} \begin{bmatrix} \boldsymbol{\psi}_{+}^{0} \\ \boldsymbol{\psi}_{-}^{h} \end{bmatrix} - \boldsymbol{M}_{1}^{-1} \begin{bmatrix} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{21} & \end{bmatrix} \begin{bmatrix} \mathbf{A}_{+}^{\mathrm{h}} \\ \mathbf{A}_{-}^{0} \end{bmatrix}, \tag{22b}$$

$$\begin{bmatrix} \boldsymbol{a}_+ \\ \boldsymbol{a}_- \end{bmatrix} = \boldsymbol{M}_{3+}\boldsymbol{\Lambda}^{-1}\boldsymbol{b} + \left(h_x \begin{bmatrix} \mathbf{0} & \\ & \mathrm{e}^{h_x\boldsymbol{\Lambda}_-} \end{bmatrix} - \boldsymbol{M}_{3+}\boldsymbol{\Lambda}^{-1}\right)\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho} - \boldsymbol{M}_{3+}\begin{bmatrix} \mathbb{X}_+^0 \\ \mathbb{X}_-^{\mathrm{L}} \end{bmatrix}, \tag{22c}$$

$$\begin{bmatrix} \boldsymbol{\psi}_+^h \\ \boldsymbol{\psi}_-^0 \end{bmatrix} = \boldsymbol{R}\boldsymbol{\Lambda}^{-1}\boldsymbol{b} + \left(h_x \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ & \end{bmatrix} - \boldsymbol{R}\boldsymbol{\Lambda}^{-1}\right)\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho} - \boldsymbol{M}_4\begin{bmatrix} \boldsymbol{a}_+ \\ \boldsymbol{a}_- \end{bmatrix}. \tag{22d}$$

Substituting $\boldsymbol{a}$ (22c) into (22d), defining

$$\boldsymbol{M}_5 = \boldsymbol{M}_4\boldsymbol{M}_{3+} = \begin{bmatrix} \boldsymbol{R}_{11}\mathrm{e}^{-h_x\boldsymbol{\Lambda}_+} & \boldsymbol{R}_{12}\mathrm{e}^{-h_x\boldsymbol{\Lambda}_-} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22} \end{bmatrix}\begin{bmatrix} \boldsymbol{I} & \\ & \mathrm{e}^{h_x\boldsymbol{\Lambda}_-} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{11}\mathrm{e}^{-h_x\boldsymbol{\Lambda}_+} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22}\mathrm{e}^{h_x\boldsymbol{\Lambda}_-} \end{bmatrix}, \tag{21f}$$

and simplifying

$$\begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ & \end{bmatrix} - \boldsymbol{M}_4\begin{bmatrix} \mathbf{0} & \\ & \mathrm{e}^{h_x\boldsymbol{\Lambda}_-} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22} \end{bmatrix}\begin{bmatrix} \boldsymbol{I} & \\ & -\mathrm{e}^{h_x\boldsymbol{\Lambda}_-} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22} \end{bmatrix}\boldsymbol{M}_{3-}, \tag{23a}$$

$$(\boldsymbol{R} - \boldsymbol{M}_5) = \begin{bmatrix} \boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22} \end{bmatrix}\begin{bmatrix} \boldsymbol{I} - \mathrm{e}^{-h_x\boldsymbol{\Lambda}_+} & \\ & \boldsymbol{I} - \mathrm{e}^{h_x\boldsymbol{\Lambda}_-} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22} \end{bmatrix}\boldsymbol{M}_2, \tag{23d}$$

the equation becomes

$$\begin{bmatrix} \boldsymbol{\psi}_+^h \\ \boldsymbol{\psi}_-^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22} \end{bmatrix}\boldsymbol{M}_2\boldsymbol{\Lambda}^{-1}\boldsymbol{b} + \begin{bmatrix} \boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22} \end{bmatrix}\left(h_x\boldsymbol{M}_{3-} - \boldsymbol{M}_2\boldsymbol{\Lambda}^{-1}\right)\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho} + \boldsymbol{M}_5\begin{bmatrix} \mathbb{X}_+^0 \\ \mathbb{X}_-^{\mathrm{h}} \end{bmatrix}. \tag{24}$$

Substituting $\begin{bmatrix} \mathbb{X}_+^0 \\ \mathbb{X}_-^{\mathrm{h}} \end{bmatrix}$ (22b) into Equation (24) the equation becomes

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\psi}_+^h \\ \boldsymbol{\psi}_-^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22} \end{bmatrix}\boldsymbol{M}_2\boldsymbol{\Lambda}^{-1}\boldsymbol{b} + \begin{bmatrix} \boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22} \end{bmatrix}\left(h_x\boldsymbol{M}_{3-} - \boldsymbol{M}_2\boldsymbol{\Lambda}^{-1}\right)\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho} \\ +\boldsymbol{M}_5\boldsymbol{M}_1^{-1}\begin{bmatrix} \boldsymbol{\psi}_+^0 \\ \boldsymbol{\psi}_-^h \end{bmatrix} - \boldsymbol{M}_5\boldsymbol{M}_1^{-1}\begin{bmatrix} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{21} & \end{bmatrix}\begin{bmatrix} \mathbf{A}_+^h \\ \mathbf{A}_-^0 \end{bmatrix}. \end{aligned} \tag{25}$$

Substituting $\begin{bmatrix} \mathbf{A}_+^h \\ \mathbf{A}_-^0 \end{bmatrix}$ (22a) into (25) and defining

$$\boldsymbol{M}_6 = \begin{bmatrix} \boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22} \end{bmatrix} - \boldsymbol{M}_5\boldsymbol{M}_1^{-1}\begin{bmatrix} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{21} & \end{bmatrix}, \tag{21g}$$

the equation becomes

$$\begin{bmatrix} \boldsymbol{\psi}_+^h \\ \boldsymbol{\psi}_-^0 \end{bmatrix} = \boldsymbol{M}_5\boldsymbol{M}_1^{-1}\begin{bmatrix} \boldsymbol{\psi}_+^0 \\ \boldsymbol{\psi}_-^h \end{bmatrix} + \boldsymbol{M}_6\boldsymbol{M}_2\boldsymbol{\Lambda}^{-1}\boldsymbol{b} + \boldsymbol{M}_6\left(h_x\boldsymbol{M}_{3-} - \boldsymbol{M}_2\boldsymbol{\Lambda}^{-1}\right)\boldsymbol{\Lambda}^{-1}\boldsymbol{\rho}, \tag{26}$$

which can be written

$$\boldsymbol{\psi_x} = \boldsymbol{B_x}\boldsymbol{\psi_{Bx}} + \boldsymbol{R_A}\boldsymbol{b} + \boldsymbol{R_B}\boldsymbol{\rho}, \tag{27}$$

where

$$\boldsymbol{B_x} = \boldsymbol{M_5}\boldsymbol{M_1^{-1}} = \boldsymbol{M_4}\boldsymbol{M_{3+}}\boldsymbol{M_1^{-1}}, \tag{28a}$$

$$\boldsymbol{R_A} = \boldsymbol{M_6}\boldsymbol{M_2}\boldsymbol{\Lambda^{-1}}, \tag{28b}$$

$$\boldsymbol{R_B} = \boldsymbol{M_6}\left(h_x\boldsymbol{M_{3-}} - \boldsymbol{M_2}\boldsymbol{\Lambda^{-1}}\right)\boldsymbol{\Lambda^{-1}} = (h_x\boldsymbol{M_6}\boldsymbol{M_{3-}} - \boldsymbol{R_A})\boldsymbol{\Lambda^{-1}}, \tag{28c}$$

and $\boldsymbol{\psi_{Bx}}$ denotes boundary conditions in the x direction. Note that $\boldsymbol{R_B}$ is not needed for the constant transverse leakage formulation. Substituting $\boldsymbol{b}$ (12b) and $\boldsymbol{\rho}$ (12c) and subsequently substituting $\boldsymbol{q_x}$ (8) noting Equation (5c),

$$\boldsymbol{\psi_x} = \boldsymbol{B_x}\boldsymbol{\psi_{Bx}} + \boldsymbol{R_A}\boldsymbol{R^{-1}}\upmu_{\mathrm{x}}^{-1}\Gamma^{-1}Q\mathbf{1} - \boldsymbol{R_A}\boldsymbol{R^{-1}}\upmu_{\mathrm{x}}^{-1}\boldsymbol{\eta_x}\mathbb{L}_x + (h_x2^{-1}\boldsymbol{R_A} - \boldsymbol{R_B})\boldsymbol{R^{-1}}\upmu_{\mathrm{x}}^{-1}\boldsymbol{\eta_x}\mathbf{r}_{\mathbf{x}}, \tag{29}$$

which can be written

$$\boldsymbol{\psi_x} = \boldsymbol{A_x} + \boldsymbol{B_x}\boldsymbol{\psi_{Bx}} + \boldsymbol{R_C}\mathbb{L}_x + \boldsymbol{R_D}\mathbf{r}_{\mathbf{x}}, \tag{30}$$

where

$$\boldsymbol{A_x} = \boldsymbol{R_A}\boldsymbol{R^{-1}}\upmu_{\mathrm{x}}^{-1}\Gamma^{-1}Q\mathbf{1}, \tag{31a}$$

$$\boldsymbol{R_C} = -\boldsymbol{R_A}\boldsymbol{R^{-1}}\upmu_{\mathrm{x}}^{-1}\boldsymbol{\eta_x}, \tag{31b}$$

$$\boldsymbol{R_D} = (h_x2^{-1}\boldsymbol{R_A} - \boldsymbol{R_B})\boldsymbol{R^{-1}}\upmu_{\mathrm{x}}^{-1}\boldsymbol{\eta_x} = -h_x2^{-1}\boldsymbol{R_C} - \boldsymbol{R_B}\boldsymbol{R^{-1}}\upmu_{\mathrm{x}}^{-1}\boldsymbol{\eta_x}. \tag{31c}$$

The cell top and bottom surface flux vectors, $\boldsymbol{\psi}_{i,j+\frac{1}{2}}$ and $\boldsymbol{\psi}_{i,j-\frac{1}{2}}$, respectively, calculated from the transverse integrated flux in the y-direction are split into cell surface incoming and outgoing flux vectors, $\boldsymbol{\psi_{By}}$ and $\boldsymbol{\psi_y}$, respectively, to give the transverse leakage as

$$\mathbb{L}_x = h_y^{-1}\boldsymbol{T_{MN}}\left(\boldsymbol{\psi}_{i,j+\frac{1}{2}} - \boldsymbol{\psi}_{i,j-\frac{1}{2}}\right) = h_y^{-1}\boldsymbol{T_{MN}}\left(\begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}\boldsymbol{\psi_{By}} - \begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}\boldsymbol{\psi_y}\right). \tag{32}$$

The mapping matrix $\boldsymbol{T_{MN}}$ is introduced to reorder the quadrature points because, as noted previously, the quadrature points are ordered with respect to μ in the x-direction and with respect to η in the y-direction. An equivalent matrix $\boldsymbol{T_{NM}}$ is defined to perform the reverse mapping. Equation (32) can be substituted into Equation (30) with $\mathbf{r}_{\mathbf{x}}$ (7d) to give

$$\boldsymbol{\psi_x} = \boldsymbol{A_x} + \boldsymbol{B_x}\boldsymbol{\psi_{Bx}} + \boldsymbol{R_C} h_y^{-1}\boldsymbol{T_{MN}}\left(\begin{bmatrix} -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}\boldsymbol{\psi_{By}} - \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}\boldsymbol{\psi_y}\right) + \boldsymbol{R_D} h_y^{-1} h_x^{-1}\boldsymbol{\alpha_x}. \tag{33}$$

The first and second terms correspond to the homogenous and particular solution, respectively, of the 1D analytical solution. The third and fourth terms correspond to the constant, and linear transverse leakage distribution, respectively, of the 2D nodal solution. Substitution of $\boldsymbol{\alpha_x}$ and the subsequent derivation is algebraically straightforward but results in long expressions that obscure the more important aspects of the remaining step.

Derivation of ANDO-C: Constant Transverse Leakage

Defining

$$\boldsymbol{C_x} = h_y^{-1}\boldsymbol{R_C}\boldsymbol{T_{MN}}\begin{bmatrix} -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{34}$$

and neglecting the fourth term in Equation (33), the solution approximating the transverse leakage as constant is

$$\boldsymbol{\psi_x} = \boldsymbol{A_x} + \boldsymbol{B_x}\boldsymbol{\psi_{Bx}} + \boldsymbol{C_x}\boldsymbol{\psi_{By}} - \boldsymbol{C_x}\boldsymbol{\psi_y}. \tag{35}$$

An equivalent expression can be found for the solution along the y as

$$\boldsymbol{\psi_y} = \boldsymbol{A_y} + \boldsymbol{B_y}\boldsymbol{\psi_{By}} + \boldsymbol{C_y}\boldsymbol{\psi_{Bx}} - \boldsymbol{C_y}\boldsymbol{\psi_x}. \tag{36}$$

The method can at this point be implemented and will converge successfully. However, as will be shown in Chapter 3, the method is prone to oscillation and slow convergence. The final simple but non-obvious step is to perform a local leakage balance with Equations (35) and (36) by first combining into a single expression

$$\begin{bmatrix} \boldsymbol{\psi_x} \\ \boldsymbol{\psi_y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A_x} \\ \boldsymbol{A_y} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B_x} & \\ & \boldsymbol{B_y} \end{bmatrix}\begin{bmatrix} \boldsymbol{\psi_{Bx}} \\ \boldsymbol{\psi_{By}} \end{bmatrix} + \begin{bmatrix} & \boldsymbol{C_x} \\ \boldsymbol{C_y} & \end{bmatrix}\begin{bmatrix} \boldsymbol{\psi_{Bx}} \\ \boldsymbol{\psi_{By}} \end{bmatrix} - \begin{bmatrix} & \boldsymbol{C_x} \\ \boldsymbol{C_y} & \end{bmatrix}\begin{bmatrix} \boldsymbol{\psi_x} \\ \boldsymbol{\psi_y} \end{bmatrix}, \tag{37}$$

then moving the unknown outgoing flux to the left-hand side of the equation, combining the second and third term,

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{C_x} \\ \boldsymbol{C_y} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi_x} \\ \boldsymbol{\psi_y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A_x} \\ \boldsymbol{A_y} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B_x} & \boldsymbol{C_x} \\ \boldsymbol{C_y} & \boldsymbol{B_y} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi_{Bx}} \\ \boldsymbol{\psi_{By}} \end{bmatrix}, \tag{38}$$

and finally solving for the outgoing flux with

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{C_x} \\ \boldsymbol{C_y} & \boldsymbol{I} \end{bmatrix}^{-1}, \tag{39a}$$

$$\boldsymbol{A} = \boldsymbol{D} \begin{bmatrix} \boldsymbol{A_x} \\ \boldsymbol{A_y} \end{bmatrix}, \tag{39b}$$

$$\boldsymbol{B} = \boldsymbol{D} \begin{bmatrix} \boldsymbol{B_x} & \boldsymbol{C_x} \\ \boldsymbol{C_y} & \boldsymbol{B_y} \end{bmatrix}, \tag{39c}$$

to give

$$\boldsymbol{\psi_{i,j}^{out}} = \boldsymbol{A} + \boldsymbol{B\psi_{i,j}^{in}}. \tag{40}$$

One of the benefits of the ANDO method is that the column vector $\boldsymbol{A}$ and matrix $\boldsymbol{B}$ can be precomputed outside the iteration loop based only on the quadrature set, material properties, and cell dimensions, making the implementation of the ANDO method highly computationally efficient in terms of FLOPS. The final closed form solution (40) simultaneously solves for the outgoing flux on all four cell faces from the incoming flux on all four cell faces with no spatial truncation error within the cell. Further, the column vector $\boldsymbol{A}$ directly gives the outgoing flux for vacuum boundary conditions whereas solving Equations (35) and (36) iteratively on a single cell could require 100's of iterations for a highly diffuse region.

Derivation of ANDO-L: Linear Transverse Leakage

A similar expression can be found under the approximation of a linear transverse leakage distribution. The linear distribution along the top and bottom faces of the cell can be calculated from the downwind flux on the top and bottom faces of adjacent cells, respectively, as

$$\psi_{i,j+\frac{1}{2}}(x,\,\Omega_m) = \begin{cases} \bar{\psi}_{i,j+\frac{1}{2}}(\Omega_m) + \frac{2}{h_x}\left(x - \frac{h_x}{2}\right)\left[\psi_{i+\frac{1}{2},j+\frac{1}{2}}(\Omega_m) - \bar{\psi}_{i,j+\frac{1}{2}}(\Omega_m)\right], & \mu_m > 0 \\ \bar{\psi}_{i,j+\frac{1}{2}}(\Omega_m) + \frac{2}{h_x}\left(x - \frac{h_x}{2}\right)\left[\bar{\psi}_{i,j+\frac{1}{2}}(\Omega_m) - \psi_{i-\frac{1}{2},j+\frac{1}{2}}(\Omega_m)\right], & \mu_m < 0 \end{cases}, \tag{41a}$$

$$\psi_{i,j-\frac{1}{2}}(x,\,\Omega_m) = \begin{cases} \bar{\psi}_{i,j-\frac{1}{2}}(\Omega_m) + \frac{2}{h_x}\left(x - \frac{h_x}{2}\right)\left[\psi_{i+\frac{1}{2},j-\frac{1}{2}}(\Omega_m) - \bar{\psi}_{i,j-\frac{1}{2}}(\Omega_m)\right], & \mu_m > 0 \\ \bar{\psi}_{i,j-\frac{1}{2}}(\Omega_m) + \frac{2}{h_x}\left(x - \frac{h_x}{2}\right)\left[\bar{\psi}_{i,j-\frac{1}{2}}(\Omega_m) - \psi_{i-\frac{1}{2},j-\frac{1}{2}}(\Omega_m)\right], & \mu_m < 0 \end{cases}, \tag{41b}$$

Substituting Equations (41) into Equation (4) and rearranging into the form of Equation (5) the slope $\boldsymbol{\alpha}_{\mathbf{x}}$ is

$$\begin{aligned} \boldsymbol{\alpha}_{\mathbf{x}} = {} & \begin{bmatrix} -\boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i,j+\frac{1}{2}} + \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & -\boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i,j-\frac{1}{2}} + \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \mathbf{0} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i+1,j+\frac{1}{2}} \\ & + \begin{bmatrix} -\boldsymbol{I} & 0 \\ 0 & \mathbf{0} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i+1,j-\frac{1}{2}} + \begin{bmatrix} \mathbf{0} & 0 \\ 0 & -\boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i-1,j+\frac{1}{2}} + \begin{bmatrix} \mathbf{0} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i-1,j-\frac{1}{2}}. \end{aligned} \tag{42}$$

The flux on the top and bottom cell faces can be split into incoming and outgoing vectors to give

$$\begin{aligned} \boldsymbol{\alpha}_{\mathbf{x}} = {} & \begin{bmatrix} -\boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \left(\begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{\psi_{By}} - \begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{\psi_{y}} \right) + \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \mathbf{0} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i+1,j+\frac{1}{2}} \\ & + \begin{bmatrix} -\boldsymbol{I} & 0 \\ 0 & \mathbf{0} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i+1,j-\frac{1}{2}} + \begin{bmatrix} \mathbf{0} & 0 \\ 0 & -\boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i-1,j+\frac{1}{2}} + \begin{bmatrix} \mathbf{0} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i-1,j-\frac{1}{2}}. \end{aligned} \tag{43}$$

Substituting $\boldsymbol{\alpha}_{\mathbf{x}}$ (43) into Equation (33)

$$\begin{aligned} \boldsymbol{\psi}^{x}_{i,j} = {} & \boldsymbol{A_x} + \boldsymbol{B_x} \boldsymbol{\psi}^{Bx}_{i,j} + \boldsymbol{R_C} h_y^{-1} \boldsymbol{T_{MN}} \left(\begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{\psi_{By}} - \begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{\psi_{y}} \right) \\ & + \boldsymbol{R_D} h_y^{-1} h_x^{-1} \begin{bmatrix} -\boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \left(\begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{\psi_{By}} - \begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{\psi_{y}} \right) \\ & + \boldsymbol{R_D} h_y^{-1} h_x^{-1} \begin{pmatrix} \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \mathbf{0} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i+1,j+\frac{1}{2}} + \begin{bmatrix} -\boldsymbol{I} & 0 \\ 0 & \mathbf{0} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i+1,j-\frac{1}{2}} + \\ \begin{bmatrix} \mathbf{0} & 0 \\ 0 & -\boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i-1,j+\frac{1}{2}} + \begin{bmatrix} \mathbf{0} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \boldsymbol{\psi}_{i-1,j-\frac{1}{2}} \end{pmatrix}. \end{aligned} \tag{44}$$

Defining

$$\boldsymbol{C^L_x} = \left(h_y^{-1} \boldsymbol{R_C} + h_x^{-1} h_y^{-1} \boldsymbol{R_D} \begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \right) \boldsymbol{T_{MN}} \begin{bmatrix} -\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}, \tag{45}$$

which is the same as $\boldsymbol{C_x}$ in Equation (34) with an additional term from the slope of the linear distribution, and

$$\boldsymbol{L_{1x}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{46a}$$

$$\boldsymbol{L_{2x}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{46b}$$

$$\boldsymbol{L_{4x}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{46c}$$

$$\boldsymbol{L_{5x}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \tag{46d}$$

$$\boldsymbol{L_{6x}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{46e}$$

$$\boldsymbol{L_{8x}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \tag{46f}$$

Equations (45-46) can be substituted into Equation (44) to give

$$\begin{aligned} \boldsymbol{\psi_x} = \boldsymbol{A_x} + \boldsymbol{B_x}\boldsymbol{\psi_{Bx}} + \boldsymbol{C_x^L}\boldsymbol{\psi_{By}} - \boldsymbol{C_x^L}\boldsymbol{\psi_y} + \boldsymbol{L_{1x}}\boldsymbol{\psi}_{y,i+1,j} + \boldsymbol{L_{2x}}\boldsymbol{\psi}_{y,i+1,j+1} \\ +\boldsymbol{L_{4x}}\boldsymbol{\psi}_{y,i-1,j+1} + \boldsymbol{L_{5x}}\boldsymbol{\psi}_{y,i-1,j} + \boldsymbol{L_{6x}}\boldsymbol{\psi}_{y,i-1,j-1} + \boldsymbol{L_{8x}}\boldsymbol{\psi}_{y,i+1,j-1}. \end{aligned} \tag{47}$$

An equivalent expression can be found for the solution along the y as

$$\begin{aligned} \boldsymbol{\psi_y} = \boldsymbol{A_y} + \boldsymbol{B_y}\boldsymbol{\psi_{By}} + \boldsymbol{C_y^l}\boldsymbol{\psi_{Bx}} - \boldsymbol{C_y^L}\boldsymbol{\psi_x} + \boldsymbol{L_{2y}}\boldsymbol{\psi}_{x,i+1,j+1} + \boldsymbol{L_{3y}}\boldsymbol{\psi}_{x,i,j+1} \\ +\boldsymbol{L_{4y}}\boldsymbol{\psi}_{x,i-1,j+1} + \boldsymbol{L_{6y}}\boldsymbol{\psi}_{x,i-1,j-1} + \boldsymbol{L_{7y}}\boldsymbol{\psi}_{x,i,j-1} + \boldsymbol{L_{8y}}\boldsymbol{\psi}_{x,i+1,j-1}, \end{aligned} \tag{48}$$

with

$$\boldsymbol{L_{2y}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{49a}$$

$$\boldsymbol{L_{3y}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{49b}$$

$$\boldsymbol{L_{4y}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \tag{49c}$$

$$\boldsymbol{L_{6y}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{49d}$$

$$\boldsymbol{L_{7y}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \tag{49e}$$

$$\boldsymbol{L_{8y}} = \boldsymbol{R_D} h_x^{-1} h_y^{-1} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{T_{MN}} \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{49f}$$

Combining Equations (47) and (48) as in Equation (37) and defining similar expressions to equations (39),

$$\boldsymbol{D^L} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{C_x^L} \\ \boldsymbol{C_y^L} & \boldsymbol{I} \end{bmatrix}^{-1}, \tag{50a}$$

$$\boldsymbol{A^L} = \boldsymbol{D^L} \begin{bmatrix} \boldsymbol{A_x} \\ \boldsymbol{A_y} \end{bmatrix}, \qquad \boldsymbol{B^L} = \boldsymbol{D^L} \begin{bmatrix} \boldsymbol{B_x} & \boldsymbol{C_x^L} \\ \boldsymbol{C_y^L} & \boldsymbol{B_y} \end{bmatrix}, \tag{50b,c}$$

$$\boldsymbol{E_1} = \boldsymbol{D^L} \begin{bmatrix} & \boldsymbol{L_{1x}} \end{bmatrix}, \qquad \boldsymbol{E_2} = \boldsymbol{D^L} \begin{bmatrix} & \boldsymbol{L_{2x}} \\ \boldsymbol{L_{2y}} & \end{bmatrix}, \tag{50d,e}$$

$$\boldsymbol{E_3} = \boldsymbol{D^L} \begin{bmatrix} \boldsymbol{L_{3y}} & \end{bmatrix}, \qquad \boldsymbol{E_4} = \boldsymbol{D^L} \begin{bmatrix} & \boldsymbol{L_{4x}} \\ \boldsymbol{L_{4y}} & \end{bmatrix}, \tag{50f,g}$$

$$\boldsymbol{E_5} = \boldsymbol{D^L} \begin{bmatrix} & \boldsymbol{L_{5x}} \end{bmatrix}, \qquad \boldsymbol{E_6} = \boldsymbol{D^L} \begin{bmatrix} & \boldsymbol{L_{6x}} \\ \boldsymbol{L_{6y}} & \end{bmatrix}, \tag{50h,i}$$

$$\boldsymbol{E_7} = \boldsymbol{D^L} \begin{bmatrix} \boldsymbol{L_{7y}} & \end{bmatrix}, \qquad \boldsymbol{E_8} = \boldsymbol{D^L} \begin{bmatrix} & \boldsymbol{L_{8x}} \\ \boldsymbol{L_{8y}} & \end{bmatrix}, \tag{50j,k}$$

to give

$$\begin{aligned} \boldsymbol{\psi_{i,j}^{out}} &= \boldsymbol{A^L} + \boldsymbol{B^L}\boldsymbol{\psi_{i,j}^{in}} + \boldsymbol{E_1}\boldsymbol{\psi}_{i+1,j} + \boldsymbol{E_2}\boldsymbol{\psi}_{i+1,j+1} + \boldsymbol{E_3}\boldsymbol{\psi}_{i,j+1} \\ &+ \boldsymbol{E_4}\boldsymbol{\psi}_{i-1,j+1} + \boldsymbol{E_5}\boldsymbol{\psi}_{i-1,j} + \boldsymbol{E_6}\boldsymbol{\psi}_{i-1,j-1} + \boldsymbol{E_7}\boldsymbol{\psi}_{i,j-1} + \boldsymbol{E_8}\boldsymbol{\psi}_{i+1,j-1}. \end{aligned} \tag{51}$$

Equation (51) has a similar form to Equation (40) except for the additional terms for flux in adjacent cells that is used to calculate the transverse leakage. Note that the matrices $\boldsymbol{E}$ are sparse and the matrix vector multiplications in Equation (51) can be implemented more efficiently. Note also that $\boldsymbol{A^L}$ and $\boldsymbol{B^L}$ in Equations (50a,b) have different values than $\boldsymbol{A}$ and $\boldsymbol{B}$ in Equations (39a,b) due to the additional linear leakage correction applied to $\boldsymbol{C_x^L}$ in Equation (45) appearing in the inverse matrix in Equation (50a). The matrices defined in Equations (50) can be precomputed outside the iteration loop for computational efficiency.

Derivation of ANDO-R: Cell Average Angular Flux Reconstruction

An expression for the average angular flux can be found by a similar derivation. Equation (14) gives the analytical nodal solution for the transverse integrated equation along one dimension for a linear transverse leakage approximation. For a constant transverse leakage, the equation becomes

$$\mathbb{X} = \boldsymbol{\Lambda}^{-\mathbf{1}}\boldsymbol{b_x} - e^{-x\boldsymbol{\Lambda}}\boldsymbol{a_x}. \tag{52}$$

Integrating over x

$$\overline{\mathbb{X}} = \boldsymbol{\Lambda}^{-\mathbf{1}}\boldsymbol{b_x} + h_x^{-1}\boldsymbol{\Lambda}^{-\mathbf{1}}(e^{-h_x\boldsymbol{\Lambda}} - \boldsymbol{I})\boldsymbol{a_x}, \tag{53}$$

the average angular flux is

$$\overline{\boldsymbol{\psi}}_{\mathrm{x}} = \boldsymbol{R}\overline{\mathbb{X}} = \boldsymbol{R\Lambda}^{-\mathbf{1}}\boldsymbol{b_x} - \boldsymbol{M_{4R}}\begin{bmatrix}\boldsymbol{a}_+\\ \boldsymbol{a}_-\end{bmatrix}, \tag{54}$$

with

$$\boldsymbol{M_{4R}} = h_x^{-1}\boldsymbol{R\Lambda}^{-\mathbf{1}}\left(\boldsymbol{I} - e^{-h_x\boldsymbol{\Lambda}}\right). \tag{55}$$

Equation (54) is of a similar form as Equation (14) and can be solved with Equations (22a-22c) in an analogous fashion. Defining solution matrices similar to Equations (21c, 21f, 28a-b, 31a-b, and 34) and using an R subscript to denote reconstruction;

$$\boldsymbol{M_{5R}} = \boldsymbol{M_{4R}}\boldsymbol{M}_{3+}, \tag{56a}$$

$$\boldsymbol{M_{6R}} = \begin{bmatrix}\boldsymbol{R}_{11} & \\ & \boldsymbol{R}_{22}\end{bmatrix} - \boldsymbol{M}_{5R}\boldsymbol{M}_1^{-1}\begin{bmatrix} & \boldsymbol{R}_{12}\\ \boldsymbol{R}_{21} & \end{bmatrix}, \tag{56b}$$

$$\boldsymbol{B_{xR}} = \boldsymbol{M_{5R}}\boldsymbol{M}_1^{-1}, \tag{56c}$$

$$\boldsymbol{R_{AR}} = \boldsymbol{M_{6R}}\boldsymbol{M}_2\boldsymbol{\Lambda}^{-\mathbf{1}} + (\boldsymbol{M}_5 - \boldsymbol{M_{5R}})\boldsymbol{\Lambda}^{-\mathbf{1}}, \tag{56d}$$

$$\boldsymbol{A_{xR}} = \boldsymbol{R_{AR}}\boldsymbol{R}^{-\mathbf{1}}\boldsymbol{\mu}_{\mathbf{x}}^{-1}\Gamma^{-1}Q\mathbf{1}, \tag{56e}$$

$$\boldsymbol{R_{CR}} = -\boldsymbol{R_{AR}}\boldsymbol{R}^{-\mathbf{1}}\boldsymbol{\mu}_{\mathbf{x}}^{-1}\boldsymbol{\eta_x}, \tag{56f}$$

$$\boldsymbol{C_{xR}} = \boldsymbol{h_y^{-1}}\boldsymbol{R_{CR}}\boldsymbol{T_{MN}}\begin{bmatrix}-\boldsymbol{I} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{I}\end{bmatrix}, \tag{56g}$$

the solution becomes

$$\overline{\boldsymbol{\psi}}_{\boldsymbol{x}} = \boldsymbol{A}_{\boldsymbol{xR}} + \boldsymbol{B}_{\boldsymbol{xR}}\boldsymbol{\psi}_{\boldsymbol{Bx}} + \boldsymbol{C}_{\boldsymbol{xR}}\boldsymbol{\psi}_{\boldsymbol{By}} - \boldsymbol{C}_{\boldsymbol{xR}}\boldsymbol{\psi}_{\boldsymbol{y}}. \tag{57}$$

From which the average scalar flux can be readily obtained

$$\boldsymbol{\phi} = \boldsymbol{W}_{\boldsymbol{x}}\overline{\boldsymbol{\psi}_{\boldsymbol{x}}} = \boldsymbol{W}_{\boldsymbol{y}}\overline{\boldsymbol{\psi}_{\boldsymbol{y}}}. \tag{58}$$

Note that $\overline{\boldsymbol{\psi}_{\boldsymbol{x}}} = \overline{\boldsymbol{\psi}_{\boldsymbol{y}}}$ only at convergence. The above expression is equivalent to reconstruction using a simple particle balance equation and is therefore no more accurate. Further, the ANDO-Reconstruction method is less computationally efficient than a particle balance reconstruction. However, the above expression may be useful in derivations employing the average angular flux, such as the LL1 transverse leakage approximation presented next.

Derivation of ANDO-LL1: LL1 Transverse Leakage

Early work on nodal methods for the transport equation relied on calculating flux moments on the cell interior and on the cell face. Several linear approximations were proposed by Walters and O'Dell in 1981 [33] to improve the accuracy and computational efficiency of the analytical nodal method. The LL1 Leakage approximation approximates both the cell interior and cell exterior angular flux with a linear distribution. In the LL1 approximation the slope of the linear distribution is determined from the cell average angular flux and angular flux on the outgoing cell face. The advantage of this approach is that a linear distribution can be establish on the cell face without additional computations to determine the distribution from adjacent cells, potentially improving the accuracy of the ANDO-Constant method without the additional cost of the ANDO-Linear method. Following the work of Walters and O-Dell [33] the angular flux distribution on the top cell face is approximated as

$$\boldsymbol{\psi}_{i,j+\frac{1}{2}}(x) = \overline{\boldsymbol{\psi}}_{i,j+\frac{1}{2}} + 2h_x^{-1}(x - h_x2^{-1})\,\boldsymbol{\alpha}_{\boldsymbol{x}}. \tag{59}$$

With the slope $\boldsymbol{\alpha}$ defined as

$$\boldsymbol{\alpha_x} = \begin{cases} \boldsymbol{\psi}_{i+\frac{1}{2},j} - \overline{\boldsymbol{\psi}}_{i,j}, & \boldsymbol{\mu} > \mathbf{0} \\ \overline{\boldsymbol{\psi}}_{i,j} - \boldsymbol{\psi}_{i-\frac{1}{2},j}, & \boldsymbol{\mu} < \mathbf{0} \end{cases}, \quad \boldsymbol{\alpha_y} = \begin{cases} \boldsymbol{\psi}_{i,j+\frac{1}{2}} - \overline{\boldsymbol{\psi}}_{i,j}, & \boldsymbol{\eta} > \mathbf{0} \\ \overline{\boldsymbol{\psi}}_{i,j} - \boldsymbol{\psi}_{i,j-\frac{1}{2}}, & \boldsymbol{\eta} < \mathbf{0} \end{cases}. \tag{60}$$

The slope of the linear leakage distribution can be determined based on the cell average angular flux value and the outgoing angular flux value. The advantage of this approach was to eliminate the need for adjacent cells to determine the linear distribution. The piecewise expression (60) can be written as a single equation to give

$$\boldsymbol{\alpha_x} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} \end{bmatrix} \boldsymbol{T_{NM}} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} \end{bmatrix} \boldsymbol{\psi_x} - \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} \end{bmatrix} \boldsymbol{T_{NM}} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} \end{bmatrix} \overline{\boldsymbol{\psi}}, \tag{61}$$

where $\overline{\boldsymbol{\psi}}$ can be determined from ANDO-R (57). The slope $\boldsymbol{\alpha_x}$ can then be substituted into equation (33) to give

$$\boldsymbol{\psi_x} = \boldsymbol{A_x} + \boldsymbol{R_\alpha A_{xR}} + \boldsymbol{B_x \psi_{Bx}} + \boldsymbol{R_\alpha B_{xR} \psi_{Bx}} + \boldsymbol{R_C} \mathbb{L}_x + \boldsymbol{R_\alpha R_{CR}} \mathbb{L}_{\boldsymbol{x}} - \boldsymbol{R_\alpha \psi_x}, \tag{62}$$

with

$$\boldsymbol{R_\alpha} = \mathbf{2} \boldsymbol{h_y^{-1} h_x^{-1} R_D} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} \end{bmatrix} \boldsymbol{T_{NM}} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I} \end{bmatrix}. \tag{63}$$

The coefficients are now the same as equations (28a, 31a, 34) from the derivation of ANDO-C with an additional higher order term:

$$\boldsymbol{A'_x} = \boldsymbol{A_x} + \boldsymbol{R_\alpha A_{xR}}, \tag{64a}$$

$$\boldsymbol{B'_x} = \boldsymbol{B_x} + \boldsymbol{R_\alpha B_{xR}}, \tag{64b}$$

$$\boldsymbol{C'_x} = \boldsymbol{C_x} + \boldsymbol{R_\alpha C_{xR}}, \tag{64c}$$

$$\boldsymbol{D'_x} = \boldsymbol{R_\alpha}. \tag{64d}$$

Finally, the local leakage balance is performed as in equations (39a-c),

$$\boldsymbol{D_{LL1}} = \begin{bmatrix} \boldsymbol{I} + \boldsymbol{D'_x} & \boldsymbol{C'_x} \\ \boldsymbol{C'_y} & \boldsymbol{I} + \boldsymbol{D'_y} \end{bmatrix}^{-\mathbf{1}}, \tag{65a}$$

$$\boldsymbol{A_{LL1}} = \boldsymbol{D} \begin{bmatrix} \boldsymbol{A'_x} \\ \boldsymbol{A'_y} \end{bmatrix}, \tag{65b}$$

$$\boldsymbol{B_{LL1}} = \boldsymbol{D}\begin{bmatrix} \boldsymbol{B'_x} & \boldsymbol{C'_x} \\ \boldsymbol{C'_y} & \boldsymbol{B'_y} \end{bmatrix}, \tag{65c}$$

to give

$$\boldsymbol{\psi_{i,j}^{out}} = \boldsymbol{A_{LL1}} + \boldsymbol{B_{LL1}}\boldsymbol{\psi_{i,j}^{in}}. \tag{66}$$

Equation (66) is the same as Equation (40) for ANDO-C except that the coefficients in the vector **A** and matrix **B** are slightly modified. However, the matrix coefficients are less accurate than the ANDO-Constant method, as will be shown later. The coefficients were also found to be less accurate through numerical experimentation.

Derivation of ANDO-C$_{2x2}$: Closed Form Solution on a Heterogenous Grid

Equation (40) gives the closed form solution of a single cell. The outgoing flux on all four cell faces can be solved exactly and simultaneously based only on the incoming flux on all four cell faces without determining any additional coefficients or solving any additional equations. This further allows for a closed form solution to be easily obtained on a 2x2 heterogenous grid. Defining the indexing matrices

$$\boldsymbol{U_1} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{U_2} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{67a,b}$$

$$\boldsymbol{U_3} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{U_4} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{67c,d}$$

$$\boldsymbol{U_{13}} = \boldsymbol{U_1} + \boldsymbol{U_3}, \qquad \boldsymbol{U_{14}} = \boldsymbol{U_1} + \boldsymbol{U_4}, \tag{67e,f}$$

$$\boldsymbol{U_{23}} = \boldsymbol{U_2} + \boldsymbol{U_3}, \qquad \boldsymbol{U_{24}} = \boldsymbol{U_2} + \boldsymbol{U_4}, \tag{67g,h}$$

and defining matrices to solve for the interior cell faces, denoted with a prime, and exterior cell faces separately,

$$\boldsymbol{T_1} = \boldsymbol{U_{13}}\boldsymbol{B_1}\boldsymbol{U_{13}} + \boldsymbol{U_{24}}\boldsymbol{B_4}\boldsymbol{U_{24}}, \qquad \boldsymbol{T'_1} = \boldsymbol{U_{24}}\boldsymbol{B_1}\boldsymbol{U_{13}} + \boldsymbol{U_{13}}\boldsymbol{B_4}\boldsymbol{U_{24}}, \tag{68a,b}$$

$$\boldsymbol{T_2} = \boldsymbol{U_{23}B_2U_{23}} + \boldsymbol{U_{14}B_3U_{14}}, \qquad \boldsymbol{T'_2} = \boldsymbol{U_{14}B_2U_{23}} + \boldsymbol{U_{23}B_3U_{14}}, \tag{68c,d}$$

$$\boldsymbol{T_3} = \boldsymbol{U_{23}B_2U_{14}} + \boldsymbol{U_{14}B_3U_{23}}, \qquad \boldsymbol{T'_3} = \boldsymbol{U_{14}B_2U_{14}} + \boldsymbol{U_{23}B_3U_{23}}. \tag{68e,f}$$

$$\boldsymbol{T_4} = \boldsymbol{U_{13}B_1U_{24}} + \boldsymbol{U_{24}B_4U_{13}}, \qquad \boldsymbol{T'_4} = \boldsymbol{U_{24}B_1U_{24}} + \boldsymbol{U_{13}B_4U_{13}}, \tag{68g,h}$$

After some algebra, the equations can be combined into a single vector for interior flux and a single vector for outgoing flux with

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{T'_4} \\ \boldsymbol{T'_3} & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{J'} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{T_4} \\ -\boldsymbol{T_3} & \boldsymbol{I} \end{bmatrix}^{-1}, \tag{69a,b}$$

The new coefficients for the 2x2 cell become

$$\boldsymbol{A'_{2\times2}} = \boldsymbol{J'}\left(\begin{bmatrix} \boldsymbol{U_{13}} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{A_1} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U_{23}} \end{bmatrix}\boldsymbol{A_2} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U_{14}} \end{bmatrix}\boldsymbol{A_3} + \begin{bmatrix} \boldsymbol{U_{24}} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{A_4}\right), \tag{70a}$$

$$\boldsymbol{A_{2\times2}} = \begin{bmatrix} \boldsymbol{U_{24}} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{A_1} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U_{14}} \end{bmatrix}\boldsymbol{A_2} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U_{23}} \end{bmatrix}\boldsymbol{A_3} + \begin{bmatrix} \boldsymbol{U_{13}} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{A_4} + \boldsymbol{JA'_{2\times2}}, \tag{70b}$$

$$\boldsymbol{B'_{2\times2}} = \boldsymbol{J'}\begin{bmatrix} \boldsymbol{T_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T_2} \end{bmatrix}, \tag{70c}$$

$$\boldsymbol{B_{2\times2}} = \begin{bmatrix} \boldsymbol{T'_1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T'_2} \end{bmatrix} + \boldsymbol{JB'_{2\times2}}. \tag{70d}$$

Thus, the solution becomes

Interior Cell Faces Exterior Cell Faces

$$\begin{bmatrix} \boldsymbol{\psi}^{x+}_{i+\frac{1}{2},j} \\ \boldsymbol{\psi}^{x-}_{i+\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j+\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j+\frac{1}{2}} \\ \boldsymbol{\psi}^{x+}_{i+\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i+\frac{1}{2},j} \\ \boldsymbol{\psi}^{y+}_{i+1,j+\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j+\frac{1}{2}} \end{bmatrix} = \boldsymbol{A'_{2\times2}} + \boldsymbol{B'_{2\times2}}\begin{bmatrix} \boldsymbol{\psi}^{x+}_{i-\frac{1}{2},j} \\ \boldsymbol{\psi}^{x-}_{i+\frac{3}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j-\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{x+}_{i-\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i+\frac{3}{2},j} \\ \boldsymbol{\psi}^{y+}_{i+1,i-\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j+\frac{3}{2}} \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{\psi}^{x+}_{i+\frac{3}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i-\frac{1}{2},j} \\ \boldsymbol{\psi}^{y+}_{i+1,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j-\frac{1}{2}} \\ \boldsymbol{\psi}^{x+}_{i+\frac{3}{2},j} \\ \boldsymbol{\psi}^{x-}_{i-\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j-\frac{1}{2}} \end{bmatrix} = \boldsymbol{A_{2\times2}} + \boldsymbol{B_{2\times2}}\begin{bmatrix} \boldsymbol{\psi}^{x+}_{i-\frac{1}{2},j} \\ \boldsymbol{\psi}^{x-}_{i+\frac{3}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j-\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{x+}_{i-\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i+\frac{3}{2},j} \\ \boldsymbol{\psi}^{y+}_{i+1,i-\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j+\frac{3}{2}} \end{bmatrix}. \tag{71a,b}$$

Defining the following mapping for consistent ordering of the angular flux vector

$$\begin{bmatrix} \boldsymbol{\psi}^{x+}_{i-\frac{1}{2},j} \\ \boldsymbol{\psi}^{x-}_{i+\frac{3}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j-\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{x+}_{i-\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i+\frac{3}{2},j} \\ \boldsymbol{\psi}^{y+}_{i+1,i-\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j+\frac{3}{2}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}^{x+}_{i-\frac{1}{2},j} \\ \boldsymbol{\psi}^{x+}_{i-\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i+\frac{3}{2},j} \\ \boldsymbol{\psi}^{x-}_{i+\frac{3}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j-\frac{1}{2}} \\ \boldsymbol{\psi}^{y+}_{i+1,i-\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j+\frac{3}{2}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\psi}^{x+}_{i+\frac{1}{2},j} \\ \boldsymbol{\psi}^{x+}_{i+\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i+\frac{1}{2},j} \\ \boldsymbol{\psi}^{x-}_{i+\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j+\frac{1}{2}} \\ \boldsymbol{\psi}^{y+}_{i+1,j+\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j+\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j+\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}^{x+}_{i+\frac{1}{2},j} \\ \boldsymbol{\psi}^{x-}_{i+\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j+\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j+\frac{1}{2}} \\ \boldsymbol{\psi}^{x+}_{i+\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i+\frac{1}{2},j} \\ \boldsymbol{\psi}^{y+}_{i+1,j+\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j+\frac{1}{2}} \end{bmatrix} \qquad \text{(72a,b)}$$

$$\begin{bmatrix} \boldsymbol{\psi}^{x+}_{i+\frac{3}{2},j} \\ \boldsymbol{\psi}^{x+}_{i+\frac{3}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i-\frac{1}{2},j} \\ \boldsymbol{\psi}^{x-}_{i-\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{y+}_{i+1,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j-\frac{1}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j-\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}^{x+}_{i+\frac{3}{2},j+1} \\ \boldsymbol{\psi}^{x-}_{i-\frac{1}{2},j} \\ \boldsymbol{\psi}^{y+}_{i+1,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{y-}_{i,j-\frac{1}{2}} \\ \boldsymbol{\psi}^{x+}_{i+\frac{3}{2},j} \\ \boldsymbol{\psi}^{x-}_{i-\frac{1}{2},j+1} \\ \boldsymbol{\psi}^{y+}_{i,j+\frac{3}{2}} \\ \boldsymbol{\psi}^{y-}_{i+1,j-\frac{1}{2}} \end{bmatrix}. \qquad \text{(72c)}$$

The solution can be written to give

Interior Cell Faces / Exterior Cell Faces

$$\boldsymbol{\psi}^{out}_{i,j} = \boldsymbol{A}'_{2x2} + \boldsymbol{B}'_{2x2}\boldsymbol{\psi}^{in}_{i,j}, \qquad \boldsymbol{\psi}^{out}_{i,j} = \boldsymbol{A}_{2x2} + \boldsymbol{B}_{2x2}\boldsymbol{\psi}^{in}_{i,j}. \qquad \text{(73a,b)}$$

The angular flux vectors are identical to the single cell formulation, except that the vectors contain twice as many angular flux values on each cell face. The sweeping algorithm remains exactly the same. Note that the interior cell faces do not need to be calculated during iteration. Further, iterating over 2x2 groups of cells requires exactly the same number of FLOPS as iterating over each cell individually and therefore does not incur any additional computational cost per iteration. Since combining the cells analytically increases the implicitness of the solution, the number of iterations is reduced and an overall saving is achieved. Finally, the process of combining cells is fully scalable. The closed form solution of a domain of 4x4 cells can be obtained just as easily by combining a 2x2 domain of cells that are themselves a combined domain of 2x2 cells. The combination procedure can be applied for any 2^n heterogenous fine mesh and can be applied to solve the problem without iteration.

Derivation of ANDO-h: Local-h Adaption Using the Closed Form Solution

The 2^n closed form solution can be used to improve the computational efficiency and convergence of the ANDO-Constant method, as will be shown in Chapter 4. The 2^n closed form solution can also be used for local mesh refinement. Several approaches were implemented to interface the coarse mesh with the local-h mesh. The simplest approach is to apply the constant flux distribution on the coarse mesh cell face to each of the coincident fine mesh cell faces. Linear distributions can be established from the nearest adjacent coarse mesh cell face, or from the downwind or upwind flux, respectively, in adjacent cells. The downwind flux was found establish a more accurate linear distribution than the other approaches. For a single 2x2 fine mesh region surrounded on all sides by a coarse mesh, with a constant flux distribution interface, the vector $\boldsymbol{A}_{\mathbf{2}\times\mathbf{2}}$ and matrix $\boldsymbol{B}_{\mathbf{2}\times\mathbf{2}}$ are reduced to

$$\boldsymbol{B_h} = 0.5\begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{I} & \boldsymbol{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{I} & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \boldsymbol{B}_{\mathbf{2}\times\mathbf{2}} \begin{bmatrix} \boldsymbol{I} & 0 & 0 & 0 \\ \boldsymbol{I} & 0 & 0 & 0 \\ 0 & \boldsymbol{I} & 0 & 0 \\ 0 & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & \boldsymbol{I} & 0 \\ 0 & 0 & 0 & \boldsymbol{I} \\ 0 & 0 & 0 & \boldsymbol{I} \end{bmatrix}, \tag{74a}$$

$$\boldsymbol{A_h} = 0.5\begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{I} & \boldsymbol{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{I} & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \boldsymbol{A}_{\mathbf{2}\times\mathbf{2}}\,, \tag{74b}$$

to give

$$\boldsymbol{\psi}_{i,j}^{out} = \boldsymbol{A_h} + \boldsymbol{B_h}\boldsymbol{\psi}_{i,j}^{in}. \tag{75}$$

Equation (75) is the same as Equation (40) for the ANDO-Constant method except that the coefficients in the vector **A** and matrix **B** are slightly modified. However, the matrix coefficients are less accurate than ANDO-C, as will be shown later. The coefficients were also found to be less accurate through numerical experimentation. Notwithstanding that local-h refinement of a single coarse mesh cell is less accurate than the ANDO-Constant method over the same cell, local-h refinement can significantly improve accuracy in a region of the domain if the fine mesh cells are defined contiguously. A 2^n cell can easily be modified to simultaneously interface with adjacent coarse mesh cells and fine mesh cells on different faces. Further, the modification to interface with coarse mesh cells reduces the size of the vector **A** and matrix **B** and can be performed as a pre-computation step to improve the computational efficiency of the iteration loop.

Derivation of ANDO-Q: Quadratic and Higher Order Transverse Leakage

The derivation of ANDO-L can easily be extended to an n^{th} order polynomial to approximate the transverse leakage. The transverse leakage approximation in Equation (5) can be written as

$$L_{i,j}(x,\Omega_m) = \mathbb{L}_{\boldsymbol{x}} + L_{rx} + r_x x + \cdots + r_{xn} x^n \ . \tag{76}$$

Equation (7) then becomes

$$\frac{\partial}{\partial x}\boldsymbol{\psi}_x + \boldsymbol{\upmu}_{\mathbf{x}}^{-1}\Sigma_t\boldsymbol{\psi}_x - \boldsymbol{\upmu}_{\mathbf{x}}^{-1}\Gamma^{-1}\Sigma_s\boldsymbol{W}_{\boldsymbol{x}}\boldsymbol{\psi}_x =$$

$$\boldsymbol{\upmu}_{\mathbf{x}}^{-1}\Gamma^{-1}Q\mathbf{1} - \boldsymbol{\upmu}_{\mathbf{x}}^{-1}\boldsymbol{\eta}_{\boldsymbol{x}}(\mathbb{L}_{\boldsymbol{x}} + \boldsymbol{L}_{\boldsymbol{rx}} + \mathbf{r}_{\mathbf{x}}x + \cdots + \boldsymbol{r}_{\boldsymbol{xn}}x^n). \tag{77}$$

Following the same algebraic simplification in Equations (9-30), the intermediate matrix $\boldsymbol{R}_{\boldsymbol{B}}$ follows a pattern defined by

$$\boldsymbol{R}_{\boldsymbol{Bn}} = (-1)^n\boldsymbol{M}_{\boldsymbol{6}}\big(n!\,\boldsymbol{M}_{\boldsymbol{2}}\boldsymbol{\Lambda}^{\boldsymbol{-1}} + \textstyle\sum_{j=1}^{n}(n+1-j)!\,(-1)^j h_x^j\boldsymbol{M}_{\boldsymbol{3}\_}\boldsymbol{\Lambda}^{(j-1)}\big)\boldsymbol{\Lambda}^{-n}, \tag{78}$$

for an n^th order polynomial approximation. It is easy to show that substitution of n=1 gives $\boldsymbol{R_B}$ defined in Equation (28c) for a linear approximation and that $\boldsymbol{R_A}$ defined in Equation (28b) for a constant approximation is simply Equation (78) with n=0. Again, Equation (31c) becomes

$$\boldsymbol{R_{Dn}} = -\boldsymbol{R_{Bn}}\boldsymbol{R}^{-\mathbf{1}}\boldsymbol{\upmu}_{\mathbf{x}}^{-1}\boldsymbol{\eta_x}, \tag{79}$$

for an n^th order approximation to give Equation (30) as

$$\boldsymbol{\psi_x} = \boldsymbol{A_x} + \boldsymbol{B_x}\boldsymbol{\psi_{Bx}} + \boldsymbol{R_{D0}}(\mathbb{L}_{\boldsymbol{x}} + \boldsymbol{L_{rx}}) + \boldsymbol{R_{D1}}\boldsymbol{r_x} + \cdots + \boldsymbol{R_{Dn}}\boldsymbol{r_{xn}}. \tag{80}$$

All that remains to define the coefficients, r_{xn}, of Equation (76), substitute into Equation (80), and solve the local leakage balance as defined for the ANDO-Linear method. This work considered a quadratic leakage approximation calculated from the flux in adjacent cells. For n=2,

$$\boldsymbol{R_{B2}} = +\boldsymbol{M_6}\big(2\boldsymbol{M_2}\boldsymbol{\Lambda}^{-\mathbf{1}} - 2h_x^2\boldsymbol{M_{3-}} + h_x^2\boldsymbol{M_{3-}}\boldsymbol{\Lambda}\big)\boldsymbol{\Lambda}^{-\mathbf{2}}. \tag{81}$$

The remaining derivation is not instructive beyond that already presented for ANDO-L and results in a lengthy expression that will not be shown here for brevity.

Derivation of ANDO-3D: ANDO-C in 3-Dimensional Cartesian Geometry

As is the case for all nodal methods, the derivation of the ANDO-Constant method can easily be extended to 3-dimensional geometry. The transport equation

$$\mu_m \frac{\partial}{\partial x}\psi(x,y,\mathrm{z},\Omega_\mathrm{m}) + \eta_m \frac{\partial}{\partial y}\psi(x,y,\mathrm{z},\Omega_\mathrm{m}) + \zeta_m \frac{\partial}{\partial z}\psi(x,y,\mathrm{z},\Omega_\mathrm{m}) + \Sigma_t\psi(x,y,\mathrm{z},\Omega_\mathrm{m}) =$$

$$\Gamma^{-1}\Sigma_s \textstyle\sum_{k=1}^{M} w_k\psi(x,y,\mathrm{z},\Omega_k) + \Gamma^{-1}Q(x,y,\mathrm{z}), \tag{82}$$

is simply transverse integrated in two dimensions rather than one and the integration is approximated on the cell face rather than on the cell edge to give

$$\mu_m \frac{\partial}{\partial x}\psi_{i,j}(x,\mathit{\Omega}_m) + \Sigma_t\psi_{i,j}(x,\mathit{\Omega}_m) - \Gamma^{-1}Q(x,y,z) =$$

$$\Gamma^{-1}\Sigma_s \textstyle\sum_{k=1}^{M} w_k\psi_{i,j}(x,\mathit{\Omega}_m) - \eta_m\mathbb{L}_{x,i,j}^{y}(\Omega_m) - \zeta_m\mathbb{L}_{x,i,j}^{z}(\Omega_m). \tag{83}$$

Where $\mathbb{L}_{x,i,j}^{y}$ is the leakage in the y direction, as was defined in (5a) and $\mathbb{L}_{x,i,j}^{z}$ is the leakage in the z direction and is analogously defined. Following the same derivation as for 2-dimensional geometry, new intermediate matrices for the third dimension are defined that parallel Equations (31b) and (34)

$$\boldsymbol{R_{CZ}} = -\boldsymbol{R_A}\boldsymbol{R}^{-1}\boldsymbol{\mu}_x^{-1}\boldsymbol{\zeta}_x, \tag{84}$$

$$\boldsymbol{C_{xz}} = h_z^{-1}\boldsymbol{R_{CZ}}\boldsymbol{T_{MZ}}\begin{bmatrix} -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{85}$$

where the mapping matrix $\boldsymbol{T_{MZ}}$ is defined for the ordering of the quadrature points in the third dimension. Performing the local leakage balance

$$\boldsymbol{D_{3D}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{C_{xy}} & \boldsymbol{C_{xz}} \\ \boldsymbol{C_{yx}} & \boldsymbol{I} & \boldsymbol{C_{yz}} \\ \boldsymbol{C_{zx}} & \boldsymbol{C_{zy}} & \boldsymbol{I} \end{bmatrix}^{-1}, \tag{86a}$$

$$\boldsymbol{A_{3D}} = \boldsymbol{D_{3D}}\begin{bmatrix} \boldsymbol{A_x} \\ \boldsymbol{A_y} \\ \boldsymbol{A_z} \end{bmatrix}, \tag{86b}$$

$$\boldsymbol{B_{3D}} = \boldsymbol{D_{3D}}\begin{bmatrix} \boldsymbol{B_x} & \boldsymbol{C_{xy}} & \boldsymbol{C_{xz}} \\ \boldsymbol{C_{yx}} & \boldsymbol{B_y} & \boldsymbol{C_{yz}} \\ \boldsymbol{C_{zx}} & \boldsymbol{C_{zy}} & \boldsymbol{B_z} \end{bmatrix}, \tag{86c}$$

to give

$$\boldsymbol{\psi_{i,j}^{out}} = \boldsymbol{A_{3D}} + \boldsymbol{B_{3D}}\boldsymbol{\psi_{i,j}^{in}}. \tag{87}$$

Equation (87) is the same as Equation (40) for ANDO-C except that the size of vector **A** and matrix **B** are increased to accommodate six cell faces rather than four cell edges.

Derivation of ANDO-MG: The Multi-Group Transport Equation

The multi-group neutron transport equation in two-dimensional cartesian geometry with isotropic scattering and constant neutron source is written in discrete ordinates from as

$$\mu_m \frac{\partial}{\partial x}\psi_g(x, y, \Omega_m) + \eta_m \frac{\partial}{\partial y}\psi_g(x, y, \Omega_m) + \Sigma_{t_g}(x, y)\psi_g(x, y, \Omega_m) =$$

$$\Gamma^{-1} \textstyle\sum_{g'=1}^{G} \Sigma_{s_{g'\to g}}(x, y) \sum_{k=1}^{M} w_k \psi_{g'}(x, y, \Omega_k) + \Gamma^{-1} Q_g(x, y, \Omega_m). \tag{88}$$

The notation is the same as in Equation (1) for the mon-energetic transport equation except that the scattering cross section, $\Sigma_{s_{g'\to g}}$, is from g' to g and is summed for all groups G, whereas the energy group was not previously considered. After transverse integration

$$\mu_m \frac{\partial}{\partial x}\psi_{g,i,j}(x, \Omega_m) + \Sigma_{t_g}\psi_{g,i,j}(x, \Omega_m) - \Gamma^{-1} \textstyle\sum_{g'=1}^{G} \Sigma_{s_{g'\to g}} \sum_{k=1}^{M} w_k \psi_{g,i,j}(x, \Omega_m) =$$

$$\Gamma^{-1} Q_g - \eta_m \mathbb{L}_{g,i,j}(x, \Omega_m), \tag{89}$$

where $\psi_{g,i,j}(x, \Omega_m)$ and $\mathbb{L}_{g,i,j}(x, \Omega_m)$ are defined analogously to Equations (3) and (4), respectively, for each group g. Equation (89) is clearly the multi-group equivalent of Equation (6) and should be vectorized to derive an expression equivalent to Equation (7). The proposed methodology is to order the terms in equation (89) from group g to G with each group ordered according to the order of the quadrature points as in Equation (7a), i.e.

$$\boldsymbol{\psi}_x = \begin{bmatrix} \boldsymbol{\psi}_{g,x} \\ \vdots \\ \boldsymbol{\psi}_{G,x} \end{bmatrix}, \boldsymbol{\psi}_{g,x} = \begin{bmatrix} \psi_{g,i,j}(x, \Omega_m) \\ \vdots \\ \psi g,_{i,j}\ (x, \Omega_\mathrm{M}) \end{bmatrix}, \text{ angular flux vector;} \tag{90a}$$

$$\boldsymbol{\mu}_x = \begin{bmatrix} \boldsymbol{\mu}_\mathbf{g,x} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{\mu}_\mathbf{G,x} \end{bmatrix}; \boldsymbol{\mu}_\mathbf{g,x} = \begin{bmatrix} \boldsymbol{\mu}_x & \\ & -\boldsymbol{\mu}_x \end{bmatrix}, \text{ with } \boldsymbol{\mu}_{g,x} = \begin{bmatrix} \mu_m & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mu_{M/2} \end{bmatrix}; \tag{90b}$$

$$\boldsymbol{\Sigma}_t = \begin{bmatrix} \Sigma_{t_{g_1}}\boldsymbol{I} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma_{t_{g_2}}\boldsymbol{I} \end{bmatrix}; \tag{90c}$$

$$\boldsymbol{\Sigma}_s = \begin{bmatrix} \Sigma_{s_{g_1\to g_1}}\boldsymbol{I} & \cdots & \Sigma_{s_{G\to g_1}}\boldsymbol{I} \\ \vdots & \ddots & \vdots \\ \Sigma_{s_{g_1\to G}}\boldsymbol{I} & \cdots & \Sigma_{s_{G\to G}}\boldsymbol{I} \end{bmatrix}; \tag{90d}$$

and similarly for the other terms to give

$$\frac{\partial}{\partial x}\boldsymbol{\psi}_x + \boldsymbol{\mu}_\mathbf{x}^{-1}\boldsymbol{\Sigma}_t\boldsymbol{\psi}_x - \boldsymbol{\mu}_\mathbf{x}^{-1}\Gamma^{-1}\boldsymbol{\Sigma}_s\boldsymbol{W}_x\boldsymbol{\psi}_x = \boldsymbol{\mu}_\mathbf{x}^{-1}\Gamma^{-1}\boldsymbol{Q} - \boldsymbol{\mu}_\mathbf{x}^{-1}\boldsymbol{\eta}_x\mathbb{L}_x. \tag{91}$$

Note that Equation (91) reduces to Equation (7) for $G = 1$. A mapping is now applied to reorder the angular flux vector by direction, rather than by group, so that the previous derivation following Equation (7) is directly applicable to multi-group. The angular flux vector is ordered with the positive direction followed by the negative direction for each group with the groups order sequentially. Instead, the vector will be order with the positive direction for each group order sequentially followed by the negative direction for each group order sequentially. Defining the mapping matrix $\boldsymbol{T_G}$ such that

$$\begin{bmatrix} \psi_{g_1+} \\ \cdots \\ \psi_{\mathrm{G}+} \\ \psi_{g_1-} \\ \cdots \\ \psi_{G-} \end{bmatrix} = \boldsymbol{T_G} \begin{bmatrix} \psi_{g_1+} \\ \psi_{g_1-} \\ \cdots \\ \psi_{G+} \\ \psi_{\mathrm{G}-} \end{bmatrix}, \tag{92}$$

Equation (91) can be rearranged to give

$$\frac{\partial}{\partial x}\boldsymbol{\psi}_x + \boldsymbol{T}_{\boldsymbol{G}}^{-\mathbf{1}}\mu_{\mathrm{x}}^{-1}(\boldsymbol{\Sigma}_{\boldsymbol{t}} - \mu_{\mathrm{x}}^{-1}\Gamma^{-1}\boldsymbol{\Sigma}_{\boldsymbol{s}}\boldsymbol{W}_{\boldsymbol{x}})\boldsymbol{T}_{\boldsymbol{G}}\boldsymbol{\psi}_x = \boldsymbol{T}_{\boldsymbol{G}}^{-\mathbf{1}}\mu_{\mathrm{x}}^{-1}(\Gamma^{-1}\boldsymbol{Q} - \boldsymbol{\eta}_{\boldsymbol{x}}\mathbb{L}_{\boldsymbol{x}}). \tag{93}$$

Note that the group mapping simplifies the derivation and implementation of ANDO multi-group as Equations (90a-d) are arranged in a straightforward manner, but that the group mapping could be achieved in the reordering of Equations (90a-d). The matrix appearing in Equation (93) is diagonalizable and the following eigen value problem can be solved,

$$\boldsymbol{T}_{\boldsymbol{G}}^{-\mathbf{1}}\mu_{x}^{-1}(\boldsymbol{\Sigma}_{\boldsymbol{t}} - \Gamma^{-1}\boldsymbol{\Sigma}_{\boldsymbol{s}}\boldsymbol{W}_{\boldsymbol{x}})\boldsymbol{T}_{\boldsymbol{G}} = \boldsymbol{R\Lambda R^{-1}}. \tag{94}$$

Again, note the similarity to Equation (11) for single group. Further, because the eigen vectors $\boldsymbol{R}$ and eigen values $\boldsymbol{\Lambda}$ have already been reordered, the derivation is identical to Equations (12) to (31) and the resulting expressions for the intermediate solution matrices are identical. The solution vector $\boldsymbol{A_x}$ (31a) and matrix $\boldsymbol{B_x}$ (28a) are unchanged from the previous derivation. The solution matrix $\boldsymbol{C_x}$ (34) is slightly modified with

$$\boldsymbol{C}_{\boldsymbol{xMG}} = h_y^{-1}\boldsymbol{R}_{\boldsymbol{C}} \begin{bmatrix} \boldsymbol{T}_{\boldsymbol{MN}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{T}_{\boldsymbol{MN}} \end{bmatrix} \begin{bmatrix} -\boldsymbol{I} & & \cdots & 0 & \\ & \boldsymbol{I} & & & \\ \vdots & & \ddots & \vdots & \\ & & & -\boldsymbol{I} & \\ 0 & & \cdots & & \boldsymbol{I} \end{bmatrix} \boldsymbol{T}_{\boldsymbol{G}}, \tag{95}$$

where again the group mapping is applied in a way that simplifies the derivation and implementation but could be avoided by reordering the terms in the matrices of Equation (95). The unbalanced equation is now of the familiar form, except for the matrices and vectors are increased by G,

$$\boldsymbol{\psi}_x = \boldsymbol{A}_{\boldsymbol{xMG}} + \boldsymbol{B}_{\boldsymbol{xMG}}\boldsymbol{\psi}_{\boldsymbol{Bx}} + \boldsymbol{C}_{\boldsymbol{xMG}}\boldsymbol{\psi}_{\boldsymbol{By}} - \boldsymbol{C}_{\boldsymbol{xMG}}\boldsymbol{\psi}_y. \tag{96}$$

Perform the local leakage balance analogously to Equations (39a-c) to give

$$\boldsymbol{\psi}_{\boldsymbol{i,j}}^{\boldsymbol{out}} = \boldsymbol{A}_{\boldsymbol{MG}} + \boldsymbol{B}_{\boldsymbol{MG}}\boldsymbol{\psi}_{\boldsymbol{i,j}}^{\boldsymbol{in}}. \tag{97}$$

Equation (97) is now indistinguishable from Equation (40) for ANDO-C and no changes need to be made to the iteration loop. The angular flux vector can be split into each group after convergence and the balanced equation used to reconstruct the scalar flux.

The extraordinary advantage to reordering the multi-group angular flux vector is that all the previous derivations for single-group are readily applicable to multi-group. ANDO-2x2 Closed Form, ADNO-LL1, ANDO local h, and ANDO-3D are directly applicable to multi-group. ANDO-Reconstruction, ANDO-Linear, and ANDO-Quadratic and nth order are applicable to multi-group simply by modifying the solution matrices, i.e. Equations (49), in an analogous way to the modification of $\boldsymbol{C}_{\boldsymbol{x}}$ in Equation (95).

Chapter 3. Implementation

This chapter discusses some of the implementation details of the method, in particular the iteration scheme. The development of the method was itself iterative with the implementation informing the formulation presented in Chapter 2.

Code Optimization and Efficiency

The first implementation is unbalanced and alternates using Equations (35) and (36), calculating the transverse leakage after each half iteration. Significant work was performed to optimize the sweeping scheme and code to achieve better computational efficiency. The resulting Unbalanced Optimized implementation reduced the CPU time 75% over the first implementation. Even with significant optimization of the implementation the method was unacceptably slow due to a poor ability to balance the transverse leakage. This observation led to the local leakage balance shown in Equations (37-40). The local leakage balance was developed independently but is equivalent to Cell Block Inversion (CBI) developed by Barros in 1990 [7-8]. The new balanced formation was first implemented with a Jacobi iteration scheme. Even with a sub-optimal iteration scheme the Balanced Jacobi implementation was a significant improvement over the Unbalanced Optimized implementation. Implementing the balanced formulation with a Gauss-Seidel iteration scheme, where the domain is swept for eight directions, significantly reduced the number of iterations at an increase in CPU time per iteration. Subsequent testing and code modification led to an optimized balance implementation that did not require additional iterations over the Balanced Gauss-Seidel implementation and required only moderately more CPU time per iteration over the Balanced Jacobi implementation. The Balanced Optimized implementation sweeps the domain along the four diagonals, updating only the outgoing angular flux on the two faces in the direction of each sweep, rather than all four faces, effectively reducing the computations per iteration by

50% without increases the number of iterations required. Table 1 and Table 2 compare the performance of each implementation for two tests cases. The first case considers a 2x2 cm region with $\Sigma_s = 0.6$ cm^{-1}, $\Sigma_t = 1$ cm^{-1}, unity source and 20x20 spatial discretization. The second case considers a thick diffusive 2x2 cm region with $\Sigma_s = 9.9$ cm^{-1}, $\Sigma_t = 10$ cm^{-1}, unity source and 20x20 spatial discretization. Note that the ANDO method converges to the same solution regardless of the implementation, so the accuracy is not considered in the comparison in Table 1 and Table 2.

Table 1: Performance Comparison for Test Case 1

Case 1: 20x20	Unbalanced Optimized	Balanced Jacobi	Balanced Gauss-Seidel	Balanced Optimized	Diamond Difference
Iterations	1839	262	19	19 (99.5%)	41
CPU Time [s]	44.1064	2.0611	0.9370	0.2127 (99.0%)	0.3792
Time per Iteration [s]	0.0240	0.0079	0.0493	0.0112	0.0090

Table 2: Performance Comparison for Test Case 2

Case 2: 20x20	Unbalanced Optimized	Balanced Jacobi	Balanced Gauss-Seidel	Balanced Optimized	Diamond Difference
Iterations	971	1369	258	258 (73.4%)	1357
CPU Time [s]	23.9111	10.6457	13.8453	3.0370 (87.3%)	12.5322
Time per Iteration [s]	0.0246	0.0078	0.0537	0.0118	0.0092

For both cases the Balanced Optimized implementation is clearly superior to the other implementations and also outperforms DD in terms of number of iterations and overall computational cost. The Balanced Optimized implementation achieved a 99% reduction in number of iterations and CPU time for the Case 1 and 73% and 87% reduction in number of iterations and CPU time, respectively, for the diffusive case.

The Balanced Optimized implementation was extended for the linear transverse leakage formulation. ANDO-Linear achieves higher accuracy at greater computational cost. Five different approaches to approximating the slope of the leakage distribution were implemented. The first approach was to average the colinear adjacent cell faces and use the upwind flux. The second approach was the same as the first except the downwind flux was used. The third approach used the perpendicular cell faces and the upwind flux while the fourth approach used the downwind flux. A fifth approach used the flux on colinear adjacent cell faces to calculate the slope of the distribution, but not the intercept. The first, third, and fifth, approaches diverged. The second approach, Linear Consistent Downwind, and fourth approach, Linear Mixed Downwind, are compared in Table 3 and Table 4 below for cases 1 and 2, respectively.

Table 3: Performance Comparison for Test Case 1, ANDO-Linear

Case 1: 20x20	Unbalanced Optimized	Balanced Optimized	Linear Consistent Downwind	Linear Mixed Downwind	Diamond Difference
Iterations	1839	19	28	104	41
CPU Time [s]	44.1064	0.2127	10.2570	37.2947	0.3792
Time per Iteration [s]	0.0240	0.0112	0.3663	0.3586	0.0090

Table 4: Performance Comparison for Test Case 2, ANDO-Linear

Case 2: 20x20	Unbalanced Optimized	Balanced Optimized	Linear Consistent Downwind	Linear Mixed Downwind	Diamond Difference
Iterations	971	258	256	237	1357
CPU Time [s]	23.9111	3.0370	95.6986	84.5016	12.5322
Time per Iteration [s]	0.0246	0.0118	0.3738	0.3565	0.0092

The final implementation of ANDO-L and the derivation presented in Chapter 2 use the consistent downwind approach to approximate the transverse leakage.

Local Leakage Balance on Single Cell

Two test cases were considered to observe the effectiveness of the local leakage balance. The problem was a single cell 0.1 cm on a side with $\Sigma_t = 1$ cm^{-1} and vacuum boundary conditions. For the first case $\Sigma_s = 0.6$ cm^{-1} and for the second $\Sigma_s = 0.9$ cm^{-1}. As previously noted, the column vector A (39b) directly gives the outgoing flux for the local leakage balance of a single cell with vacuum boundary. Figure 1 and Figure 2 show the unbalanced formulation slowly converging to the balanced solution. For the diffusive case in Figure 2 the unbalanced formulation is seen to take hundreds of iterations to converge.

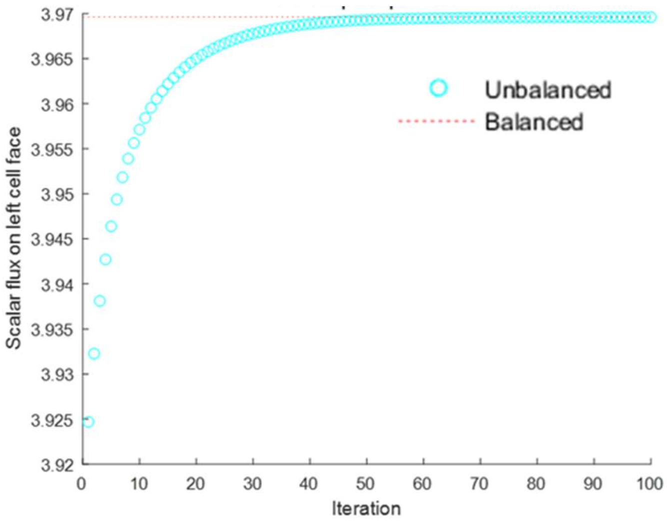


Figure 1: Local Leakage Balance Case 1

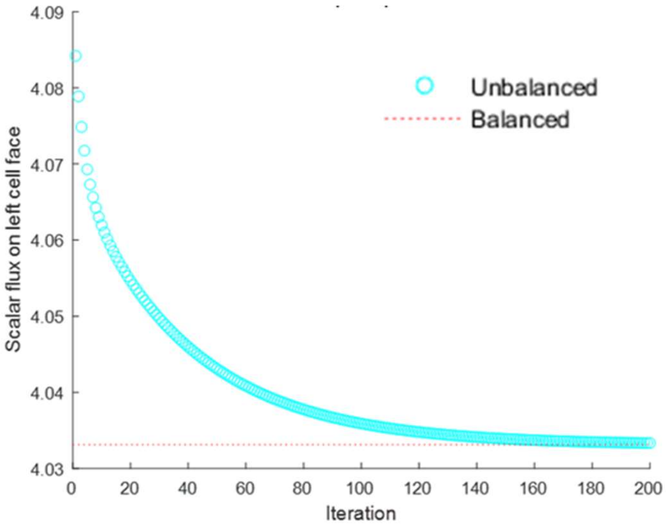


Figure 2: Local Leakage Balance Case 2

Relaxation of Unbalanced ANDO

The unbalanced implementation of ANDO was observed to oscillate and converge slowing. Underrelaxation and Overrelaxation were implemented for both the transverse leakage and the angular flux. Experimentation by trial and error was performed to determine an optimal value for the relaxation. One case is shown in Figure 3 and Figure 4 for 1x1 cm region with $\Sigma_t = 1$ cm^{-1}, $\Sigma_s = 0.6$ cm^{-1} discretized into a 10x10 mesh and is representative of the overall results.

Underrelaxation of 0.5 for either the transverse leakage or the angular flux reduced oscillations but also slowed convergence. Overrelaxation of 1.2 improved convergence rate but increased instability. Neither underrelaxation nor overrelaxation significantly improved convergence. However, local leakage balance significantly improved convergence.

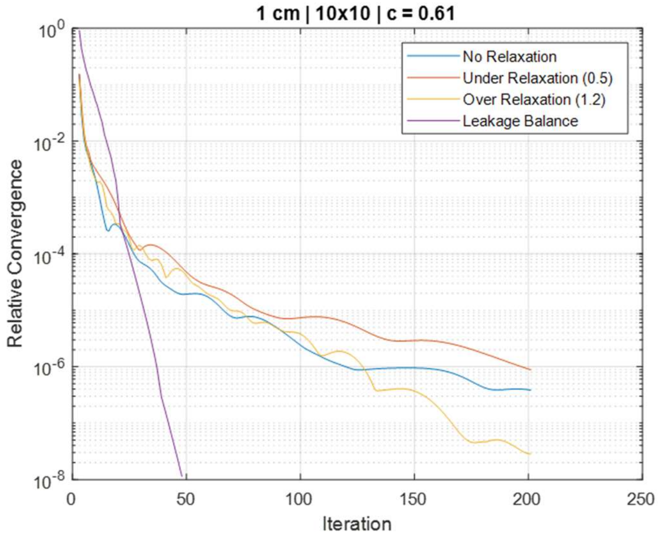


Figure 3: Relaxation of Transverse Leakage

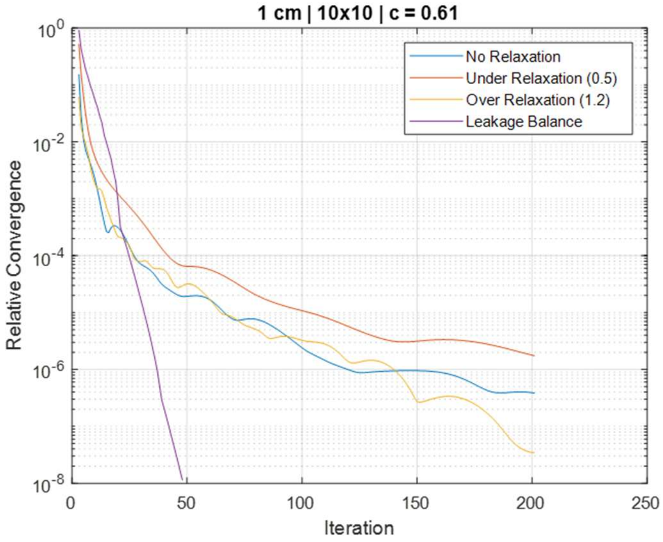


Figure 4: Relaxation of Angular Flux

Scalar Flux Reconstruction

Unlike source iteration methods, the scalar flux does not need to be calculated inside the iteration loop of ANDO. The average scalar flux in each cell is reconstructed from the cell face angular flux after convergence using the balance equation

$$\phi_{i,j} = \frac{Q - \frac{\Sigma_i^{\mathrm{M}} \mu_i \left(\psi_{i+\frac{1}{2},j} - \psi_{i-\frac{1}{2},j}\right) w_i}{\mathrm{h}_x} - \frac{\Sigma_i^{\mathrm{M}} \eta_i \left(\psi_{i,j+\frac{1}{2}} - \psi_{i,j-\frac{1}{2}}\right) w_i}{\mathrm{h}_y}}{\Sigma_a}. \tag{2.1}$$

Alternatively, ANDO-R can be used to reconstruct the cell average scalar fluxing using equation (57). The balance equation and ANDO-R are equally accurate, but the balance equation requires fewer FLOP's. The average scalar flux over multiple cells can be calculated either by the average of the scalar flux in each cell or by averaging the angular flux for each boundary and using reconstruction. Because ANDO is conservative, both approaches will yield the same result.

Sparse Matrix

It is worth mentioning that the 2Mx2M matrix **B** is a full matrix, preventing sparse matrix techniques from being employed. The moderately large size of the array limits the computational efficiency of ANDO, although it also improves the accuracy and rate of convergence. Only a few values in each row of **B** have a large magnitude relative to rest of the row. In an attempt to reduce the overall computational cost, the matrix **B** was posed as a sparse matrix containing only the largest values for the first few iterations. However, the total number of iterations increased and did not reduce the computational cost overall. Since the source term is included in the eigen value decomposition the final matrix **B** is ultimately full. However, the matrices **E** used in ANDO-L and ANDO-Q to establish the leakage distribution are sparse and are implemented in a way that reduces computational cost by $2\mathrm{M}^2$.

Recovery of 1D Analytical Solution

To assess the accuracy of ANDO, a region with reflected boundary conditions on top and bottom was considered to reduce the problem to an infinite slab. The accuracy of ANDO is limited only by the assumption that the transverse leakage is constant across the cell face. For top and bottom reflected boundary conditions this assumption is accurate, and ANDO reduces to the 1D analytical solution. Figure 5 shows the L1 error for a 1x1 cm region with $\Sigma_t = 1, \Sigma_s = 0.6$. ANDO is seen to have no spatial truncation error down to the machine rounding epsilon of 10^{-16} for the infinite slab, as expected.

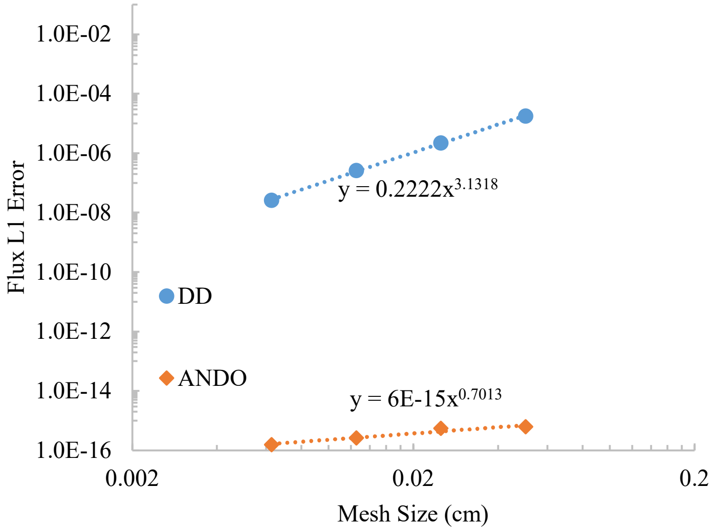


Figure 5: L1 Flux Error for Top and Bottom Reflected Boundary

It can also easily be shown mathematically that the ANDO method reduces to the 1D analytical solution for reflected boundaries. The outgoing flux is

$$\boldsymbol{\psi}_{i,j}^{x} = \boldsymbol{A}_x + \boldsymbol{B}_x\boldsymbol{\psi}_{i,j}^{Bx} + \boldsymbol{C}_x\boldsymbol{\psi}_{i,j}^{By} - \boldsymbol{C}_x\boldsymbol{\psi}_{i,j}^{y} \tag{2.2}$$

For reflected boundaries $\boldsymbol{\psi}_{i,j}^{By} = \boldsymbol{\psi}_{i,j}^{y}$ and the expression simplifies to

$$\boldsymbol{\psi}_{i,j}^{x} = \boldsymbol{A}_x + \boldsymbol{B}_x\boldsymbol{\psi}_{i,j}^{Bx} \tag{2.3}$$

One will find that if the solution outlined in Chapter 2 is performed for the 1D equation in slab geometry without transverse integration than all the transverse leakage terms will drop out and the exact same vector $\boldsymbol{A_x}$ and matrix $\boldsymbol{B_x}$ will result. Further, the analytical SN solution in slab geometry in Ref. [32] can be solved for the outgoing flux on the left and right boundaries and rearranged to give the same vector $\boldsymbol{A_x}$ and matrix $\boldsymbol{B_x}$.

Optimal Matrix coefficients of ANDO-C

The vector $\boldsymbol{A}$ and matrix $\boldsymbol{B}$ for ANDO-C contain 10N coefficients, were N is the number of quadrature points. Assuming no non-linear relations between angular flux values, as is the case for the transport equation used as the governing equation of ANDO-C, a minimum of 10N coefficients are needed to relate every incoming angular flux value to every outgoing angular flux value. Further, one set of coefficients will have optimal accuracy. Assuming that 1) the optimal set of coefficients will recover the 1D analytical solution in reflected opposing boundaries and 2) the transverse angular flux in reflected opposing boundaries is a function of the flux in the 1D analytical solution, it can be shown that only one set of $\boldsymbol{A_x}$, $\boldsymbol{A_y}$, $\boldsymbol{B_x}$, $\boldsymbol{B_y}$, $\boldsymbol{C_x}$, and $\boldsymbol{C_y}$ will give the most accurate set of coefficients. Starting with the system of equations

$$\boldsymbol{\psi_x = A_x + B_x\psi_{Bx} + C_x\psi_{By} - C_x\psi_y}, \tag{2.4a}$$

$$\boldsymbol{\psi_y = A_y + B_y\psi_{By} + C_y\psi_{Bx} - C_y\psi_x}, \tag{2.4b}$$

$$\begin{bmatrix} \boldsymbol{A_1} \\ \boldsymbol{A_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{C_x} \\ \boldsymbol{C_y} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A_x} \\ \boldsymbol{A_y} \end{bmatrix}, \tag{2.4c}$$

$$\begin{bmatrix} \boldsymbol{B_{11}} & \boldsymbol{B_{12}} \\ \boldsymbol{B_{21}} & \boldsymbol{B_{22}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{C_x} \\ \boldsymbol{C_y} & \boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{B_x} & \boldsymbol{C_x} \\ \boldsymbol{C_y} & \boldsymbol{B_y} \end{bmatrix}, \tag{2.4d}$$

$$\boldsymbol{\psi^{out} = A + B\psi^{in}}, \tag{2.4e}$$

$$\boldsymbol{\psi_x = A_1 + B_{11}\psi_{Bx} + B_{12}\psi_{By}}, \tag{2.4f}$$

$$\boldsymbol{\psi_y = A_2 + B_{21}\psi_{Bx} + B_{22}\psi_{By}}, \tag{2.4g}$$

Solve for outgoing flux for reflected boundaries and the transverse outgoing flux, the following system of 8 equations in 8 unknowns can be defined

$$\boldsymbol{B_x} = \boldsymbol{B_{11}} + \boldsymbol{B_{12}}(\boldsymbol{I} - \boldsymbol{B_{22}})^{-1}\boldsymbol{B_{21}}, \tag{2.5a}$$

$$\boldsymbol{B_y} = \boldsymbol{B_{21}}(\boldsymbol{I} - \boldsymbol{B_{11}})^{-1}\boldsymbol{B_{12}} + \boldsymbol{B_{22}}, \tag{2.5b}$$

$$(\boldsymbol{I} - \boldsymbol{B_x})^{-1}(\boldsymbol{C_x} - \boldsymbol{C_x}\boldsymbol{B_y}) = (\boldsymbol{I} - \boldsymbol{B_{11}})^{-1}\boldsymbol{B_{12}}, \tag{2.5c}$$

$$\left(\boldsymbol{I} - \boldsymbol{B_y}\right)^{-1}(\boldsymbol{C_y} - \boldsymbol{C_y}\boldsymbol{B_x}) = (\boldsymbol{I} - \boldsymbol{B_{22}})^{-1}\boldsymbol{B_{21}}, \tag{2.5d}$$

$$\boldsymbol{B_x} = \boldsymbol{B_{11}} + \boldsymbol{C_x}\boldsymbol{B_{21}}, \tag{2.5e}$$

$$\boldsymbol{B_y} = \boldsymbol{C_y}\boldsymbol{B_{12}} + \boldsymbol{B_{22}}, \tag{2.5f}$$

$$\boldsymbol{C_x} = \boldsymbol{B_{12}} + \boldsymbol{C_x}\boldsymbol{B_{22}}, \tag{2.5g}$$

$$\boldsymbol{C_y} = \boldsymbol{C_y}\boldsymbol{B_{11}} + \boldsymbol{B_{21}}. \tag{2.5h}$$

The equations are either linearly independent and have only one solution or have a linear dependency and have infinitely many solutions. Equations (2.5a,b) can be expressed using Equations (2.5e-h) and the system is reduced to 6 equations in 8 unknowns. However, as was shown earlier, the vector $\boldsymbol{A_x}$ and matrix $\boldsymbol{B_x}$ are the same as the 1D analytical solution

$$\boldsymbol{B_x} = \boldsymbol{B_{Analytical}}, \tag{2.6a}$$

$$\boldsymbol{B_y} = \boldsymbol{B_{Analytical}}, \tag{2.6b}$$

and from the above system of equations it is seen that ANDO-C is guaranteed to recover the analytical solution regardless of the values of $\boldsymbol{C_x}$ and $\boldsymbol{C_y}$. Further, the system reduces to 6 equations in 6 unknowns so that only one set of $\boldsymbol{B_{11}}$, $\boldsymbol{B_{12}}$, $\boldsymbol{B_{21}}$, $\boldsymbol{B_{22}}$, $\boldsymbol{C_x}$, and $\boldsymbol{C_y}$ will satisfy the assumed properties of the optimal set of coefficients. It was found that $\boldsymbol{B_{11}}$, $\boldsymbol{B_{12}}$, $\boldsymbol{B_{21}}$, $\boldsymbol{B_{22}}$, $\boldsymbol{C_x}$, and $\boldsymbol{C_y}$ defined by ANDO-C satisfy the above system of equations. Therefore, under the approximation of constant transverse leakage, the vector $\boldsymbol{A}$ and matrix $\boldsymbol{B}$ for ANDO-C already contain the optimal set of coefficients and no modification to the set of coefficients will improve the accuracy of

ANDO-C. This is consistent with the numerical results for ANDO-LL1 and local-h refinement of a single coarse mesh cell that both give slightly modified $\boldsymbol{A}$ and $\boldsymbol{B}$ that are less accurate.

Optimal Refinement for Improved Convergence and Reduced CPU Time

As mentioned in Chapter 2 and as will be shown in the results of Chapter 3, the ANDO-2x2 closed-form solution can be used to improve convergence and reduce CPU time per iteration. It must be noted, however, that the ANDO-Refined method is not an acceleration scheme in the usual sense, although acceleration of the solution is certainly achieved and similar terminology will be used. It should be further emphasized that acceleration techniques, such as DSA or CMFD, could be applied on top of the ANDO-Refined method for a compounding acceleration effect. The ANDO method is free from spatial truncation error within the computational cell and the exact nodal solution of the cell is obtained for the given cell boundary conditions. The ANDO-Refined method uses the ADNO-2x2 closed-form solution to analytically solve the interior cell faces of a group of cells. The group of cells will be referred to as the fine mesh, for example, a 2x2 fine mesh is a group of 4 cells. Thus, the exact nodal solution of the group of cells is obtained only from cell boundary conditions on the periphery of the group. This improves convergence by 1. Propagating boundary information across the group in fewer iterations and 2. Increasing the analyticity of the fine mesh solution. The entire computational domain is a coarse mesh were each node is a fine mesh group of cells. It is emphasized that only one mesh level is being solved. The fine mesh solution within each coarse mesh cell is precomputed outside the iteration loop and iteration is preformed over the coarse mesh. The total number of computational cells does not change, nor does the numerical method, so the exact same solution will be achieved as using ANDO-C for the same spatial discretization. Remarkably, the CPU time per iteration actually decreases using the ANDO-Refined method even though the number of FLOP's is the same as ANDO-C. However, additional computational cost is incurred in the pre-calculation. Indeed, the number if iterations and CPU time per iteration could be reduced to 0 if the entire domain were a single coarse mesh cell, but the pre-calculation would take much longer than solving the problem iteratively. Therefore, it was desirable to determine an optimal refinement that would reduce the CPU time required for iteration without significant additional CPU time required for pre-calculation. The test problem is a 2x2 cm region with $\Sigma_s = 9.9$ cm^{-1}, $\Sigma_t = 10$ cm^{-1} and unity source. The discretization of the domain ranged from 8x8 to 128x128 cells in total solved with fine mesh refinements ranging from 2x2 to 32x32, where fine mesh refers to the size of the group of cells within each coarse mesh. An S_{12} angular discretization was used and the convergence criterion was 10^{-16}. Figure 6 shows the number of iterations to convergence as a function of the number of grid points for each fine mesh refinement.

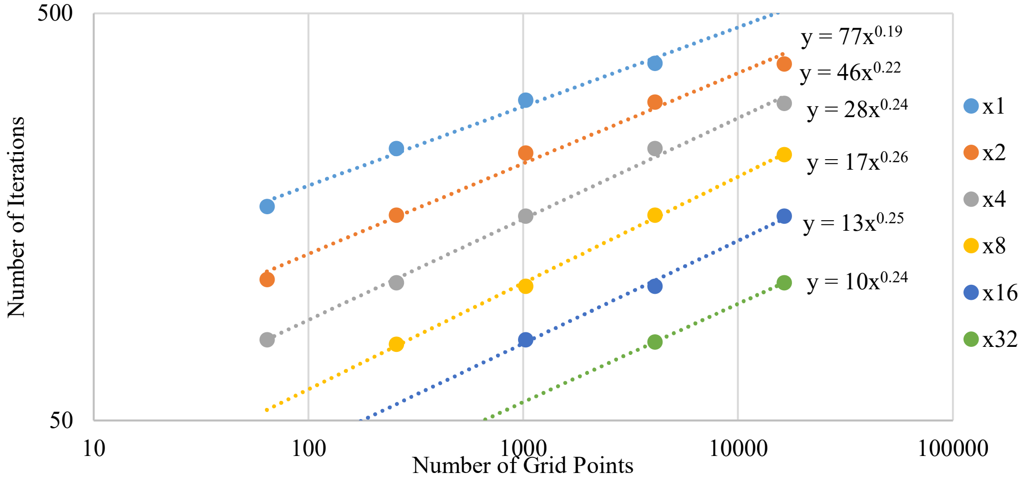


Figure 6: Number of Iterations to Convergence by Fine Mesh Grid Size

As expected, the number of iterations to convergence decreases as the number of cells in the fine mesh increases, approaching 1 iteration as the number of coarse mesh cells approaches 1. In fact, plotting the same data as above but for the number of coarse mesh cells it is seen in Figure 7 that the number of iterations to convergence is independent of the number of cells in the fine mesh.

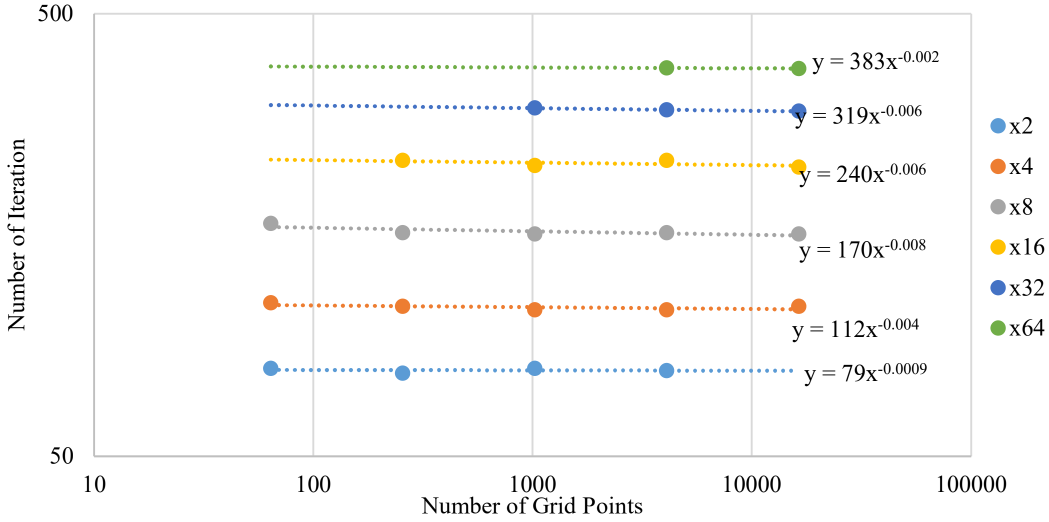


Figure 7: Number of Iterations to Convergence by Coarse Mesh Grid Size

The number of iterations decreased linearly with refinement and the precomputation cost increased exponentially with local refinement. Surprisingly, the CPU time per iteration does not continue to decrease with refinement as shown in Figure 8.

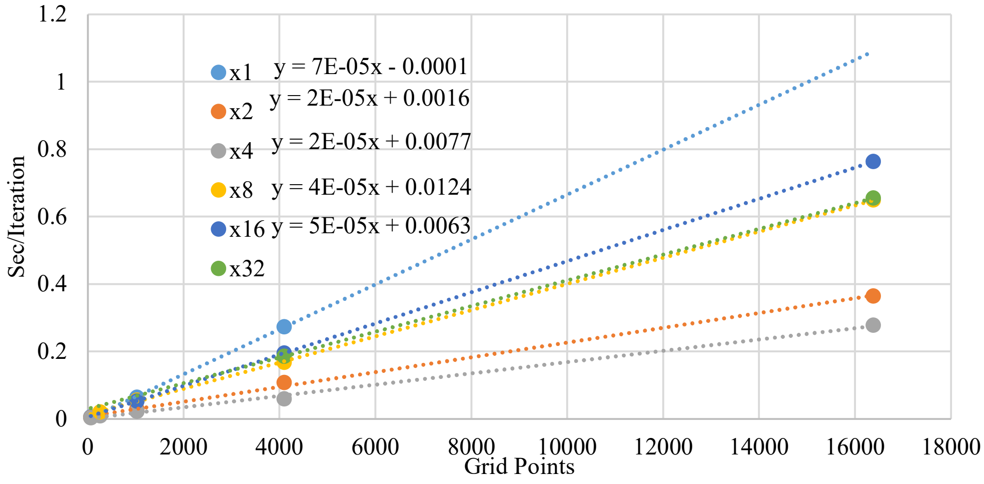


Figure 8: CPU Time per Iteration by Coarse Mesh Grid Size

Although refinement consistently reduces the CPU time per iteration compared to no refinement, a refinement of 4x4 is seen to have the lowest CPU time per iteration. It was found that for a different, more powerful computer a refinement of 8x8 was optimal. The ANDO refined method reduces

CPU time per iteration because memory access becomes more efficient as the size of contiguously stored data increases. Until, however, the entire matrix cannot be read into memory at once, at which point memory access becomes slower. The optimal refinement, therefore, is the largest matrix the given computing system can read at once. The size of the matrix depends on the size of the angular discretization, spatial discretization, and energy discretization. For the test cases and computing resources used in this work, a 4x4 refinement was found to be optimal in term of total CPU time. The reduction in CPU time per iteration alone compensates for the additional pre-calculation cost even without consideration to the CPU time saved by the reduced number of iterations.

How to Implement the ANDO Method

Although the derivation of the ANDO method is lengthy, the additional algebra greatly simplifies implementation because the solution matrices can be pre-computed outside the iteration loop using the equations presented in Chapter 2.

1. Reorder the quadrature points $\Omega_m = (\mu_m, \eta_m)$ and weights w_m such that $\mu_m > \mu_{m+1}$ for $m = 1,2,\dots,\frac{M}{2}$ and $\mu_{m+\frac{M}{2}} = -\mu_m$ and equivalently for η_n for the y direction. See appendix for algorithm.
2. Create the square matrices $\boldsymbol{\mu}_{\mathbf{x}}$, $\boldsymbol{\eta}_{x}$, $\boldsymbol{\eta}_{\mathbf{y}}$, $\boldsymbol{\mu}_{y}$ as defined in Equation (7e,f).
3. Create the transformation matrices $\boldsymbol{T_{MN}}$ and $\boldsymbol{T_{NM}}$ such that $\boldsymbol{\psi}_x(\Omega_m) = \boldsymbol{T_{MN}}\boldsymbol{\psi}_y(\Omega_n)$ and $\boldsymbol{\psi}_y(\Omega_n) = \boldsymbol{T_{NM}}\boldsymbol{\psi}_x(\Omega_m)$. See appendix for algorithm.
4. Set up Equation (11) and perform eigen value decomposition to determine $\boldsymbol{\Lambda}$ and $\boldsymbol{R}$.
5. Define the material matrices in Equations (21a-g).
6. Define the intermediate matrices in Equations (28b-c, 31b).
7. Define the solution matrices in Equations (28a, 31a, 34).

8. Perform the local leakage balance defined in Equations (39a-c).

9. Repeat 4-8 for the y direction, replacing $\boldsymbol{\mu_x}$ with $\boldsymbol{\eta_y}$, $\boldsymbol{\eta_x}$ with $\boldsymbol{\mu_y}$, and h_x with h_y.

10. Repeat 4-9 for each cell of unique material properties or unique dimension.

11. Split the resulting vector $\boldsymbol{A}$ and matrix $\boldsymbol{B}$ into fourths by row. This is equivalent to solving for the outgoing flux on each cell face separately. For M quadrature points:

```
ARgt = A(1:M/4);            BRgt = B(1:M/4,:);
ALft = A(M/4+1:M/2);        BLft = B(M/4+1:M/2,:);
ATop = A(M/2+1:M*3/4);      BTop = B(M/2+1:M*3/4,:);
ABot = A(M*3/4+1:M);        BBot = B(M*3/4+1:M,:);
```

12. The suggested algorithm for the iteration loop is:

```
for i = 2:Nx+1
    for j = 2:Ny+1
        psi(1:M/4,i+1,j)       = ARgt + BRgt*psi(:,i,j);
        psi(M/2+1:M*3/4,i,j+1) = ATop + BTop*psi(:,i,j);
    end
end

for j = Ny+1:-1:2
    for i = 2:Nx+1
        psi(1:M/4,i+1,j)       = ARgt + BRgt*psi(:,i,j);
        psi(M*3/4+1:M,i,j-1)   = ABot + BBot*psi(:,i,j);
    end
end

for j = Nx+1:-1:2
    for i = Ny+1:-1:2
        psi(M/4+1:M/2,i-1,j)   = ALft + BLft*psi(:,i,j);
        psi(M*3/4+1:M,i,j-1)   = ABot + BBot*psi(:,i,j);
    end
end

for j = 2:Ny+1
    for i = Nx+1:-1:2
        psi(M/4+1:M/2,i-1,j)   = ALft + BLft*psi(:,i,j);
        psi(M/2+1:M*3/4,i,j+1) = ATop + BTop*psi(:,i,j);
    end
end
```

Chapter 4. Numerical Results

This chapter presents the results of numerical experiments that demonstrate the performance of the ANDO method. First, the ANDO method is shown to have superior accuracy and computational efficiency compared to the well-established Diamond Difference (DD) method on two homogenous test cases. Next, the ANDO method is shown to be robust, possessing both positivity preserving and asymptotic preserving. A more realistic heterogenous case is then presented to demonstrate accuracy, computational efficiency, and robustness of the ANDO method. The results of the Well-Logging problem benchmark case are presented to demonstrate the applicability of the ANDO method to realistic transport problems and productions codes. Local-p adaption is shown to improve accuracy for the Well-Logging problem. Local-h adaption is shown to improve computational efficiency for the Fixed Source problem. Finally, the ANDO method is shown to be easily extended to 3-dimensional geometry and to multi-group calculations.

Spatial Discretization Convergence

Two cases are considered to demonstrate the accuracy and computational efficiency of the ANDO method in comparison to the DD method. Both cases are a homogenous 2×2-cm region with vacuum boundary conditions on all sides. The S_{12} level symmetric quadrature set is used for angular discretization. A uniform spatial discretization is used. The L1 error is obtained by comparing the numerical flux values to a sufficiently fine reference. The result for the first case with total macroscopic cross section $\Sigma_t = 1$ cm^{-1} and scattering cross section $\Sigma_s = 0.6$ cm^{-1} are shown in Figure 9. The second case considers a diffuse region with macroscopic cross section $\Sigma_t = 10$ cm^{-1} and scattering cross section $\Sigma_s = 9.9$ cm^{-1}, shown in Figure 10. The results show that the ANDO method has greater accuracy than DD in spatial discretization for both Case 1 and Case 2. The ANDO-Constant and ANDO-Linear methods have comparable accuracy when the spatial discretization is large while the ANDO-Linear method has a higher order of accuracy. Higher order transverse leakage approximations do not significantly improve the accuracy of the ANDO method.

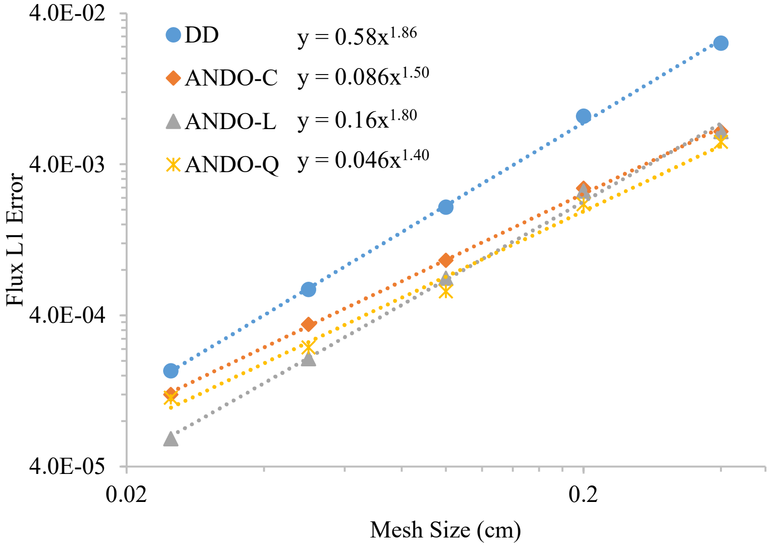


Figure 9: Flux L1 Error (Σ_t=1, c=0.6)

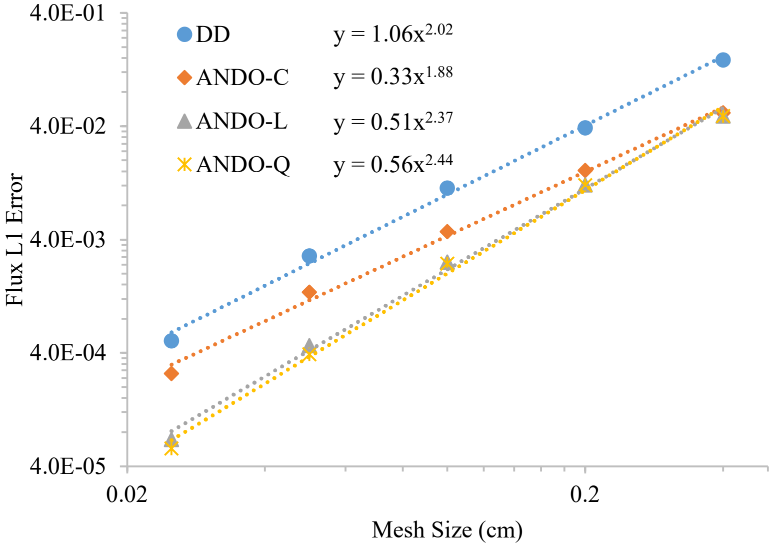


Figure 10: Flux L1 Error (Σ_t=10, c=0.99)

Angular Discretization Convergence

The same two cases are considered to demonstrate the angular discretization convergence. A 40x40 mesh is used for spatial discretization. The L1 Error is determined by comparing the scalar flux values in each cell to a spatially refined reference of the same quadrature refinement. The results for case 1 and 2 are shown in Figure 11 and Figure 12, respectively.

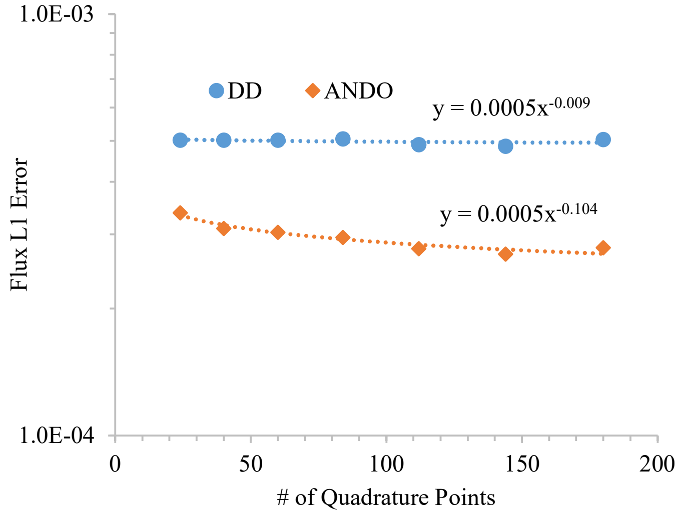


Figure 11: Flux L1 Error (Σ_t=1, c=0.6) Angular Discretization

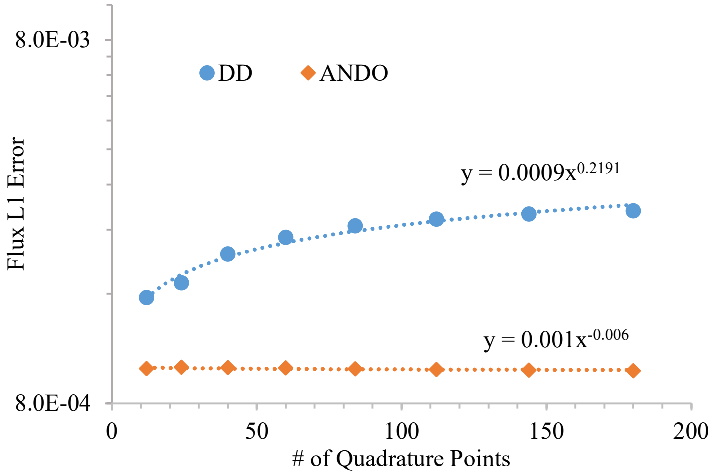


Figure 12: Flux L1 Error (Σ_t=10, c=0.99) Angular Discretization

The ANDO method is seen to be more accurate than the DD method. The accuracy of the ANDO method improves with angular refinement for the Case 1. The accuracy of the ANDO method for the diffusive case 2 is seen to be nearly independent of angular refinement. The CPU time per iteration for both methods for the angular discretization refinement is shown in Figure 13.

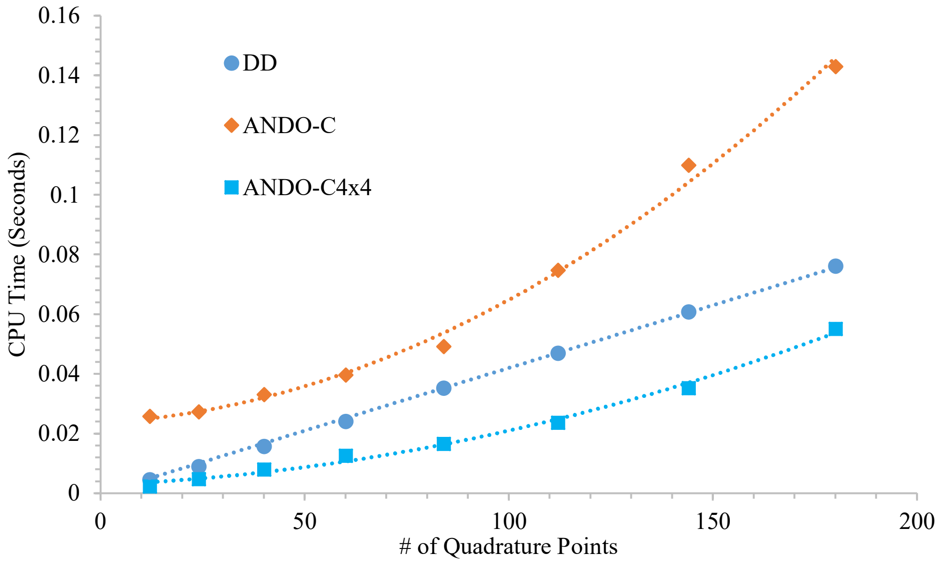
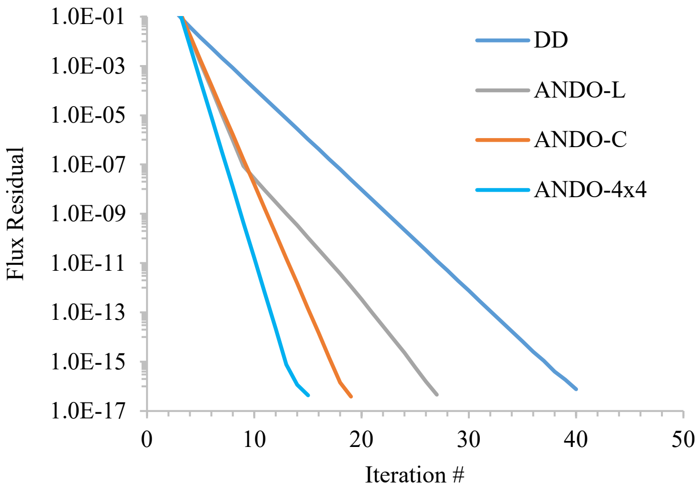


Figure 13: CPU Time per Iteration as a Function of Quadrature Points

The number of Floating-Point Operations (FLOP) required per iteration increases with the number of quadrature points. For M quadrature points, the FLOP count of DD scales with M whereas the FLOP count of ANDO scales with M^2, as evidenced in the CPU time required per iteration. The results show that the FLOP count for a 4x4 local refinement also scales with M^2 but does so at a reduced rate compared to the ANDO-Constant method. Thus, local refinement not only reduces the CPU time for a given spatial refinement but also minimized the additional computational cost associated with increasing the number of quadrature points.

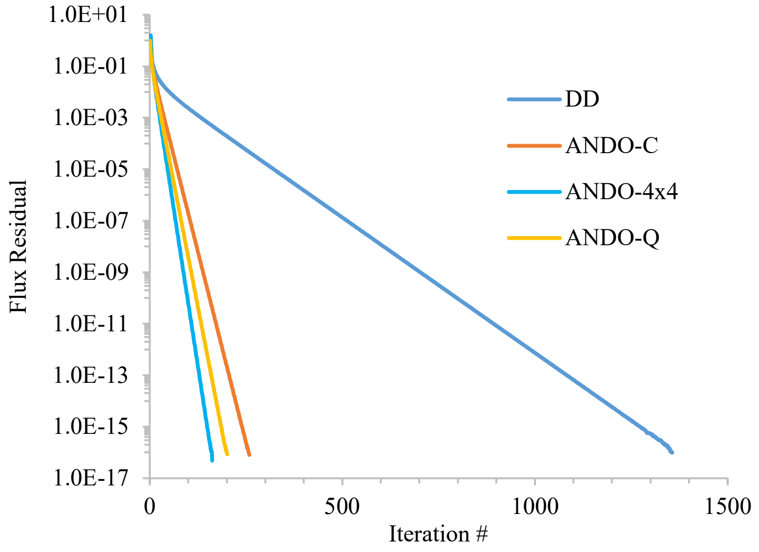
Transport Sweeping Convergence

Next, the ANDO method is shown to be rapidly convergent with a standard Gauss-Seidel type sweeping scheme. The residual flux error on a 20×20 mesh for Case 1 and Case 2 is shown in Figure 14 and Figure 15Figure 15, respectively. Note that the ANDO method iterates on the cell surface flux during the sweep, while the DD method uses source iteration (SI). For the case of $\Sigma_t =$ 1 cm^{-1} and $c = 0.6$ cm^{-1}, the ANDO method requires half as many iterations as the DD method to converge the solution within the machine rounding epsilon of 10^{-16}. The superior convergence is especially prominent for highly diffusive problems.



Note: The curve for ANDO-Linear and ANDO-Quadratic overlap

Figure 14: Sweeping Convergence (Σ_t=1, c=0.6)



Note: The curve for ANDO-Constant and ANDO-Linear overlap

Figure 15: Sweeping Convergence (Σ_t=10, c=0.99)

The ANDO-Constant method is approximately 1.5 times slower per iteration than the DD method but requires less than half as many iterations to converge to a prescribed criterion, so an overall time saving is achieved. The additional computation cost of the ANDO method arises from the absorption of the source term to the left-hand side of the equation before integration. The addition cost, however, is compensated by a significant reduction in the number of iterations required compared to source iteration. The number of iterations to convergence decreases with increasing local refinement and can be reduced to 1 if local refinement is applied to the entire domain. However, in terms of CPU time, it is generally faster to solve the problem iteratively. Therefore, the sweeping convergence of local refinement shown is for minimal CPU time. Further, the DD method is approximately 1.5 times slower per iteration than the ANDO-4x4 method. Thus, local refinement not only reduces the number of iterations but also reduces the CPU time per iteration.

Computational Complexity

A third case similar to the first two was considered to determine the computational complexity of the ANDO method. The problem is a 2x2 cm homogenous domain with Σ_t = 1 cm^{-1}, Σ_s = 0.9 cm^{-1}, constant unity source and vacuum boundary conditions. The domain is discretized into a uniform mesh of varying refinement. The number of iterations and CPU time required to achieve a convergence of 10^{-16} was measured to determine the computation complexity. The results are tabulated in Table 5 and shown in Figure 16.

Table 5: Computational Complexity Comparison

Mesh	Number of Grid Points (m)	Diamond Difference		ANDO			
		Number of Iterations (n)	Complexity $(n \times m)$	Constant		Linear	
				Iterations	Complexity	Iterations	Complexity
20×20	20^2	43	17 200	29	11 600	28	11 200
40×40	40^2	43	68 800	30	48 000	33	52 800
80×80	80^2	43	275 200	31	198 400	39	249 600
160×160	160^2	43	1 100 800	32	819 200	43	1 100 800

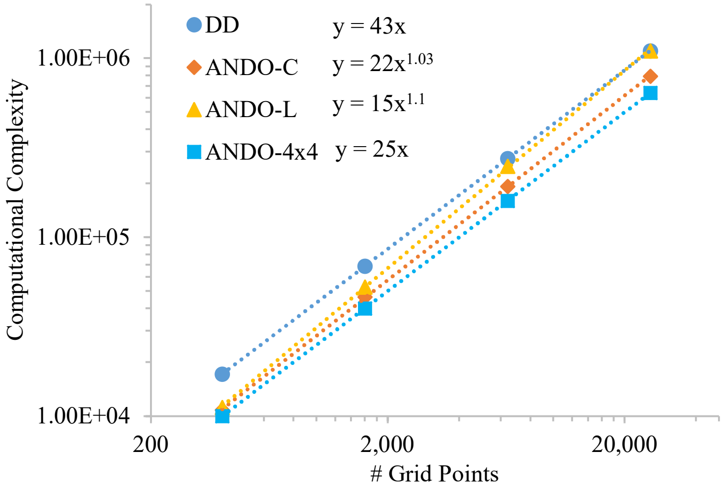


Figure 16: Computational Complexity Versus Number of Grid Points

The ANDO method is seen to have a lower computational complexity compared to the DD method. The ANDO-Constant and ANDO-Linear methods are seen to have a nearly linear computational complexity. The ANDO method with local refinement has the lowest computational complexity and the ANDO-Linear method has a higher computational complexity than the ANDO-Constant method. Despite a higher computational cost per iteration, the ANDO-Constant method achieves a significant time saving compared to the DD method, as shown in Figure 17. However, an overall time saving is not achieved for the ANDO-Linear method due to higher CPU time per iteration. Local refinement has the lowest computational complexity the lowest CPU time per iteration resulting in the greatest time saving.

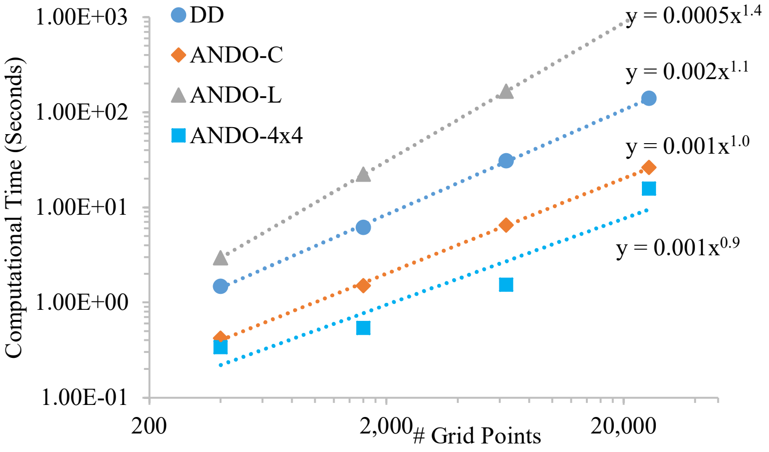


Figure 17: Computational Time Versus Number of Grid Points

The computational cost can also be calculated directly from the Floating-Point Operations (FLOP) required per iteration for each method. For m quadrature points and N grid points, the DD method requires 25mN FLOP per iteration, the ANDO-Constant method requires 16m2N FLOP, and the ANDO-Linear method requires an additional 8mN FLOP. The FLOP's of each method scales linearly with number of gird points, as is shown in Figure 17. The FLOP's increase linearly with number of quadrature points for the DD method and increase quadratically for the ANDO method, as was shown in Figure 13. For the test cases considered, the ANDO method was found to require significantly less computer resources to solve the same problem and, as was shown earlier, to a greater accuracy.

Positivity Preserving

One of the shortcomings of the DD method and other high-order S_N methods is the potential to produced non-physical negative flux values near material boundaries which can lead to erroneous or misleading results. The test problem consists of a 5x5 cm region with high absorbing region in the center. Figure 18 shows the ANDO-Constant method is robust and positivity preserving for this case, whereas the DD method and other non-positivity preserving methods would produce negative values in the interior region.

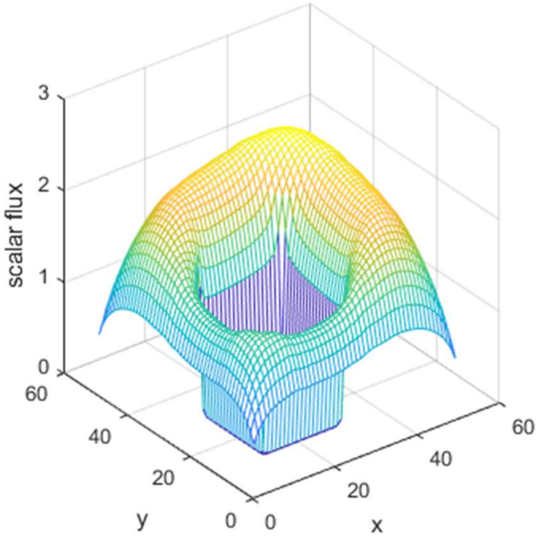


Figure 18: Positivity Preserving

Asymptotic Preserving

Another property of robust numerical methods is asymptotic preserving. The solution of the transport equation tends to the diffusion equation for optically thick computation cells. That is, when the total cross section is high and the absorption cross section and external source are low. The first test problem is a 1/16x1/16 cm region with material properties parametrically defined as $\Sigma_t = \frac{1}{\epsilon}$, $\Sigma_s = \frac{1}{\epsilon} - 0.8\epsilon$, and $Q = \epsilon$. The problem becomes thick and diffusive as $\epsilon \to 0$ and the solution asymptotically approaches the solution to the diffusion equation. Figure 19 shows that both the DD method and the ANDO method possess the diffusion limit and are asymptotically preserving.

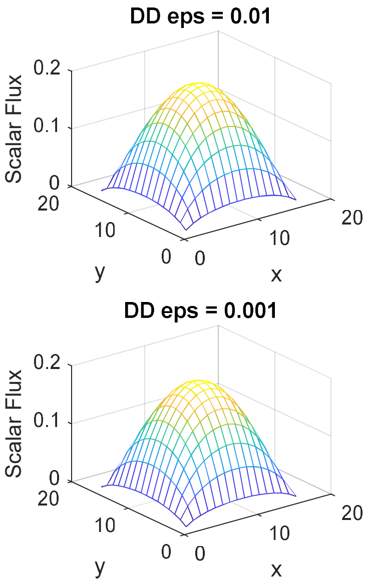


Figure 19: Asymptotic Preserving Case 1

More significantly, because the ANDO-Constant method is asymptotically preserving and free from spatial truncation error within the computational cell, local refinement is also robust and can rapidly give the solution no matter how diffusive the problem. The ANDO method results shown in Figure 16 were obtained in one iteration on a 1 cell coarse mesh with 16x16 local refinement and took approximately three seconds to compute, regardless of the value of ϵ. However, methods such as the DD method that use source iteration require hundreds of thousands of iterations and several hours to converge. Further, although acceleration schemes can be used to improve convergence, they may fail for highly diffusive problems. The ANDO method, however, requires no acceleration and was able to solve the problem in a single iteration up to scattering ratio c = 0.999999999999992 at ϵ =10^{-7}.

The second problem uses the same domain and parametrically defined material properties. The difference is an incident angular flux on the left side of the domain introducing a thin boundary layer. The ANDO method is able to resolve the boundary layer as shown in Figure 20 even on a coarse mesh whereas greater spatial refinement would be required for the DD method.

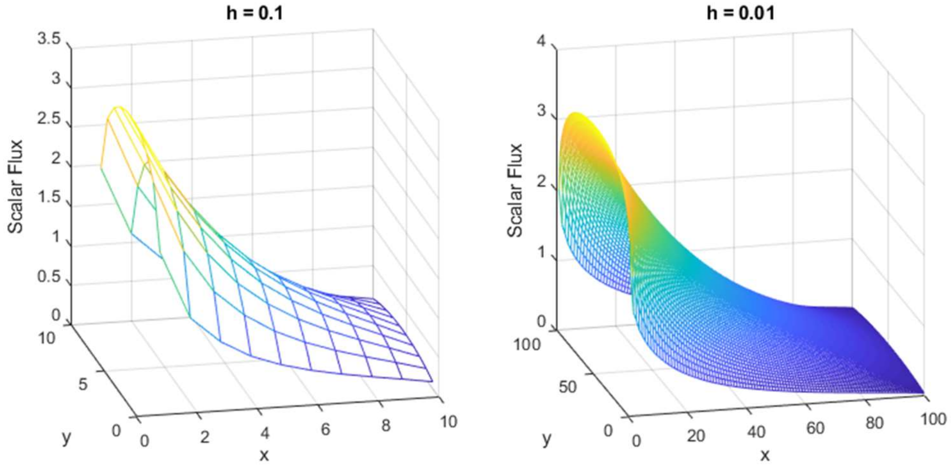


Figure 20: Asymptotic Preserving Case 2

Heterogenous Test Problem

A third case was considered to demonstrate the accuracy of the ANDO method on a heterogenous domain. The problem is shown in Figure 21, as described in Ref. [34].

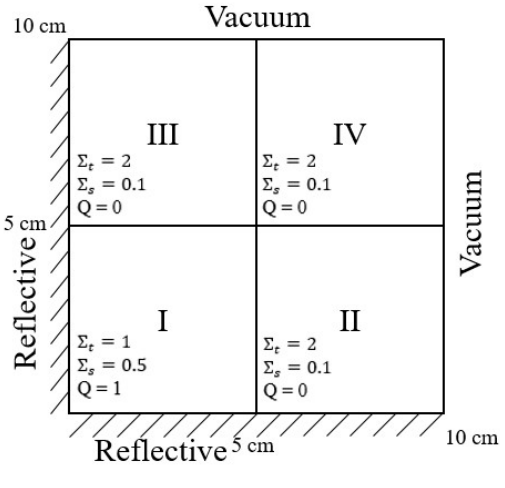


Figure 21: Problem Setup

The L1 error is shown in Figure 22. The ANDO method has a significantly higher accuracy than the DD method for this heterogenous case, which is consistent with the results for the homogenous cases. The ANDO-Linear method has comparable accuracy to the ANDO-Constant method and has a higher order of accuracy, which is consistent with the results for the homogenous case. The linear leakage approximation did not significantly improve the accuracy of the ANDO method.

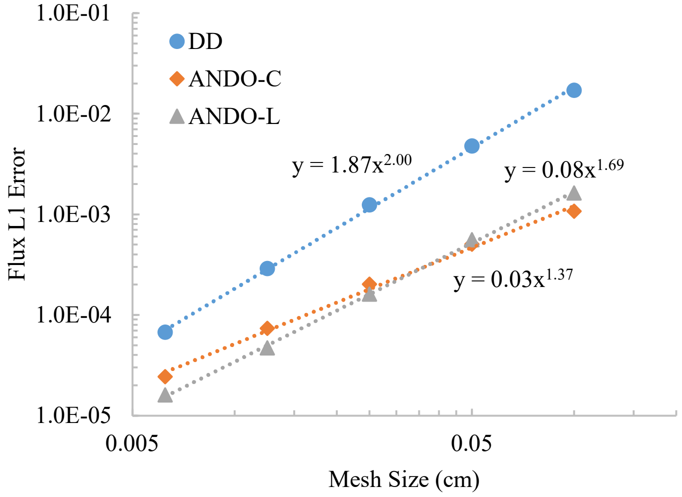


Figure 22: Flux L1 Error for Heterogenous Case

The DD method required twice as many iterations to converge while the ANDO method required more time per iteration. The total CPU time for the ANDO-Constant method was nearly half that required by the DD method for a given spatial refinement, as shown in Figure 23. In addition, considering the greater accuracy of the ANDO method, a coarser spatial discretization could be used to resolve the solution to the same precision. For example, to achieve three digits of accuracy DD required a 40x40 mesh and took 7.6 seconds. To achieve the same accuracy the ANDO-Constant method required only a 10x10 mesh and took 0.69 seconds. An overall time saving could thus be achieved to obtain a prescribed accuracy. The ANDO-Linear method requires a greater computational cost compared to the ANDO-Constant method that is not justified in greater accuracy of this case.

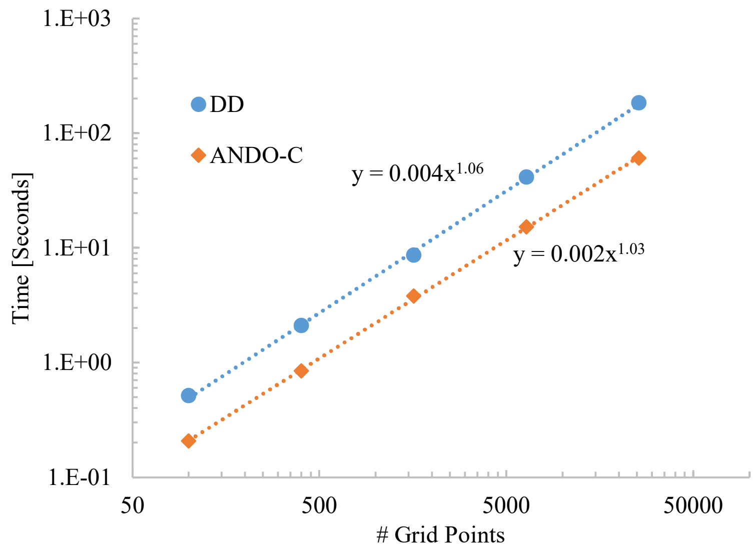


Figure 23: CPU Time versus Number of Grid Points Heterogenous Case

This four-quadrant heterogenous test case is frequently used in the literature as a numerical benchmark. The average scalar flux in each region is computed for varying spatial refinements, noting that the flux in region II and region III is identical. For this deterministic neutron transport benchmark, the solution is taken to be that of a deterministic method with a sufficiently refined mesh. The performance of the ANDO method presented in this work will be presented in comparison to early nodal methods presented in Azmy, Ref. [34], and the recent ANDO method presented in Picoloto et. al. Ref [28]. First, the results for an S_4 quadrature set with 10^{-4} convergence criteria are presented in Table 6 in comparison to the LN method. The results show that the ANDO method presented in this work has comparable accuracy to the LN method and achieves an excellent reduction in the number of iterations with corresponding reduction in CPU time.

Table 6: Average Scalar flux Comparison.

	This work				LN [34] (1988)			
Mesh	Region I	Region II	Region IV	Iteration #	Region I	Region II	Region IV	Iteration #
10x10	1.677	4.15E-02	2.06E-03	7	1.676	4.16E-02	1.99E-03	19
20x20	1.676	4.16E-02	2.02E-03	7	1.676	4.16E-02	1.99E-03	21
40x40	1.676	4.16E-02	2.00E-03	8	1.676	4.16E-02	1.99E-03	19

Second, the results for a 2×2 region of varying angular refinement are presented in Table 7 in comparison to the ANDO method in Ref [28]. Both methods solve the closed-form solution on the 2×2 mesh. Thus, no iteration is required, and the solution is free from convergence error. Further, both analytical nodal methods are free from spatial truncation error within the computation cell. Therefore, since both methods use a constant transverse leakage approximation both methods have comparable accuracy. However, the ANDO method presented in this work gives a truly closed form solution that does not require the solution of a large coefficient matrix, unlike the previous ANDO method. This results a huge reduction in computation cost. The previous ANDO method was reported to solve the problem in ~9 seconds whereas the new ANDO method required less than 0.03 seconds for the largest quadrature order S_{16}.

Table 7: Average Scalar flux Comparison of Closed Form Solution

Region	Sn	This Work	ANDO [28] (2015)
I	2	1.656	1.655
	4	1.679	1.678
	6	1.682	1.682
	8	1.684	1.683
	12	1.685	1.685
	16	1.686	1.685
II	2	4.39E-02	4.38E-02
	4	4.13E-02	4.12E-02
	6	4.09E-02	4.08E-02
	8	4.06E-02	4.06E-02
	12	4.05E-02	4.04E-02
	16	4.04E-02	4.04E-02
IV	2	2.75E-03	2.74E-03
	4	2.03E-03	2.03E-03
	6	1.93E-03	1.92E-03
	8	1.88E-03	1.87E-03
	12	1.84E-03	1.84E-03
	16	1.83E-03	1.82E-03

Well-Logging Problem Benchmark

The Well-Logging Problem is frequently used in the literature as a numerical benchmark. Well logging is used to measure the composition of rock formations by the insertion of a probe into a bore hole. Neutrons emitted from a neutron source at the end of the probe scatter in the surrounding material and are measured at two detector locations along the probe to infer the material composition. The ratio of the detector counts is thus the primary figure of merit. The well logging problem presents a challenge for transport calculations because it is dominated by scattering. The two-dimensional well logging problem setup and material properties were taken from [34] and is shown in Figure 24.

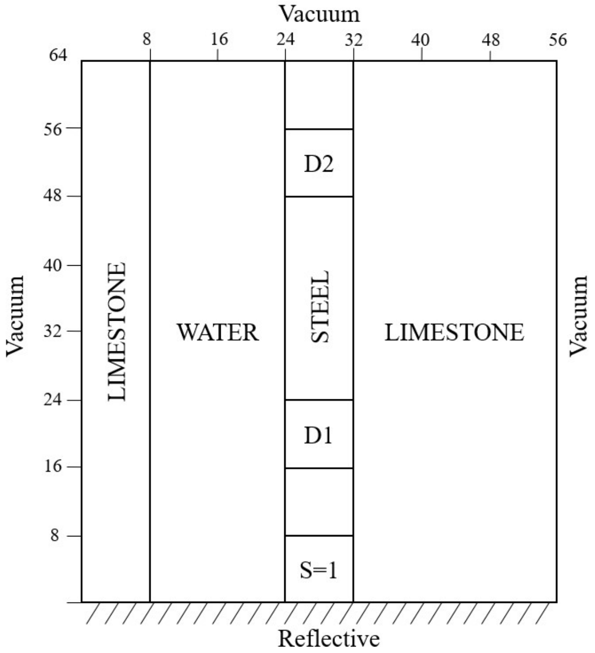


Figure 24: Well-Logging Problem Setup

An S_6 quadrature set was used for the angular discretization and the convergence criteria was set to 10^{-6} for comparison to results reported in the literature. The L1 flux error plotted in Figure

25 shows the ANDO-Linear method is significantly more accurate than the DD method and the ANDO-Constant method for this case.

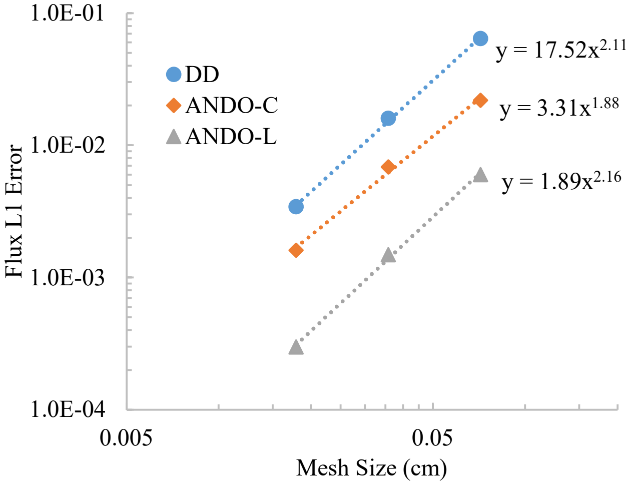


Figure 25: Flux L1 Error (Well-Logging Problem)

The scalar flux in each detector is tabulated in Table 8 for varying spatial refinement for the ANDO-Constant, ANDO-Linear, and ANDO-Quadratic methods with the LN method and DD method included for comparisons. The accuracy is in comparison to the solution of a deterministic method on a sufficiently refined mesh. The results show that the ANDO method achieves an almost tenfold reduction in the number of iterations with accompanying reduction in CPU time. The results indicate that the ANDO-Constant method is less accurate than the LN method for the same spatial refinement but that an overall time saving can still be achieved if a greater spatial refinement is used for the ANDO-Consent method. However, it should be noted that the LN method produces non-physical negative flux values on the coarse mesh whereas the ANDO-Constant method remains strictly positivity preserving. The ANDO-Linear method further improves the accuracy of the ANDO method while still achieving a time saving. However, the ANDO-Quadratic method does not significantly improve the accuracy of the ANDO method relative to the ANDO-Linear method. The higher order leakage approximations have the additional benefit of reducing the number of iterations.

Table 8: Comparison of the Numerical Solutions for the Well-Logging Problem

Mesh	D_1	D_2	D_1/D_2	Number of iterations	CPU time (minutes)
LN [34]					
7x8	1.647	0.01374	119.9	344	5.7
14x16	1.732	0.01260	137.5	390	27.4
28x32	1.718	0.01247	137.8	414	115.8
56x64	1.717	0.01247	137.7	454	490.7
DD					
7x8	0.887	-0.00296	-299.4	218	0.19
14x16	1.548	0.01025	151.1	227	0.31
28x32	1.672	0.01182	141.5	230	1.59
56x64	1.705	0.01227	139.0	231	5.95
ANDO-Constant					
7x8	1.348	0.00938	143.7	31	0.05
14x16	1.603	0.01142	140.4	43	0.28
28x32	1.682	0.01215	138.5	56	1.45
56x64	1.707	0.01237	138.0	70	7.23
112x128	1.714	0.01243	137.9	83	36.96
ANDO-Linear					
7x8	1.745	0.01291	135.2	30	0.15
14x16	1.736	0.01253	138.5	40	0.99
28x32	1.720	0.01248	137.9	50	5.44
56x64	1.717	0.01246	137.8	61	28.03
ANDO-Quadratic					
7x8	1.514	0.0089	170.0	29	0.15
14x16	1.678	0.0117	143.0	38	1.08
28x32	1.708	0.0123	139.0	47	5.96
56x64	1.714	0.0124	138.1	52	24.10
ANDO-L Local-p Adaption					
7x8	1.653	0.01204	137.3	31	0.12
14x16	1.696	0.01217	139.4	41	0.56
28x32	1.709	0.01236	138.3	53	2.91
56x64	1.714	0.01243	137.9	65	14.85
112x128	1.716	0.01245	137.9	74	73.60

The results present a tradeoff between the lower CPU time of the ANDO-Constant method and the greater accuracy of the ANDO-Linear method. Local-p adaption can be used to gain the improved accuracy of the ANDO-Linear method without incurring a significantly greater computational cost. To demonstrate the local-p adaptivity of the ANDO method, the ANDO-Linear method was applied for cells along the central steel region of the domain and one cell on either side with the ANDO-Constant method applied elsewhere in the domain. The results show that local-p adaption significantly improved the accuracy of the ANDO-Constant method without significantly increasing the total computational cost. Other local-p adaptions were considered, as well as local-p adaption with ANDO-Quadratic. The results shown had the best tradeoff between accuracy and CPU time.

Fixed Source Problem and Local-h Adaption

The fixed source problem setup is a central source region bound on one side by a high absorption region and on the other by a high scattering region, as shown in Figure 26.

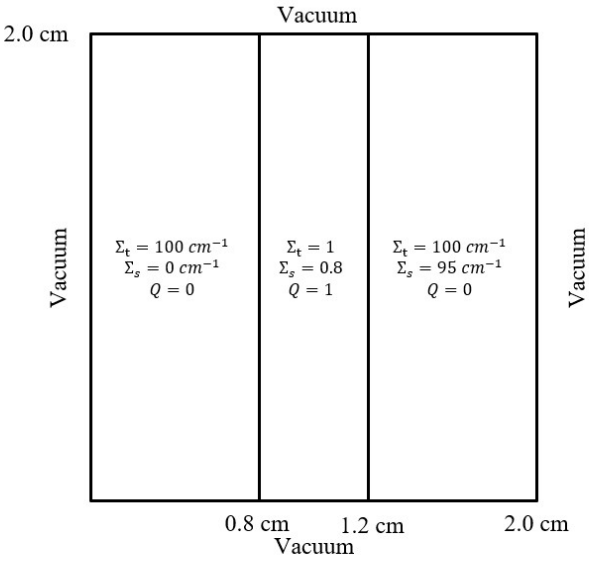


Figure 26: Fixed Source Problem Setup

The sharp material boundaries present a challenge for numerical method and can result in negative flux values for non-positivity persevering methods. Further, resolution of the central peak also presents a challenge can result in an underprediction of maximum value. However, determination of the peak scalar flux is of utmost importance in nuclear reactor design. The scalar flux profile on a 50×50 mesh and a 200×200 mesh for ANDO-C is shown in Figure 27.

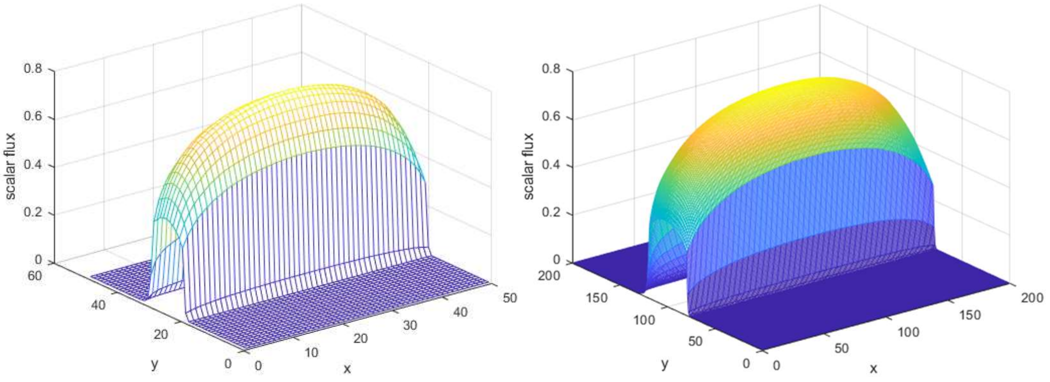


Figure 27: Scalar Flux Profile for the Fixed Source Problem

The ANDO-Constant method is seen to have excellent resolution of the flux profile, producing a smooth solution free from non-physical negative flux values. Further, the peak scalar flux calculated for each spatial discretization agree within three digits of accuracy. However, the structured cartesian mesh results in significant wasteful refinement outside the central region of interest. To demonstrate the local-h adaptivity of the ANDO method, the problem was discretized into a 50×50 coarse mesh and solved using the ANDO-Constant method. The problem was then solved using ANDO-Constant on a 200×200 mesh. Next, a 4×4 local-h fine mesh was applied to every cell in the 50×50 coarse mesh, which is equivalent to the 200×200 spatial discretization. The solution of the local-h fine mesh is precomputed using the ANDO-2x2 closed-form solution. The ANDO-4x4 method shown in earlier results is equivalent to local-h refinement in every cell of the computational domain and was shown to significantly improve convergence and CPU time per iteration. Finally, a 4×4 local-h fine mesh was applied to every coarse mesh cell in the source region and one coarse mesh cell on either side. The performance of each local-h adaption case is tabulated in Table 9. Consistent with easier results, the ANDO-4x4 method significantly reduced the computational cost of ANDO-C on 200×200 mesh. Local-h adaption of the entire computational domain resulted in a similar time reduction. Further, the results show the local-h adaption in the source region significantly reduces the number of gird points without compromising the accuracy of the central peak. Local-h adaption thus reduced the computational cost of the ANDO-Constant method by an additional 93% beyond the already significant reduction in computational cost compared to other methods.

Table 9: Comparison of Local-h Adaption on the Fixed Source Problem

# Grid Points	Coarse Mesh	Fine Mesh	Local h 4^2	Peak Flux	Iterations	Time [s]	Time per Iteration
2500	50^2	1^2	None	0.7769	18	2.02	0.112235
40000	200^2	1^2	None	0.7765	32	59.69	1.865464
40000	50^2	4^2	None	0.7765	18	8.03	0.446224
40000	50^2	1^2	Whole Domain	0.7765	18	9.27	0.515152
11500	50^2	1^2	Source Region	0.7765	18	3.90	0.216983

3-Dimensional Numerical Results

The derivation of the ANDO method can easily be extended to 3-dimensional cartesian geometry, as was demonstrated in Chapter 3. The implementation of the ANDO method can also easily be extended to 3-dimensions, as will be demonstrated by example in this section. The homogenous Case 1, with $\Sigma_t = 1$ cm^{-1} and $\Sigma_s = 0.6$ cm^{-1}, was solved for a $2 \times 2 \times 2$ cm volume with 20 uniformly spaced computational cells in each dimension. The S_{10} quadrature set was used for the angular discretization. The 3D flux profile is shown in Figure 28.

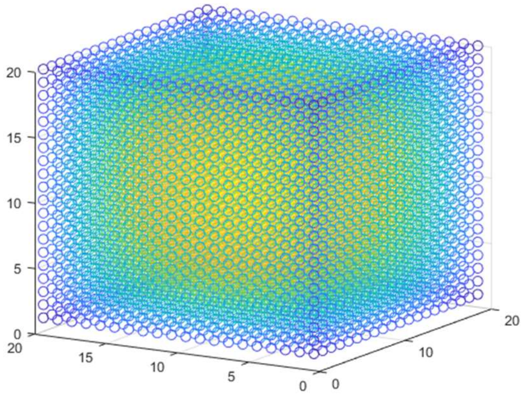


Figure 28: Case 1 3D Flux Profile

The 2D flux profile along each mid-plane of the 3D solution is identical, within machine rounding error. Further, the ANDO method in 3D recovers the ANDO method solution in 2D for top and bottom reflected boundaries.

The Well-Logging Problem was also run in 3D geometry. An S_6 quadrature set was used for the angular discretization and the convergence criteria was set to 10^{-6} for comparison. The two-dimensional geometry was taken as the xz-midplane along the y-axis and the yz-midplane along the x-axis. The 3D flux profile is shown in Figure 29 and the numerical results are tabulated in Table 10. The flux profile and numerical results are qualitatively reasonable, but no mono-energetic 3D well-logging benchmark could be found for a quantitative comparison.

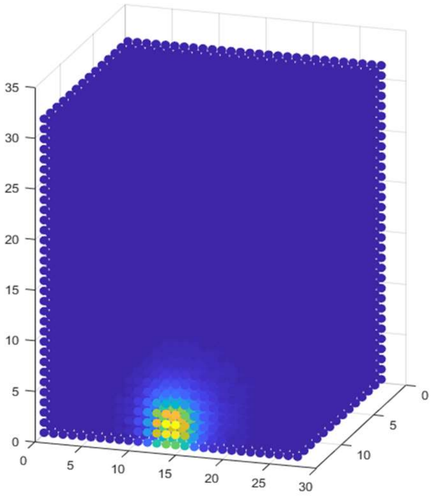


Figure 29: Well-Logging 3D Flux Profile

Table 10: 3D Numerical Results for the Well-Logging Problem

Mesh	D$_1$	D$_2$	D$_1$/D$_2$	Number of iterations	CPU time (minutes)
ANDO-Constant 3D					
7x8	0.12168	0.00077	157.6	22	0.794
14x16	0.30934	0.00113	273.7	28	8.77
28x32	0.34840	0.00129	270.8	34	85.4
56x64	0.36100	0.00138	260.7	22	823

Multi-Group Numerical Results

The derivation of the ANDO method can easily be extended to multi-group, as was demonstrated in Chapter 3. The implementation of the ANDO method can also easily be extended multi-group, as will be demonstrated by example in this section. The test problem consisted of two groups with identically defined total and group scattering cross sections and unity source, which is equivalent to each group having the same energy. Although this is not physically meaningful, it provides a sense check of the method and code as the scalar flux profile for each group should be identical within machine rounding error. This was found to be the case.

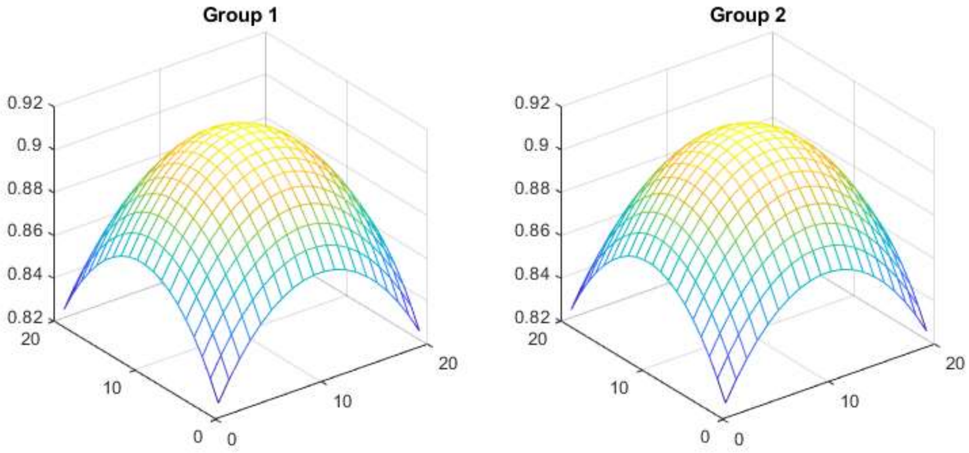


Figure 30: Multi-Group Test Case 1 Flux Profile

For the second sense check, different material properties were used for each respective group. As expected, the flux profile for each group was no longer identical, as shown in the Figure 28. Further, the material properties for each respective group were then switched, such that the flux profiles should be identical but associated with the other group. The difference between the flux profile for the same material properties assigned to a different group was found to be the same within machine rounding error. The L1 error for varying spatial refinements was calculated for each respective group and is shown in Figure 32. The results suggest that the ANDO multi-group method has high accuracy, but a lower order of accuracy than single group. The ANDO multi-group method was also rapidly convergent for this test problem requiring only 8 iteration to reach the machine rounding epsilon of 10^{-16}.

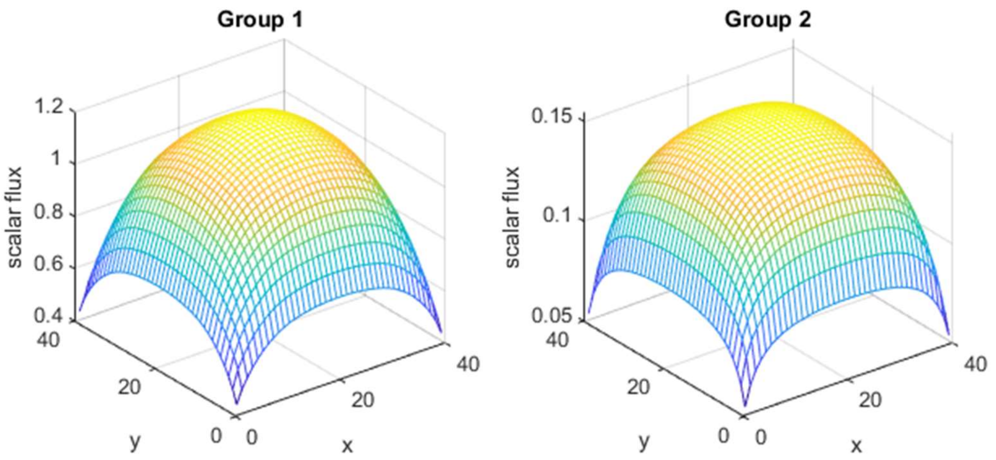


Figure 31: Multi-Group Test Case 2 Flux Profile

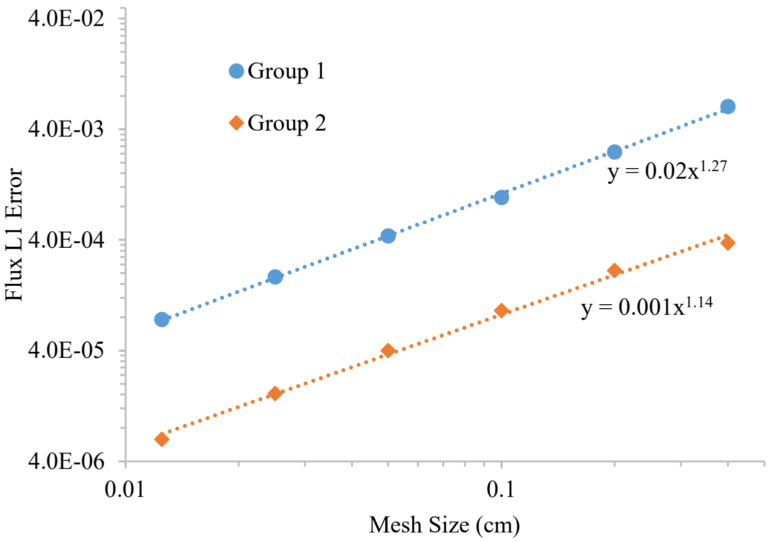


Figure 32: Flux L1 Error Multi-Group Test Case

The results are qualitatively reasonable as the solution is converging with spatial refinement, but time limitations prevented a more in-depth quantitative analysis of ANDO multi-group or comparison to a benchmark.

Chapter 5. Conclusion

Conclusions

The ADNO method is free from spatial truncation error within the computational cell and is limited in accuracy only by the approximation of the transverse leakage, as are all analytical nodal methods. It was shown through numerical experiments that the ANDO method recovers the 1D analytical solution for opposing reflected boundaries. Recovery of the 1D analytical solution was further demonstrated mathematically. The ANDO method has a much greater accuracy than traditional methods based on the finite difference method, such as the DD method. A linear transverse leakage approximation further improves the accuracy of the ANDO method with constant transverse leakage, especially for problems dominated by the ray-effect. However, the ANDO-Quadratic method was not found to be appreciably more accurate than the ANDO-Linear method. Based on the results, higher order approximations do not appear to significantly improve the accuracy of the ANDO method beyond a linear approximation. Further, although the ANDO-Linear method is more accurate, the additional computational cost is often not justified by improved accuracy. Instead, the ANDO-Constant method could be used with a greater spatial discretization to solve the problem in the same amount of time. The ANDO-Refined method has exactly the same accuracy as ANDO-Constant but requires fewer iterations and less time per iteration.

The ANDO method is rapidly convergent compared to traditional methods based on source iteration, such as the DD method. Additionally, the closed form solution for any heterogenous 2^n mesh can be easily obtained to solve the problem without iteration. Further, it was shown that the ANDO-2x2 closed-form solution can be used to improve the convergence and the CPU time per iteration of the ANDO method. The rapid convergence is due to the absorption of the source term in the derivation. The ANDO method requires no acceleration scheme to achieve rapid convergence. However, because the ANDO method iterates over the cell edge values it is not immediately amendable to acceleration schemes that rely on source iteration. The ANDO method has nearly linear computational complexity. The computational complexity of the ANDO method scales with the size of the coarse mesh, regardless of the discretization used in the fine mesh. This effectively improves the convergence of the ANDO-Refined method. The optimal ANDO-Refined fine mesh in terms of CPU time was found to be 4x4 cells, although the optimal fine mesh in terms of iterations is the entire domain.

The ANDO method is based on matrix-vector multiplication, which is highly optimized on modern computing systems. Because of this, the ANDO method is comparable in terms of CPU time per iteration to traditional methods based on finite difference, such as DD, even though a larger number of floating-point operations are needed. Further, the ANDO-2x2 closed form solution reduces the CPU time per iteration compared to the ANDO-Constant method even though the same number of floating-point operations are being performed. This is because memory access becomes more efficient as the size of the contiguously stored data increases. The optimal ANDO-Refined fine mesh refinement is thus limited by the memory capacity of the computing system.

The ANDO method is also highly computationally efficient in that every angular flux value on each cell faces is linearly dependent on every other angular flux value. Under the assumption that there are no non-linear relations between angular flux values, as is the case for the linearized Boltzmann equation used as the governing equation, matrix vector multiplication is the most computationally efficient. Further, it was demonstrated mathematically that the ANDO-Constant method is the most accurate set of coefficients under the constant transverse leakage approximation. The simplicity of the final closed-form solution vector $\boldsymbol{A}$ and matrix $\boldsymbol{B}$ of the ANDO-Constant method improves the accuracy, rate of convergence, CPU time per iteration, and ease of implementation.

Pre-calculation of the solution matrices improves the computational efficiency of the iteration loop. Only unique cells need to be precomputed, rather than every cell in the domain. This makes pre-calculation more efficient and reduces the memory requirements. Further, only the cell interface angular flux values and the precomputed solution matrices must be stored, no additional coefficients need to be stored in memory. The main iteration loop is highly efficient as only matrix vector operations are performed. Reading and writing to memory are also efficiently implemented. The ANDO method does not require any complex math operations or sweeping schemes be programmed in implementation of the method.

Further, the ANDO method is local-p adaptive, allowing for the greater accuracy of the ANDO-Linear method to be employed judiciously to reduce computational cost. Local-h adaption can be used to reduce the number of grid points while still maintaining a desired level of spatial discretization in specific regions. The ANDO method was shown to be robust and possess both asymptotic and positivity preserving. The ANDO method is asymptotic preserving because it is free from spatial truncation error within the computational cell. The ANDO method is positive preserving because it is a conservative method.

The ANDO method can also be easily extended to higher order approximations using the derivation presented in this work. However, the results of the ANDO-Quadratic method suggest that higher order approximations may not achieve significantly greater accuracy. Higher order approximation beyond constant may therefore have limited use, especially when the greater computational cost is considered. As is the case of all nodal methods, the ANDO method can easily be extended to three dimensional cartesian geometry by transverse integration along the additional axis. The ANDO method in three dimensional cartesian geometry was demonstrated in the derivation and numerical results. The ANDO method can also easily be extended to multi-group calculations as was demonstrated in the derivation and numerical results.

Ease of implementation is another benefit of the ANDO method. A single computer code was developed to implement the ANDO method and all its variations presented in this work. The code can be used for an arbitrary level symmetric quadrature order. The code can be used for any user defined heterogenous domain and for non-uniform mesh spacing. Reflective boundaries can be set on any side of the domain for problems with symmetry. ANDO-Reconstruction can be used to reconstruct the scalar flux, or the balance equation can be used for reconstruction. The code is local-p adaptive; any cell can be user defined to use ANDO-C, ANDO-L, ANDO-LL1, or ANDO-Q. The code is local-h adaptive; any cell can be user defined to use any 2^n local refinement. The code can be used for 3-dimensional problems with constant leakage approximation. The code can also automatically use local refinement to speed up certain problems without additional work on the part of the user. The results for ANDO-Refined presented in Chapter 4 are for this automatic speed up feature. All of the above features can also be used with an arbitrary number of groups.

Future Work

Extension of the ANDO method to solve k_{eff} eigen value problems would be the necessary next step to make the ANDO method useful for production codes. Because the ANDO method does not rely on source iteration, acceleration schemes based on source iteration are not amenable to the ANDO method. Therefore, the development of a transport-based acceleration scheme could improve the convergence of the ANDO method. Extension of the ANDO method to a triangular mesh would greatly improve the viability of the method for use in modern computer codes that are not restricted by cartesian geometry. Although the transverse integration used in the ANDO method is dependent on a cartesian coordinate system, nodal methods have been derived on triangular meshes. Finally, it could not be shown definitively that the coefficients used in the ANDO-Constant method are the most accurate possible. Further investigation could prove this to be the case or could uncover an even more accurate set of solution coefficients.

Summary

The results of this work demonstrate that the ANDO method possesses many favorable properties and has great potential for implementation in commercial codes. The ADNO-Constant method is free from spatial truncation error within the computational cell. It was shown numerically and mathematically that the ANDO-Constant method recovers the 1D analytical solution. The ANDO method gives a truly closed-form solution that solves for all outgoing angular flux values from all incoming angular flux values simultaneously. It was also shown mathematically that the coefficients of the solution matrices of the ANDO-Constant method are the most accurate. Further, the ANDO method can give the closed-form solution on any 2^n heterogenous domain. The ANDO method is limited in accuracy only by the approximation of the transverse leakage. It was shown that higher order approximations improve the accuracy of the ANDO method. It was also shown that the ANDO method can easily be extended to any n^{th} order polynomial approximation. However, when total CPU time is considered, the ANDO-Constant method is favorable. Local-p adaption can be used to improve the accuracy of the solution in specific regions for a minor computational penalty. The ANDO method is rapidly convergent and has nearly linear computational complexity. Iteration over groups of cells using the 2x2 closed-form solution effectively reduces the number of grid points and further improves convergence. Local-h adaption can also be used to reduce the number of grid points. The high computational efficiency of the ANDO method achieves excellent performance in terms of CPU time per iteration. Iteration over groups of cells using the 2x2 closed-form solution further improves the computational efficiency.

The ANDO method is also robust, possessing both asymptotic preserving and positivity preserving. The ANDO method can easily be extended to 3-Dimensional cartesian geometry. The ANDO method can easily be extended to multi-group transport problems.

The high accuracy, fast convergence, and computational efficiency of the ANDO method can significantly reduce the computation time required for deterministic transport calculations. Higher order approximations, local-p, and local-h adaption can further improve the accuracy of the ANDO method. The 2x2 closed-form solution can further reduce both the number of iterations and the CPU time per iteration. With further development, the ANDO method could find widespread application in production codes.

Bibliography

[1] A. BADRUZZAMAN," An efficient algorithm for nodal transport solutions in multidimensional geometry," Nucl. Sci. Eng. 89, 281 (1985).

[2] R.D. LAWRENCE, "Progress in nodal methods for the solution of the neutron diffusion and transport equations," Progress in Nuclear Energy. Vol. 17, Issue 3, Pages 271-301 (1986).

[3] E.E. LEWIS, and W.F. MILLER, "Computational methods of neutron transport," United States, (1984).

[4] M. R. WAGNER, "A nodal discrete-ordinates method for the numerical solution of the multidimensional transport equation," Proc. Conf. Computational Methods in Nuclear Engineering, p. 4-117, Williamsburg, VA, 23-25 April, American Nuclear Society (1979).

[5] R. D. LAWRENCE and J. J. DORNING, "A Discrete Nodal Integral Transport Theory Method for Multidimensional Reactor Physics and Shielding Calculations," Proc. Conf. Advances in Reactor Physics and Shielding, Sun Valley, Idaho, September 14-19, 1980, p. 840, American Nuclear Society (1980).

[6] W. F. WALTERS, "Augmented Weighted-Diamond Form of the Linear-Nodal Scheme for Cartesian Coordinate Systems," *Nucl. Sci. Eng.*, 92, 192 (1986).

[7] R. C. DE BARROS, and, E. LARSEN, "A Numerical Method for Multigroup Slab-Geometry Discrete Ordinates Problems with no Spatial Truncation Error," Transport Theory and Statistical Physics, Vol 20, Issue 5-6, Pages 441-462 (1991).

[8] R. C. DE BARROS, and, E. LARSEN, "A Numerical Method for One-Group Slab-Geometry Discrete Ordinates Problems with No Spatial Truncation Error," Transport Theory and Statistical Physics, Vol 104, Issue 3, Pages 199-208 (1990).

[9] R. C. DE BARROS, "A Spectral-Nodal Method for the Solution of Discrete Ordinates Problems in One and Two-Dimensional Cartesian Geometry," Ph. D. Dissertation University of Michigan, (1990).

[10] J. A. M., de MELLO, R. C. BARROS, "An Exponential Spectral Nodal Method for One-Speed x, y-Geometry Deep Penetration Discrete Ordinates Problems," Ann. Nucl. Energy., 29, 1855 (2002).

[11] MENEZES, W.A., ALVES FILHO, H. AND BARROS, R.C. (2018) 'An analytical nodal method for energy multi-group discrete ordinates transport calculations in two-

dimensional rectangular geometry’, Int. J. Nuclear Energy Science and Technology, Vol. 12, No. 1, pp.66–80.

[12] M.T. VILHENA, L.B. BARICHELLO, “A New Analytical Approach to Solve the Neutron Transport Equation,” Nuclear Physics and Radiation Physics, Vol. 56, Pages 334-336 (1991).

[13] L.B. BARICHELLO, M.T. VILHENA, “A General Analytical Approach to the One-Group, One-Dimensional Transport Equation,” Nuclear Physics and Radiation Physics, Vol. 58(3), Pages 182-184 (1993).

[14] M.T. VILHENA, L.B. BARICHELLO, “An analytical solution for the multigroup slab geometry discrete ordinates problems,” Transport Theory and Statistical Physics, Vol. 24, Issue 9, Pages 1337-1352 (1995).

[15] J. R. ZABADAL, M.T. VILHENA, L.B. BARICHELLO, Proc. 9th Brazilian Meeting on Reactor Physics and Thermal Hydraulics, p.90 (1993).

[16] J. R. ZABADAL, M.T. VILHENA, L.B. BARICHELLO, “Solution of the Three-Dimensional One-Group Discrete Ordinates Problems by The LTSN Method”, Annals of Nuclear Energy, Vol. 22, Issue 2, Pages 131-134 (1995).

[17] J. R. ZABADAL, M.T. VILHENA, L.B. BARICHELLO, “An Analytical Solution for the Two-Dimensional Discrete Ordinates Problem in a Convex Domain”, Progress in Nuclear Energy, Vol. 31, Issue 3, Pages 225-228 (1997).

[18] C. H. BATISTELA, M. T. VILHENA, V. BORGES, “Criticality by the LTSN Method”, Journal of Nuclear Science and Technology, Vol. 34, Issue 6, Pages 603-606 (1997)

[19] R.M.F. VARGAS, M.T.M.B. De VILHENA, “Analytical Solution of the Discrete Ordinates Problem by the Decomposition Method,” Annals of Nuclear Energy, 24, (1997).

[20] G. ADOMIAN, “A Review of the Decomposition Method in Applied Mathematics,” Journal of Mathematical Analysis and Applications, Vol. 125, Issue 2, Pages 501-544 (1988).

[21] L. B. BARICHELLO, R. D. GARCIA, C. E. Siewert, “A Spherical-Harmonics Solution for Radiative-Transfer Problems with Reflecting Boundaries and Internal Sources”, Journal of Quantitative Spectroscopy and Radiative Transfer, Vol. 60, Issue 2, Pages 247-260 (1998).

[22] L. B. BARICHELLO, C. E. Siewert, “The FN Method for Spectral-Line Formation by Completely Noncoherent Scattering”, Journal of Quantitative Spectroscopy and Radiative Transfer, Vol. 60, Issue 2, Pages 261-276 (1998).

[23] L. B. BARICHELLO, C. E. Siewert, “A Discrete-Ordinates Solution for a Non-Grey Model with Complete Frequency Redistribution”, Journal of Quantitative Spectroscopy and Radiative Transfer, Vol. 62, Issue 6, Pages 665-675 (1999).

[24] L. B. BARICHELLO, C. E. Siewert, “A New Version of The Discrete-Ordinates Method”, Proceedings of the 2nd International Conference on Computational Heat and Mass Transfer, Federal University of Rio de Janeiro, Brazil, October 22-26 (2001).

[25] B. N. Datta, “Numerical Linear Algebra and Applications”. Brooks/Cole Publishing Co, Pacific Grove, USA (1995).

[26] L. B. BARICHELLO, L. C. CABRERA, J. F. PROLO FILHO, “An Analytical Approach for a Nodal Scheme of Two-Dimensional Neutron Transport Problems,” Annals of Nuclear Energy, Vol. 38, Issue 6, Pages 1310-1317 (2011).

[27] L. B. BARICHELLO, L. C. CABRERA, J. F. PROLO FILHO, “An Analytical Discrete Ordinates Solution for Two Dimensional Problems Based on Nodal Schemes,” 2009 International Nuclear Atlantic Conference, Rio de Janeiro, RJ, Brazil, September 27 to October 2 (2009).

[28] C. B. PICOLOTO, A. TRES, R. D. DE CUNHA, L. B. BARICHELLO, “Two-Dimensional Neutron Transport Problems with Reflective Boundary Conditions: An Analytical Approach,” 2013 International Nuclear Atlantic Conference, Recife, PE, Brazil, November 24-29 (2013).

[29] C. B. PICOLOTO, A. TRES, R. D. DE CUNHA, L. B. BARICHELLO, “Closed-Form Solutions for Nodal Formulations of Two Dimensional Transport Problems in Heterogeneous Media,” Annals of Nuclear Energy, Vol. 86, Pages 65-71 (2015).

[30] C. B. PICOLOTO, R. D. DE CUNHA, R. C. BARROS, L. B. BARICHELLO, “An Analytical Approach for Solving a Nodal Formulation of Two-Dimensional Fixed-Source Neutron Transport Problems with Linearly Anisotropic Scattering,” Annals of Nuclear Energy, Vol. 98, Pages 193-201 (2017).

[31] S.R. CROMIANSKI, K. RUI, L.B. BARICHELLO, “A study on boundary fluxes approximation in explicit nodal formulations for the solution of the two-dimensional neutron transport equation,” Progress in Nuclear Energy. Vol. 110, Pages 354-363 (2019).

[32] D. WANG and T. BYAMBAAKHUU, “A New Analytical SN Solution in Slab Geometry," Trans. AM. Nucl. Soc., 117 (2017).

[33] W.F. WALTERS, R.D. O’Dell, “Nodal methods for discrete-ordinates transport problems in (x,y) geometry,” Conference: ANS/ENS joint topical meeting on mathematical methods in nuclear engineering, Munich, F.R. Germany, 27 April (1981).

[34] Y. AZMY, “The Weighted Diamond-Difference Form of Nodal Transport Methods,” Nucl. Sci. Eng. 98, 29 (1988).

[35] J. ROCHELEAU, D. WANG, "A Novel Analytical Nodal Method for Solution of the SN Transport Equation," 2020 ANS Summer Meeting, Phoenix, AZ, June 7-11, 2020.

Appendix A: Other ANDO Approaches

Several other approaches attempting to establish a linear distribution for the transverse leakage using only the constant incoming angular flux on each cell face were derived. Just as the LL1 approximation, the goal is to derive a coefficient matrix that is the same size as ANDO-C but is more accurate. However, none of these approaches would improve the accuracy of ANDO-C because, as was shown in Chapter 3, ANDO-C already defines the most accurate set of coefficients possible for a constant leakage approximation. A brief description of other ANDO approaches that were attempted is provided in this Appendix along with a brief explaination of the shortcoming of the respective methods.

ANDO-LL1 C: Establish a linear distribution from the cell center angular flux and outgoing flux. This approach is different from the ANDO-LL1 method because the cell center angular flux, rather than the cell average angular flux is used. This method was never implemented because it was determined that the modification of the solution matrix by the ANDO-LL1 method is less accurate than ANDO-C and that the ANDO-LL1 C method would be less accurate. Further, the cell center angular flux in the x and y direction is often not equal.

ANDO-C Analytical: Determine the flux distribution in the x-direction analytically. This approach introduced a non-linearity that prevents the ODE from being solved.

ANDO-E: Establish exponential distribution from adjacent cells. This approach was never implemented because it is not clear what value should be used for the exponent. The eigen values are too large and can only be applied to the eigenspace solution. The material cross-sections appear to be too small based on preliminary testing.

ANDO-C Exponential average: Determine an exponential distribution on the cell face from the average angular flux on the cell face. Determination of the value to use for the exponent, as described above, presents an issue. An attempt to derive and implement this method resulted in a matrix inversion that could not be evaluated due to poor scaling. Implementation of this method failed at matrix inversion and the method was abandoned.

ANDO-C: cell average angular flux approach. In the ANDO method, only the cell edge angular flux is evaluated, although the angular flux at any point along the centerline of the cell can be readily calculated. Because the cell centerline in the x direction and y direction, respectively, cross at the cell center, a method was devised to improve convergence by introducing additional equations relating the cell center angular flux calculated in each respective direction. However, the cell center angular flux is not equal, and the additional relation is invalid. Further, implementation of this method failed at matrix inversion and the method was abandoned.

ANDO-L: cell average angular flux approach. A method was devised to determine the slope of the transverse leakage by introducing an additional relation between the cell average angular flux calculated in each direction. However, as noted earlier, the cell average angular flux calculated in the x direction and in the y direction, respectively, is equal only at convergence so the additional relation is invalid. Further, implementation of this method failed at matrix inversion and the method was abandoned.

Additionally, none of these other approaches would improve the accuracy of ANDO-C. It is shown in Chapter 3 that the coefficients of the ANDO-C method are already the most accurate under a constant leakage approximation and no other method, therefore, could achieve greater accuracy for the same number of coefficients. Attempts at improving the convergence by modifying the coefficients is also unlikely to succeed because the ANDO method already gives the exact nodal solution subject to the cell boundary conditions.

Appendix B: Additional Implementation Details

As emphasized previously, the ANDO method is easy to implement because all the details are handled analytically in extensive algebraic calculation. However, correct and consistent ordering of the quadrature points is of utmost importance. Explicit descriptions and code snippets are provided for the mapping matrices to supplement the implementation details provided at the end of chapter 3.

The quadrature points:

The following code snippet is provided to generate the vectors $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ from the cosine angles $\Omega_\mathrm{i} = (\mu_i, \eta_i)$ for the level symmetric quadrature set. The angles are assumed order in increasing magnitude in the vector SN with weights W and assumed to be of length n for S_n.

For S_{12}:

```
n  = 12;
SN = [0.1672126,0.4595476,0.6280191,0.7600210,0.8722706,0.9716377];
w  = [0.0707626, 0.0558811, 0.0373377, 0.0502819, 0.0258513];
W  = w([1 2 2 3 4 3 3 5 5 3 2 4 5 4 2 1 2 3 3 2 1]);
```

The algorithm to uniquely pair the quadrature points Ω is

```
Sn = n*(n+2)/8;
F = zeros(1,Sn);  fi=1;
R = zeros(1,Sn);  ri=1;

for i = 1:n/2
   for f = 1:i
       F(fi) = SN(f);
       fi=fi+1;
   end
   for r = i:-1:1
       R(ri) = SN(r);
       ri=ri+1;
   end
end

mu  = [-R  F  R -F];
eta = [-F -R  F  R];
wgt = [ W  W  W  W];
```

The map $\boldsymbol{T_{NM}}$ and $\boldsymbol{T_{MN}}$:

The quadrature points are ordered with respect to μ in the x direction and with respect to η in the y direction. Such an ordering simplifies the derivation but necessitates a mapping for the transverse leakage terms. Let the matrix $\boldsymbol{T_M}$ map the points from the original ordering Ω to ordering μ and $\boldsymbol{T'_M}$ map the points from ordering μ to ordering Ω. The following code snippet constructs the mapping matrices for the vector of quadrature points μ of length Sn, where Sn = 84 for S_{12}.

```
mu_o = mu;
Sn_half = Sn/2;
for m = 1:Sn_half
    [~, index] = max(mu_o);
    TM(m, index) = 1;
    TM_(index, m) = 1;
    mu_o(index) = 0;
end
for m = Sn_half+1:Sn
    [~, index] = min(mu_o);
    TM(m, index) = 1;
    TM_(index, m) = 1;
    mu_o(index) = 0;
end
```

Where $\boldsymbol{T_M}$ = `TM` and $\boldsymbol{T'_M}$ = `TM_`. The same algorithm can be used to construct the mapping matrices $\boldsymbol{T_N}$ and $\boldsymbol{T'_N}$ for η. Finally, multiply the mapping matrices as follows to get $\boldsymbol{T_{MN}}$ and $\boldsymbol{T_{NM}}$:

$$\boldsymbol{T_{MN} = T_M T'_N}$$

```
MN = TM*TN_;
```

$$\boldsymbol{T_{NM} = T_N T'_M}$$

```
NM = TN*TM_;
```

The mapping matrices can now be used in the matrices defined in Chapter 2.

The map $\boldsymbol{T_G}$:

For ANDO-Multi-group, the solution process and implementation are kept the same as the mono-energetic solution by the application of the group mapping matrix $\boldsymbol{T_G}$.

$$\begin{bmatrix} \psi_{g_1+} \\ \cdots \\ \psi_{G+} \\ \psi_{g_1-} \\ \cdots \\ \psi_{G-} \end{bmatrix} = \boldsymbol{T_G} \begin{bmatrix} \psi_{g_1+} \\ \psi_{g_1-} \\ \cdots \\ \psi_{G+} \\ \psi_{G-} \end{bmatrix}$$

The angular flux values are ordered in the positive and negative direction by group increasing by group to simplify setting up problem but are ordered in the positive direction increasing by group and the negative direction by group to simplify the derivation. It is not difficult to construct or code the group mapping matrix. The matrices for two and three group are explicitly shown below and can be hardcoded if the number of groups is known *a priori*.

$$\boldsymbol{T_G} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix}$$

$$\boldsymbol{T_G} = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}$$

$\boldsymbol{T_G} = \boldsymbol{T_G^{-1}}$ and, therefore, $\boldsymbol{I} = \boldsymbol{T_G^2}$

For completeness the following code snippet is provided to construct the group mapping matrix for an arbitrary number of groups Ng.

```
SG = Sn*Ng;
Sn_half = Sn/2;
TG = zeros(SG);
for G = 1:Ng
    for g = 1:Ng
       TG(((G-1)*Sn_half*Ng+(g-1)*Sn_half+1):((G-1)*Sn_half*Ng+(g-
       1)*Sn_half+Sn_half),(G-1)*Sn_half+(g-1)*Ng*Sn_half+1):((G-1)*Sn_half+(g-
       1)*Ng*Sn_half+Sn_half) ) = eye(Sn_half);
    end
end
```

where $\boldsymbol{T_G}$ = `TG`.