A Two-Grid, Nonlinear Diffusion Acceleration Method for the S_N Equations with Neutron Upscattering

by

Marissa Ramirez Zweiger

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Nuclear Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Rachel Slaybaugh, Chair
Professor Phillip Colella
Professor Massimiliano Fratoni

Summer 2018

Contents

Chapter 1

Introduction

The field of computational neutronics is generally concerned with modeling neutrons in a nuclear system. These systems include nuclear reactors or shielding problems where there exists a fixed source of neutrons. The materials used in these systems have a variety of properties including how they effect the energy of the neutron. Many materials that are commonly used in nuclear applications, particularly those containing hydrogen, exhibit a property known as upscattering, where lower energy neutrons are "bounced" to higher energies. Mathematically, a scattering matrix representing the scattering cross sections from group g to g$'$ with the highest energy in group 1 and the lowest in group G would be upper triangular if there is no upscattering. Only the entries $[\mathrm{g}, \mathrm{g}' \geq \mathrm{g}]$ are nonzero as without upscattering, the scattering from a lower energy group to a higher energy group is zero. A triangular matrix has nice computational properties that make the solving of the problem easier. However, when upscattering is present, the triangularity of the matrix is broken and other solution techniques must be employed. The most commonly used method in neutron transport calculations for problems with upscattering is known as the Gauss-Seidel Method.

In this work, we explore a method known as nonlinear diffusion acceleration (NDA). It is also known as coarse-mesh finite difference and is a well-known technique applied to accelerate the scattering convergence in neutronics calculations. In multigroup neutronics problems, NDA is effective in conjunction with Gauss-Seidel (GS) iteration in energy if there is little upscattering. However, when upscattering manifests, which is common with thermal neutrons, the efficiency of GS-NDA degrades as extra iterations are required for convergence in energy [8].

To remedy the issue in GS with upscattering, an energy two-grid (TG) acceleration scheme was first developed to approximate iteration error by solving a one-group diffusion-like equation with artificial material properties generated by using the scattering eigenspectrum [2]. Later, a transport TG (TTG) method was developed that approximates the energy error using a consistent S_N solver in multi-D [3]. Inspired by the previous studies, we derive a TG scheme for the NDA equation with GS iteration.

1.1 Previous Work

To accelerate the convergence of source iteration in transport calculations, Nonlinear Diffusion Acceleration (NDA) was developed [**?**]. It pairs a lower order equation with a higher order, drift diffusion equation. As the higher order equation is not conservative, formally the method is inconsistent: The scalar flux and current may not be equal upon convergence, although they are in the limit as the spatial mesh is refined. However, second order accuracy can still be maintained with the high-order equation generally giving a more accurate shape and the conservative low-order equation giving a more accurate magnitude. The combination of the two equations has been found to be more accurate than either equation alone. [9]

Since the development of NDA, it has been tried with a variety of higher order equations, [9][10] as well as different spatial discretizations [9][**?**]. In this work we explore NDA with the Self-Adjoint Angular Flux Equation (SAAF) and a continuous finite element discretization.

The steady-state transport equation is dependent on space, angle, and energy. It is often solved via a series of nested iterations. The various iteration methods and how they are used with each other are described in detail in Ch. 3. The Nonlinear Diffusion Acceleration happens in the source iteration, where angle and energy are fixed. When energy is discretized into more than one energy group, an outer layer of iteration is introduced. One of the most commonly used methods for iterating over energy groups is known as Gauss-Seidel. Gauss-Siedel is guaranteed to converge, however in problems with significant upscattering the time it takes to reach convergence can become arbitrarily slow.

A number of techniques to accelerate Gauss-Seidel convergence have been developed [2] [3], although to our knowledge none have yet been paired with NDA.The primary techniques used in commercial transport software rely on a rebalance scheme and coarse-mesh finite-difference diffusion. Although these methods are widely used and successful for acceleration, they are very sensitive to the coarse mesh size. Rebalancing with too fine a mesh may be divergent, and an overly coarse mesh degrades performance. [3]

While for standard problems, the proper mesh size is generally well understood, this is not the case for shielding problems. Adams and Morel developed an upscatter acceleration scheme known as the Two-Grid method. An estimation of the error at each Gauss-Seidel iteration is calculated using a collapsed in energy, one-group diffusion equation and energy eigenvector of the Gauss-Seidel iteration matrix. These make up the correction term to the scalar flux at each group which is applied at each iteration. In one dimensional calculations, Adams and Morel have demonstrated their method to be very efficient for thermal upscattering problems [2].

1.2 The Boltzmann Transport Equation

The angular neutron flux of a reactor, ψ can be described by the steady state Boltzmann Transport equation.

$$[\hat{\Omega}\cdot\nabla+\Sigma(\vec{r},E)]\psi(\vec{r},\hat{\Omega},E)=\chi(E)\int_0^\infty dE'\nu\Sigma_f(\vec{r},E')\int_{4\pi}d\hat{\Omega}'\psi(\vec{r},\hat{\Omega},E') \\ +\int_0^\infty dE'\int_{4\pi}d\hat{\Omega}'\Sigma_s(\vec{r},E'\rightarrow E,\hat{\Omega}'\cdot\hat{\Omega})\psi(\vec{r},\hat{\Omega}',E') \tag{1.1}$$

where $\hat{\Omega}$ represents the angle; $\vec{r}$, the position vector; E, the energy; Σ, the total macroscopic cross-section; Σ_f, the macroscopic fission cross-section; Σ_s, the macroscopic scattering cross section; χ, the energy distribution; and ν, the average number of neutrons per fission.

We apply an incident boundary condition, where for $\hat{n}\cdot\hat{\Omega}<0$, with $\hat{n}$ being the outward normal on the boundary $\partial\mathcal{D}$

$$\psi(\vec{r},\hat{\Omega},E)=\psi^{inc}(\vec{r},\hat{\Omega},E),\quad \vec{r}\in\partial\mathcal{D} \tag{1.2}$$

1.2.1 Forms of the Transport Equation

There are are several forms of the transport equaution that are of interest in the field of nuclear energy. In this work we present all methods in fixed-source form, however they can all be easily extended to k-eigenvalue form for criticality calculations.

Fixed Source Form

In the fixed source form, we assume there is no fission and that there is an external neutron source, Q. We use the same boundary conditions as above.

$$[\hat{\Omega}\cdot\nabla+\Sigma(\vec{r},E)]\psi(\vec{r},\hat{\Omega},E)= \\ \int_0^\infty dE'\int_{4\pi}d\hat{\Omega}'\Sigma_s(\vec{r},E'\rightarrow E,\hat{\Omega}'\cdot\hat{\Omega})\psi(\vec{r},\hat{\Omega}',E')+Q \tag{1.3}$$

k-Eigenvalue Form

If the chain reaction is self-sustaining and time-independent, the reactor is known as "critical." To maintain criticality, the asymptotic neutron distribution must not be changing over time. This behavior can be described by a parameter, k, the ratio of neutrons in two successive generations. We scale ν in Eq. (1.1) by k to express the deviation from critical. This gives the following equation,

$$[\hat{\Omega}\cdot\nabla+\Sigma(\vec{r},E)]\psi(\vec{r},\hat{\Omega},E)=\frac{\chi(E)}{4\pi k}\int_0^\infty dE'\nu\Sigma_f(\vec{r},E')\int_{4\pi}d\hat{\Omega}'\psi(\vec{r},\hat{\Omega},E') \\ +\int_0^\infty dE'\int_{4\pi}d\hat{\Omega}'\Sigma_s(\vec{r},E'\rightarrow E,\hat{\Omega}'\cdot\hat{\Omega})\psi(\vec{r},\hat{\Omega}',E') \tag{1.4}$$

Again using the same boundary conditions. (1.4) is simply an eigenvalue problem that with some algebraic manipulation can be thought of in the form $A\psi=\frac{1}{k}F\psi$ and solved via standard eigenvalue solvers.

1.3 Space-Angle Approximations of Interest

In this work we are primarily concerned with the fixed source form of the transport equation, although all of our methods can be used with the eigenvalue form by replacing the fixed source Q with $\frac{1}{k}\nu\Sigma_f$. For ease we assume our scattering and fixed sources are isotropic, which gives the following form

$$[\hat{\Omega}\cdot\nabla+\Sigma(\vec{r},E)]\psi(\vec{r},\hat{\Omega},E)= \int_0^\infty \frac{1}{4\pi}\Sigma_s(\vec{r},E'\rightarrow E)dE'\int_{4\pi}d\hat{\Omega}'\psi(\vec{r},\hat{\Omega}',E')+\frac{1}{4\pi}Q \tag{1.5}$$

This equation gives the angular flux. To find the scalar flux, ϕ, we must integrate over all directions.

$$\phi(\vec{r},E)=\int_{4\pi}\phi(\vec{r},\hat{\Omega},E)d\Omega. \tag{1.6}$$

1.3.1 Self-Adjoint Angular Flux

Discretizing the traditional transport equation (1.5) in space, particularly using the finite element method presents a number of challenges [7]. A finite element spatial discretization is generally favorable as it produces as symmetric positive-definite (SPD) linear system. SPD systems can be solved using a number of well-studied, robust solution techniques. In order to make use of these properties, we must rearrange the standard transport equation into a form that is more computationally friendly, a self-adjoint form [7].

The monoenergetic version of the Self-Adjoint Angular Flux equation appropriate for S_n calculations is as follows:

$$-\vec{\Omega}\cdot\nabla\frac{1}{\Sigma_t}\vec{\Omega}\cdot\nabla\psi+\Sigma_t\psi=\frac{1}{4\pi}[\Sigma_s\phi+Q-\vec{\Omega}\cdot\nabla\frac{(\Sigma_s\phi+Q)}{\Sigma_t}] \tag{1.7}$$

Where $\vec{\Omega}$ are the angles, Σ_t is the total cross section, ψ is the angular flux, Σ_s is the scattering cross section, ϕ is the scalar flux, and Q is the fixed source.

1.3.2 The Diffusion Equation

To simplify even further, we employ a commonly used approximation known as the diffusion equation. To derive, assume all neutrons have the same energy and consider the neutron balance within an infinitesimal volume centered at a point, r. Under steady state conditions, neutron conservation requires

$$\textit{neutrons leaking out} + \textit{neutrons absorbed} = \textit{source neutrons emitted}. \tag{1.8}$$

We describe the neutrons leaking out as the rate of the current, J, in all directions, the neutrons absorbed is the absorption cross section times the scalar flux, $\Sigma_a\phi$, and the source neutrons are represented by the source variable, Q.

$$\nabla\cdot\vec{J}(\vec{r})+\Sigma_a(\vec{r})\phi(\vec{r})=Q(\vec{r}) \tag{1.9}$$

Using Fick's Law which relates the current to the flux, $\vec{J}(\vec{r}) = -D(\vec{r})\nabla\phi(\vec{r})$ where $D = 1/3\Sigma_t$, we get the diffusion approximation

$$-\nabla \cdot D(\vec{r})\nabla\phi(\vec{r}) + \Sigma_a\phi(\vec{r}) = Q(\vec{r}). \tag{1.10}$$

While the diffusion equation is much easier to solve, due to the assumptions made in the Fick's Law approximation, it is not valid near boundaries where material properties change dramatically, near localized sources, or in strongly absorbing media [6].

1.3.3 Nonlinear Diffusion Acceleration

This work presents an acceleration to a method known as Nonlinear Diffusion Acceleration (NDA). NDA reformulates the transport equation as a correction to the diffusion equation and uses a two step process to solve. For reference, we repeat the derivation of the low-order NDA equation found in [9] with small modifications assuming no fission source and vacuum boundary conditions. Consider the first order, one-group, fixed-source, steady-state $S_N$transport equation with isotropic scattering.

$$\vec{\Omega} \cdot \nabla\psi\left(\vec{r}\right) + \Sigma_{\mathrm{t}}\left(\vec{r}\right)\psi = \frac{1}{4\pi}\Sigma_{\mathrm{s}}\left(\vec{r}\right)\phi\left(\vec{r}\right) + \frac{1}{4\pi}Q\,. \tag{1.11}$$

Integrate over all angles to obtain the zeroth moment equation

$$\nabla \cdot \vec{J} + \Sigma_a\phi = Q \tag{1.12}$$

where $\vec{J} = \int_{4\pi}\vec{\Omega}\psi$ and Σ_a is the absorption cross section. Now consider the first moment equation:

$$\nabla \cdot \overleftrightarrow{P} + \Sigma_t J = 0 \tag{1.13}$$

where $\nabla \cdot \overleftrightarrow{P} = \int_{4\pi}\vec{\Omega}\vec{\Omega} \cdot \nabla\psi$. It can be rewritten as:

$$J = -\frac{1}{\Sigma_t}\nabla \cdot \overleftrightarrow{P}. \tag{1.14}$$

By adding and subtracting the diffusion coefficient, $D = \frac{1}{3\Sigma_t}$, times the gradient of the flux, it takes the form of a correction to Fick's Law.

$$J = -D\nabla\phi + D\nabla\phi - \frac{1}{\Sigma_t}\nabla\overleftrightarrow{P} = -D\nabla\phi - \vec{\mathbf{D}}\phi \tag{1.15}$$

where

$$\vec{\mathbf{D}}(\psi) = \frac{\int_{4\pi}[\frac{1}{\Sigma_t}\vec{\Omega}\vec{\Omega} \cdot \nabla\psi] - D\nabla\phi^{ho}}{\phi^{ho}}. \tag{1.16}$$

Where $\psi^{ho} = \int_{4\pi}\psi(\Omega, \vec{r})d\Omega$ where ψ is the solution of the higher order equation. Plugging (1.15) into (1.12) we have the following NDA equation:

$$\nabla \cdot (-D\nabla\phi - \vec{\mathbf{D}}\phi) + \Sigma_a\phi = Q\,. \tag{1.17}$$

Because NDA uses the solution from a higher-order in calculating the drift vector, it will not give an exact correction. While NDA maintains much of the accuracy of the higher-order equation when compared to diffusion, the two are not exactly equal.

1.3.4 Coupling NDA and SAAF

In this work, we use SAAF as the higher order equation and NDA to accelerate the convergence of source iteration. The implementation of the algorithm is outlined below:

1. Intitialize system, by setting $\vec{\mathbf{D}}$ to 0 and solving (1.17) to get ϕ^0
2. Loop Until Convergence:
   (a) Solve (1.7) for ψ^l using ϕ^{l-1} on RHS.
   (b) Calculate drift vector, (1.16), using ψ^l
   (c) Solve (1.17) for ϕ^l
   (d) Check ϕ^{l-1}, ϕ^l for convergence
3. Return ϕ

While we are able to replicate the results of [10], showing a significant reduction in the number of source iterations necessary when using NDA with SAAF as compared to SAAF alone, NDA only acclerates one layer of iteration. The presence of upscattering introduces another layer of iteration: Gauss-Seidel iteration in energy. In the following section we derive an acceleration scheme for the outer layer of iteration.

1.4 Methods of Discretization

The angular flux, ψ, is a function of space, angle, and energy. In the solution process, each one of those dimensions is discretized. There are several choices that have to be made regarding discretizations. In this work, we endeavor to show equation forms that are discretization agnostic as well as showing formulations unique to the particular discretization methods we chose to implement.

1.4.1 Angular Discretization

Angular discretization on the left hand side of (1.1) is handled via the discrete ordinates (S_N) method, a finite-element collocation method [5]. We assume the sources are isotropic and do not perform any additional expansion, however they can be expanded via Spherical Harmonics and the methods will still hold.

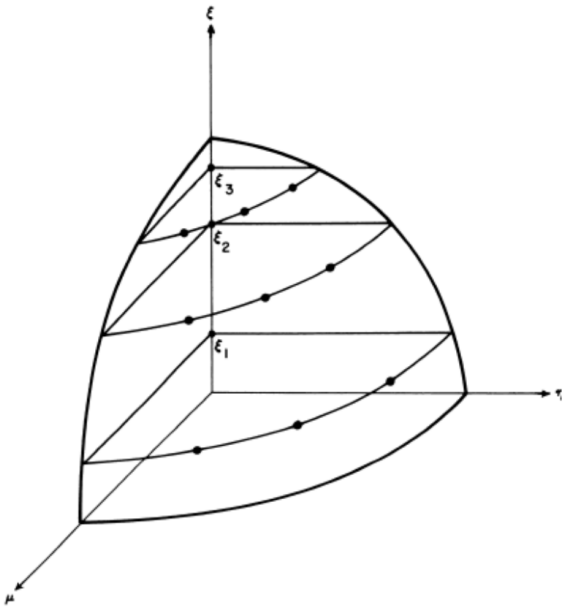


Figure 1.1: Equally-Weighted Gauss-Chebyshev Quadrature Points [5]

The S_N method evaluates the equation at number of discrete angles or "ordinates" and then sums over all angles with their given weights to perform a quadrature. To choose quadrature points, an octant of the unit sphere is discretized into several levels. At each level several nodes are chosen. We use a Gauss-Chebyshev angular quadrature set, which can be thought of as a product set, combining a one dimensional Gaussian Quadrature along the polar angles and an equally-weighted Chebyshev quadrature along the azimuthal angles [4]. For a set of angles, M, a function integrated in angle is discretized as follows:

$$\int \Omega f(\Omega) d\Omega = \sum_{\mathrm{m}=1}^{M} w_{\mathrm{m}} f_{\mathrm{m}}. \tag{1.18}$$

The discretized, steady-state, S_N transport equation is given as follows,

$$\vec{\Omega}_{\mathrm{m}} \cdot \nabla \psi_{\mathrm{m}} (\vec{r}) + \Sigma_{\mathrm{t}} (\vec{r}) \psi_{\mathrm{m}} = \frac{1}{4\pi} \Sigma_{\mathrm{s}} (\vec{r}) \phi (\vec{r}) + \frac{1}{4\pi} Q \tag{1.19}$$

where m is the angular index and $\phi = \sum_{\mathrm{m}=1}^{M} \omega_{\mathrm{m}} \psi_{\mathrm{m}}$.

1.4.2 Energy Discretization

In our treatment of energy, we divide the full energy spectrum into several energy groups. By convention, the highest energy group is given the index, 1, with the index number going up until it reaches the lowest energy group. In expanding to multiple energy groups, we must take into account scattering from one group, g′ to another, g, denoted as $\mathrm{g}' \rightarrow \mathrm{g}$.

The energy discretized, steady-state, transport equation is

$$\vec{\Omega}\cdot\nabla\psi_{\mathrm{g}}\left(\vec{r}\right)+\Sigma_{\mathrm{t,g}}\left(\vec{r}\right)\psi_{\mathrm{g}}=\frac{1}{4\pi}\sum_{\mathrm{g}'=1}^{G}\Sigma_{\mathrm{s,g}'\to\mathrm{g}}\left(\vec{r}\right)\phi_{\mathrm{g}'}\left(\vec{r}\right)+\frac{1}{4\pi}Q_{\mathrm{g}}. \tag{1.20}$$

1.4.3 Spatial Discretization

In this work, we choose to discretize in two dimensions, assuming uniformity in the third, however all formulations could be extended to be truly 3D. Spatial discretization methods for the transport equation are usually performed using commonly known differential equation discretization techniques such as the finite difference, finite volume, or finite element methods. In this work we discretize using the finite element method on triangular elements (described in detail in Appendix A), however TG-NDA can be used with any spatial discretization. We give the weak forms of NDA and SAAF. Their full derivations can be seen in the appendix.

Weak Form of Multigroup NDA

Given a function space $W_{\mathcal{D}}$, for all ψ^* in $W_{\mathcal{D}}$, Find $\psi_{\mathrm{g}}\in W_{\mathcal{D}}$ such that

$$\begin{aligned}\left(D_{\mathrm{g}}\nabla\psi_{\mathrm{g}}^{k+1/2},\nabla\psi^*\right)_{\mathcal{D}}+\left(\vec{\mathbf{D}}_{\mathrm{g}}\psi_{\mathrm{g}}^{k+1/2},\nabla\psi^*\right)_{\mathcal{D}}+\left(\sigma_{r,\mathrm{g}}\varphi_{\mathrm{g}}^{k+1/2},\psi^*\right)_{\mathcal{D}}=\\ \left(\sum_{\mathrm{g}'=1}^{\mathrm{g}-1}\sigma_{\mathrm{s,g}'\to\mathrm{g}}\psi_{\mathrm{g}}^{k+1/2},\psi^*\right)_{\mathcal{D}}+\left(\sum_{\mathrm{g}'=\mathrm{g}+1}^{\mathrm{G}}\sigma_{\mathrm{s,g}'\to\mathrm{g}}\psi_{\mathrm{g}}^{k},\psi^*\right)_{\mathcal{D}}+\left(Q_{\mathrm{g}},\psi_{\mathrm{g}}^*\right)_{\mathcal{D}}\end{aligned} \tag{1.21}$$

Weak Form of Multigroup SAAF

Given a function space $W_{\mathcal{D}}$, for all ψ_{g}^* in $W_{\mathcal{D}}$, Find $\psi_{\mathrm{g}}\in W_{\mathcal{D}}$ such that,

$$\begin{aligned}\left(\vec{\Omega}\frac{1}{\sigma_{t,rg}}\vec{\Omega}\cdot\vec{\nabla}\psi_{\mathrm{g}},\vec{\nabla}\psi*\right)_{\mathcal{D}}-\left(\vec{\Omega}\cdot\hat{n}\frac{1}{\sigma_{t,\mathrm{g}}}\vec{\Omega}\cdot\vec{\nabla}\psi_{\mathrm{g}},\psi*\right)_{\Gamma}+\left(\sigma_{t,\mathrm{g}}\psi_{\mathrm{g}},\psi*\right)_{\mathcal{D}}=\\ \left(q_{\mathrm{g}},\psi*\right)_{\mathcal{D}}+\left(\vec{\Omega}\frac{q}{\sigma_{t,\mathrm{g}}},\vec{\nabla}\psi*\right)_{\mathcal{D}}-\left(\vec{\Omega}\cdot\hat{n}\frac{q_{\mathrm{g}}}{\sigma_{t,\mathrm{g}}},\psi*\right)_{\Gamma}\end{aligned} \tag{1.22}$$

where $q=\int_{4\pi}(\Sigma_{s,\mathrm{g}\to\mathrm{g}'}\phi_{\mathrm{g}}+Q)$.

Chapter 2

Two-Grid, Nonlinear Diffusion Acceleration

2.1 Derivation of TG-NDA

A condensed version of this derivation first appeared in our previous work [**?**]. In this chapter we rederive the formulation with more detailed explanation.

Consider the first order, fixed-source, multigroup, steady state, S_N transport equation with isotropic scattering and internal source, where g is the group index and m is the angle index:

$$\vec{\Omega}_{\mathrm{m,g}} \cdot \nabla \psi_{\mathrm{m,g}} \left(\vec{r}\right) + \Sigma_{\mathrm{t,g}} \left(\vec{r}\right) \psi_{\mathrm{m,g}} = \frac{1}{4\pi} \sum_{\mathrm{g'=1}}^{\mathrm{G}} \Sigma_{\mathrm{s,g'\to g}} \left(\vec{r}\right) \phi_{\mathrm{g'}} \left(\vec{r}\right) + \frac{1}{4\pi} Q_{\mathrm{g}} \,. \tag{2.1}$$

Where $\phi_{\mathrm{g}} = \sum_{\mathrm{m=1}}^{M} w_{\mathrm{m}} \psi_{\mathrm{m,g}}$. Here, we assume vacuum boundary conditions. Using a similar technique as Peterson et al. used for the one group case, we multiply by angular weights and sum to get the zeroth moment equation [9].

$$\nabla \cdot J_{\mathrm{g}} + \Sigma_{r,\mathrm{g}} \phi_g = \sum_{\substack{\mathrm{g'=1} \\ \mathrm{g'\neq g}}}^{G} \Sigma_{s,\mathrm{g'\to g}} \phi_{\mathrm{g'}} + Q_{\mathrm{g}} \tag{2.2}$$

where $J_{\mathrm{g}} = \sum_{\mathrm{m=1}}^{M} w_{\mathrm{m}} \vec{\Omega} \psi_{\mathrm{m,g}}$ and the removal cross section, $\Sigma_{r,\mathrm{g}} = \Sigma_{t,\mathrm{g}} - \Sigma_{s,\mathrm{g\to g}}$. Now consider the first moment equation:

$$\nabla \cdot \overleftrightarrow{P}_{\mathrm{g}} + \Sigma_{t,\mathrm{g}} J_{\mathrm{g}} = 0 \tag{2.3}$$

where $\nabla \cdot \overleftrightarrow{P}_{\mathrm{g}} = \Omega_{m,\mathrm{g}} \Omega_{m,\mathrm{g}} \nabla \psi_{\mathrm{g}}$. It can be rewritten as:

$$J_{\mathrm{g}} = -\frac{1}{\Sigma_{t,\mathrm{g}}} \nabla \cdot \overleftrightarrow{P}_{\mathrm{g}}. \tag{2.4}$$

By adding and subtracting the diffusion coefficient, $D_{\mathrm{g}} = \frac{1}{3\Sigma_{t,\mathrm{g}}}$, times the gradient of the flux, it takes the form of a correction to Fick's Law.

$$J_{\mathrm{g}} = -D_{\mathrm{g}} \nabla \phi_{\mathrm{g}} + D_{\mathrm{g}} \nabla \phi_{\mathrm{g}} - \frac{1}{\Sigma_{t,\mathrm{g}}} \nabla \cdot \overleftrightarrow{P}_{\mathrm{g}} = -D_{\mathrm{g}} \nabla \phi_{\mathrm{g}} - \vec{\mathbf{D}}_{\mathrm{g}} \phi_{\mathrm{g}} \tag{2.5}$$

where

$$\vec{\mathbf{D}}_{\mathrm{g}}(\psi_{\mathrm{g}}) = \frac{\sum_{\mathrm{m}=1}^{M} w_{\mathrm{m}}[\frac{1}{\Sigma_t}\vec{\Omega}_{\mathrm{m}}\vec{\Omega}_{\mathrm{m}} \cdot \nabla\psi_{\mathrm{m,g}} - D_{\mathrm{g}}\nabla \sum_{\mathrm{m}=1}^{M} w_{\mathrm{m}}\psi_{\mathrm{m,g}}]}{\sum_{\mathrm{m}=1}^{M} w_{\mathrm{m}}\psi_{\mathrm{m,g}}}. \tag{2.6}$$

Substituting (2.5) in (2.2) we have the following NDA equation:

$$\nabla \cdot (-D_{\mathrm{g}}\nabla\phi_{\mathrm{g}} - \vec{\mathbf{D}}_{\mathrm{g}}\phi_{\mathrm{g}}) + \Sigma_{r,\mathrm{g}}\phi_{\mathrm{g}} = \sum_{\mathrm{g}'\neq\mathrm{g}} \Sigma_{\mathrm{s,g}'\to\mathrm{g}}\phi_{\mathrm{g}'} + Q_{\mathrm{g}}\,. \tag{2.7}$$

The two grid method for upscatter acceleration involves two calculations at each iteration. We will refer to these intermediate calculations as half iterations. To add in the upscatter acceleration we must determine an equation for the error correction. We start with the $k + 1/2$ iteration, expressed as:

$$\begin{aligned}\nabla \cdot (-D_{\mathrm{g}}\nabla\phi_{\mathrm{g}}^{k+1/2} - \vec{\mathbf{D}}_{\mathrm{g}}\phi_{\mathrm{g}}^{k+1/2}) + \Sigma_{r,\mathrm{g}}\phi_{\mathrm{g}}^{k+1/2} = \\ \sum_{\mathrm{g}'=1}^{\mathrm{g}-1} \Sigma_{\mathrm{s,g}'\to\mathrm{g}}\phi_{\mathrm{g}}^{k+1/2} + \sum_{\mathrm{g}'=\mathrm{g}+1}^{\mathrm{G}} \Sigma_{\mathrm{s,g}'\to\mathrm{g}}\phi_{\mathrm{g}'}^{k} + Q_{\mathrm{g}}\,.\end{aligned} \tag{2.8}$$

We next define two forms of error: The error of the scalar flux at each GS iteration, $\epsilon_{\mathrm{g}}^{k+1/2}$, and the error in the scattering source from upscattering at each iteration, $R_{\mathrm{g}}^{k+1/2}$.

$$\epsilon_{\mathrm{g}}^{k+1/2} := \phi_{\mathrm{g}} - \phi_{\mathrm{g}}^{k+1/2} \quad \text{and} \quad R_{\mathrm{g}}^{k+1/2} := \sum_{\mathrm{g}'=\mathrm{g}+1}^{\mathrm{G}} \Sigma_{\mathrm{s,g}'\mathrm{g}} \left(\phi_{\mathrm{g}'}^{k} - \phi_{\mathrm{g}'}^{k+1/2}\right)\,. \tag{2.9}$$

$\epsilon_{\mathrm{g}}^{k+1/2}$ can be further broken into the spatial component ϕ_ϵ and scattering spectrum ξ_{g} as defined in [2, 3].

$$\epsilon_{\mathrm{g}}^{k+1/2} = \phi_{\epsilon}^{k+1/2}(\vec{r})\,\xi_{\mathrm{g}}, \quad \sum_{\mathrm{g}=1}^{G} \xi_{\mathrm{g}} = 1. \tag{2.10}$$

We calculate ξ using the procedure outlined in [3]. Performing a Fourier analysis of the Gauss-Seidel iteration process, assuming isotropic scattering in an infinite medium, gives the following eigenproblem.

$$(\mathbf{T} - \mathbf{S}_L - \mathbf{S}_D)^{-1}\mathbf{S}_U\xi = \rho(\lambda)\xi, \tag{2.11}$$

where

$$\begin{aligned}\mathbf{T} &= diag(\mathbf{T}) = \text{matrix of total cross sections by group}\\ \mathbf{S}_L &= lower(\mathbf{S}) = \text{zeroth moments of the downscattering cross sections}\\ \mathbf{S}_D &= diag(\mathbf{S}) = \text{zeroth moments of within group scattering cross sections}\\ \mathbf{S}_U &= upper(\mathbf{S}) = \text{zeroth moments of the upscattering cross sections.}\end{aligned}$$

$\rho(\lambda)$ represents the eigenvalues of the system. ξ is the eigenvector. The ξ we are interested in is the eigenvector corresponding to the maximum eigenvalue of the system. ξ_{g} is the g-th entry of that eigenvector. ξ is material dependent and is recaluculated for each material in the problem.

Subtracting Eq. (2.8) from Eq. (2.7) and adding and subtracting $R_{\mathrm{g}}^{k+1/2}$ gives an exact formulation of the error, but for the sake of efficiency, we approximate the error using the following diffusion approximation:

$$\nabla \cdot (-D_{\mathrm{g}} \nabla \epsilon_{\mathrm{g}}^{k+1/2} - \vec{\mathbf{D}}_{\mathrm{g}} \epsilon_{\mathrm{g}}^{k+1/2}) + \Sigma_{r,\mathrm{g}} \epsilon_{\mathrm{g}}^{k+1/2} = \sum_{\mathrm{g}' \neq \mathrm{g}} \Sigma_{\mathrm{s},\mathrm{g}' \to \mathrm{g}} \epsilon_{\mathrm{g}'}^{k+1/2} - R_{\mathrm{g}}^{k+1/2} \,. \tag{2.12}$$

Using the eigenvector calculated in (2.11), we split the error into spatial and energy components and then integrate over all groups to collapse into a one group equation.

$$\nabla \cdot (-\langle D\xi \rangle \nabla \phi_{\epsilon}^{k+1/2} - \left\langle \vec{\mathbf{D}} \xi \right\rangle \phi_{\epsilon}^{k+1/2}) + \langle \Sigma_a \rangle \phi_{\epsilon}^{k+1/2} = - \left\langle R^{k+1/2} \right\rangle \,, \tag{2.13}$$

where $\langle X \rangle = \sum_{\mathrm{g}=1}^{G} X_{\mathrm{g}}$ and $\Sigma_{a,\mathrm{g}} = \Sigma_{r,\mathrm{g}} \xi_{\mathrm{g}} - \sum_{\mathrm{g}' \neq \mathrm{g}} \Sigma_{s,\mathrm{g}' \to \mathrm{g}} \xi_{\mathrm{g}'}$.

By solving (2.13) with the same spatial discretization as the NDA solve, we correct the scalar flux from the previous Gauss Seidel iteration as $\phi_{\mathrm{g}}^{k+1} = \phi_{\mathrm{g}}^{k+1/2} + \epsilon_{\mathrm{g}}^{k+1/2}$.

Chapter 3

Iterative Solvers

3.1 Iterative Methods for Neutron Transport

Solving the transport equation involves several nested iterative solvers. To discuss these iterative methods it is useful to think of the transport equation in operator form.

$$\mathbf{L}\psi = \mathbf{S}\psi + \mathbf{Q} \tag{3.1}$$

where $\mathbf{L}$ is the matrix formed by discretizing the left-hand side, $[\hat{\Omega} \cdot \nabla + \Sigma(\vec{r}, E)]\psi(\vec{r}, \hat{\Omega}, E)$, using finite elements or any other discretization method and $\mathbf{S}\psi$ and $\mathbf{Q}$ are matrices representing the source terms.

Linear Solver

The inner-most layer of iteration is the linear solve in which we treat the transport equation as a system of linear equations and solve for a vector, ψ, which represents the flux at several points in space. In order to do so, it must take the form $\mathbf{A}x = b$ where b is a constant vector, but we see in this case our $b = \mathbf{S}\psi + \mathbf{Q}$ which is dependent on ψ in the scattering source. To remedy this we choose ψ on the right hand side to be a constant. The method by which we choose this constant, called source iteration, is the next level of iteration, detailed below. Now that b is a constant vector, we are able to perform a linear solve. Solving $\mathbf{A}x = b$ is well studied problem, and there are many iterative methods to choose from. Most have already been implemented in forms optimized for performance in various linear algebra libraries such as LAPACK.

"Inner" Iterations

The inner iterations solve the space-angle component for each energy group. As was mentioned above, the scattering term on the right hand side of the transport equation (1.1) depends on the angular flux, ψ. The inner iterations, iterate on a guess of that angular flux so that it can be treated as a constant for the linear solves. The traditional method is called source iteration, which is explained in more detail below. A more advanced alternative to source iteration that is starting to gain popularity is a Krylov solver.

Multigroup or "Outer Iterations"

The multigroup iterations solve the energy component. When we have more than one energy group, we solve each group independently, performing source iteration on each one. In doing so, we must include the scattering from all other energy groups to the current group multiplied by the flux of the other groups. When there is no "upscattering", meaning there is no scattering from a lower energy group to a higher energy group, we can solve each group sequentially, starting with the highest energy group, without any problems. No other groups scatter to group one, the highest energy group, the next group is only dependent on the flux from group one, which we just solved for, and so on. In the case of upscattering, an outer layer of iteration is added to converge the scattering source. The most common multigroup solver is called the Gauss-Seidel method. Other alternatives include Jacobi or multigroup Krylov.

The three formulations that are discussed, NDA, TG-NDA, and SAAF can be solved using any combination of the solvers listed above. We will explain in detail the solvers that were used in our implementation.

3.2 Implemented Solvers

3.2.1 Linear Solver

For our linear solver we used SciPy's conjugate gradient solver [**?**] [**?**]. CG solves the system Ax = b assuming A is a real, symmetric, positive-definite matrix. By using the finite element method, we are guaranteed that our matrix A satisfies those constraints, so CG is an applicable method for our problem.

3.2.2 Within Group Solver - NDA

The inner solves are handled by NDA which is a correction to source iteration. In source iteration, the ψ on the right hand side of the equation starts with an initial guess. At each iteration, the linear system is solved and ψ is updated with the result of that solve. This continues until the values converge.

Algorithm 1 Source Iteration

while $res > tol$ **do** ▷ Iteration Index k
  Set up FEM discretization of Eq.
  $\mathbf{S}\psi \leftarrow \sigma_s \phi^{k-1}$ ▷ Assumes one group, see multigroup below.
  Solve system $\mathbf{L}\psi = \mathbf{S}\psi + \mathbf{Q}$
  $res \leftarrow max(|\psi^k - \psi^{k-1}|)$

NDA adds an extra step to this process by first calculating the angular flux using a higher order equation and then solving the lower order equation using the result of the higher order equation to compute a correction. The result of the lower order solve is then used as the guess on the right hand side during the next iteration.

1. Intitialize system, by setting $\vec{\mathbf{D}}$ to 0 and solving (1.17) to get ϕ^0
2. Loop Until Convergence:
   (a) Solve (1.7) for ψ^l using ϕ^{l-1} on RHS.
   (b) Calculate drift vector, (1.16), using ψ^l
   (c) Solve (1.17) for ϕ^l
   (d) Check ϕ^{l-1}, ϕ^l for convergence
3. Return ϕ

3.2.3 Multigroup Solver - Gauss Seidel

When dealing with a multigroup problem, the scattering source is dependent not only on the flux from the current group, but from other groups as well. In the case that there is no upscattering, meaning that the flux from a lower energy group does not contribute to any higher energy group, we can solve sequentially, starting with the highest energy group and using the fluxes as we calculate them. In the case of upscattering, we must choose an initial guess for the fluxes of lower energy groups and then iterate until we find a convergent value. We use a method inspired by the Gauss-Seidel method for linear equations. When iterating, we use the flux calculated in this iteration for the higher energy groups and the flux calculated in the previous iteration for the lower energy groups. The pseudocode of the algorithm is given below.

Algorithm 2 Outer Iterations: Gauss Seidel

while $res > tol$ **do** ▷ Iteration Index l
  for $g \in G$ **do**
    calculate scattering source:
    $\mathbf{S}\psi = \sigma_{gg}\phi_g^{l+1} + \sum_{g'=1}^{g-1} \sigma_{gg'}\phi_{g'}^{l+1} + \sum_{g'=g+1}^{G} \sigma_{gg'}\phi_{g'}^{l}$
    procedure SOURCE ITERATION ON GROUP g
  $res \leftarrow max(|\phi^l - \phi^{l-1}|)$ ▷ Check if sol. for each group has converged
return ϕ

3.2.4 TG-NDA

TG-NDA acclerates convergence by applying a correction at two different layers of iteration. NDA applies a correction using the drift vector at each source iteration. Two-Grid provides a correction at each Gauss-Seidel iteration. The full scheme as implemented is illustrated below with the correction terms highlighted in blue.

Text

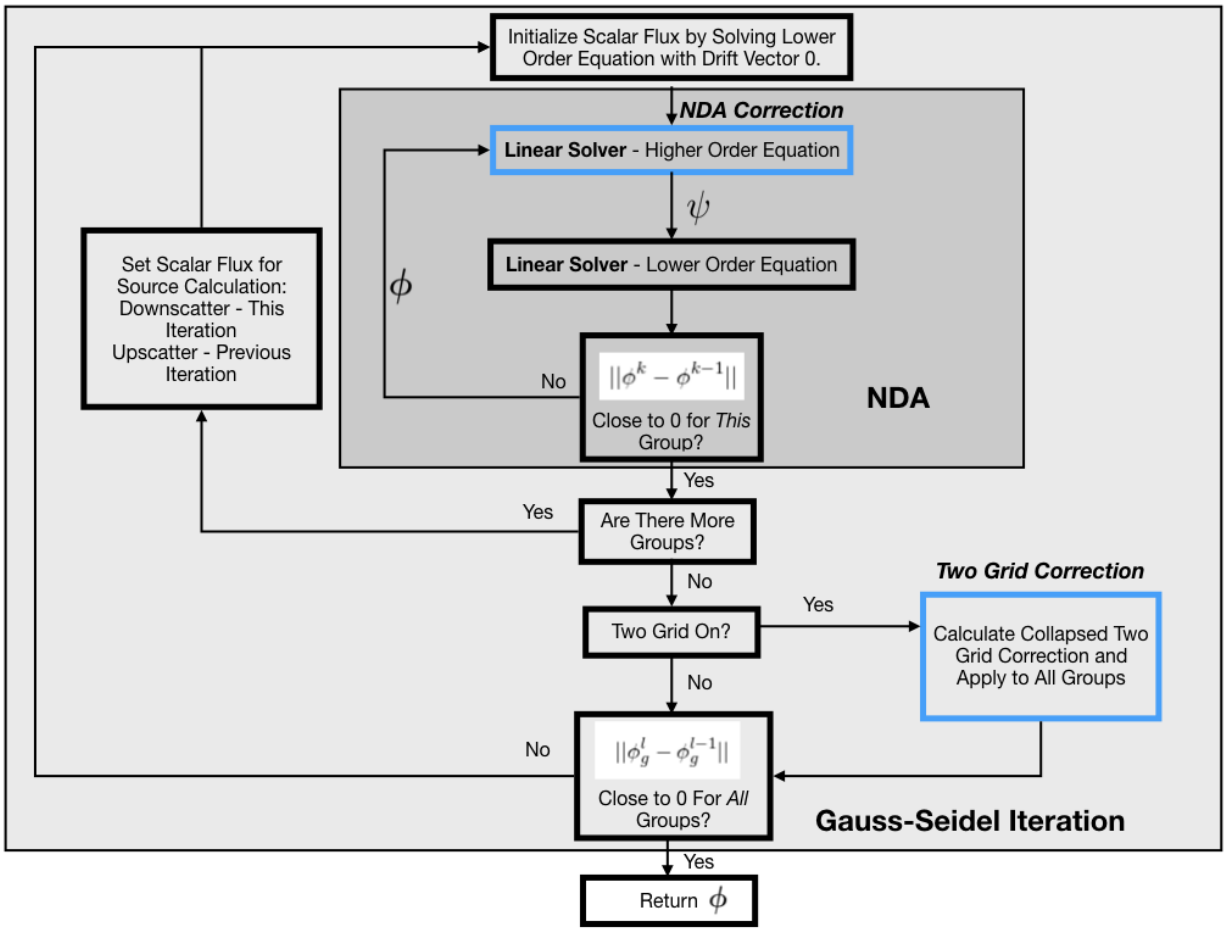


Figure 3.1: Full Iterative TG-NDA Scheme

Chapter 4

Testing & Validation

We used a number of tests to validate the code. There are a number of unit tests designed to test individual functions in the code. For integration testing we selected a suite of simple problems all using a single material on a square domain for which we could derive or approximate analytic solutions. We varied the material properties to trigger use of the various solvers. These problems were a single energy group with no scattering, a single energy group with scattering, and two energy groups with scattering.

4.1 NDA/SAAF Agreement

We also tested the agreement between the NDA+SAAF and SAAF alone to ensure we were replicating expected behavior. As the SAAF equation is non-conservative, its solution does not necessarily agree with the low order equation it is coupled with. However, the two solutions should approach each other as the number of mesh cells increases [10]. We tested to see if our code replicated this behavior in a number of ways. First we simply compare the mean difference in the SAAF and NDA solutions for scalar flux as number of mesh cells increases. We notice a consistent rate of convergence.

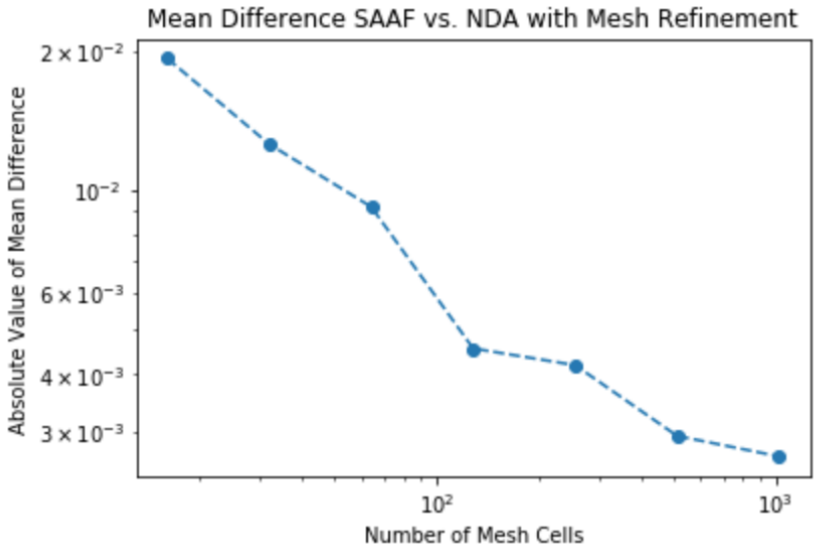


Figure 4.1: SAAF/NDA Agreement with Mesh Refinement

Second we look at the error in the absorption rate, $\int_{\mathcal{D}} \Sigma_a \phi$, in NDA as compared to a fine mesh SAAF solution. This, too, steadily decreases as the number of mesh cells increase.

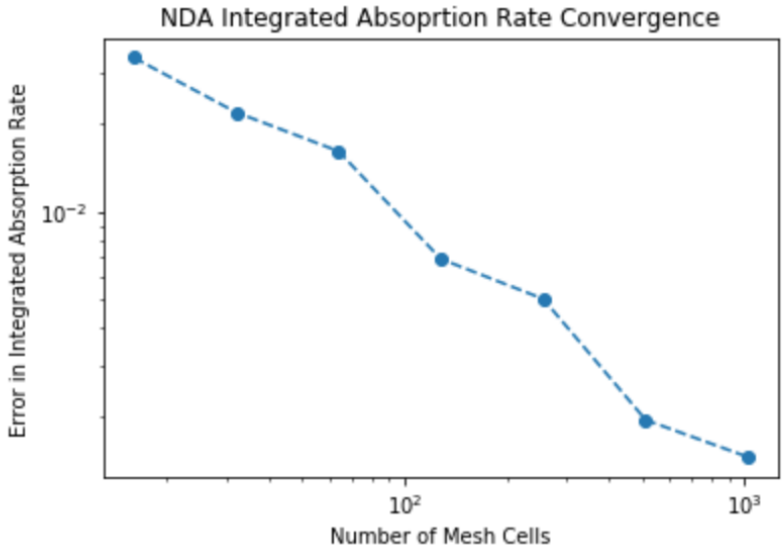


Figure 4.2: Error in NDA Absorption Rate as Mesh Refines

Finally, we plot a 1-D line out of all three methods. For this problem we chose one test material with one group and some scattering. This is more of a gut check than a formal check of convergence. From this plot we can clearly see that NDA has provided a correction to diffusion, producing a solution that is much closer to the higher order equation than diffusion by itself.

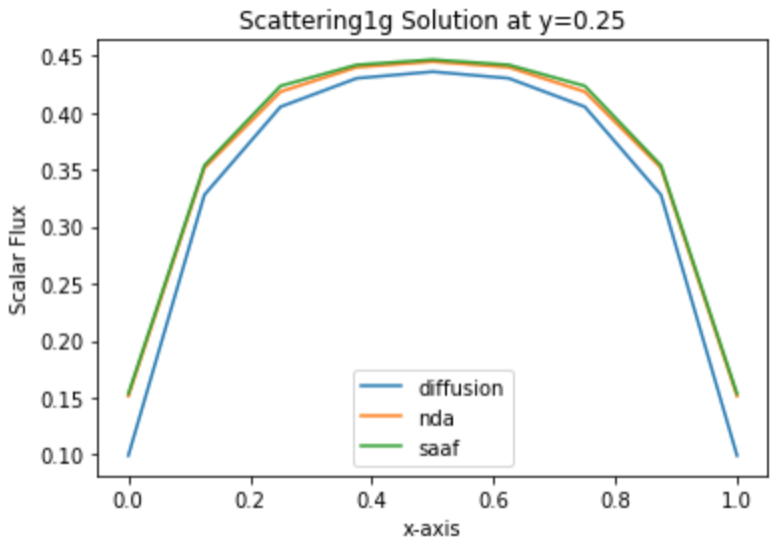


Figure 4.3: Comparison of Diffusion, SAAF, and NDA

From the results of these three checks, we determine that we are in agreement with previously published work on SAAF+NDA and have implemented a working version.

Chapter 5

Numerical Results

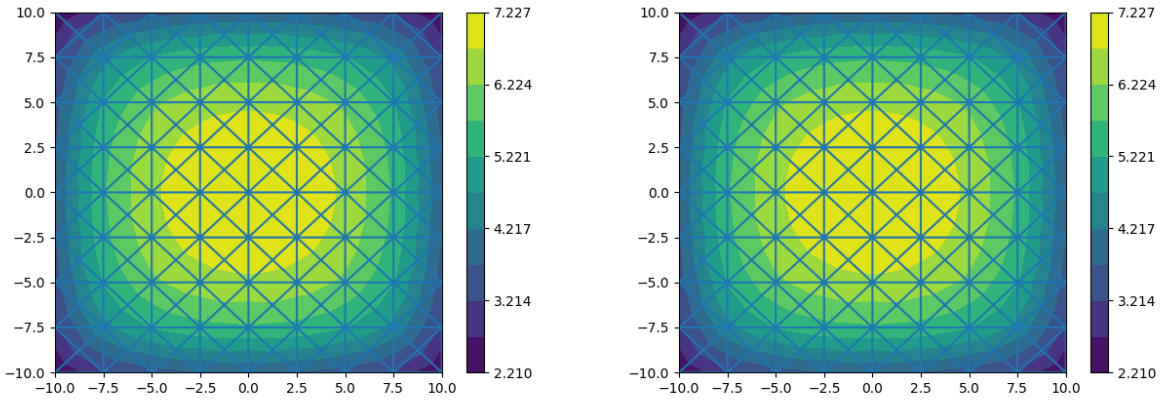
5.1 One Material

For the first test problem we used a single material with upscattering throughout a 20 by 20 domain with a constant source. We used cross sections for the seven group moderator material in the C5G7 benchmark problem. [1]. The problems were run on a single processor of a MacBook Pro.

	Runtime (s)	GS Iterations
TG-NDA	5465.00732	9
NDA	14513.23	31

The two-grid method provides a considerable acceleration of the Gauss-Seidel method. While when using TG-NDA each iteration takes slightly longer as the correction term must be calculated, it more than makes up for it with a considerable decrease in the number of iterations necessary to reach convergence.

Importantly, the accleration in convergence came at no cost to accuracy. As can be seen in the figure below, the NDA and TG-NDA solutions have the same values (up to tolerance). In the interest of space, we only show the results from the highest energy group, but the NDA/TG-NDA agreement held for all energy groups in our test problems.



(a) Highest Energy Group, NDA

(b) Highest Energy Group, TG-NDA

Figure 5.1: Comparison of NDA/TG-NDA in Flux Value for One Material Problem

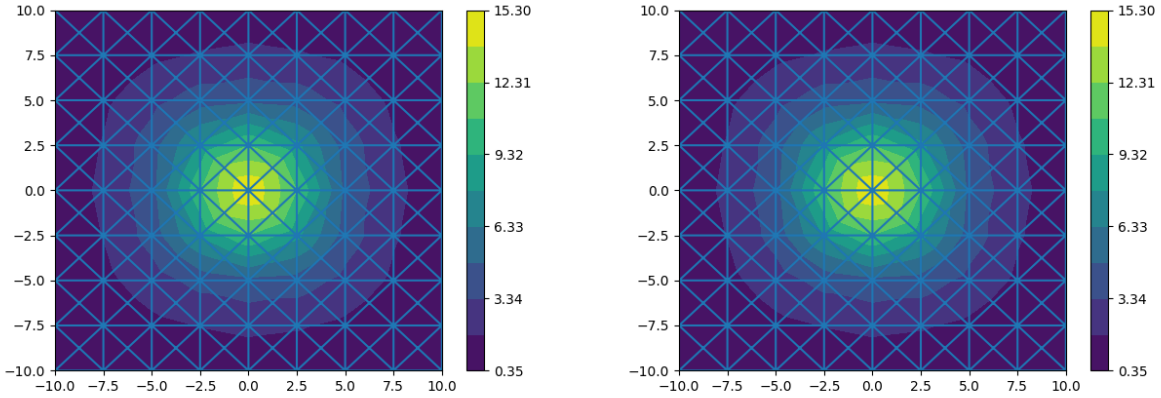
5.2 Two Materials

The second problem consists of two materials in a concentric geometry with a box source in the center. The first material, located in the center and outer layer, is the C5G7 moderator material used above and the second material has the same total cross sections and pattern of upscatter, but with higher absorption and lower total scattering. Both materials have seven groups. There is a box source in the center that emits 70% in the highest energy group, 20% in the second highest, and 10% in the third.

Figure 5.2: Geometry of Two-Material Test Problem

	Runtime (s)	GS Iterations
TG-NDA	4221.92	8
NDA	10381.52	25

Again, TG-NDA showed a significant improvement over the unaccelerated Gauss Seidel, taking roughly a third of the number of iterations. Again, NDA and TG-NDA agree in terms of flux values. Here we also only show results for the highest energy group.



(a) Highest Energy Group, NDA

(b) Highest Energy Group, TG-NDA

Figure 5.3: Comparison of NDA/TG-NDA in Flux Value for Two Material Problem

5.3 Reproducibility

The code used to run these experiments is hosted online at www.github.com/mzweig/gallo. The version used is tagged as masters-thesis. The geometry inputs `origin-centered10` and material input `c5g7mod` were used for the first test problem, and the geometry input `iron-water10` and material input `mod-water` were used for the second problem.

Chapter 6

Conclusion & Future Work

In this work we derived and implemented a two grid acceleration scheme for the nonlinear diffusion acceleration equations with neutron upscattering. In our tests problems, we observe a reduction in the number of Gauss-Seidel iterations necessary to resolve the upscattering of roughly a factor of three. The current implementation is in a lightweight code developed in python that is only designed as a proof of concept. We intend to implement a heavier-duty C++ version into the Slaybaugh Lab's neutron transport code, Bay Area Radiation Transport. With that implementation we can begin to test larger problems and understand how the method scales.

6.1 Possible Extensions to TG-NDA

There are a number of modifications to our implementation that could be interesting to explore with TG-NDA.

6.1.1 Other Discretization Schemes

P_N Angular Discretization

Our derivation is specific to the S_N equations. It is possible to derive the higher order equation using P_N instead [**?**] and see if there is any effect on the convergence behavior of TG-NDA.

Discontinuous Finite Elements

In our implementation we use continuous finite elements to discretize in space. Discretizing NDA using discontinuous finite elements, as done in [**?**], would change the form of TG-NDA and could change Gauss-Seidel convergence behavior.

6.1.2 Other Higher Order Equations

We chose to pair NDA with SAAF, however other equations could be used as well, such as the even-parity equation [**?**].

6.1.3 Applications to Criticality Problems

In this work we only focused on fixed-source problems for shielding applications, however many materials commonly used in nuclear reactors, such as water, heavy water, or graphite, also exhibit significant upscattering. Performing criticality calculations using our method would be a straightforward extension involving only an additional later of iteration to perform the eigenvalue calculation.

Appendix A

FEM on an Unstructured Triangular Grid

A.1 Continuous FEM on an Unstructured Grid

To discretize in space we will be using the continuous finite element method with bilinear basis functions on an unstructured triangular mesh. The mesh is generated with the Triangle library [**?**] via a Delaunay triangulation algorithm.

To integrate over the triangles, we have implemented a 2nd degree Gaussian Quadrature approximation over the standard triangle,

$$\iint_{T_{st}} f(\xi,\eta)d\xi d\eta \approx \frac{1}{6}\left[f\left(0,\frac{1}{2}\right)+f\left(\frac{1}{2},0\right)+f\left(\frac{1}{2},\frac{1}{2}\right)\right]. \tag{A.1}$$

We are working with an unstructured grid of triangles, so we must additionally map nodes on the standard triangle to an arbitrary element. This is done by the `gaussNodes()` function in this `FEGrid` class. The mapping is given as follows

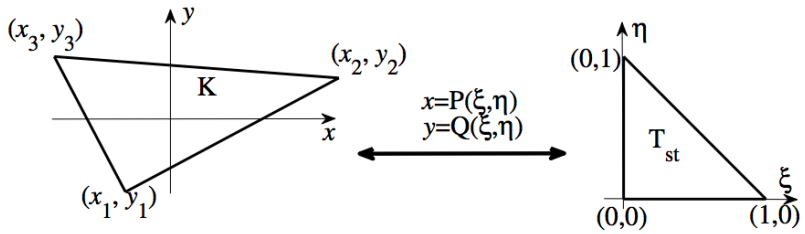


Figure A.1: Linear mapping between an element K and the standard triangle

$$x = P(\xi,\eta) = x_1 + \xi(x_2 - x_1) + \eta(x_3 - x_1) \tag{A.2}$$
$$y = Q(\xi,\eta) = y_1 + \xi(y_2 - y_1) + \eta(y_3 - y_1) \tag{A.3}$$

After applying the transformation we have,

$$\iint_K F(x,y)dxdy = \iint_{T_{st}} F(P(\xi,\eta),Q(\xi,\eta))|J(\xi,\eta)|d\xi d\eta \tag{A.4}$$

where $|J(\xi,\eta)|$ is the Jacobian of the transform which is equal to $2Area_K$. This gives the final quadrature rule as

$$\iint_K F(x,y)dxdy = 2A_k \iint_{T_{st}} F(P(\xi,\eta),Q(\xi,\eta))d\xi d\eta \tag{A.5}$$

$$\approx \frac{1}{6}\left[F\left(P(0),Q(\frac{1}{2})\right) + F\left(P(\frac{1}{2}),Q(0)\right) + F\left(P(\frac{1}{2}),Q(\frac{1}{2})\right)\right]. \tag{A.6}$$

A.2 Weak Form of NDA

Similar to the higher order equation, we discretize Eq. 4 via the continuous finite element method by multiplying by a test function and integrating over the domain. This gives the following weak form:

Find $\varphi_{d,g} \in W_D$ such that

$$\left(D_{\mathrm{g}}\nabla\varphi_{\mathrm{g}}^{k+1/2},\nabla\varphi_{\mathrm{g}}^{*}\right)_{\mathcal{D}} + \left(\vec{\mathbf{D}}_{\mathrm{g}}\varphi_{\mathrm{g}}^{k+1/2},\nabla\varphi_{\mathrm{g}}^{*}\right)_{\mathcal{D}} + \left(\sigma_{r,\mathrm{g}}\varphi_{\mathrm{g}}^{k+1/2},\varphi_{\mathrm{g}}^{*}\right)_{\mathcal{D}} =$$
$$\left(\sum_{\mathrm{g}'=1}^{\mathrm{g}-1}\sigma_{\mathrm{s},\mathrm{g}'\to\mathrm{g}}\varphi_{\mathrm{g}}^{k+1/2},\varphi_{\mathrm{g}}^{*}\right)_{\mathcal{D}} + \left(\sum_{\mathrm{g}'=\mathrm{g}+1}^{\mathrm{G}}\sigma_{\mathrm{s},\mathrm{g}'\to\mathrm{g}}\varphi_{\mathrm{g}}^{k},\varphi_{\mathrm{g}}^{*}\right)_{\mathcal{D}} + \left(S_{\mathrm{f,g}},\varphi_{\mathrm{g}}^{*}\right)_{\mathcal{D}} \tag{A.7}$$

The error (Eq. 8) is similarly discretized via CFEM.

$$\left(\langle D\xi\rangle\nabla\varphi_{\epsilon}^{k+1},\nabla\varphi^{*}\right)_{\mathcal{D}} + \left(\left\langle\vec{\mathbf{D}}\xi\right\rangle\varphi_{\epsilon}^{k+1},\nabla\varphi^{*}\right)_{\mathcal{D}} + \left(\langle\sigma_r\rangle\,\phi_{\epsilon}^{k+1},\varphi^{*}\right)_{\mathcal{D}}$$
$$= \left(\left\langle R^{k+1}\right\rangle,\varphi^{*}\right)_{\mathcal{D}} + \left(\langle S_{\mathrm{f,g}}\rangle,\varphi^{*}\right)_{\mathcal{D}} \tag{A.8}$$

A.3 Weak Form of the Higher Order Equation

We apply a finite element discretization to the higher order equation, SAAF, by first multiplying Eq. 1.7 by a test function $\psi*$ and integrating over the domain D.

$$\left(-\vec{\Omega}\cdot\vec{\nabla}\frac{1}{\sigma_t}\vec{\Omega}\cdot\vec{\nabla}\psi,\psi*\right)_D + (\sigma_t\psi,\psi*)_D = (Q,\psi*)_D - \left(\vec{\Omega}\cdot\vec{\nabla}\frac{Q}{\sigma_t},\psi*\right)_D \tag{A.9}$$

Integrating by parts,

$$\left(\vec{\Omega}\frac{1}{\sigma_t}\vec{\Omega}\cdot\vec{\nabla}\psi,\vec{\nabla}\psi*\right)_D - \left(\vec{\Omega}\cdot\hat{n}\frac{1}{\sigma_t}\vec{\Omega}\cdot\vec{\nabla}\psi,\psi*\right)_\Gamma + (\sigma_t\psi,\psi*)_D =$$
$$(Q,\psi*)_D + \left(\vec{\Omega}\frac{Q}{\sigma_t},\vec{\nabla}\psi*\right)_D - \left(\vec{\Omega}\cdot\hat{n}\frac{Q}{\sigma_t},\psi*\right)_\Gamma$$

Bibliography

[1] Benchmark on deterministic transport calcuations without spatial homogenisation: Mox fuel assembly 3-d extension case. Technical report, Nuclear Energy Agency, Organisation for Economic Co-Operation and Development, 2005.

[2] B. T. Adams and J. E. Morel. A two-grid acceleration scheme for the multigroup s_n equations with neutron upscattering. *Nuclear Science and Engineering*, 115:253–264, 1993.

[3] T. M. Evans, K. T. Clarno, and J. E. Morel. A transport acceleration scheme for multigroup discrete ordinates with upscattering. *Nuclear Science and Engineering*, 165:292–304, 2010.

[4] Joshua Jarrell. *An Adaptive Angular Discretization Method for Neutral-Particle Transport in Three Dimensional Geometries*. PhD thesis, Texas AM University, 2010.

[5] Kaye D. Lathrop and Bengt G. Carlson. Discrete ordinates angular quadrature of the neutron transport equation. Technical report, Los Alamos National Laboratory, 1965.

[6] E. E. Lewis and Jr. W. F. Miller. *Computational Methods of Neutron Transport*. Wiley-Interscience, 1993.

[7] J.E. Morel and J. M. McGhee. A self-adjoint angular flux equation. *Nuclear Science and Engineering*, 132:312–325, 1999.

[8] H. Park, D. A. Knoll, and C. K. Newman. Nonlinear acceleration of transport criticality problems. *Nuclear Science and Engineering*, 172:52–65, 2012.

[9] Jacob R. Peterson, Hans R. Hammer, Jim E. Morel, Jean C. Ragusa, and Y. Wang. Conservative nonlinear diffusion acceleration applied to the unweighted least-squares transport equation in moose. In *ANS M&C2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*. Nashville, TN, April, 2015.

[10] Yaqi Wang. Nonlinear diffusion acceleration for the multigroup transport equation discretized with s_n and continuous fem with rattlesnake. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering*, May.