Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

Steven Paul Hamilton	Date

Numerical Solution of the k-Eigenvalue Problem

By

Steven Paul Hamilton
Doctor of Philosophy

Mathematics and Computer Science

Michele Benzi, Ph.D.
Advisor

Kevin Clarno, Ph.D.
Committee Member

James Nagy, Ph.D.
Committee Member

Alessandro Veneziani, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Numerical Solution of the k-Eigenvalue Problem

By

Steven Paul Hamilton
M.S. in Nuclear Engineering, Georgia Institute of Technology, 2007
B.S. in Nuclear and Radiological Engineering, Georgia Institute of Technology, 2006

Advisor: Michele Benzi, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Mathematics and Computer Science
2010

Abstract

Numerical Solution of the k-Eigenvalue Problem
By Steven Paul Hamilton

Obtaining solutions to the k-eigenvalue form of the radiation transport equation is an important topic in the design and analysis of nuclear reactors. Although this has been an area of active interest in the nuclear engineering community for several decades, to date no truly satisfactory solution strategies exist. In general, existing techniques are either slow to converge for difficult problems or suffer from stability and robustness issues that can cause solvers to diverge for some problems. This work provides a comparison between a variety of methods and introduces a new strategy based on the Davidson method that has been used in other fields for many years but never for this problem. The Davidson method offers an alternative to the nested iteration structure inherent to standard approaches and allows expensive linear solvers to be replaced by a potentially cheap preconditioner. To fill the role of this preconditioner, a strategy based on a multigrid treatment of the energy variable is developed. Numerical experiments using the 2-D NEWT transport package are presented, demonstrating the effectiveness of the proposed strategy.

Numerical Solution of the k-Eigenvalue Problem

By

Steven Paul Hamilton
M.S. in Nuclear Engineering, Georgia Institute of Technology, 2007
B.S. in Nuclear and Radiological Engineering, Georgia Institute of Technology, 2006

Advisor: Michele Benzi, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Mathematics and Computer Science
2010

Acknowledgments

I want to begin by thanking the Department of Energy and the Krell Institute for supporting my graduate work though a Computational Science Graduate Fellowship. Without the opportunity granted by this program, I certainly would not be where I am today. It has been a wonderful experience meeting so many fellow computational scientists and learning about their diverse areas of interest.

I want to thank all of my committee members for taking the time to serve on the committee, your advice and suggestions have made a tremendous difference. I especially want to thank Kevin Clarno for the years of mentoring that he has provided. Without his suggestion I never would have given any thought to pursuing a mathematics degree at Emory; I am certainly a better scientist for having taken that path.

I am truly indebted to my advisor, Michele Benzi, for agreeing to take me in as his graduate student. I entered Emory without any rigorous mathematics background and I cannot even begin to describe how much I have learned from him.

Many thanks are due to Jim Warsa of Los Alamos National Laboratory for mentoring me during a summer practicum at the lab. My understanding of the transport equation was greatly enhanced through our interactions. Tom Evans of Oak Ridge National Laboratory provided many fruitful discussions that helped to shape and improve the methods developed in my dissertation work. I also want to thank Matt Jessee and Dan Ilas, also of Oak Ridge National Laboratory, for assisting with the development of the problems to test the methods.

Last, and certainly not least, I want to express my appreciation of my wife Beth for her continuous love and support.

for Beth

Contents

1 Introduction **1**

2 Neutron Transport Equation **5**

2.1 Continuous 5

2.1.1 Boundary Conditions 6

2.1.2 Scattering Integral 7

2.2 Discretizations 9

2.2.1 Energy 10

2.2.2 Angle 11

2.2.3 Space 13

2.2.4 Alternate Discretizations 14

2.3 Discrete Formulation 15

2.3.1 Linear System 15

2.3.2 k-Eigenvalue Problem 16

2.4 Structure 18

2.4.1 Solution Vector 18

2.4.2 Transport Matrix 19

2.4.3 Scattering and Fission Matrices 21

2.4.4 Transfer Matrices 22

2.4.5 Implementation Considerations 24

2.5 Spectral Properties 26

2.5.1 Transport Operators 26
2.5.2 k-Eigenvalue Problem 28

3 Eigensolvers **33**
3.1 Fixed Point Methods 34
3.1.1 Power Iteration 34
3.1.2 Shifted Power Iteration 35
3.1.3 Rayleigh Quotient Iteration 36
3.1.4 Newton's Method 37
3.2 Subspace Methods 38
3.2.1 Arnoldi's Method 40
3.2.2 Generalized Davidson Method 41
3.2.3 Davidson Preconditioners 42
3.2.4 Reduced Memory Subspace Methods 46
3.2.5 Schur Forms 48
3.2.6 Restarting Subspace Methods 49

4 Transport Solvers **52**
4.1 Monoenergetic Solvers 52
4.1.1 Richardson Iteration 53
4.1.2 Diffusion Synthetic Acceleration 53
4.1.3 Other Linear Preconditioners 56
4.1.4 Nonlinear Acceleration 56
4.1.5 Krylov Methods 58
4.2 Multigroup Solvers 59
4.2.1 Block Gauss-Seidel 60
4.2.2 Upscatter Acceleration 60
4.2.3 Krylov Methods 62
4.3 Eigensolvers 64
4.3.1 Power Iteration 64

4.3.2 Shifted Iterations 65
4.3.3 Nonlinear Acceleration 65
4.3.4 Krylov Methods 66
4.3.5 Newton's Method 67
4.4 Parallelization 68
4.4.1 Domain Decomposition 68
4.4.2 KBA 69
4.4.3 Unstructured Sweeps 70

5 Multigrid-in-Energy **72**
5.1 Basic Structure 73
5.2 Grid Transfer 74
5.3 Smoothing Iterations 75
5.4 Coarse Angle Approximation 77
5.5 Parameter Selection 78

6 Numerical Results **88**
6.1 Test Problems 89
6.2 Parametric Studies 95
6.2.1 Spatial Refinement 96
6.2.2 Angular Refinement 99
6.2.3 Energy Refinement 100
6.2.4 Scattering Order Refinement 101
6.2.5 Subspace Dimensions 103
6.3 Linear Solvers 107

7 Conclusion **112**

A Linear Solver Parametric Studies **116**

Bibliography **120**

List of Figures

2.1 Geometry for spectrum computation 28
2.2 Transport spectrum, L fully inverted 29
2.3 Transport spectrum, L diagonal blocks inverted 30
2.4 k-eigenvalue spectrum 32

5.1 Infinite medium MGE dependence on ω 81
5.2 Infinite medium preconditioned spectra 82
5.3 MGE dependence on ω 84

6.1 HTTR base configuration 90
6.2 HTTR CMFD equivalent configuration 91
6.3 HTTR dominant eigenvector 92
6.4 C5G7 base configuration 93
6.5 C5G7 dominant eigenvector 94
6.6 GMRES convergence with fission contribution 111

List of Tables

5.1 MGE pre- and post-smoothing steps, HTTR 85
5.2 MGE pre- and post-smoothing steps, C5G7 86
5.3 MGE coarse angle smoothing, HTTR 87
5.4 MGE coarse angle smoothing, C5G7 87

6.1 Eigensolver performance, base configurations 95
6.2 Problem size for spatial refinement 96
6.3 Eigenvalue convergence for spatial refinement 97
6.4 Performance for spatial refinement, HTTR 98
6.5 Performance for spatial refinement, C5G7 98
6.6 CMFD performance for spatial refinement, HTTR 98
6.7 CMFD performance for spatial refinement, C5G7 99
6.8 Eigenvalue convergence for angular refinement 100
6.9 Performance for angle refinement, HTTR 100
6.10 Performance for angle refinement, C5G7 101
6.11 Convergence for energy refinement 101
6.12 Performance for energy refinement, HTTR 102
6.13 Performance for energy refinement, C5G7 102
6.14 Convergence for P_N order refinement 103
6.15 Performance for P_N order refinement, HTTR 104
6.16 Performance for P_N order refinement, C5G7 104
6.17 Davidson performance for subspace size variation, HTTR . . . 105

6.18 Davidson performance for subspace size variation, C5G7 . . . 106
6.19 Comparison of linear solver performance for base cases 108
6.20 Linear solver performance, HTTR with shifts 109
6.21 Linear solver performance, C5G7 with shifts 110

A.1 Linear solver performance for angle refinement, C5G7 117
A.2 Linear solver performance for angle refinement, HTTR 117
A.3 Linear solver performance for energy refinement, C5G7 117
A.4 Linear solver performance for energy refinement, HTTR 118
A.5 Linear solver performance for spatial refinement, C5G7 118
A.6 Linear solver performance for spatial refinement, HTTR 118
A.7 Linear solver performance for P_N order refinement, C5G7 . . . 119
A.8 Linear solver performance for P_N order refinement, HTTR . . 119

Chapter 1

Introduction

The numerical solution of the movement of radiation through matter is a topic of great interest in a number of fields. In the medical community modeling radiation transport is used to design treatment plans for radiation cancer therapy as well as assisting in imaging processes, in geophysics it is used to reconstruct subterranean material compositions from sensor data for applications such as natural gas and oil well logging, and in astrophysics it is possible to simulate the events surrounding the explosion of supernovae. However, in this study we will be interested in the use of radiation transport as it pertains to the distribution of neutrons in the core of a nuclear reactor. In particular we are interested in solving the k-eigenvalue problem. The solution to this equation describes how far the system is from achieving criticality – the state where the absorption and escape of neutrons are perfectly balanced by the production of new neutrons through fission events. This task is cast mathematically as an eigenvalue problem in which the dominant eigenvalue describes the amount of multiplication in the system and the corresponding eigenvector provides the distribution of neutrons not only according to their spatial location but also according to their energy and direction of travel. Detailed knowledge of this distribution is necessary in

order to design and operate reactors in a safe and efficient manner.

Although significant effort has been put into the solution of this problem over the past several decades, the desire to continue to improve existing designs has pushed current methods beyond their practical limits. A new Department of Energy project, the Consortium for the Advanced Simulation of Light Water Reactors (CASL) [85], has taken on the task of creating a 'virtual reactor,' a computational toolkit with the aim of improving our understanding of the interactions taking place within an operating nuclear reactor with a level of detail that has never before been achievable. This project has the potential to help in resolving key issues that have proven to be limiting factors in achieving greater performance from commercial reactors. Increasing the understanding of such phenomena could lead to improvements in the form of increasing the power output of reactors, extending the operating life of existing reactors, and increasing the level of fuel burnup that is feasible, thereby reducing the amount of spent nuclear fuel that must be stored. These techniques will also prove invaluable in the design of the next generation of nuclear reactors.

There are two classes of methods available for the solution of problems in radiation transport. In their simplest form, Monte Carlo methods determine desired quantities by modeling the paths of individual particles and averaging the impact of very large numbers of such histories. The selection of particular events within a given history is determined by sampling from known probability distributions through the use of random number generators. Due to their ability to avoid discretizing the problem with respect to problem variables (space, angle, energy), Monte Carlo methods avoid the introduction of many errors and are thus frequently highly touted for their accuracy. The downside of such methods, however, lies in the statistical nature of the process, frequently requiring enormous numbers of particle histories to obtain sufficiently accurate results. It is frequently advantageous to

use Monte Carlo methods if only global quantities (such as a single eigenvalue) are required; when detailed local information is required the cost of achieving acceptable statistical results tend to be prohibitive. Because access to such local information is frequently desired, Monte Carlo methods are less commonly used in large scale applications and we turn our attention to deterministic methods, the topic of this dissertation. Deterministic methods approach the problem by directly discretizing the problem with respect to each element of the phase space to obtain a system of equations that are then solved using some numerical algorithm.

In this study, the primary goals are to provide an overview of the methods frequently used in production-level radiation transport software for the solution of the k-eigenvalue problem and offer potential alternatives to such techniques with the aim of improving the quality of available solvers. It should be noted that the task of improving solvers is not limited to merely reducing the computational effort required to solve a problem (although this is certainly a desirable feature), but also increasing the level of reliability and robustness. Indeed, there are a number of methods available that exhibit superb performance in certain problem regimes but display significant degradation (or even fail) for other types of problems. Thus, methods that are competitive with the best available solvers but maintain this performance over a larger range of problems would be quite promising. Furthermore, the development of solvers that require few, if any, problem-dependent parameter selections would allow such methods to reach a broader user base, including those that have a limited understanding of the underlying mechanics of a particular solver. By bringing ideas and strategies from the mathematical community that have been generally unknown to the transport community to approach this problem, we hope to create a step in the direction of developing such solution methods.

The remainder of this dissertation is organized as follows. In Chapter 2

we introduce the radiation transport problem with particular focus on the k-eigenvalue problem and discuss techniques for discretization as well as certain numerical properties of the resulting linear systems. In Chapter 3 an overview of eigensolvers amenable to finding particular eigenvalue/eigenvector pairs of very large matrices is provided. Chapter 4 offers a glimpse into the state of the art in computational radiation transport, including a discussion of both linear and eigenvalue solvers. Chapter 5 provides the development of a novel preconditioner for the transport equation that revolves around a multigrid treatment of the energy variable. Numerical results for several representative test problems are discussed in Chapter 6 with a particular focus on describing the performance of the newly developed preconditioner combined both with eigensolvers traditionally employed for transport as well as some not previously explored. Concluding remarks and suggestions for areas of continued development are offered in Chapter 7.

Chapter 2

Neutron Transport Equation

2.1 Continuous

For a position $\mathbf{r}$, direction vector $\hat{\Omega}$, and energy E, the source driven Boltzmann neutron transport equation in a multiplying medium can be written as

$$\hat{\Omega} \cdot \nabla\psi + \sigma\psi = \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}'\, \sigma_s\psi + \chi \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}'\, \nu\sigma_f\psi + \mathbf{q} \quad (2.1)$$

where $\psi = \psi(\mathbf{r}, \hat{\Omega}, E)$ is the angular flux, $\sigma = \sigma(\mathbf{r}, E)$ is the total cross section, $\sigma_s = \sigma_s(\mathbf{r}, \hat{\Omega}' \rightarrow \hat{\Omega}, E' \rightarrow E)$ is the scattering cross section, $\chi = \chi(\mathbf{r}, E)$ is the energy distribution of fission neutrons, $\nu = \nu(\mathbf{r}, E)$ is the average neutron production per fission, $\sigma_f = \sigma_f(\mathbf{r}, E)$ is the fission cross section and $\mathbf{q} = \mathbf{q}(\mathbf{r}, \hat{\Omega}, E)$ is an external source term.

The left side of (2.1) describes neutrons being removed from the current element of phase space through streaming and collision interactions. The right side describes neutrons being introduced into the current element of phase space through scattering from every energy and angle, fission events occurring at all energies and angles, and through the external source.

When operating a nuclear reactor, we would like to have the losses of neutrons through absorption and leakage be perfectly offset by the production of new neutrons through fission events and thus no external source would be necessary. However, it is very difficult to achieve an *exact* balance in practice. To enforce a numerical balance between the production and removal of neutrons, we divide the fission term by a parameter k, resulting in the k-eigenvalue problem

$$\hat{\Omega} \cdot \nabla \psi + \sigma \psi = \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \, \sigma_s \psi + \frac{1}{k} \chi(E) \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \sigma_f \psi \qquad (2.2)$$

after removal of the source term. Thus, when $k = 1$ an exact balance is attained and the system will maintain a steady state distribution. When $k < 1$ the removal terms exceed production and the neutron population will decrease in time and the system is termed subcritical. When $k > 1$ production exceeds removal and the neutron population will increase in time, resulting in a supercritical system. Numerically, k is an eigenvalue and ψ is its corresponding eigenvector. In particular, we are interested in estimating the largest value of k (and the corresponding ψ) such that (2.2) is satisfied, as this will indicate the behavior of the system after a long time.

2.1.1 Boundary Conditions

In order for (2.2) to be well-posed, it is necessary to place conditions on the angular flux on the external boundary of the problem, denoted by Γ. The simplest such condition is the vacuum boundary, which specifies that no neutrons enter the problem domain from the outside, i.e.

$$\psi(\mathbf{r}, \hat{\Omega}, E) = 0, \quad \mathbf{r} \in \Gamma, \ \hat{\Omega} \cdot \hat{\mathbf{n}} < 0 \qquad (2.3)$$

where $\hat{\mathbf{n}}$ is the outward unit normal vector on Γ. Vacuum boundary conditions are thus homogeneous Dirichlet conditions. Prescribed source (non-homogeneous Dirichlet) conditions frequently appear in the solution of the

source driven problem, but do not appear in the k-eigenvalue problem. Vacuum boundaries are typically used when the computational boundary represents the exterior of the physical system.

Another commonly used boundary condition is the reflecting boundary

$$\psi(\mathbf{r}, \hat{\Omega}, E) = \psi(\mathbf{r}, \hat{\Omega}', E), \quad \mathbf{r} \in \Gamma, \ \hat{\Omega} \cdot \hat{\mathbf{n}} < 0 \tag{2.4}$$

where $\hat{\Omega}' = \hat{\Omega} - 2(\hat{\Omega} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$ is the angle of specular reflection. Reflecting boundaries are frequently used to reduce a large problem to one of smaller size by exploiting axes of symmetry in the problem. Furthermore, it is common to impose reflecting boundaries on opposing sides to model an infinite array of a particular object (e.g. a single fuel pin or fuel assembly). Other boundary conditions such as white or periodic boundaries are occasionally used but will not be described here since they are less common and do not present any fundamental difficulty not seen in the vacuum or reflecting boundaries.

2.1.2 Scattering Integral

At this point in time a few comments about the scattering integral appearing in (2.2) are in order. The scattering cross section $\sigma_s = \sigma_s(\mathbf{r}, \hat{\Omega}' \to \hat{\Omega}, E' \to E)$ is a value that must be determined experimentally. Clearly it is not possible to obtain the probability of a neutron scattering from every possible direction on the unit sphere to every other direction, so some simplification must be made. First we note that the scattering cross section does not depend on the incident and exiting angles themselves, but merely on the change in angle during a collision. We can thus write

$$\sigma_s(\mathbf{r}, \hat{\Omega}' \to \hat{\Omega}, E' \to E) = \sigma_s(\mathbf{r}, \mu_0, E' \to E), \tag{2.5}$$

where $\mu_0 = \hat{\Omega}' \cdot \hat{\Omega}$ is the cosine of the scattering angle. It is then standard practice to expand the angular dependence of this quantity in terms of

Legendre polynomials so that

$$\sigma_s(\mathbf{r}, \mu_0, E' \to E) = \sum_{\ell=0}^{\infty} \sigma_{s,\ell}(\mathbf{r}, E' \to E) P_\ell(\mu_0), \tag{2.6}$$

where P_ℓ is the Legendre polynomial of degree ℓ and $\sigma_{s,\ell}$ is the ℓ^{th} angular moment of the scattering cross section. These angular moments are typically tabulated in large data libraries for use in computer codes. Obviously only a finite number of moments can be used and it usually suffices to include only a small number of terms (the commonly used ENDF data sets contain scattering moments up to degree 5 [1]).

With the expression in (2.6), we can now write the scattering integral (suppressing the dependence on the spatial location $\mathbf{r}$) as

$$\begin{aligned} & \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \, \sigma_s(\hat{\Omega}' \to \hat{\Omega}, E' \to E) \psi(\hat{\Omega}', E') \qquad (2.7) \\ = & \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \sum_{l=0}^{N_M} \sigma_{s,\ell}(E' \to E) P_\ell(\hat{\Omega}' \cdot \hat{\Omega}) \psi(\hat{\Omega}', E') \\ = & \int_0^\infty dE' \sum_{l=0}^{N_M} \sigma_{s,\ell}(E' \to E) \int_{4\pi} d\hat{\Omega}' \, P_\ell(\hat{\Omega}' \cdot \hat{\Omega}) \psi(\hat{\Omega}', E'), \end{aligned}$$

where N_M is the order of the scattering expansion. The addition theorem for Legendre polynomials states that

$$P_\ell(\hat{\Omega}' \cdot \hat{\Omega}) = \frac{1}{2\ell + 1} \sum_{m=-\ell}^{\ell} Y^*_{\ell,m}(\hat{\Omega}') Y_{\ell,m}(\hat{\Omega}), \tag{2.8}$$

where $Y_{\ell,m}$ is the spherical harmonic of degree ℓ and order m (because the quantities of interest are real–valued, it is standard to use the real spherical harmonics). Using the property $Y_{\ell,-m}(\hat{\Omega}) = (-1)^m Y_{\ell,m}(\hat{\Omega})$, this can be rewritten as

$$P_\ell(\hat{\Omega}' \cdot \hat{\Omega}) = \frac{1}{2\ell + 1} \sum_{m=0}^{\ell} (2 - \delta_{m,0}) Y^*_{\ell,m}(\hat{\Omega}') Y_{\ell,m}(\hat{\Omega}), \tag{2.9}$$

where $\delta_{i,j}$ is the Kronecker delta function. With this representation, the last integral in (2.7) becomes

$$\begin{aligned} & \int_{4\pi} d\hat{\Omega}'\, P_\ell(\hat{\Omega}'\cdot\hat{\Omega})\psi(\hat{\Omega}',E') \qquad (2.10)\\ = & \int_{4\pi} d\hat{\Omega}' \frac{1}{2\ell+1}\sum_{m=0}^{\ell}(2-\delta_{m,0})Y^*_{\ell,m}(\hat{\Omega}')Y_{\ell,m}(\hat{\Omega})\psi(\hat{\Omega}',E') \\ = & \frac{1}{2\ell+1}\sum_{m=0}^{\ell}(2-\delta_{m,0})Y_{\ell,m}(\hat{\Omega})\int_{4\pi} d\hat{\Omega}' Y^*_{\ell,m}(\hat{\Omega}')\psi(\hat{\Omega}',E') \\ = & \frac{1}{2\ell+1}\sum_{m=0}^{\ell}(2-\delta_{m,0})Y_{\ell,m}(\hat{\Omega})\phi_{\ell,m}(E'), \end{aligned}$$

where $\phi_{\ell,m}(E') \equiv \int_{4\pi} d\hat{\Omega}' Y^*_{\ell,m}(\hat{\Omega}')\psi(\hat{\Omega}',E')$ is the spherical harmonic coefficent of the angular flux (we will frequently refer to this as an angular flux moment or simply a flux moment). Now (2.7) can be fully written as

$$\begin{aligned} & \int_0^\infty dE' \sum_{l=0}^{N_M}\sigma_{s,\ell}(E'\to E)\int_{4\pi} d\hat{\Omega}'\, P_\ell(\hat{\Omega}'\cdot\hat{\Omega})\psi(\hat{\Omega}',E') \qquad (2.11)\\ = & \int_0^\infty dE' \sum_{l=0}^{N_M}\frac{\sigma_{s,\ell}(E'\to E)}{2\ell+1}\sum_{m=0}^{\ell}(2-\delta_{m,0})Y_{\ell,m}(\hat{\Omega})\phi_{\ell,m}(E'). \end{aligned}$$

2.2 Discretizations

The transport equation of (2.1) or (2.2) is a continuous function of space, angle, and energy. In order to obtain a system of equations amenable to solution on a computer, we must discretize this problem with respect to each of these independent variables. Although there are a large variety of techniques available in the literature for discretizing each of these quantities, to simplify the subsequent discussion we will only introduce some of the more commonly used methods here.

2.2.1 Energy

By far the most prevalent discretization of the energy variable in the transport equation is the multigroup method. Suppose that the highest energy acheivable in a particular environment is given by E_{max}. Then we can divide the interval $[0, E_{max}]$ into G subintervals (or groups) $[E_g, E_{g-1})$, where $E_G = 0$ and $E_0 = E_{max}$ (it is common convention that the smallest indexed groups represent the highest energies) and thus

$$\begin{aligned} \hat{\Omega} \cdot \nabla\psi + \sigma\psi &= \sum_{g'=1}^{G} \int_{E'_g}^{E_{g'-1}} dE' \int_{4\pi} d\hat{\Omega}' \, \sigma_s \psi \\ &+ \frac{1}{k}\chi(E) \sum_{g'=1}^{G} \int_{E'_g}^{E_{g'-1}} dE' \int_{4\pi} d\hat{\Omega}' \nu\sigma_f \psi. \end{aligned} \tag{2.12}$$

Instead of the continuous energy ψ, we are now interested in finding the group averaged angular flux $\psi^g = \int_{E_g}^{E_{g-1}} dE \; \psi(E)$ for each energy group. It would be desirable to define group total cross sections σ_g such that the interaction rate within each energy group for the multigroup quantities is the same as the original continuous formulation, i.e.

$$\sigma_g \psi_g = \int_{E_g}^{E_{g-1}} dE \; \sigma(E)\psi(E). \tag{2.13}$$

This condition is trivially satisfied if we select

$$\sigma_g = \frac{\int_{E_g}^{E_{g-1}} dE \; \sigma(E)\psi(E)}{\psi_g}, \tag{2.14}$$

however since ψ is an unknown quantity it is common practice to replace ψ in (2.14) with an approximate energy spectrum $f(E)$, giving

$$\sigma_g = \frac{\int_{E_g}^{E_{g-1}} dE \; \sigma(E)f(E)}{\int_{E_g}^{E_{g-1}} dE \; f(E)}. \tag{2.15}$$

The weighting function f is usually selected by performing a 0-D or 1-D calculation on a simplified geometry with a very fine energy structure (typically greater than 10,000 energy points). A similar process aiming to preserve reaction rates for scattering and fission events leads to the definition of the multigroup scattering and fission cross sections as

$$\sigma_{s,g'\to g} = \frac{\int_{E_{g'}}^{E_{g'-1}} dE' \int_{E_g}^{E_{g-1}} dE\ \sigma_s(E'\to E) f(E')}{\int_{E_{g'}}^{E_{g'-1}} dE\ f(E)} \tag{2.16}$$

$$\text{and } (\nu\sigma_f)_g = \frac{\int_{E_g}^{E_{g-1}} dE\ \nu(E)\sigma_f(E) f(E)}{\int_{E_g}^{E_{g-1}} dE\ f(E)}, \tag{2.17}$$

respectively (note that to preserve the fission production rate it is necessary to include ν inside the integral and thus $\nu\sigma_f$ is computed as a single quantity). If we further define

$$\chi_g = \int_{E_g}^{E_{g-1}} dE\ \chi(E), \tag{2.18}$$

then integration of (2.12) from E_g to E_{g-1} and substituting the appropriate multigroup quantities yields the multigroup transport equation

$$\hat{\Omega}\cdot\nabla\psi_g + \sigma_g\psi_g = \sum_{g'=1}^{G}\int_{4\pi} d\hat{\Omega}'\ \sigma_{s,g'\to g}\psi_{g'} + \frac{1}{k}\chi_g\sum_{g'=1}^{G}\int_{4\pi} d\hat{\Omega}'\ (\nu\sigma_f)_{g'}\psi_{g'}. \tag{2.19}$$

2.2.2 Angle

One of the oldest and still most commonly used methods for discretizing the angular variable in the transport equation is the discrete ordinates (or S_N) method [22]. Supporting the ubiquity of the discrete ordinates approach, Lewis and Miller [72] dedicate two full chapters to its development while providing only cursory treatment of other angular discretizations. The discrete ordinates method is a collocation method: (2.1) or (2.2) is enforced at only a finite number of discrete angles which we denote by $\{\hat{\Omega}_n\}_{n=1}^{N}$ and

we write $\psi_n = \psi(\hat{\Omega}_n)$. The discrete ordinates equations for the fixed source problem can be written as

$$\hat{\Omega}_n \cdot \nabla \psi_n + \sigma \psi_n = \tilde{\mathbf{q}}_n, \quad 1 \leq n \leq N, \tag{2.20}$$

where the right hand size contains both the original source term evaluated at the discrete ordinate directions and the scattering contribution of (2.11):

$$\tilde{\mathbf{q}}_n = \int_0^\infty dE' \sum_{l=0}^{N_M} \frac{\sigma_{s,\ell}(E' \to E)}{2\ell + 1} \sum_{m=0}^{\ell} (2 - \delta_{m,0}) Y_{\ell,m}(\hat{\Omega}_n) \phi_{\ell,m}(E') + \mathbf{q}_n, \tag{2.21}$$

where the integral defining $\phi_{\ell,m}$ must now be approximated by a quadrature formula, i.e.

$$\phi_{\ell,m} = \int_{4\pi} d\hat{\Omega} Y^*_{\ell,m}(\hat{\Omega}) \psi(\hat{\Omega}) \approx \sum_{n=1}^{N} w_n Y^*_{\ell,m}(\hat{\Omega}_n) \psi_n. \tag{2.22}$$

The selection of the discrete ordinates $\hat{\Omega}_n$ and the corresponding weights w_n (referred to as a quadrature set) has been the subject of much study. It is usually desirable to have the set of quadrature angles be symmetric with respect to reflections and rotations of 90° with respect to the coordinate planes (in the case of hexagonal geometries, invariance with respect to 60° rotations may be more appropriate). The use of these so-called level symmetric quadratures is especially vital when employing the reflecting boundary conditions described in Section 2.1.1, as it guarantees that the angle of reflection on a Cartesian boundary will also be contained in the quadrature set. An even number of angles is invariably used because this avoids issues in applying boundary conditions for angles parallel to problem boundaries [113]. The remaining degrees of freedom in the selection of the discrete ordinates and their weights are typically used to maximize the accuracy of the quadrature set (usually accomplished by having the set correctly integrate the maximum number of spherical harmonics for a given order).

2.2.3 Space

Unlike the energy and angular treatment where a single method has largely proved to be the most common, a large variety of spatial discretizations have been introduced to the transport literature. Because the discussions in the remainder of this work are relatively independent of the spatial discretization and so as not to place preference on a single method, we elect to only briefly mention some of the more commonly used approaches and point the reader to the relevant literature for more detailed treatment.

One of the earliest and still most prevalent is a finite difference approach frequently referred to as the diamond difference approximation [72, 48]. The limitation of the finite difference method to structured meshes and a desire for increased accuracy has led to the popularity of finite element discretizations and in particular discontinuous Galerkin methods [96, 61, 123]. Transport solvers based on the method of characteristics [21] have also found much favor for their ease in handling complicated geometries, particularly with regards to nuclear reactor analysis [52, 58, 111]. In addition, the slice balance approach (or short characteristics method), a variant of the subcell balance methods [4], has gained significant attention in the nuclear reactor community [39, 49, 30, 54]. Although we do not discuss them in this work, formulations of the transport equation that are self-adjoint (at least for the monoenergetic problem) are sometimes preferred [103, 25] and a description of possible spatial discretizations for such formulations is given in [79].

It is also worth mentioning that many spatial discretizations suffer from a failure to maintain positivity of the computed solution when the spatial mesh is not sufficiently refined [69]. Instabilities in solution procedures (in particular nonlinear acceleration techniques) can arise from such negativities, so strategies have been developed to reduce or eliminate them. Such techniques typically result in nonlinear discretizations such as the nonlinear characteristics [119], exponential discontinuous [120], or fixup based strategies [53].

2.2.4 Alternate Discretizations

Although the multigroup, discrete-ordinates equations have been largely favored within the transport community, we find it necessary to briefly mention a few of the alternatives. The leading competitor to the discrete-ordinates angular discretization is the P_N method, based on a spherical harmonics expansion of the angular variable. This approach, however, has proved problematic in practice as high angular orders are often required for problems with significant anisotropy and solutions tend to be plagued with Gibbs-type oscillations in the vicinity of material discontinuities and problem boundaries [72]. Where the P_N approach has found favor, however, is in connection with second order forms of the transport equations such as the even-parity equations [36] and the self-adjoint angular flux equations [81]. This approach has been the basis for the EVENT [37] and PARAFISH [117] transport codes (among others). Additional angular discretizations that have been proposed include finite element approaches [32] and wavelet expansions [20]. Simultaneous treatment of both the spatial and angular discretization with finite elements (known as phase-space finite elements [74, 78]) has also appeared in the literature.

While every production-level deterministic transport solver of which we are aware is based on the multigroup energy treatment, a few studies have investigated the use of basis functions other than the piecewise-constant approach in the multigroup method. The use of wavelets as a basis was investigated in [130] for the non-smooth resonance energy region. In [40], an orthogonal polynomial expansion was presented as another viable alternative.

2.3 Discrete Formulation

In this section we fix a notation for the matrices appearing in the discrete transport equation. We try to remain as consistent as possible with established convention in the transport literature. We represent the discretized streaming and collision operator by L. The scattering integral of (2.11) is represented with three matrices: D represents the computation of spherical harmonic moments from an angular flux, S contains the scattering cross section data, and M represents the calculation of a discrete angular flux from a set of flux moments. The fission integral is treated in a manner identical to scattering except the matrix of fission data F replaces S.

2.3.1 Linear System

Equation (2.1) can now be written as

$$L\psi = MSD\psi + MFD\psi + \mathbf{q}, \tag{2.23}$$

where ψ and $\mathbf{q}$ now represent discrete approximations to the appropriate continuous quantities. This represents a linear system of equations (usually referred to as a fixed-source problem) to be solved for the vector ψ. Frequently in these problems the fission term will be dropped, either because fission is not being accounted for or because the fission data is simply added to the scattering data to produce effective scattering cross sections, allowing the equation to be written in a more standard matrix form as

$$(L - MSD)\psi = \mathbf{q}. \tag{2.24}$$

Multiplying this equation through by DL^{-1} and defining $\phi = D\psi$ produces the integral form of the transport equation

$$(I - DL^{-1}MS)\phi = DL^{-1}\mathbf{q}. \tag{2.25}$$

If the angular flux is desired, it can be recovered from the solution to (2.25) by the relationship $\psi = L^{-1}MS\phi + \mathbf{q}$. Because the combination of operators $DL^{-1}M$ commonly appear together (and in fact this combination frequently appears as a single computational routine in transport codes), some sources (e.g. [41]) designate the combination by a single operator T so that (2.25) becomes

$$(I - TS)\phi = DL^{-1}\mathbf{q}. \tag{2.26}$$

It is frequently convenient to write the fixed-source transport equation in the equivalent block form

$$\begin{bmatrix} L & MS \\ -D & I \end{bmatrix} \begin{bmatrix} \psi \\ \phi \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix}. \tag{2.27}$$

It is easy to see that (2.24) and (2.25) are simply the Schur complements of this system in the $(1,1)$ and $(2,2)$ blocks, respectively. Clearly this formulation of the problem would never be used for a numerical implementation as it is even larger than the formulation involving the angular flux, but it is nonetheless useful from a theoretical standpoint and may be useful in deriving algorithms and alternate formulations of the problem.

2.3.2 k-Eigenvalue Problem

Using the same notation, the discretized k-eigenvalue problem can be written as

$$(L - MSD)\psi = \frac{1}{k}MFD\psi. \tag{2.28}$$

As with the linear system, multiplying by DL^{-1} results in the integral formulation

$$(I - DL^{-1}MS)\phi = \frac{1}{k}DL^{-1}MF\phi \tag{2.29}$$

or

$$(I - TS)\phi = \frac{1}{k}TF\phi. \tag{2.30}$$

It is also possible to write a blocked form of the k-problem as

$$\begin{bmatrix} L & MS \\ -D & I \end{bmatrix} \begin{bmatrix} \psi \\ \phi \end{bmatrix} = \frac{1}{k} \begin{bmatrix} 0 & MF \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \phi \end{bmatrix}. \tag{2.31}$$

While (2.30) has the form of a generalized eigenvalue problem, it may sometimes be preferable to have it in the form of a standard eigenvalue problem. This can be accomplished by noting that the matrix on the left hand side is always nonsingular [43, 24], so that we can write

$$(I - TS)^{-1} TF\phi = k\phi. \tag{2.32}$$

To reduce the problem even further we make use of the fact that the fission matrix F can be written as the product of rectangular matrices as $F = \chi f^T$. Here χ contains the multigroup fission spectrum given by (2.18) and f^T contains the multigroup fission data of (2.17). This separation is possible due to the fact that the energy of a neutron resulting from a fission event is independent of the energy of the neutron inducing the fission (in contrast to standard scattering where the incident and exiting energies are strongly connected). Multiplying (2.32) by f^T and defining $\Gamma = f^T\phi$ results in

$$f^T (I - TS)^{-1} T\chi\Gamma = k\Gamma, \tag{2.33}$$

which we refer to as the fission source formulation. Such a formulation may be advantageous because the eigenvector has a much smaller size (it is a function of space but not energy) although the performance of numerical algorithms is expected to remain unchanged with this formulation because it has the exact same non-zero eigenvalues as (2.30) and (2.32) (for any matrices A and B such that the products AB and BA are defined and square, the nonzero eigenvalues of AB and BA will be identical, including multiplicity [59]).

2.4 Structure

In this section we investigate the matrix structure of the fully discretized integral transport equation (2.30). We denote the number of spatial unknowns by N_S, the number of energy groups by N_E, the number of angles by N_A and the number of spherical harmonic moments by N_M ($N_M = (N_L + 1)^2$, where N_L is the Legendre order used to expand the angular dependence of the scattering cross section). Thus the length of the vector ϕ is given by $N_S \cdot N_E \cdot N_M$ (independent of N_A).

2.4.1 Solution Vector

The particular structure of the matrices appearing in the k-eigenvalue problem depend on how the elements of the solution vector are ordered. We will briefly describe the ordering convention that we have adopted, which is quite standard and very natural with respect to the solver structure in most transport codes. For systems involving the angular flux ψ, we list all of the fluxes for the first group together, then all fluxes for the second group and so forth, producing

$$\psi = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_{N_E} \end{bmatrix}, \tag{2.34}$$

where the subscript indicates the energy group index. Within an energy group, we elect to group all elements for a given angle together, i.e.

$$\psi_g = \begin{bmatrix} \psi_g^1 \\ \vdots \\ \psi_g^{N_A} \end{bmatrix}, \tag{2.35}$$

where the superscript indicates the angle index. Thus ψ_g^n contains the angular flux at every spatial location for the g^{th} energy group and the n^{th} angle.

For the scalar flux, the ordering is essentially the same except that in place of an angular index we have the index of the spherical harmonic moment. This gives

$$\phi = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_{N_E} \end{bmatrix} \tag{2.36}$$

with

$$\phi_g = \begin{bmatrix} \phi_g^1 \\ \vdots \\ \phi_g^{N_M} \end{bmatrix}, \tag{2.37}$$

where this superscript provides the index of the spherical harmonic moment and each ϕ_g^ℓ contains a value for each spatial position. Wherever possible, we will continue to use the convention that subscripts will denote energy indices and superscripts will denote either angle or moment indices (the distinction should be clear from the context). Any deviations from this convention will be specifically noted.

2.4.2 Transport Matrix

Although the *s*ize of the system in (2.30) does not depend on N_A, the dimension of the transport matrix L is $N_S \cdot N_E \cdot N_A$. Multiplying by the integral transport matrices requires (effectively) inverting this matrix and because this action typically dominates the overall computational effort, the cost of a matrix-vector product tends to be proportional to $N_S \cdot N_E \cdot N_A$ (independent of N_M). Because the inversion of L dominates the overall cost, understanding the structure of this matrix is extremely important to the development of an efficient method.

Because L represents a discretization of the streaming and collision operators that contain no coupling between different energy groups, it can be

written as a block diagonal matrix with N_G diagonal blocks each of size $(N_S \cdot N_A) \times (N_S \cdot N_A)$, i.e.

$$L = \begin{bmatrix} L_1 & & \\ & \ddots & \\ & & L_{N_E} \end{bmatrix}. \tag{2.38}$$

Thus the problem of solving a linear system with L is reduced to the task of solving N_E smaller decoupled linear systems.

The structure of each L_g is dependent on the boundary conditions that are enforced. If the vacuum boundary conditions of (2.3) are used then L_g contains no coupling between different angles and is itself a block diagonal matrix containing N_A diagonal blocks each of size $N_S \times N_S$. If the reflecting boundary conditions of (2.4) are used, however, the situation becomes more complicated. Now we have a full matrix

$$L_g = \begin{bmatrix} L_g^{1,1} & \cdots & L_g^{1,N_A} \\ \vdots & \ddots & \vdots \\ L_g^{N_A,1} & \cdots & L_g^{N_A,N_A} \end{bmatrix} \tag{2.39}$$

where the off-diagonal blocks are very sparse matrices that couple the outgoing boundary angular flux for one direction to the incoming angular flux of another direction. There will be no more than 2 nonzero off-diagonal blocks in 2-D and no more than 3 in 3-D. If there are no pairs of opposite reflecting boundaries (e.g. reflection on both the left and right boundaries) then there will always be a permutation of (2.39) that is block lower triangular. If at least one pair of opposite reflecting boundaries is present then no permutation to block lower triangular form will exist.

The innermost level of structure involves the within-group, single angle diagonal block matrices $L_g^{i,i}$. For the vast majority of problems, these matrices are 'psychologically lower triangular,' i.e. there is a permutation of

the matrix that is lower triangular (for finite element discretizations with multiple unknowns per spatial cell they will be block lower triangular with the diagonal block size corresponding to the number of unknowns per cell). For structured mesh transport problems, determining the 'sweep order' (the permutation vector that produces a lower triangular ordering of the matrix) is generally trivial. For unstructured meshes, the determination of the lower triangular order is accomplished by viewing the mesh as a directed graph for each angle of interest and performing a traversal of this graph. For arbitrary 2-D meshes with convex elements and 3-D structured meshes a lower triangular ordering is guaranteed to exist. In 3-D, however, meshes consisting only of convex elements (and indeed even those consisting only of simplices) can potentially have cycles in their corresponding directed graph (see Figure 11 of [94] for an example of such a case). Under these circumstances, some technique must typically be used to identify and attempt to break the cycle (see [93] for a discussion of such cycle detection). For the purposes of this work, we will assume that all meshes are acyclic and thus each $L_g^{i,i}$ is effectively lower triangular. The process of solving a linear system with such a lower triangular matrix is commonly referred to in the transport literature as 'transport sweeping.'

2.4.3 Scattering and Fission Matrices

Because the scattering and fission matrices only couple different energy groups at a specific spatial location, their structure is much simpler to describe. The scattering matrix takes the form

$$S = \begin{bmatrix} S_{1,1} & \cdots & S_{1,N_E} \\ \vdots & \ddots & \vdots \\ S_{N_E,1} & \cdots & S_{N_E,N_E} \end{bmatrix}, \tag{2.40}$$

where each $S_{g,g'}$ is simply a diagonal matrix containing the scattering cross sections from energy group g' to group g (including anisotropic scattering) at each spatial location. If S is permuted so as to place all energy groups for a single spatial unknown together, the result is a block diagonal matrix with dense $N_E \times N_E$ blocks containing the full group-to-group scattering cross sections at a particular spatial location (this is more in line with the way the S would typically be stored).

The fission matrix F has the same structure as S except that the blocks have additional structure. $F_{g,g'}$ is a diagonal matrix that contains the product of the component of the fission spectrum in energy group g, χ_g, with the component of the fission cross section in group g', $(\nu\sigma_f)_{g'}$. This means that if F is permuted to cluster energy groups together, the result is a block diagonal matrix where the diagonal blocks are $N_E \times N_E$ rank-1 matrices (at spatial locations where no fissionable material is present there will be zero blocks). Furthermore, because fission events are isotropic in angle, locations corresponding to higher order Legendre moments will be zero. Thus the rank of F is given by the number of spatial locations where fissionable material is present and is much smaller than the total number of unknowns in the problem.

2.4.4 Transfer Matrices

Next we turn our focus to the angular transfer matrices D and M. The angular restriction matrix D is an $(N_S \cdot N_E \cdot N_M) \times (N_S \cdot N_E \cdot N_A)$ matrix that converts an angular flux vector to a vector of flux moments. At the outermost level it is a block diagonal matrix consisting of N_E identical blocks, i.e.

$$D = \begin{bmatrix} \tilde{D} & & \\ & \ddots & \\ & & \tilde{D} \end{bmatrix}, \tag{2.41}$$

where $\tilde{D}$ is an $(N_S \cdot N_M) \times (N_S \cdot N_A)$ matrix of the form

$$\tilde{D} = \begin{bmatrix} \tilde{D}^{1,1} & \cdots & \tilde{D}^{1,N_A} \\ \vdots & \ddots & \vdots \\ \tilde{D}^{N_M,1} & \cdots & \tilde{D}^{N_M,N_A} \end{bmatrix}. \tag{2.42}$$

Each matrix $\tilde{D}^{i,j}$ is a multiple of the identity matrix of dimension N_S, with the multiple being the spherical harmonic coefficient corresponding to the i^{th} moment and j^{th} angle (including the appropriate quadrature weights).

The angular prolongation matrix M can likewise be represented as

$$M = \begin{bmatrix} \tilde{M} & & \\ & \ddots & \\ & & \tilde{M} \end{bmatrix}, \tag{2.43}$$

where N_E identical matrices appear along the main diagonal, each of the form

$$\tilde{M} = \begin{bmatrix} \tilde{M}^{1,1} & \cdots & \tilde{M}^{1,N_M} \\ \vdots & \ddots & \vdots \\ \tilde{M}^{N_A,1} & \cdots & \tilde{M}^{N_A,N_M} \end{bmatrix}. \tag{2.44}$$

As with $\tilde{D}$, these matrices are multiples of the $N_S \times N_S$ identity matrix with the multiple now determined by evaluating the j^{th} spherical harmonic at the i^{th} angle.

2.4.5 Implementation Considerations

It is now possible to write (2.28) in the form

$$\begin{bmatrix} (L_1 - \tilde{M}S_{1,1}\tilde{D}) & \cdots & -\tilde{M}S_{1,N_E}\tilde{D} \\ \vdots & \ddots & \vdots \\ -\tilde{M}S_{N_E,1}\tilde{D} & \cdots & (L_{N_E} - \tilde{M}S_{N_E,N_E}\tilde{D}) \end{bmatrix} \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_{N_E} \end{bmatrix} = \frac{1}{k} \begin{bmatrix} \tilde{M}F_{1,1}\tilde{D} & \cdots & \tilde{M}F_{1,N_E}\tilde{D} \\ \vdots & \ddots & \vdots \\ \tilde{M}F_{N_E,1}\tilde{D} & \cdots & \tilde{M}F_{N_E,N_E}\tilde{D} \end{bmatrix} \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_{N_E} \end{bmatrix}. \tag{2.45}$$

Correspondingly, (2.30) can be written as

$$\begin{bmatrix} I - \tilde{D}L_1^{-1}\tilde{M}S_{1,1} & \cdots & -\tilde{D}L_1^{-1}\tilde{M}S_{1,N_E} \\ \vdots & \ddots & \vdots \\ -\tilde{D}L_{N_E}^{-1}\tilde{M}S_{N_E,1} & \cdots & I - \tilde{D}L_{N_E}^{-1}\tilde{M}S_{N_E,N_E} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_{N_E} \end{bmatrix} = \frac{1}{k} \begin{bmatrix} \tilde{D}L_1^{-1}\tilde{M}F_{1,1} & \cdots & \tilde{D}L_1^{-1}\tilde{M}F_{1,N_E} \\ \vdots & \ddots & \vdots \\ \tilde{D}L_{N_E}^{-1}\tilde{M}F_{N_E,1} & \cdots & \tilde{D}L_{N_E}^{-1}\tilde{M}F_{N_E,N_E} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_{N_E} \end{bmatrix}. \tag{2.46}$$

Frequently we may want to compute the action $\mathbf{y} = A\mathbf{x}$, where A is the matrix on the left hand side of (2.46). This can be accomplished by breaking $\mathbf{x}$ and $\mathbf{y}$ into N_E vectors of length $N_S \cdot N_M$ and carrying out the multiplication separately for each energy group (this process can in fact be completely parallelized with respect to energy). To compute the i^{th} component of the result vector, the process appears as

$$\mathbf{y}_i = \mathbf{x}_i - \tilde{D}L_i^{-1}\tilde{M}\left(\sum_{j=1}^{N_E} S_{i,j}\mathbf{x}_j\right). \tag{2.47}$$

This implies that each L_i must be exactly inverted in order to perform such a matrix-vector product. As noted previously, when reflecting boundary conditions are present in the problem there will generally be no permutation of

these L_i matrices that is lower triangular. Solving linear systems with these matrices must therefore be done iteratively, introducing a nested iteration to simply perform a matrix-vector multiplication. As an alternative, it is possible to eliminate this inner iteration at a cost of storing a few additional elements to the solution vector. To illustrate this, let us for a moment consider a monoenergetic form of (2.28). We can rewrite this in a form analogous to (2.31) as

$$\begin{bmatrix} L_D & -MS & -L_O \\ -D & I & 0 \\ -R & 0 & I \end{bmatrix} \begin{bmatrix} \psi \\ \phi \\ \psi_b \end{bmatrix} = \frac{1}{k} \begin{bmatrix} 0 & MF & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \phi \\ \psi_b \end{bmatrix}, \tag{2.48}$$

where L_D and L_O contain the diagonal and off-diagonal blocks of L, respectively, ψ_b contains the entries of the angular flux vector on the problem boundary, and R is a restriction matrix that selects elements on this boundary. Now we can eliminate the angular fluxes on the interior of the domain but retain the terms on the boundary, producing

$$\begin{bmatrix} I - DL_D^{-1}MS & -DL_D^{-1}L_O \\ -RL_D^{-1}MS & I - RL_D^{-1}L_O \end{bmatrix} \begin{bmatrix} \phi \\ \psi_b \end{bmatrix} = \frac{1}{k} \begin{bmatrix} DL_D^{-1}MF & 0 \\ RL_D^{-1}MF & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \psi_b \end{bmatrix}. \tag{2.49}$$

Now observe that performing a matrix-vector product with the matrices appearing in this equation involves solving linear systems with only the diagonal blocks of L (which are lower triangular) and thus requires no inner convergence iteration. This alternate integral formulation of the k-eigenvalue problem may offer significant computational savings when only matrix-vector products are necessarily at the small price of storing the angular flux along reflecting boundaries. It is expected that certain numerical properties of these matrices will be affected by this formulation, but the savings in computational effort will likely outweigh any deficiencies. This process is easily

extended to the multigroup problem by performing the same process for each energy group. The same process can also be applied to the fixed point problem of (2.24).

2.5 Spectral Properties

In order to design an efficient k-eigenvalue solver, it is necessary to have some understanding of the behavior of the eigenvalues. Because most solvers rely on solving linear systems with the matrices appearing in the transport equation, it is important to study the behavior the eigenvalue distribution of these matrices as well as the distribution of eigenvalues of the k-eigenvalue problem itself.

2.5.1 Transport Operators

Relatively little information has appeared in the transport literature concerning the behavior of the eigenvalues of the operators appearing in the transport equation. While this may seem strange considering the large body of literature pertaining to the transport equation, the complete spectrum has not been important until recently. Historically, fixed point iterations have been used to solve the transport equation and the convergence of such methods does not depend on the complete distribution of eigenvalues but rather only on a small number of very select values. As subspace methods become more common for use on the transport problem, the eigenvalue distribution will be more important as the convergence of these methods typically depends on the entire distribution of eigenvalues.

For the case of monoenergetic 1–D transport with isotropic scattering, it was shown in [43] that the continuous integral transport operator is self–adjoint, positive, and a compact perturbation to the identity and thus its

eigenvalues are discrete and lie within a bounded component of the positive real line. Similar observations were made in [102] for multidimensional transport, again only for the monoenergetic equation with isotropic scattering. In [25], Chang demonstrated that the transport equation discretized with discontinuous Galerkin finite elements could be slightly modified so as to retain the previously mentioned properties of the continuous operator, but again these findings were limited to the energy independent, isotropic scenario.

Because the extension of such results to the more complicated (and realistic) situations of the multigroup equations with anisotropic scattering is likely to be intractable, we choose to simply compute the entire spectrum for a small sample problem in order to make a few observations. We consider a small 2–D problem containing a 5×5 array of fuel pins surrounded by a small water reflector containing 296 spatial cells as shown in Figure 2.1. Figure 2.2 shows the complete spectrum of the matrix $(I - DL^{-1}MS)$ for this problem with 12 angles (an S_4 level symmetric quadrature), and 4 energy groups with isotropic or anisotropic scattering. The spatial discretization is the step characteristics method available through the NEWT [38] transport solver. When only isotropic scattering is included, the spectrum is entirely real and contained in the interval $(0, 1]$, consistent with the previously mentioned results. However, the inclusion of anisotropic scattering breaks this property, causing eigenvalues to appear with nonzero imaginary part. The real part of all eigenvalues is still contained within the same interval $(0, 1]$.

Figure 2.3 shows the spectrum for the same problem except that the matrix is now modified so that only the diagonal blocks of the matrix L are inverted and angular fluxes on reflecting boundaries are now stored in the solution vector. Now even with only isotropic scattering, the spectrum extends into the complex plane and furthermore the real component of the eigenvalues is no longer bounded above by 1. The addition of anisotropic scattering further adds to the eigenvalues appearing off of the real line and increases the largest

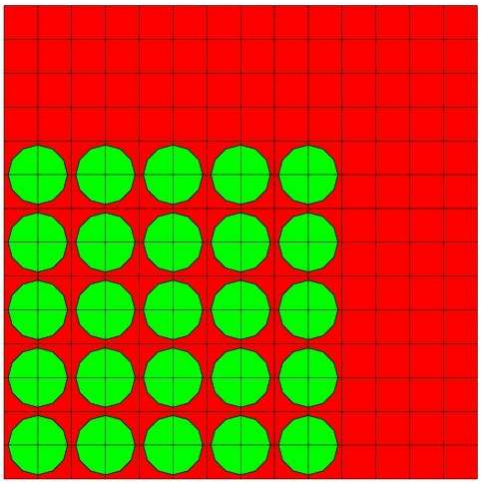
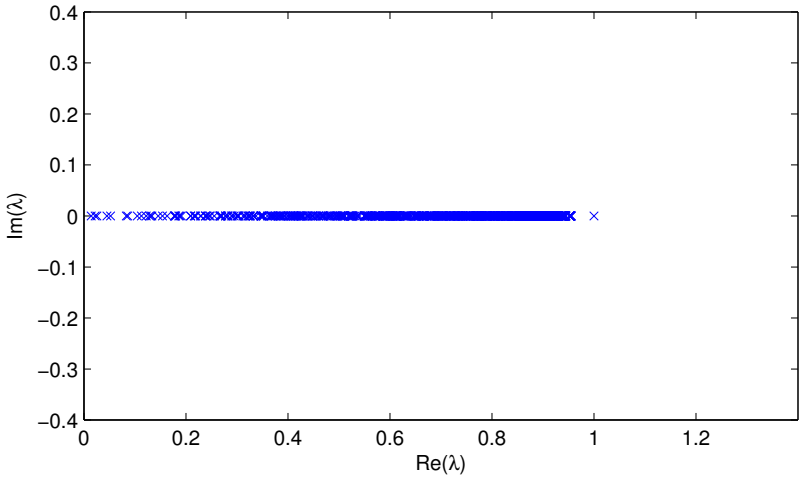


Figure 2.1: Geometry for spectrum computation.

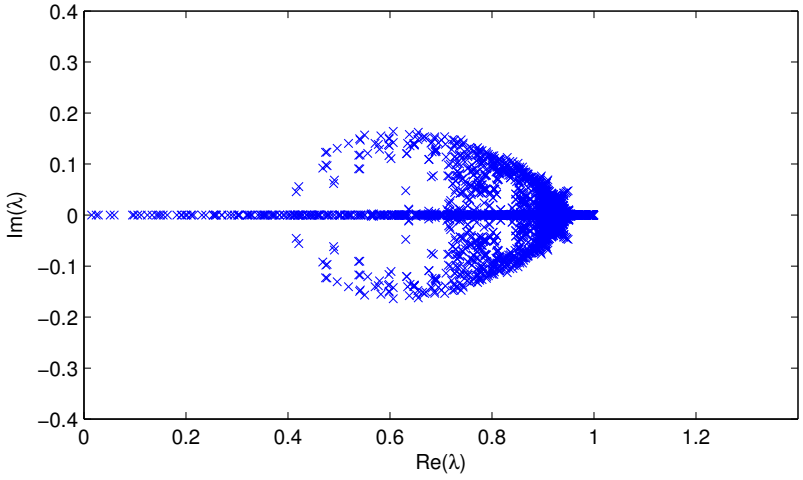
eigenvalue to approximately 1.3. This behavior could be very important when subspace type methods are used to solve linear systems with these matrices.

2.5.2 k-Eigenvalue Problem

Analogous to the situation with the spectrum of the transport operators, the transport community has never been interested in the behavior of the entire spectrum of the k-eigenvalue problem because the convergence of the standard eigensolver, power iteration, does not depend on the entire spectrum. As subspace solvers gain popularity, the spectrum as a whole becomes a more important factor. To begin with, we can say a few things about the number of nonzero eigenvalues by looking at the form of the problem given in (2.32). Clearly the number of nonzero eigenvalues of the overall operator cannot be greater than the rank of the matrix F. As was noted in 2.4.3, the fission matrix can be permuted to create a block diagonal matrix with $N_S \cdot N_M$ diagonal blocks that are each $N_E \times N_E$ rank-1 matrices (or all zero matrices).

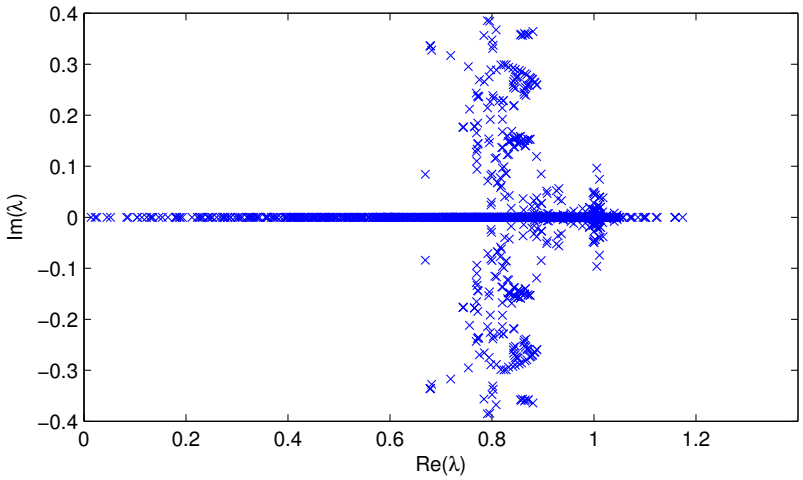


(a) Isotropic Scattering

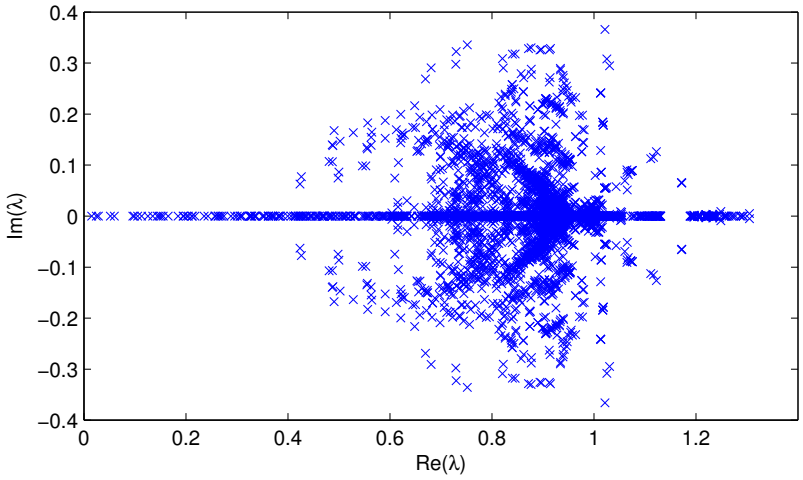


(b) Anisotropic Scattering

Figure 2.2: Eigenvalue spectrum of transport matrix A with L fully inverted.



(a) Isotropic Scattering

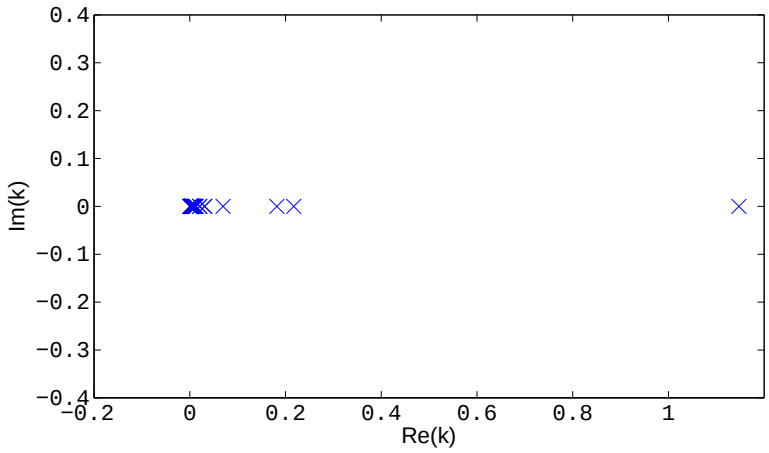


(b) Anisotropic Scattering

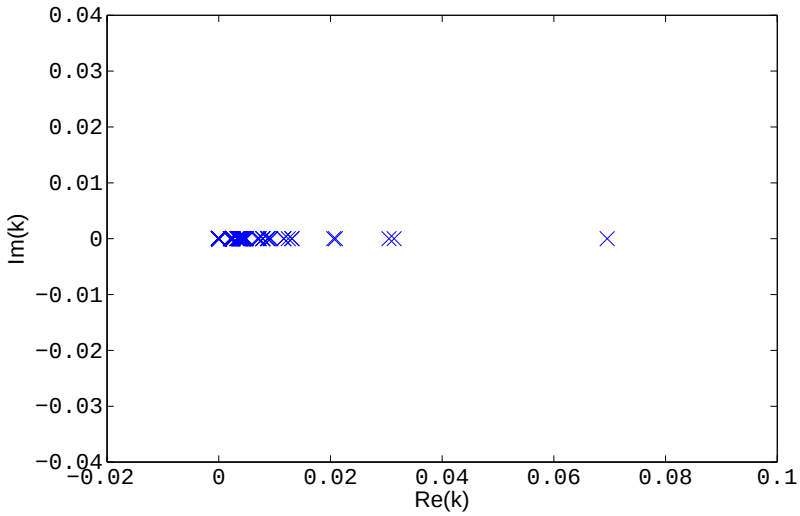
Figure 2.3: Eigenvalue spectrum of transport matrix A with only diagonal blocks of L inverted.

Additionally, the blocks corresponding to spatial locations with no fissionable material will be entirely zero, as will blocks corresponding to higher order anisotropic moments since fission is modeled is a purely isotropic occurrence. This places an upper bound on the number of nonzero eigenvalues as the number of spatial locations in the problem that contain fissionable material. The number of nonzero k-eigenvalues will therefore be considerably smaller than the size of the matrices.

Figure 2.4 shows the full k-eigenvalue spectrum at two different levels of zoom for the same problem from the previous section. This spectrum is fundamentally identical regardless of whether isotropic or anisotropic scattering is used. Due to the small size of this problem it is expected that the distribution may deviate significantly from problems of actual interest (for instance it is known that in many problems the second largest eigenvalue is very close to the largest), however certain properties are expected to carry over. First, note that all of the eigenvalues lie along the non-negative portion of the real line. This is somewhat unexpected considering the eigenvalues of the transport matrix are generally complex. Previous studies have concluded that the dominant eigenvalue is real, positive, and simple [75, 17], though we are not aware of any such guarantees for the remaining portion of the spectrum. Also of note in the spectrum is the fact that there are 100 nonzero eigenvalues, which is exactly the number of spatial cells containing fissionable materials (25 fuel pins with 4 cells per pin). This observation is consistent with the preceding discussion on the expected number of nonzero eigenvalues.



(a) Full Spectrum



(b) Detail Near Origin

Figure 2.4: Spectrum of k-eigenvalues.

Chapter 3

Eigensolvers

In this chapter, we discuss a variety of techniques that may be suitable for solution of the k-eigenvalue problem. In particular, we are interested in those methods that only require access to the relevant operators via matrix-vector products due to the fact that matrix entries will typically not be accessible in radiation transport solvers. We consider both solvers that search for solution to the standard eigenvalue problem

$$\mathcal{A}\mathbf{x} = k\mathbf{x} \tag{3.1}$$

and to the generalized eigenvalue problem

$$A\mathbf{x} = \lambda B\mathbf{x}. \tag{3.2}$$

Solvers directed at the generalized eigenvalue problem will be directly applicable to (2.30), where we are interested in finding the smallest $\lambda \equiv \frac{1}{k}$ and its corresponding eigenvector. For solvers that are applicable only to the standard eigenvalue problem, it is be necessary to use the standard eigenvalue formulation of (2.32) where we are interested in the largest k-eigenvalue.

3.1 Fixed Point Methods

The first class of eigensolvers to be considered are fixed point iterations, that is iterations in which the next iterate is entirely determined based on the single previous iterate. Thus, such iterations can be written in the form

$$\left(\lambda^{k+1}, \mathbf{x}^{k+1}\right) = f\left(\lambda^k, \mathbf{x}^k\right). \tag{3.3}$$

3.1.1 Power Iteration

Let us consider (3.1) and assume that the matrix $\mathcal{A}$ is diagonalizable and that there exists a unique eigenvalue of maximum modulus. Thus we can define a basis for $\mathcal{C}^N$ consisting of the eigenvectors $\mathbf{v}_i$ of $\mathcal{A}$, i.e. $\{\mathbf{v}_i\}_{i=1}^N$ is a basis for $\mathcal{C}^N$ and $\mathcal{A}\mathbf{v}_i = k_i\mathbf{v}_i$ for each i and the k_i have been ordered such that $|k_1| > |k_2| \geq ... \geq |k_N|$. Then any vector $\mathbf{x}_0 \in \mathcal{C}^N$ can be written as

$$\mathbf{x}_0 = \sum_{i=1}^{N} c_i\mathbf{v}_i \tag{3.4}$$

where the c_i are scalars. Applying the matrix $\mathcal{A}$ to this vector yields

$$\mathcal{A}\mathbf{x}_0 = \mathcal{A}\sum_{i=1}^{N} c_i\mathbf{v}_i = \sum_{i=1}^{N} c_i\mathcal{A}\mathbf{v}_i = \sum_{i=1}^{N} c_i k_i\mathbf{v}_i \tag{3.5}$$

and repeating this m times gives

$$\mathcal{A}^m\mathbf{x}_0 = \sum_{i=1}^{N} c_i k_i^m\mathbf{v}_i. \tag{3.6}$$

Separating the first term and dividing by k_1^m, we see that

$$\frac{1}{k_1^m}\mathcal{A}^m\mathbf{x}_0 = c_1\mathbf{v}_1 + \sum_{i=2}^{N} c_i\left(\frac{k_i}{k_1}\right)^m\mathbf{v}_i. \tag{3.7}$$

Because $|k_1| > |k_i|$ for each i, every term in the summation on the right hand side tends to zeros as m tends to infinity and in this limit we obtain

$$\frac{1}{\lambda_1^m}\mathcal{A}^m\mathbf{x}_0 \to c_1\mathbf{v}_1. \tag{3.8}$$

Thus for any initial guess, repeatedly multiplying by the matrix $\mathcal{A}$ produces a sequence of vectors that lie increasingly along the direction of the dominant eigenvector. The rate of convergence of this sequence of vectors depends on how rapidly the terms in the summation on the right hand side of (3.7) tend to zero. Since $\left|\frac{k_2}{k_1}\right| > \left|\frac{k_i}{k_1}\right|$ for any $i > 2$, this convergence will asymptotically behave as $\left|\frac{k_2}{k_1}\right|^m$. When $|k_2| \ll |k_1|$, we expect that convergence will be obtained quickly. However, for difficult problems it may happen that $k_2 \approx k_1$ and therefore the convergence will be quite slow. We refer to this ratio $\rho(\mathcal{A}) \equiv \left|\frac{k_2}{k_1}\right|$ as the dominance ratio of the matrix $\mathcal{A}$ and will occasionally use this as a rough estimate of the difficulty of an eigenvalue problem.

3.1.2 Shifted Power Iteration

If a reasonable approximation to the desired eigenvalue is available, it may be possible to use that information to improve the standard power method. Suppose that it is known that $\mu \approx \lambda_1$, then we can rewrite the original generalized eigenproblem as

$$A\mathbf{x} - \mu B\mathbf{x} = \lambda B\mathbf{x} - \mu B\mathbf{x} \tag{3.9}$$

or equivalently

$$(A - \mu B)\mathbf{x} = (\lambda - \mu)B\mathbf{x}. \tag{3.10}$$

Assuming that μ is not an exact eigenvalue of the original problem, the matrix on the left side of this equation is invertible and thus we can write

$$(A - \mu B)^{-1}B\mathbf{x} = \tau\mathbf{x}, \tag{3.11}$$

where $\tau \equiv \frac{1}{\lambda - \mu}$. Thus we have created a standard eigenvalue problem from a generalized eigenvalue problem, but we have shifted the system before applying the inverse. Applying the standard power method now to this

problem leads to convergence to the largest eigenvalue τ_1 at the rate

$$\rho\left((A-\mu B)^{-1}B\right) = \left|\frac{\tau_2}{\tau_1}\right| = \left|\frac{\lambda_1-\mu}{\lambda_2-\mu}\right|. \tag{3.12}$$

If the shift parameter μ is much closer to the desired eigenvalue than to any other eigenvalue then the convergence rate will be significantly improved. However, care must be taken since a poor selection of μ will result in convergence to the wrong eigenvalue. Because it is frequently difficult to identify a sufficiently accurate approximation to the desired eigenvalue a priori combined with the fact that the matrix $(A-\mu B)$ may be nearly singular and thus potentially difficult to invert, this shifted power iteration is not always a practical option.

3.1.3 Rayleigh Quotient Iteration

A natural extension to the shifted power iteration is to allow the shift parameter to vary at each iteration, usually by setting it equal to the current eigenvalue estimate. Thus the eigenvector update takes the form

$$\mathbf{x}^{(k+1)} = (A-\lambda^{(k)}B)^{-1}B\mathbf{x}^{(k)}. \tag{3.13}$$

If the eigenvalue is then updated according to the generalized Rayleigh quotient

$$\lambda^{(k+1)} = \frac{\mathbf{x}^*B\mathbf{x}}{\mathbf{x}^*A\mathbf{x}} \tag{3.14}$$

then the result is called (generalized) Rayleigh quotient iteration. It can be shown that this process results in convergence to an eigenpair quadratically for nonsymmetric matrices (and even cubically for symmetric matrices). There are two potential drawbacks to the use of Rayleigh quotient iteration. First, the linear system that must be solved at each iteration approaches a singular system as the iteration progresses, a fact that may pose a significant problem for some linear solvers. The other difficulty lies in the fact that

convergence to any specific eigenvalue cannot be guaranteed, and thus one may end up with a different solution than the one that is desired. These difficulties notwithstanding, a great deal of effort has been spent in studying the convergence behavior of Rayleigh quotient iteration [105, 83, 132].

3.1.4 Newton's Method

One way to approach the solution of an eigenvalue problem is to simply view it as an arbitrary nonlinear equation to be solved (the nonlinearity is due to the product of the eigenvalue and eigenvector, both unknowns). We can rephrase the solution of an eigenvalue problem as finding a solution to the nonlinear equation

$$f(\mathbf{x}, \lambda) = \begin{cases} (A - \lambda B)\mathbf{x} & = \mathbf{0}, \\ \frac{1}{2} - \frac{1}{2}\mathbf{x}^*\mathbf{x} & \end{cases} \tag{3.15}$$

where the second condition forces the eigenvector to have unit norm. From this formulation it seems natural to consider the use of Newton's method. The Jacobian matrix for this function can be written as

$$J(\mathbf{x}, \lambda) = \begin{bmatrix} (A - \lambda B) & -B\mathbf{x} \\ -\mathbf{x}^* & 0 \end{bmatrix}. \tag{3.16}$$

It was demonstrated in [92] (albeit for the standard eigenvalue problem) that this Jacobian is nonsingular even when λ and $\mathbf{x}$ are an eigenpair. Thus, Newton's method can be expected to converge quadratically [60]. The task of solving a linear system with the Jacobian closely resembles solving the linear system that appears in Rayleigh quotient iteration and thus the same difficulties should be expected, namely that the difficulty of solving such linear systems will increase in difficulty as the iteration progresses.

One possible issue with the use of Newton's method as an eigensolver arises from the fact that any eigenpair of the original problem will solve (3.15) and

thus there is no way to guarantee that the algorithm will converge to the specific eigenvalue of interest. Generally it will be necessary to perform a few iterations of a more robust iteration (such as power iteration) to provide an initial guess that is sufficiently close to the desired eigenpair, however this process may interfere with the overall efficiency of the solver.

3.2 Subspace Methods

For some problems it may be beneficial to use more than one vector in order to determine the next estimate for the desired iterate. One commonly used strategy is to construct a vector subspace of increasing dimension from which the next iterate will be selected. Suppose that $\mathcal{V} \subset \mathcal{C}^N$ is a subspace of dimension $m \ll N$ and we wish to extract an estimate for the desired eigenvector from this subspace and that $\mathcal{W} \subset \mathcal{C}^N$ is another subspace, also of dimension m. Now suppose that we want the eigenvalue relationship $\mathcal{A}\mathbf{v} = k\mathbf{v}$ (or $A\mathbf{v} = \lambda B\mathbf{v}$ for the generalized eigenvalue problem) to be satisfied for some vector $\mathbf{v} \in \mathcal{V}$ and scalar $k \in \mathcal{C}$ (or λ) in a weak sense when 'tested' against vectors $\mathbf{w} \in \mathcal{W}$, i.e.

$$(\mathcal{A}\mathbf{v}, \mathbf{w}) = (k\mathbf{v}, \mathbf{w}) \tag{3.17}$$

$$\text{or } (A\mathbf{v}, \mathbf{w}) = (\lambda B\mathbf{v}, \mathbf{w}) \tag{3.18}$$

for the standard or generalized eigenvalue problem, respectively. Equivalently, the goal is to find a vector $\mathbf{v} \in \mathcal{V}$ and a $k \in \mathcal{C}$ such that the eigenvalue residual is orthogonal to $\mathcal{W}$. If $\mathcal{W} = \mathcal{V}$ then this is a Ritz-Galerkin procedure; if $\mathcal{W} \neq \mathcal{V}$ then it is a Petrov-Galerkin procedure. If V is an orthogonal basis for $\mathcal{V}$, then any vector $\mathbf{v} \in \mathcal{V}$ can be written as $\mathbf{v} = V\mathbf{u}$ for some $\mathbf{u} \in \mathcal{C}^m$. If W is similarly an orthogonal basis for $\mathcal{W}$ then the solutions to (3.17) and

(3.18) satisfy

$$W^*\mathcal{A}V\mathbf{u} = kW^*V\mathbf{u} \tag{3.19}$$

$$\text{or } W^*AV\mathbf{u} = \lambda W^*BV\mathbf{u}. \tag{3.20}$$

We will refer to (3.19) and (3.20) as the projected eigenproblems. For the standard eigenvalue problem it is natural to take the spaces $\mathcal{V}$ and $\mathcal{W}$ to be biorthogonal, i.e. $W^*V = I$ so that (3.19) becomes the standard eigenvalue problem $W^*\mathcal{A}V\mathbf{u} = k\mathbf{u}$. Furthermore, when $\mathcal{A}$ is Hermitian the selection $\mathcal{W} = \mathcal{V}$ leads to certain optimality conditions being satisfied [98, Ch. IV]. For the generalized eigenvalue problem, there is generally no explicit motivation to select the subspaces to be orthogonal or even biorthogonal (although it may result in a smaller storage requirement).

Since the matrices appearing in (3.19) and (3.20) are of size $m \times m$, these low-dimensional eigenproblems can typically be solved quickly using (for instance) the QR or QZ methods [47]. Assuming that the subspaces $\mathcal{V}$ and $\mathcal{W}$ are chosen appropriately, the eigenvalues that satisfy the projected eigenproblems are typically good approximations to extremal eigenvalues of the original problem and the vectors $\mathbf{v} = V\mathbf{u}$ are approximations to the corresponding eigenvectors [98]. These approximations are commonly referred to as Ritz values and Ritz vectors, respectively (occasionally Petrov values/vectors when $\mathcal{W} \neq \mathcal{V}$). The primary differences between the subspace methods to be described in this section lie in the methods for selecting the subspaces $\mathcal{V}$ and $\mathcal{W}$ (though the implementations may be significantly different in order to exploit special structure of the particular method). Thus we can describe a general subspace eigenvalue solver with the very simple algorithm given by Algorithm 1.

Algorithm 1 General Subspace Eigenvalue Algorithm

while Not converged **do**
  Expand $\mathcal{V}$
  Expand $\mathcal{W}$
  Solve projected eigenproblem (3.19) or (3.20)
  Select desired Ritz/Petrov pairs
end while

3.2.1 Arnoldi's Method

One of the best known and widely used subspace eigenvalue algorithms is Arnoldi's method [9]. For the standard eigenvalue problem (3.1), Arnoldi's method selects both subspaces $\mathcal{V}$ and $\mathcal{W}$ to be the Krylov subspace

$$\mathcal{K} = \text{span}\{\mathbf{v}_0, \mathcal{A}\mathbf{v}_0, \mathcal{A}^2\mathbf{v}_0, \ldots\} \tag{3.21}$$

for some initial vector $\mathbf{v}_0$. This subspace selection gives the problem a special structure that can be exploited to reduce the computational costs of the method. In particular, the projection matrix $V^*\mathcal{A}V$ will be an upper Hessenberg matrix when V is an orthogonal basis for the Krylov subspace. This means that updating the projection matrix after expanding the subspace requires fewer operations and also solving the projected eigenvalue problem will be less costly (the QR algorithm that is typically used to solve the projected problem usually begins by reducing the matrix to upper Hessenberg form [47]). If the cost of a matrix-vector product is small and the dimension of the subspace is allowed to become large, the reduction in computational effort by exploiting the structure may be significant. For problems where the operator action is costly and the subspace size relatively small, however, these savings will likely be inconsequential.

The exploitation of the Krylov structure comes at an additional price when the matrix is not known exactly. In order to maintain the stability and accu-

racy of the iteration (largely due to the fact that the residual is approximated as a byproduct of the structure), it is generally necessary for matrix-vector products to be computed to a high level of accuracy or the final computed eigenvalues and eigenvectors may not be sufficiently accurate. Because the operator in the k-eigenvalue problem involves the inverse of a matrix that cannot be computed directly, each Arnoldi iteration will therefore require solving a linear system to a high level of accuracy. Some relief from these strict requirements is available by appealing to so-called inexact or relaxed Krylov subspace approaches. It was observed in [18] that it is possible to reduce the accuracy of the matrix-vector products as the iteration progresses while still maintaining the overall accuracy of the method. A theoretical justification of this phenomenon as well as rigorous bounds for the allowable error in intermediate matrix-vector products were provided in [106] and approaches for projection-type eigenvalue problems were addressed in [104]. The essential idea is that the error in the inner matrix-vector product may increase proportionally to the inverse of the current *outer* residual norm. The particular bounds, however, involve matrix norms that are unlikely to be available in practice and must therefore be conservatively approximated.

3.2.2 Generalized Davidson Method

An alternative to the Krylov subspace structure of Arnoldi's method was given by Davidson for the solution of eigenvalue problems in computational chemistry [34]. The crux of the Davidson method is that given an approximate eigenpair $(\mu, \mathbf{v})$, we should search for a correction $\mathbf{t}$ to the eigenvector estimate so that the eigenvalue relationship is (approximately) satisfied at the next iteration, i.e.

$$\mathcal{A}(\mathbf{v} + \mathbf{t}) = \mu(\mathbf{v} + \mathbf{t}). \tag{3.22}$$

Rearranging this relationship gives the correction equation

$$(\mathcal{A} - \mu I)\mathbf{t} = -\mathbf{r} \tag{3.23}$$

where $\mathbf{r} = (\mathcal{A} - \mu I)\mathbf{v}$ is the eigenvalue residual at the current iterate. It was then proposed to approximate the matrix $(\mathcal{A} - \mu I)$ on the left hand side of (3.23) with a preconditioner M. In the original paper this preconditioner was taken to be the diagonal of $\mathcal{A}$; non-diagonal preconditioners lead to what are commonly known as generalized Davidson methods [82, 55, 33]. Later generalizations have applied a Davidson-type idea to nonsymmetric matrices [101] as well as to the generalized eigenvalue problem, where the correction equation becomes

$$M\mathbf{t} = -\mathbf{r} \tag{3.24}$$

where the residual is now given by $\mathbf{r} = (A - \mu B)\mathbf{v}$ and $M \approx (A - \mu B)$. Somewhat confusingly, the term 'generalized Davidson method' or simply 'Davidson method' has also been used to describe any of these situations. For the remainder of this dissertation, we use the term 'Davidson method' for brevity to imply a generalized Davidson method in the broadest sense (nonsymmetric generalized eigenvalue problem with arbitrary preconditioner) and will specialize as necessary.

3.2.3 Davidson Preconditioners

We find it appropriate at this point to make a few observations about specific selections for the preconditioner with a Davidson method. We will see that several well known methods can be viewed as specializations of Davidson's method.

Arnoldi's Method

Applying Davidson's method to the generalized eigenvalue problem, it is necessary to approximately solve (3.24). One possibility is to make the

selection $M = A$. Observe that this implies $\mathbf{t} = -M^{-1}\mathbf{r} = -A^{-1}\mathbf{r} = -A^{-1}(A - \lambda B)\mathbf{v} = \lambda A^{-1}B\mathbf{v} - \mathbf{v}$. Since this vector will be made orthogonal to a subspace that contains the vector $\mathbf{v}$ and then normalized, this is equivalent to the selection $\mathbf{t} = A^{-1}B\mathbf{v}$. If this choice is made at every iteration, the result is that the basis vectors span the Krylov subspace for the matrix $A^{-1}B$. Thus, in exact arithmetic the iterates are identical to those obtained with the Arnoldi method. Some authors have referred to such a method as an 'inefficient' implementation of Arnoldi's method (or Lanczos in the symmetric case) due to the fact that it does not exploit the Krylov structure and thus the projected problems (3.19) or (3.20) will be unstructured rather than having the upper Hessenberg or tridiagonal structure of the Arnoldi and Lanczos processes, respectively. We would like to offer a slightly different interpretation of the situation: the Davidson method is not confined to the rigid structure of the Krylov subspace methods. Because the Davidson method does not rely on any particular structure (the residual is computed explicitly at each iteration), we are free to solve the necessary linear system to a very relaxed tolerance without adversely affecting the accuracy of computed values. In this respect, we suggest that such a Davidson method can be viewed as an inexact Arnoldi method. For very sparse problems where orthogonalization and other costs associated with the projected problem are not small compared to the cost of a matrix-vector product, this approach may not be advantageous. However, when the operator action is very expensive and orthogonalization negligible in comparison, the ability to apply the matrix inexactly may pay off.

Olsen's Method

It has occasionally been observed that the (generalized) Davidson method may stagnate, particularly if the preconditioner too closely approximates $(A - \mu B)$. This can easily be seen by considering the limiting case, where

$M = (A - \mu B)$. Here the correction equation $M\mathbf{t} = -\mathbf{r}$ becomes $(A-\mu B)\mathbf{t} = -\mathbf{r} = -(A - \mu B)\mathbf{v}$, the solution of which is clearly given by $\mathbf{t} = -\mathbf{v}$. Thus we are attempting to expand the subspace by a vector that already lies in the subspace and no progress can be made.

A remedy for the stagnation issue was proposed by Olsen for the solution of full configuration-interaction problems in computational chemistry [88]. This approach was to force the subspace expansion to be orthogonal to the current iterate by introducing the correction equation

$$(I - \mathbf{v}\mathbf{v}^*)M\mathbf{t} = -\mathbf{r} \tag{3.25}$$

(note that by the Galerkin process that produced the iterate $\mathbf{v}$, the residual $\mathbf{r}$ is necessarily orthogonal to $\mathbf{v}$, thus the problem is well-posed). The solution to this equation is given by $\mathbf{t} = \varepsilon M^{-1}\mathbf{v} - M^{-1}\mathbf{r}$, where $\varepsilon = \frac{\mathbf{v}^* M^{-1}\mathbf{r}}{\mathbf{v}^* M^{-1}\mathbf{v}}$ is chosen so that $\mathbf{v}^*\mathbf{t} = 0$. Although presumably more stable, a downside to Olsen's method is that two applications of the preconditioner must be performed at each iteration, rather than just one for the generalized Davidson method. If the preconditioner is very simple (e.g. the diagonal preconditioner used in the original Davidson method) then this presents a minor increase in computational cost, however it may be a significant consideration for more expensive preconditioners. Thus the gain in stability (that for many problems may be negligible) must be weighed against the additional computational cost of an extra preconditioner application.

Jacobi-Davidson Method

One subspace method that has attracted much attention in the recent mathematical literature is the Jacobi-Davidson method [109]. Like Olsen's method (and in fact it has been noted that Olsen's method is a particular case of Jacobi-Davidson), the Jacobi-Davidson algorithm was developed to alleviate issues related to stagnation of Davidson's method. The idea is to take

$M = (A - \mu B)$ in (3.24), but restrict the operator to a subspace orthogonal to the current eigenvector estimate. This produces the Jacobi-Davidson correction equation

$$(I - \mathbf{v}\mathbf{v}^*)(A - \mu B)(I - \mathbf{v}\mathbf{v}^*)\mathbf{t} = -\mathbf{r} \tag{3.26}$$

(the original Jacobi-Davidson method was suggested for the standard eigenvalue problem, though subsequent studies [107] also considered the generalized eigenvalue problem, producing the correction equation shown here). The Jacobi-Davidson method has been viewed alternatively as an inexact Newton method [108] or as an inexact Rayleigh quotient iteration [132]. Consistent with these interpretations, Jacobi-Davidson typically exhibits quadratic convergence. Particular implementations of Jacobi-Davidson have been described [44], and alternative approaches for expanding the subspaces have also been studied in depth [110].

Because the projections cause the operator in (3.26) to be dense even if A and B are sparse, the use of iterative methods (and Krylov methods in particular) is vital. Some techniques for preconditioning the correction equation have been suggested [110], although the task of developing effective preconditioners remains an area of continued activity.

Accelerated Rayleigh Quotient Iteration

Although it does not exactly fit into the Davidson framework, another option for subspace expansion uses the generalized Rayleigh quotient direction, i.e.

$$\mathbf{t} = (A - \mu B)^{-1} B\mathbf{v} \tag{3.27}$$

This subspace expansion strategy has been referred to as Rayleigh quotient iteration with subspace acceleration [108]. However, due to the already rapid convergence properties of Rayleigh quotient iteration on its own, acceleration

hardly seems necessary. Instead, we prefer to think of this as Rayleigh quotient iteration with subspace *stabilization*. The incorporation of the Rayleigh quotient direction should ensure rapid convergence, while the subspace nature of the process should help guarantee that this convergence is directed towards the particular eigenvalue of interest. Because of the relationship between the Jacobi-Davidson method and Rayleigh quotient iteration, it is expected that the behavior will be similar for many problems.

3.2.4 Reduced Memory Subspace Methods

A potential difficulty in the use of subspace type methods for large-scale eigenvalue problems lies in the significant amount of storage that is generally necessary to carry out the iteration. In general, one has to explicitly form the projected matrices appearing in (3.19) or (3.20). If $\mathbf{w}$ and $\mathbf{v}$ are the newest additions to their respective subspaces, in the (Jacobi-)Davidson method the projected matrices in the generalized problem is typically updated as

$$W^*AV = \begin{bmatrix} \tilde{W}^*A\tilde{V} & \tilde{W}^*A\mathbf{v} \\ \mathbf{w}^*A\tilde{V} & \mathbf{w}^*A\mathbf{v} \end{bmatrix} \tag{3.28}$$

and

$$W^*BV = \begin{bmatrix} \tilde{W}^*B\tilde{V} & \tilde{W}^*B\mathbf{v} \\ \mathbf{w}^*B\tilde{V} & \mathbf{w}^*B\mathbf{v} \end{bmatrix}, \tag{3.29}$$

where $W = [\tilde{W}\mathbf{w}]$ and $V = [\tilde{V}\mathbf{v}]$. Note that the $(1,1)$ blocks of each matrix here should be stored from the previous iteration and requires no additional computation so that only the final row and column of each matrix needs to be computed at each iteration (this is a difference between a Davidson solver and an Arnoldi approach because in the Arnoldi method the Krylov structure ensures that the dot products in the bottom row are zero). The $(2,1)$ block of each of these matrices indicates the need to compute the products of A and B with each of the vectors in $\tilde{V}$ (for the standard eigenvalue problem

only a product with $\mathcal{A}$ is required). Since these vectors have been computed at previous iterations, it is standard to store not only the basis vectors V and W but also V_A and V_B, the vectors of V multiplied by the matrices A and B, respectively. However, this approach requires storing as many as four sets of basis vectors for the generalized eigenproblem which may be prohibitive for very large problems. Furthermore, because the eigenvalues and eigenvectors of the projected eigenproblems may have nonzero imaginary part even if the original matrices are real-valued (and in fact this may occur even in the case where the original eigenproblem has an entirely real-valued spectrum), the stored basis vectors must be complex-valued. The number of real-valued floating point numbers that must be stored is therefore on the order of $8Nk$, where N is the size of the original matrices and k is the maximum allowed dimension of a subspace.

One simple choice to reduce the required memory is to take the test subspace $\mathcal{W}$ equal to the search subspace $\mathcal{V}$. This brings the necessary storage down to $6Nk$ real values. Furthermore, there are strategies [118] that allow only real arithmetic to be used by appealing to the real Schur or generalized real Schur forms [47]. One such strategy is outlined in Section 3.2.5. The use of real arithmetic in storing the basis vectors reduces the required storage by a factor of two, bringing the total down to only $3Nk$ values. For problems where this requirement is still a burden, it may be possible to reduce the storage requirement even further at the cost of a few extra matrix-vector products, provided that it is possible to perform multiplications with the matrix adjoint. To see this, observe that the $(2,1)$ block of (3.28) can be written as

$$\begin{aligned} \mathbf{w}^* A\tilde{V} &= \left((\mathbf{w}^* A\tilde{V})^*\right)^* \\ &= \left(\tilde{V}^* A^* \mathbf{w}\right)^* \\ &= (A^*\mathbf{w})^*\tilde{V}. \end{aligned} \tag{3.30}$$

Thus the need to store the basis vectors V_A has been replaced by a single adjoint matrix-vector product at each iteration. The same argument holds for eliminating the storage of V_B in favor of multiplying by B^* once per iteration. If all of the preceding modifications are made then only a single set of real basis vectors must be stored in order to implement a subspace eigenvalue method (an eight-fold reduction relative to the general case). Further reduction in memory requirements (in addition to or instead of any of these modifications) can be achieved by periodically restarting the method, as will be described in 3.2.6.

3.2.5 Schur Forms

In describing the general form of a subspace eigenvalue algorithm, we have so far merely stated that the projected eigenvalue problem should be solved and the Ritz pair most closely satisfying the selection criteria should be chosen. In practice, care should be taken that this problem is solved in a numerically stable way. In [44] a detailed algorithm is proposed that utilizes a Schur form (or generalized Schur form for the generalized eigenvalue problem) for the solution of the projected problem. The orthogonal transformations used in such Schur decompositions are preferred not only for the numerical stability but also because they facilitate restarting techniques such as those to be described in 3.2.6.

Another consideration arises due to the potential appearance of complex eigenvalues due to nonsymmetric matrices. Indeed, it is possible for such complex eigenvalues to appear in the projected problem even if the original eigenproblem consists entirely of real eigenvalues. For problems where it is known a priori that the eigenvalue or eigenvalues of interest are real valued, it is usually highly desirable to avoid the use of complex arithmetic as this essentially doubles the number of floating point operations that must be

performed (as well as doubling the storage requirements). It is possible to avoid such complex arithmetic by making use of a real Schur form [47] rather than the standard Schur form. The real Schur form (and generalized real Schur form) is very closely related to the standard Schur form except rather than a unitary reduction to a triangular form where the eigenvalues appear on the main diagonal of the triangular matrix, the real Schur form involves an orthogonal reduction to a block triangular form in which the diagonal blocks are either 1×1 or 2×2. The 1×1 blocks contain the real eigenvalues and the 2×2 blocks represent complex conjugate pairs of eigenvalues. The real Schur form thus allows complex arithmetic to be entirely avoided (unless the complex eigenvalues need to be determined) and because it involves only orthogonal transformations the stability properties of the standard Schur form are retained. A Jacobi-Davidson implementation that uses the real Schur form can be found in [118]; the Davidson solver that is used to obtain numerical results later in this work uses an analogous implementation.

3.2.6 Restarting Subspace Methods

For large problems, the amount of memory required for a subspace eigenvalue solver may be prohibitively large, even if the simplifications described in 3.2.4 are utilized. It is therefore often necessary in practice to set a maximum allowable size for the subspaces and restart the algorithm when this maximum size is reached. Restarting generally consists of determining one or more vectors from the current subspace that retain as much valuable information about the approximate solution while de-emphasizing unwanted vectors.

One of the most well known restarting procedures is the implicit restarting process [112, 71], originally suggested for the Arnoldi method but subsequently used for other subspace eigensolvers. If m is the maximum allowable subspace dimension and j is the number of eigenvalues that are sought, im-

plicit restarting applies $m-j$ iterations of the implicitly shifted QR algorithm [47] with the unwanted Ritz values as shifts. This in effect minimizes the presence of the unwanted Ritz vectors. It has been observed that it is occasionally beneficial to retain more vectors after a restart than simply the number of desired eigenvalues. To take this into account, thick restarting procedures have been proposed that allow for a specified number of vectors to be kept [114]. Explicit restarting strategies have also been used that simply retain the portion of the current subspace that appears to be most relevant rather than removing the unwanted portion as with implicit restarting techniques.

Considerations for the generalized eigenvalue problem are generally slightly different than in the standard case. The method described in [44] is a natural use of the generalized Schur decomposition that is used to solve the projected eigenproblem. For the projected eigenproblem 3.20, the generalized Schur decomposition computes upper triangular matrices S and T (these will be block upper triangular if the real Schur form is used) and unitary matrices Q and Z such that

$$\begin{aligned} W^*AV &= QSZ^* \\ \text{and} \quad W^*BV &= QTZ^* \quad . \end{aligned} \tag{3.31}$$

This decomposition can be ordered so that the Ritz values most closely matching the selection criteria appear at the upper left portion of S and T (an algorithm for performing such a reordering can be found in [47]). Now we can rearrange (3.31) to form

$$\begin{aligned} S &= Q^*W^*AVZ = (WQ)^*A(VZ) \\ \text{and} \quad T &= Q^*W^*BVZ = (WQ)^*B(VZ) \quad . \end{aligned} \tag{3.32}$$

The columns of VZ and WQ now contain the ordered right and left Ritz vectors, respectively. The natural selection to restart is to set V and W equal to the first j columns of their respective matrices so that $V \leftarrow VZ(1:j)$

and $W \leftarrow WQ(1:j)$. The parameter j can be chosen to retain the desired number of vectors. With this selection, the leading $j \times j$ submatrices of S and T can be stored as W^*AV and W^*BV, respectively. This allows the structure of (3.28) and (3.29) to be maintained and thus the projected matrices do not need to be recomputed upon restarting.

If, however, a Ritz Galerkin procedure has been used (for instance, to reduce memory costs as described in 3.2.4), an additional complication arises. Observe that in this case the equivalent of (3.32) becomes

$$\begin{aligned} S &= Q^*V^*AVZ = (VQ)^*A(VZ) \\ \text{and} \quad T &= Q^*V^*BVZ = (VQ)^*B(VZ) \quad . \end{aligned} \tag{3.33}$$

To continue with a Ritz Galerkin process, we would like to proceed with the first j columns of VZ as the new set of basis vectors. But if this is done then the leading $j \times j$ submatrices of S and T are no longer equal to V^*AV and V^*BV. In order to proceed, it becomes necessary to explicitly compute the projected matrices before continuing. This requires computing each of the j^2 entries of each projected matrix, where each entry requires performing a dot product with two vectors of length N. The use of a Ritz Galerkin method on the generalized eigenvalue problem therefore incurs an additional computational cost at each restart which may be very large, especially if the matrices A and B are very sparse or a large restart dimension is desired. This computational burden must be weighed against the benefit associated with the reduced memory costs of the Ritz Galerkin process when determining the best option for a given application.

Chapter 4

Transport Solvers

In this chapter we describe the techniques that are most commonly used for solving various forms of the discretized radiation transport equation. First we describe methods for solving linear systems (fixed source problems) involving the transport equation. These techniques are of significant interest on their own as these linear systems appear in radiation shielding applications, medical imaging, cancer therapy and oil well logging in addition to many other applications, but are especially relevant to the current discussion of k-eigenvalue solvers since the solution of a linear system is nearly always a requisite component. We also provide a description of methods that have been used for solving the k-eigenvalue problem. A brief discussion of techniques used to parallelize transport computations is also included, as parallelization can significantly impact algorithm selection.

4.1 Monoenergetic Solvers

A large portion of the radiation transport literature has been focused on the monoenergetic transport equation. The reason for this is largely because most multigroup solvers are based around a block Gauss-Seidel iteration,

as will be discussed in the next section, and thus the multigroup problem is reduced to the task of solving a sequence of monoenergetic equations. Although in the current study our aim is to reduce the dependence on such nested iterations, we nonetheless find it important to discuss some of the standard strategies employed for these problems.

4.1.1 Richardson Iteration

Perhaps the simplest technique, and one that is very natural for the transport equation, is to use a Richardson iteration of the form

$$\phi^{m+1} = DL^{-1}(MS\phi^m + \mathbf{q}) \tag{4.1}$$

(note that the right hand side could be written as $DL^{-1}MS\phi^m + \mathbf{b}$, where $\mathbf{b} = DL^{-1}\mathbf{q}$, but the form shown above more closely reflects a standard implementation). This strategy is typically known as source iteration (or iteration on the scattering source) in the transport community [48]. Source iteration has the physical interpretation that the iterate obtained after m iterations is the exact distribution of all neutrons that have undergone at most $(m-1)$ collisions. This interpretation, however, points directly to the fundamental drawback of source iteration: for a system in which the probability of a particle scattering is much higher than its probability of being absorbed it may require a large number of iterations to produce a sufficiently converged solution. Numerically, it can be shown that the rate of convergence is dictated by the scattering ratio $c = \frac{\sigma_s}{\sigma}$. In particular, c is an upper bound for the spectral radius of the iteration matrix $DL^{-1}MS$ [48].

4.1.2 Diffusion Synthetic Acceleration

Due to the tendency for source iteration to converge slowly for many problems, a number of methods have been developed to accelerate the convergence

of the solution process. By far the most studied of these methods is diffusion synthetic acceleration (DSA). DSA has its roots in the synthetic method of Kopp [64] and has been the subject of almost continual development in the nearly five decades since.

The essential idea behind DSA is to approximate the monoenergetic transport equation with a low order angular approximation, namely diffusion. This can be rigorously justified by first expanding the angular component of the angular flux of the continuous angle transport equation in spherical harmonics as

$$\psi(\mathbf{r},\hat{\Omega}) = \sum_{\ell=0}^{N_M}(2\ell+1)\sum_{m=-\ell}^{\ell}\phi_{\ell,m}(\mathbf{r})Y_{\ell,m}(\hat{\Omega}). \tag{4.2}$$

Inserting this into the transport equation, multiplying in turn by each $Y^*_{\ell,m}$ and integrating over the unit sphere gives an infinite set of coupled equations known as the P_N equations. Introducing the closure condition $\phi_{2,m} = 0$ (strictly speaking it is only necessary to set the gradient of the second degree moments to zero) results in the P_1 equations

$$\nabla\cdot\mathbf{J}(\mathbf{r}) + (\sigma-\sigma_{s,0})\phi_{0,0}(\mathbf{r}) = \mathbf{q}_0 \tag{4.3}$$

$$\frac{1}{3}\nabla\phi_{0,0} + (\sigma-\sigma_{s,1})\mathbf{J}(\mathbf{r}) = \mathbf{q}_1 \tag{4.4}$$

where $\mathbf{J}(\mathbf{r})$ is the current vector given by

$$J_k(\mathbf{r}) = \int_{4\pi} d\hat{\Omega}\ (\hat{\mathbf{n}}_k\cdot\hat{\Omega})\psi(\mathbf{r},\hat{\Omega}), \quad k = x,y,z \tag{4.5}$$

($\hat{\mathbf{n}}_k$ is the unit vector in the k direction) and $\mathbf{q}_i$ is the i^{th} moment of the external source term. Setting $\mathbf{q}_1 = \mathbf{0}$, (4.4) can be rewritten in the form of a Fick's law as

$$\mathbf{J}(\mathbf{r}) = -\frac{1}{3(\sigma-\sigma_{s,1})}\nabla\phi_{0,0}. \tag{4.6}$$

Finally, substituting this expression back into (4.3) yields the diffusion equation

$$-\nabla\cdot D\nabla\phi_{0,0} + (\sigma-\sigma_{s,0})\phi_{0,0} = \mathbf{q}_0, \tag{4.7}$$

where the diffusion coefficient is defined as $D = \frac{1}{3(\sigma - \sigma_{s,1})}$.

It has been shown that in the limit as the scattering ratio approaches unity and the total cross section tends to infinity (a limit known as the asymptotic diffusion limit and precisely the regime where source iteration exhibits the worst performance), the solution to this diffusion equation satisfies the scalar flux for the original transport equation [10]. While extremely effective for certain problems, it was observed that such diffusion acceleration had a tendency to lose its effectiveness or even diverge for other problems, particularly those with coarse spatial meshes [95]. This led to the discovery that if the diffusion equation is discretized in a manner consistent with the spatial discretization of the transport equation then it is possible to obtain an acceleration scheme that is unconditionally effective [6, 7]. This observation sparked a long line of research concerning the development of unconditionally effective DSA strategies including the development of a four step procedure to obtain a consistent diffusion equation from the discretized transport equation [67, 68, 66]. Subsequent analyses have solidified the understanding of DSA by placing it within a rigorous linear algebra framework as a preconditioned Richardson iteration [43, 10] and demonstrating through Fourier analysis that the spectral radius of the preconditioned DSA iteration matrix is bounded by $\rho_{DSA} \approx 0.2247c$ [5], a reduction of more than a factor of four relative to the unaccelerated iteration!

Although very promising in theory, in practice it can be difficult to formulate consistent DSA methods for many discretizations, especially in 2-D or 3-D. The resulting equations are frequently themselves so difficult to solve that several studies have focused exclusively on finding methods for solving the diffusion equations [124]. An alternative to the fully consistent formulations has been the class of partially-consistent DSA techniques such as those found in [121] and later improved in [122]. The development of efficient diffusion acceleration methods remains an area of active research.

4.1.3 Other Linear Preconditioners

The primary idea behind DSA is the use of a low angular order approximation (namely diffusion) to accelerate the convergence of the high order transport calculation. An alternative approach to DSA is to use a low order angular approximation that is itself a transport problem. Examples of this include S_2 Synthetic Acceleration (S_2SA) and the more general Transport Synthetic Acceleration (TSA) [5], both of which employ lower order discrete ordinates discretizations to approximate the original problem. The advantage of this approach is that it is no longer necessary to carefully construct a low order discretization because the low order problem is the same as the high order problem except with fewer angles. This also simplifies the creation of the preconditioner as many of the routines from the high order problem can be reused. Other preconditioners include the adjacent cell preconditioner [12] and the cell-wise block-Jacobi [97], both of which rely on the inversion of small portions of a larger transport problem.

4.1.4 Nonlinear Acceleration

Another commonly used class of acceleration methods consist of a low-order approximation to the transport equation that is formulated in a nonlinear manner based on the current approximation to the solution vector. In coarse mesh rebalance (CMR), coarse spatial regions are defined with the goal of improving the current iterate with a new flux estimate that satisfies a balance of particles within each coarse region [72]. To this end, coarse region multiplicative correction factors are defined such that the accelerated solution vector is the product of the unaccelerated estimate and the correction factor, that is

$$\psi^{k+1}(\mathbf{r}) = \psi^{k}(\mathbf{r})\mathbf{g}_i, \quad 1 \leq i \leq N_C, \tag{4.8}$$

where i is the index of the coarse mesh region to which $\mathbf{r}$ belongs and N_C is the number of coarse regions. The correction factors are then determined by substituting (4.8) into the transport equation and integrating over angle and each coarse region to obtain a system of N_C equations. Because this system of equations is sparse and (potentially) much smaller than the original problem, it can generally be solved using direct linear algebra techniques. An analysis of the convergence behavior of CMR can be found in [23].

Related to CMR is the coarse mesh finite difference (CMFD) method. Rather than enforcing a particle balance, CMFD imposes that the accelerated solution vector should satisfy a finite difference diffusion equation in which the diffusion coefficients are computed based on the current solution estimate. Unlike CMR, in which the coarse mesh regions can take an arbitrary shape, CMFD generally requires that the coarse regions are structured to allow convenient formulation of the finite difference equations. A convergence analysis for CMFD can be found in [70]. A comparison of CMR and CMFD can be found in [29] and a generalized coarse mesh rebalance technique is developed in [129] that unifies the two methods into a single framework. It is of interest to note that CMR and CMFD fall into the framework of aggregation/disaggregation methods that have been used in the mathematical community for many years, particularly for problems involving Markov chains [76, 27]. A discussion of convergence properties of aggregation/disaggregation methods for such problems can be found in [73].

It has been demonstrated that both CMR and CMFD can be quite effective under certain situations and both techniques have been implemented in production transport codes [31, 131]. Although the ideal convergence behavior may be very good, the actual performance generally depends quite strongly on the selection of the size of the coarse mesh regions; selecting the regions to be either too large or too small can lead to significant degradation in performance and even divergence.

Not too unlike the CMFD approach is the quasi-diffusion (QD) approach first formulated by Gol'din [46], in which a diffusion-like equation (in 2-D and 3-D the equation is similar to a tensor diffusion equation) is obtained by manipulating the first and second angular moments of the transport equation. The nonlinearity of QD is introduced through the so-called Eddington factor (or Eddington tensor) that plays a role analogous to a diffusion coefficient. As with DSA, the quasi-diffusion equation must be discretized in a manner that is consistent with the transport spatial discretization in order for the effectiveness of the scheme to be retained, a task that may be quite challenging for some discretizations and especially so on unstructured meshes. The development of efficient QD methods remains an area of continued research [128].

4.1.5 Krylov Methods

The conjugate gradient (CG) method [57] has been proposed as an alternative to standard solvers for the transport equation as far back as 1988, when Faber and Manteuffel observed that it can lead to much faster convergence than source iteration [43]. A downside to CG for the transport equation is that it requires the matrix to be symmetric, a property not typically possessed by the discrete transport matrices. However, symmetrization strategies have been proposed (i.e. [25]) to remedy this issue and thus allow for the use of CG. Such symmetrization strategies are generally not valid for problems with anisotropic scattering and thus have found little favor in the wider transport community.

Nonsymmetric Krylov solvers such as GMRES [100], on the other hand, have started to gain significant attention in recent years [51, 90, 87]. The ability to apply such methods directly to any (linear) spatial discretization without the need for symmetrization is a very attractive property. The addi-

tional computational effort of GMRES versus CG due to orthogonalization is likely to be minimal because of the already large computational cost of performing matrix–vector products. The extra storage requirements might be more significant, although the rapid rate of convergence tends to keep the necessary subspace size small.

One particularly appealing feature of Krylov methods involves their behavior when preconditioned with DSA. As noted previously, DSA methods must be formulated very carefully to retain their efficiency. It was observed in [125] that an inconsistent DSA typically retains its efficiency when used as a preconditioner to a Krylov method even in situations when it would normally be rendered ineffective.

4.2 Multigroup Solvers

In many physical systems, it is generally only possible for particles to lose energy in the course of a scattering event (this is called downscattering). In this case the scattering matrix S (and thus the entire transport matrix) will be block lower triangular. Solving a linear system with such a matrix can be accomplished by simply performing a block elimination procedure that involves solving a monoenergetic transport equation in each energy group. These solves can be performed using the strategies described in the previous section. However, when the thermal energy of the background material is comparable to the energy of the radiation particles it becomes possible for a particle to gain energy in a collision, a process known as upscattering. This upscattering causes the transport matrix to have nonzero blocks above the main diagonal. When the behavior of particles at low energies is important to the overall behavior of the system, upscattering can be a serious impediment to the solution of multigroup problems. For the remainder of this work we will assume that upscattering is significant and must be treated, a property

that is true for many nuclear reactor designs.

4.2.1 Block Gauss-Seidel

If upscattering is only weakly present, the block lower triangular portion of the transport matrix will be dominant. Thus the use of a block Gauss-Seidel iteration will lead to rapid convergence in many instances. The diagonal blocks form a series of monoenergetic transport equations of the form

$$(I - \tilde{D}L_g^{-1}\tilde{M}S_{gg})\phi_g^{m+1} = \tilde{\mathbf{q}}_g, \tag{4.9}$$

where

$$\tilde{\mathbf{q}}_g = \tilde{D}L_g^{-1}\tilde{M}\left(\sum_{g'<g} S_{g'g}\phi_{g'}^{m+1} + \sum_{g'>g} S_{g'g}\phi_{g'}^{m} + \mathbf{q}_g\right) \tag{4.10}$$

and $\mathbf{q}_g$ is the component of the original right hand side in group g. These monoenergetic equations can be solved using the techniques outlined in the previous section. Occasionally it may be desirable to perform a backward Gauss-Seidel iteration after the forward Gauss-Seidel sweep (forming a symmetric Gauss-Seidel iteration), though the performance gain from such an approach is likely to be fairly small.

4.2.2 Upscatter Acceleration

For systems in which the behavior of neutrons at low energies is important, a significant amount of upscattering will occur. Numerically, this indicates that the Gauss-Seidel iteration is failing to capture a significant portion of the physics in the problem. In fact, the spectral radius of the Gauss-Seidel iteration matrix can approach unity for problems with highly scattering materials (in [3] it was observed that the Gauss-Seidel spectral radius was 0.9984 for graphite and 0.9998 for heavy water with a particular 69 group cross section library). Similar to the use of Richardson iteration for the monoenergetic

equations, such an approach can require an exorbitant number of iterations to fully converge.

One of the few methods developed to accelerate Gauss-Seidel iterations is the two-grid method proposed by Morel in 1993 [3]. The method is motivated by a Fourier analysis that indicates that the slow convergence of Gauss-Seidel is caused by a single error mode that is nearly isotropic in angle and has a nearly constant energy distribution. The two-grid method seeks to annihilate this single error mode by collapsing the multigroup transport problem to a monoenergetic diffusion equation. The diffusion discretization is obtained through the standard techniques available in the DSA literature. The collapse to a monoenergetic problem is achieved by using a material-dependent restriction operator that consists of the eigenvector corresponding to the dominant eigenvalue of the Gauss-Seidel iteration matrix. This vector approaches a Maxwellian energy distribution as the Gauss-Seidel spectral radius approaches unity (a Maxwellian distribution can be viewed as the steady-state energy distribution of particles scattering in an infinite, purely scattering medium). In the ideal case, the two-grid method completely removes the error component corresponding to the dominant error mode and the convergence rate is then dictated by the magnitude of the second largest Gauss-Seidel eigenvalue, which is generally much smaller than the dominant eigenvalue. A similar method was proposed nearly simultaneously in [11], differing primarily in the choice of the restriction and prolongation operators.

The behavior of the two-grid method largely parallels that of DSA techniques for the monoenergetic problem. Most notably, it relies on the availability of a diffusion discretization that is consistent with the underlying transport discretization and its performance tends to significantly degrade (or diverge) in the presence of large material discontinuities. In order to address these drawbacks, a modified approach termed the transport two-grid

(TTG) was developed in [41] that uses the same transport discretization as the original problem. Even this approach, however, still suffers from convergence issues and has the potential to diverge for problems where the energy spectrum is not well represented by its infinite medium equivalent.

4.2.3 Krylov Methods

Historically, Krylov methods have not been used in the solution of the multigroup equations and little discussion of such an approach appears in the transport literature. This is presumably due to the belief that for large problems the memory requirements of a subspace approach will be prohibitive. However, such an approach has recently been employed in the Denovo [42] transport solver. Typically a Krylov method would only be used to solve the portion of the multigroup problem that contains upscatter (the 'upscatter block') and a block elimination procedure would be used for the block lower triangular part of the matrix, although early results indicate that it may be more efficient (albeit more memory intensive) to solve the entire multigroup problem with the Krylov method [35].

Due to the low cost of performing a matrix multiplication with the multigroup matrix relative to a Gauss-Seidel iteration, it is possible that the use of a Krylov method with no preconditioner could lead to an efficient solver. However, the memory requirement alone could render this impractical and present the need for preconditioning. Due to the density of the coefficient matrices, many standard methods for the iterative solution of sparse linear systems are not available [16] and either block approximations to the problem or physics-based approximations will be necessary. One possibility for preconditioning is to use the block lower triangular portion of the multigroup matrix as a preconditioner ('block Gauss-Seidel preconditioning'), which would cause the matrix-vector product to replicate the structure of a Gauss-Seidel

iteration that may be appealing from the standpoint of leveraging existing code when implementing a new solver. However, such an approach would typically require the use of a flexible Krylov solver (such as FGMRES [99]) because the preconditioner is computed iteratively and thus changes from one iteration to the next. Another disadvantage of the use of block Gauss-Seidel as a preconditioner to a Krylov method versus the use of iterative block Gauss-Seidel is due to the fact that the block Gauss-Seidel iteration operates on the current estimate of the solution vector, which is converging as the outer iteration progresses. Thus the right hand sides for the monoenergetic problems become very similar from one iteration to the next, indicating that the solution estimate from the previous iteration will provide an excellent initial guess for the current iteration. The result is that the cost of performing a Gauss-Seidel iteration will decrease by a significant amount over the course of the solution process, leading to an apparent acceleration of the iterations and potentially a large reduction in computational cost. When used as a preconditioner, the right hand sides for the monoenergetic equations are formed by direction vectors that are independent from one iteration to the next so that no good initial guess is available and thus the cost of an iteration remains constant throughout the process.

Other potential options for multigroup preconditioners are the two-grid and transport two-grid methods described in the previous section. The downside to these methods as preconditioners is that they must be used in conjunction with a block Gauss-Seidel preconditioner, increasing the computational cost of each iteration. Additionally, these methods have been designed specifically for use with an iterative method where convergence is entirely dictated by the spectral radius of the iteration matrix. The convergence of Krylov methods, however, is not so straightforward and generally depends on the behavior of the entire spectrum. Thus it is not immediately obvious that a method that seeks to reduce the spectral radius of the iteration matrix will necessarily

function well as a preconditioner.

4.3 Eigensolvers

We now turn our attention to the solution of the k-eigenvalue problem (2.30). While descriptions of many eigensolvers were provided in Chapter 3, we will focus on aspects of eigensolvers as they have typically been applied in the transport community.

4.3.1 Power Iteration

Because the k-eigenvalue problem can be converted into the standard eigenvalue problem (2.32) in which we are interested in only the dominant eigenvalue (and its corresponding eigenvector), the use of power iteration is a natural choice. All that is needed is to repeatedly multiply a starting vector by the matrix

$$\mathcal{A} = (I - TS)^{-1} TF \tag{4.11}$$

until convergence is reached. However, this means that each iteration requires solving a linear system with the multigroup transport equation. As described in the previous section, this is frequently accomplished using a block Gauss-Seidel iteration and in fact it is common to use only a single Gauss-Seidel iteration per outer iteration, resulting in the iteration

$$(I - TS_L)\,\phi^{m+1} = T\left(S_U + \frac{1}{k^{(m)}}F\right)\phi^m \tag{4.12}$$

where $k^{(m)}$ is the current eigenvalue estimate and S_L and S_U are the lower and strictly upper triangular portions of the scattering matrix, respectively. Although this approach may result in slightly slower convergence in terms of the number of outer (power) iterations, it is usually more than compensated by the elimination of an entire level of the iteration structure.

4.3.2 Shifted Iterations

In order to address issues related to slow convergence that generally plague the power method, some authors have been led to consider the use of the shifted power method (or inverse iteration) for the k-eigenvalue problem [8]. As was shown in Chapter 3, this approach results in a more rapidly convergent method. However, each iteration has been made more difficult because the linear system that must be solved now includes fission as well as scattering. Because elements of the fission matrix tend to be large above the main diagonal, a block Gauss-Seidel iteration on this linear system will converge much more slowly (if at all). This difficulty, combined with the fact that it may be quite difficult to arrive at a sufficiently accurate eigenvalue approximation to the true eigenvalue, has traditionally kept the shifted power iteration from being widely adopted.

A natural extension to the shifted power iteration is to dynamically update the shift parameter by setting it equal to the current eigenvalue estimate, resulting in a generalized Rayleigh quotient iteration. Now not only is the linear system that must be solved more difficult than in the original power method, but the difficulty increases with each outer iteration. Block Gauss-Seidel cannot be expected to converge for this problem and more advanced techniques will be necessary. Although not widely used in the transport community (likely due to the focus on block Gauss-Seidel for solving linear systems), this strategy is beginning to attract some attention [102].

4.3.3 Nonlinear Acceleration

Just as with the solution of linear systems involving the transport equation, convergence for basic solvers for the k-eigenvalue problem tends to be very slow for many problems of interest. With the power method, this tends to happen when a problem contains multiple large, loosely coupled regions [5].

The same techniques that are typically used to accelerate the convergence of linear solvers have also been employed for the solution of the k-eigenvalue problem. Thus there are adaptations of CMR and CMFD in which the low order problem is an eigenvalue problem rather than a linear system [29, 38]. The aggregation/disaggregation class to which these methods belong have also been used for the solution of other eigenvalue problems [28]. In particular, they have been used with a great deal of success for the problem of determining the stationary distribution of Markov chains [56, 115]. Similarly, there exists a variant of the quasi-diffusion method that can be used to accelerate the solution of the k-eigenvalue problem [128].

4.3.4 Krylov Methods

In recent years the Arnoldi method [9], and in particular the Implicitly Restarted Arnoldi Method [71] has started to draw attention for the solution of the k-eigenvalue problem. It was observed in [126] that IRAM is a powerful alternative to power iteration for problems with only downscattering (i.e. where the matrix S is lower triangular) but experiences some difficulties for the more general case with upscattering included. This is because each Arnoldi iteration requires applying the matrix

$$\mathcal{A} = (I - TS)^{-1} TF \tag{4.13}$$

to a given vector and in order to maintain the structure of the Arnoldi method (and thus the accuracy of the computed eigenvalue) it is necessary to apply this matrix to a very high level of accuracy. Because block Gauss-Seidel was being used to solve the necessary linear system, the result was a high cost per iteration. This is in contrast with the power method where very inexact applications of the matrix are generally permissible (for an extreme example of this see the flattened power iteration described in [45], where no within-group iterations are performed at all). More recent implementations

of the Arnoldi method (such as that in [42]) suggest that the use of a Krylov method to solve the linear system may greatly increase the performance of IRAM. We note in passing that other applications of Krylov methods to the k-eigenvalue problem using the Orthomin(1) algorithm have been described in [50] and [77].

4.3.5 Newton's Method

Another recently proposed solver for the k-eigenvalue problem is Newton's method. In terms of the transport operators, the function to be solved can be written as

$$f(\phi;\lambda) = \begin{cases} [I - T(S + \lambda F)]\phi \\ \frac{1}{2} - \frac{1}{2}\phi^T\phi \end{cases} = \mathbf{0}. \tag{4.14}$$

The Jacobian of this system is given by

$$J(\phi;\lambda) = \begin{bmatrix} I - T(S + \lambda F) & -TF\phi \\ -\phi^T & 0 \end{bmatrix}. \tag{4.15}$$

Because the Jacobian cannot actually be formed explicitly but matrix-vector products can be performed (somewhat) inexpensively, this situation is very suitable for Newton-Krylov [19] methods. This approach (along with a look at the related Jacobian-Free Newton Krylov or JFNK method [63]) was studied in depth in [45]. The JFNK approach was also considered in [89] for the related diffusion k-eigenvalue problem. Although favorable results have been reported for many problems, a significant drawback exists in that convergence to the desired eigenpair is not guaranteed. Clearly any eigenvalue/eigenvector pair will satisfy (4.14) and thus there is generally no reason to expect convergence to the dominant pair. Indeed, studies have shown that convergence to higher order harmonics is observed in practice. The only remedy for this situation is to improve the quality of the initial guess, usually by performing several iterations of the power method before beginning the Newton iterations. Even with improved starting vectors, the results in [45] illustrate the

complex nature of the convergence of Newton's method. This difficulty has generally prevented Newton's method from achieving widespread acceptance.

4.4 Parallelization

Due to the large dimensionality of the radiation transport equation, performing computations in parallel is an absolute necessity for large (and frequently even for moderately sized) problems. This is especially true in 3-D, but even in 2-D many transport solvers rely on the ability to parallelize the problem. Although shared memory parallelism has been used, large scale parallelism always comes in the form of distributed memory architectures and will be the focus of this discussion. Parallel strategies can be devised to decompose the problem with respect to any element of phase space: space, angle, or energy. Decomposition in angle or energy requires duplicating the entire spatial mesh on each processor, presenting a hefty burden on memory resources. Instead, decomposing the problem in space is nearly always relied upon.

4.4.1 Domain Decomposition

The simplest and perhaps most commonly used parallel strategy is frequently referred to as spatial domain decomposition or parallel block Jacobi [84]. In this strategy, the spatial domain is broken up into a number of smaller sections and each processor is assigned one such section. Each processor proceeds by solving the transport equation on its own domain, with communication between processors taking place in the form of exchanging boundary angular flux data. The primary advantages of spatial domain decomposition are its ease of implementation and its ability to handle complex processor boundaries naturally. There are, however, some significant drawbacks to this approach. First of all, the rate of convergence is dependent on the number

of spatial domains used. As the number of domains is increased, each domain becomes smaller and thus more sensitive to the boundary data being exchanged. This results in a degradation in the overall convergence behavior as the number of processors is increased. Thus, domain decomposition is generally only a viable option for small parallel systems (significant degradation is usually already noticeable at fewer than 100 processors) [94]. An additional difficulty with the parallel block Jacobi approach is the issue of load balancing. Because each processor is attempting to solve the solve a full transport problem on its portion of the problem, some processors may have a more difficult region of the problem and thus require more work to complete their tasks. The amount of work is not necessarily proportional to the number of computational cells that makes it very difficult to evenly split up the task. Imbalances in the computational load will lead to some processors sitting idle for some portion of the execution time, wasting valuable resources.

4.4.2 KBA

One parallel strategy that has been very successful for structured (i.e. Cartesian) mesh problems is the method of Koch, Baker and Alcouffe [14, 15]. The KBA algorithm creates a 2-D decomposition of the problem (in 3-D each domain consists of a column spanning the entire height of the problem) and actually parallelizes the sweeping process involved in inverting the (pychologically) lower triangular matrix L described in 2.4.2. To solve this system for a given direction, the work of each processor depends on each processor that is upwind of it. Thus the KBA algorithm starts with the domain that has no upwind neighbors (located in a corner) and performs a transport sweep across its domain. That processor then passes its boundary information to neighboring processors that can then perform a sweep on their domain. Each

processor does not start work until it has received all of its boundary data from upwind neighbors. Once a processor has completed its sweep for one angle, it can then begin the same sweeping process for another angle in the same octant. Because the sweep ordering will be the same for each angle in a given octant, each angle can be processed with exactly the same strategy. If the number of angles is very large then it is possible for a pipeline to build up so that all (or at least a large percentage) of the processors are able to be sweeping some angle at the same time. This means that the efficiency of KBA may be quite low for a low angular order but will increase as the angular order is increased. It is also possible to start the sweeping process for different octants at the same time by starting at each corner of the problem. In this case, collisions will occur when a processor is able to begin work on more than one direction at the same time. An analysis of different strategies for optimally handling such collisions can be found in [13] and an algorithm with a provable worst-case scenario in [65]. Because L is being directly inverted with the KBA algorithm, the outer iteration counts will not depend on the number of processors used for a given problem.

In practice, KBA has been used effectively for large-scale transport problems and is the basis for the Denovo transport code [42]. Parallel efficiencies exceeding 70% have been observed for up to 3600 processors, significantly better than can be achieved with domain decomposition strategies.

4.4.3 Unstructured Sweeps

For transport solvers using an unstructured mesh, the task of parallelizing the sweeping process is more difficult. In particular, the sweep ordering can be completely different for different angles, even those in the same octant. Additionally, when the boundaries between processor domains are not structured, it is possible to have cycles in the sweep ordering across spatial

domains and thus some processors may have to send and receive information multiple times to complete the sweep for a single angle. These factors tend to significantly degrade the performance of unstructured mesh algorithms, however some work has been done to achieve scalable algorithms [91, 93, 94]. These strategies typically revolve around a KBA-like decomposition, although spatial domains will contain 'ragged' edges due to the unstructured mesh. Geometric and/or graph partitioning heuristics are then used to prioritize the work order of each processor in an attempt to minimize the overall idle time of processors. Though unable to match the performance of KBA, these techniques have demonstrated reasonable scaling performance ($> 50\%$ efficiency) on upwards of 1000 processors.

Chapter 5

Multigrid-in-Energy

A vital part of any k-eigenvalue solution strategy is the ability to efficiently approximate the solutions to linear systems of the form

$$(I - DL^{-1}MS)\phi = \mathbf{q}. \tag{5.1}$$

This is obviously true for methods such as the Power Method or Arnoldi's method that are based upon a direct conversion of the problem to the standard eigenvalue formulation as in (2.32) but is also true for the Davidson method, in which case it may be possible to make use of a very rough approximation as a preconditioner for use in the subspace expansion of (3.24).

Virtually all current transport solvers employ some form of nested iteration procedure as described in Chapter 4. This complicated structure typically comes at a significant computational overhead, reducing the overall efficiency (and potentially also the accuracy) of the method. Thus it is our goal to develop a strategy for solving multigroup transport problems that minimizes the need for such nested iterations.

As a starting point for this new method we take the two-grid acceleration methods [3, 41] described previously, as they are essentially the only methods targeted at the multigroup equations for problems with significant upscat-

tering. Perhaps the most noticeable drawback to these methods are their dependence on a block Gauss-Seidel iteration that can endure a hefty computational cost on its own. However, if a less expensive iteration (such as Richardson iteration) is used then numerous error modes will persist and the two-grid acceleration will be ineffective because it is only capable of damping a single mode. Rather than using a single course energy grid, the approach we propose is to use a hierarchy of energy group structures so that a range of error modes can be addressed. To minimize the cost of this strategy, rather than fully solving a problem on a given energy grid we propose to merely relax, or smooth, the error. Thus the iteration takes the form of a multigrid method [116]. To further accelerate the convergence of this approach and to alleviate any issues with consistency, we propose that this approach be used as a preconditioner to a Krylov subspace method. While previous studies have applied multigrid methods to the spatial [26] and angular [80] variables in the transport equation, to our knowledge no published work has investigated the use of a multigrid approach for the energy component. We refer to this method as a multigrid-in-energy (MGE) preconditioner.

5.1 Basic Structure

A wide variety of different multigrid cycles have been proposed. The simplest of these is the standard V-cycle that starts at the finest grid level and progressively moves towards the coarsest grid through the use of restriction operators with smoothing iterations taking place at each level. Once the coarsest level is reached, the V-cycle returns to the original fine grid through the use of prolongation operators, again applying a smoother at each intermediate level. In a 'textbook' multigrid implementation, an exact solve would be performed on the coarsest level. More complicated schemes such as W-cycles and F-cycles involve visiting coarser levels more often than the

finer grids, performing extra work on the computationally inexpensive coarse grids. In our method, we elect to use only the basic V-cycle due to its simplicity and smaller computational cost per cycle.

For the energy treatment of the transport equation, each 'grid' corresponds to an energy group structure. If the finest grid (the original problem) has N_E energy groups, then the second grid will have approximately $\frac{N_E}{2}$ groups, the third approximately $\frac{N_E}{4}$ and so forth. When an odd number of groups are present at a given level, two options are available: one can choose three of the fine grid groups to combine into a single coarse group or alternatively choose one fine grid group to remain unchanged on the coarse grid. We (somewhat arbitrarily) elect to use the latter approach in our method. The coarsest grid will generally have only a single energy group, but because this monoenergetic equation still has the full spatial resolution of the original problem a direct solve is not feasible here. We elect instead to simply perform the same smoothing operation that is performed on the intermediate grids. Such a strategy is feasible because MGE is being designed as a preconditioner and not an iterative solver. A multigrid iterative solver would require that an exact (or at least high accuracy) linear solve be performed on the coarsest level. In order to fully define the method, we must specify the two components that largely define any multigrid method: how to transfer errors between grids and how to damp (smooth) errors on a particular grid.

5.2 Grid Transfer

In the previously mentioned two-grid methods, transfer between the multigroup and monoenergetic problems is achieved through a material-dependent energy spectrum that for most problems provides a good approximation to the shape of the dominant block Gauss-Seidel error mode. At first glance it seems that a similar approach might be possible for MGE, to use a function

that in some respect represents the 'worst' error mode. However, we would like to stress that such an approach is not appropriate in this context since any approach that targets a single mode is unlikely to be effective on other modes. In order to operate effectively on all error modes, the appropriate transfer operations are simply an even weighting procedure. Thus the problem on a coarse grid is defined in terms of the collapsed cross sections given by

$$\sigma_g^C = \frac{1}{2}(\sigma_{2g-1}^F + \sigma_{2g}^F) \tag{5.2}$$

and

$$\sigma_{s,g'\to g}^C = \frac{1}{2}(\sigma_{s,(2g-1)\to(2g-1)}^F + \sigma_{s,(2g-1)\to(2g)}^F + \sigma_{s,(2g)\to(2g-1)}^F + \sigma_{s,(2g)\to(2g)}^F) \tag{5.3}$$

where the superscripts C and F indicate the coarse and fine grid levels, respectively, and the notation of (2.15) and (2.16) has been used. It should be noted that while standard multigrid methods generally use higher order grid transfer operators [116], this is not necessary in the MGE approach because no derivatives with respect to energy appear in the transport equation.

5.3 Smoothing Iterations

The smoothing iterations in a multigrid method frequently consist of a few iterations of an inexpensive fixed point iteration [127] such as a Jacobi or Gauss-Seidel iteration. Because the transport matrices are dense and individual matrix entries are not accessible, such approaches are not practical for use in the MGE method. However, blocked versions of these methods are a possibility. If we split the scattering matrix into its strictly lower triangular, diagonal, and strictly upper components (S_L, S_D, and S_U, respectively) then the block Jacobi iteration can be written as

$$(I - TS_D)\phi^{k+1} = T(S_L + S_U)\phi^k + \mathbf{b} \tag{5.4}$$

and block Gauss-Seidel as

$$[I - T(S_D + S_L)]\phi^{k+1} = TS_U\phi^k + \mathbf{b}. \tag{5.5}$$

Due to the fact that the diagonal portion of the scattering matrix appears on the left hand side of these equations, each iteration of one of these methods requires solving a monoenergetic transport problem for each energy group. This is likely to introduce a significant computational burden. Alternatively, it is possible to simply perform a Richardson iteration on the full multigroup transport problem, which can be written as

$$\phi^{k+1} = TS\phi^k + \mathbf{b}. \tag{5.6}$$

In this strategy, no within-group problems need to be solved and thus each iteration will require only a very small computational cost, although it should be expected to be somewhat less effective. It is also possible to slightly modify the Richardson iteration to include the strictly lower triangular portion of the scattering on the left hand side while keeping all other terms on the right hand side, giving

$$(I - TS_L)\phi^{k+1} = T(S_D + S_U)\phi^k + \mathbf{b}. \tag{5.7}$$

This has the same computational cost as the Richardson iteration while still allowing some portion of the scattering to be treated implicitly. We refer to this approach as a modified Gauss-Seidel iteration and adopt this strategy for use with MGE. With any of the above procedures, it is possible to add a damping operation of the form

$$\tilde{\phi}^{k+1} = \omega\phi^{k+1} + (1 - \omega)\phi^k, \tag{5.8}$$

where $\tilde{\phi}^{k+1}$ is a refinement of ϕ^{k+1} and ω is a relaxation parameter typically taken in the interval $(0, 2)$. Such relaxations are common in multigrid methods [127] and potentially offer a significant improvement if the parameter can be chosen appropriately.

5.4 Coarse Angle Approximation

It frequently occurs in the convergence of iterative methods for the transport equation that the dominant error modes are only weakly dependent on angle (this can happen even if the solution vector itself is highly anisotropic). Thus it may be possible to represent the error using a much smaller number of angles than is required to represent the solution itself. We propose to substitute a low angular order problem (say S_2) for the original S_N problem when performing the multigrid cycle. Although this may result in some loss of effectiveness with respect to the error reduction per multigrid cycle, we feel that this will be more than offset by the significant improvement in the computational cost per iteration. Because the cost of performing a matrix-vector product (and also of performing any of the proposed smoothing iterations) is directly proportional to the number of angles in the problem, significant savings may be possible. If the S_2 equations are used for the low order problem then the reduction in computational effort per iteration is given by $\frac{8}{N(N+2)}$ in either 2-D or 3-D, so the savings becomes even more noticeable as the angular order is increased. We note that the use of a coarse angular problem to approximate a high angular order is certainly not new. In fact almost all of the techniques described in Chapter 4 are based upon this approach, including the DSA and TSA strategies. The transport two-grid method introduced in [41] also mentions the possibility of using an S_2 or S_4 problem during the acceleration step.

One potential issue related to the use of a coarse angular problem in the multigrid cycle arises in problems with reflecting boundaries. As discussed in 2.4.2, it is often advantageous to add reflecting boundary angular fluxes to the solution vector to avoid having to perform an additional iteration in order to multiply by the transport matrix. If this is done, however, then the residual upon which the multigrid cycle operates necessarily contains angular

information on the problem boundary. In order to apply a multigrid cycle with a coarse angular discretization, it is necessary to first convert the high order angular fluxes to the low angular order and at the end of the cycle the reverse must occur. We propose that this angular transfer is performed using spherical harmonic moments as an intermediate. We feel this is a natural selection because the coefficients are already available due to their use in the computation of the scattering term. For the restriction from S_N to S_2, this process appears as

$$\psi_{S_2} = \mathrm{M_2 D_N} \psi_{S_N} \tag{5.9}$$

and the prolongation from S_2 to S_N is given by

$$\psi_{S_N} = \mathrm{M_N D_2} \psi_{S_2} \tag{5.10}$$

where D_N and M_N are the angular restriction and prolongation operators for the S_N quadrature set and D_2 and M_2 for the S_2 set. In problems where the boundary conditions tend to dominate the convergence behavior (typically problems modeling small regions with reflecting boundary conditions on all sides), this approach may not be sufficient to ensure rapid convergence. In such circumstances we recommend that a small number of inexpensive iterations (e.g. Richardson or modified Gauss-Seidel) be performed using the full angular order problem after the conclusion of the multigrid cycle. Because this addition may significantly increase the cost of the overall iteration, it is recommended that this only be done for problems where it is expected to really be necessary. Alternatively, an increased angular order (say S_4 or S_6) could be used in the multigrid cycle.

5.5 Parameter Selection

For the MGE approach to truly be a viable method, we must describe how the various parameters should be selected. First the relaxation parameter,

ω, for the smoothing iteration must be determined. We first consider the behavior of the smoother itself and then look at its performance within a multigrid framework. For the case of a damped Richardson iteration applied to an arbitrary matrix A, the iteration matrix is given by $(I - \omega A)$ [48]. If λ_i are the eigenvalues of the matrix A, then the eigenvalues of the iteration matrix are given by $(1 - \omega\lambda_i)$. The optimal convergence rate will be attained by selecting ω so that the spectral radius of the iteration matrix is minimized, i.e.

$$\omega_{\text{opt}} = \arg\min_{\omega} \left(\max_i |1 - \omega\lambda_i| \right). \tag{5.11}$$

If all of the eigenvalues λ_i are real and positive then the spectral radius is provided by $\max(|1 - \omega\lambda_{\min}|, |1 - \omega\lambda_{\max}|)$. Convergence will occur when both $|1 - \omega\lambda_{\min}|$ and $|1 - \omega\lambda_{\max}|$ are less than unity and the minimum will occur when $|1 - \omega\lambda_{\min}| = |1 - \omega\lambda_{\max}|$, which leads to

$$\omega_{\text{opt}} = \frac{2}{\lambda_{\min} + \lambda_{\max}}. \tag{5.12}$$

In general all eigenvalues of the transport equation will not be real, but the same result will hold provided that the complex components are relatively small. From Figures 2.2 and 2.3 we see that $\lambda_{\min} \approx 0$ and $\lambda_{\max} \approx 1$ when boundary angular fluxes are not included in the solution vector and $\lambda_{\max} \approx 1.3$ when they are included. These values indicate that the optimal relaxation parameter should be $\omega \approx 2$ without the inclusion of boundary fluxes and $\omega \approx 1.5$ with their inclusion. In both cases the predicted optimal value is quite close to the cutoff where the iteration fails to be convergent, so it seems reasonable to make a selection slightly smaller to guarantee convergence.

In the simplified scenario of an infinite medium problem, we can directly look at the effect that MGE has on the spectrum of the transport operator. In this case, the transport equation is independent of both space and angle, reducing the size of the problem to only the number of energy unknowns. The same infinite medium strategy was used in [3] to look at the convergence of

the two-grid method. Rather than computing the entire multigrid operator, we simply apply a two-level operator where an exact solve is used on the coarse level. Analyzing a two grid method rather than the multigrid operator is consistent with observations in [127] concerning the determination of multigrid convergence factors. We use a V-cycle with three pre-smoothing and three post-smoothing Richardson iterations (combined with a relaxation parameter) to analyze the performance. The problem consists of water (with a natural isotopic distribution) with 238 group cross sections provided by the SCALE [2] package. Figure 5.1 shows the spectral radius, ρ, of the iteration matrix (i.e. $I - M^{-1}A$, where M^{-1} is the two grid operator) as a function of the relaxation parameter ω. The curve has a distinct convex shape with a minimum at around $\omega = 1.3$. This is an interesting result because it differs dramatically from the optimum value of the Richardson iteration alone. The spectral radius at this minimum is less than 0.1, indicating that the multigrid method does an excellent job of smoothing all error modes (also in contrast to the standalone Richardson iteration which would have a spectral radius quite close to 1 even with an optimal selection of the relaxation parameter). Although minimizing the spectral radius of the iteration matrix does not necessarily guarantee optimal performance when using MGE as a preconditioner, it is expected that such a choice will still lead to very good performance because the small value of the spectral radius indicates a strong clustering of the entire spectrum.

Figure 5.2 displays the full spectrum of the original coefficient matrix for this infinite medium problem as well as the impact of preconditioning with either the lower triangular portion of the matrix or with MGE. The MGE plot was computed using the value $\omega = 1.3$ as suggested by the previous discussion. The original matrix has eigenvalues distributed along the real line between 0 and 1 with no strong clustering evident. Preconditioning with the lower triangular portion of the matrix tends to cluster the spectrum around

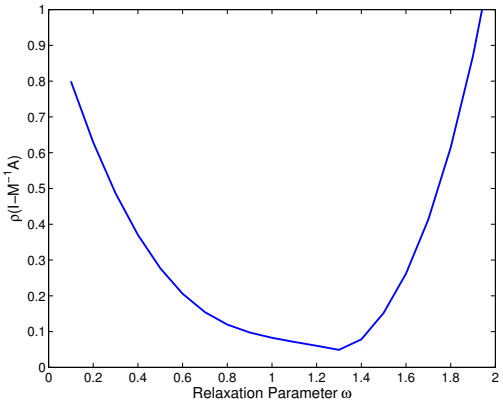
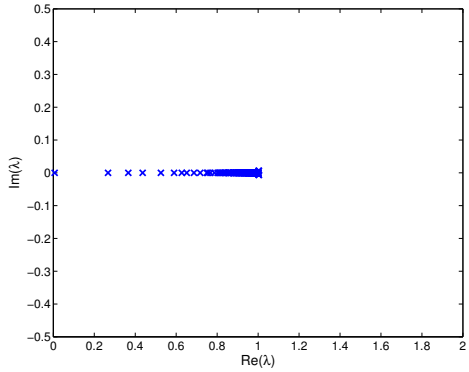


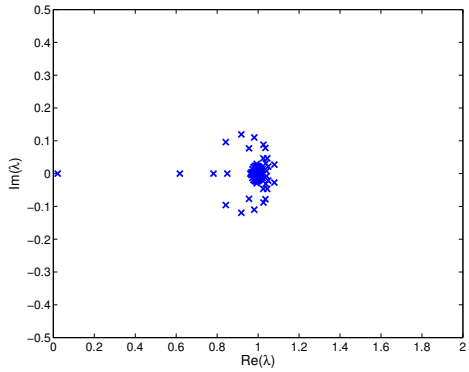
Figure 5.1: Dependence of MGE error reduction on parameter ω.

1 although the clustering is not very tight and several eigenvalues remain outside of the cluster, including one very close to the origin. This eigenvalue near the origin indicates that a preconditioned Richardson iteration with this preconditioner (i.e. a Gauss-Seidel iteration) would be very slow to converge. It is precisely this error mode that is targeted by the two-grid approaches discussed in the previous chapter. The spectrum of the matrix preconditioned by MGE with optimal relaxation parameter is quite remarkable; all of the eigenvalues are clustered very closely around 1. This spectrum indicates that a Krylov method utilizing this preconditioner would converge very rapidly. Although the clustering is unlikely to be so dramatic outside of an infinite medium calculation, these results nonetheless indicate the potential of the MGE approach.

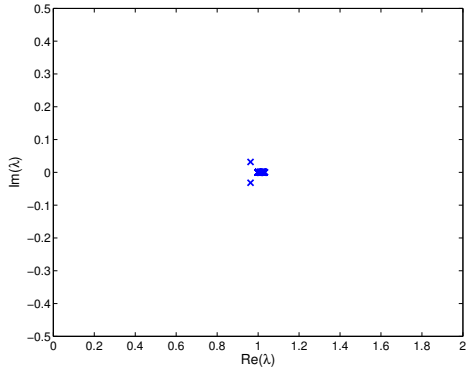
Figure 5.3 shows the iteration counts required for convergence of a Davidson eigensolver with MGE as the preconditioner for a variety of different problems. The C5G7 and HTTR problems will be described in Chapter 6 and the PWR Assembly and Fuel Pin problems are just smaller portions of



(a) Coefficient matrix A



(b) Block lower triangular preconditioner



(c) MGE preconditioner, $\omega = 1.3$

Figure 5.2: Preconditioned infinite medium spectra for different preconditioners.

the C5G7 problem modeled with reflecting boundary conditions on all sides. The Slab problem is a 5 cm thick region of boron (a strong neutron absorber) surrounded by 1 cm thick regions of water and highly enriched uranium. It is 7 cm in height and contains vacuum boundaries on all sides. The iteration counts shown are those required to reduce the eigenvalue residual to 10^{-6}. Despite the varying nature of these problems (and the varying boundary condition treatments), they display very similar trends. With the exception of the Fuel Pin problem, all of the curves display a minimum at $\omega \approx 1.3$, completely consistent with the infinite medium prediction. The shape of these convergence curves appears to be quite similar to the infinite medium behavior of Figure 5.1, although the convergence actually appears to be less sensitive to the selection of the relaxation parameter. For the C5G7, PWR Assembly, and HTTR problems, the rapid deterioration at around $\omega = 1.5$ results from the smoothing iteration becoming divergent, exactly as would be expected according to the Richardson convergence analysis (angular fluxes on the problem boundary are included in the solution vector for these problems). Because the slab problem has vacuum boundaries on all sides, no such deterioration is seen and the convergence behavior remains very favorable across a wide range of ω values. The dramatically different behavior for the Fuel Pin problem is that due to the very small physical size of the domain (the problem is only 1.26 cm across), the effects of the boundary conditions become far more significant. It is very likely that the largest eigenvalue of the multigroup transport matrix for this problem is much larger than the value of 1.3 from Figure 2.3, leading to a smaller convergence region for the Richardson iteration. Based on these results combined with the infinite medium analysis, we propose that a value of $\omega = 1.3$ be used as the default value with the understanding that a different selection will be necessary for very small spatial domains. It should be noted that the problems of the most interest in the transport community are generally at least the size of a fuel

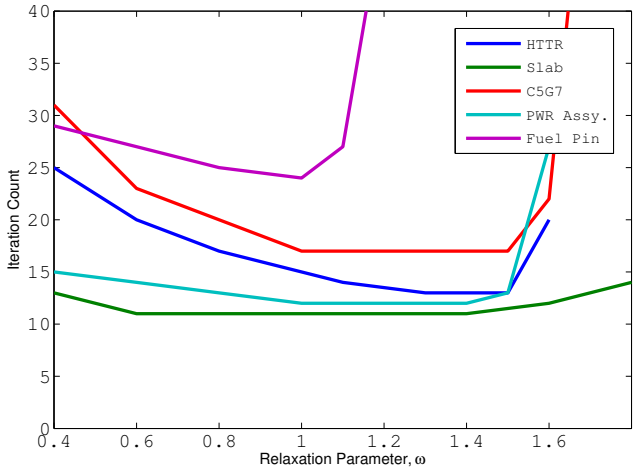


Figure 5.3: Davidson/MGE iteration counts with varying relaxation parameter ω.

assembly and potentially much larger, so that the boundary condition issues in the Fuel Pin problem are unlikely to be of much importance.

The next task is to determine the best selections for the number of pre- and post-smoothing steps to apply during the multigrid cycle. Tables 5.1 and 5.2 show the computational times (and iteration counts) to achieve a residual norm of 10^{-6} for various parameter selections for the HTTR and C5G7 problems, respectively. Because the computational time per iteration varies with the number of smoothing steps, it is the timing information that should be considered (although for an equivalent computational time, a smaller iteration count is generally preferable since it offers a smaller memory requirement). As a general trend, it is evident that post-smoothing steps are more effective at reducing iteration counts than pre-smoothing steps and, in fact, performing more than 1 or 2 pre-smoothing steps generally offers no additional reduction in iterations. As long as at least 2-3 post-smoothing it-

Table 5.1: Davidson/MGE timings in seconds (iteration counts) with varying pre- and post-smoothing steps, HTTR problem.

	ν_1				
ν_2	0	1	2	3	4
1	85.6(29)	85.3(28)	85.6(27)	85.8(26)	88.2(26)
2	59.8(19)	61.9(19)	63.9(19)	62.6(18)	64.5(18)
3	50.7(15)	52.3(15)	54.0(15)	52.2(14)	53.7(14)
4	47.2(13)	48.6(13)	50.0(13)	51.3(13)	48.7(12)
5	46.3(12)	47.6(12)	48.9(12)	50.2(12)	51.5(12)
6	45.1(11)	46.3(11)	47.5(11)	48.7(11)	49.9(11)

erations are performed, the required time is actually quite insensitive to the particular cycle selection with most combinations resulting in no more than a $10-15\%$ increase relative to the optimal combination. We propose using 1 pre-smoothing and 4 post-smoothing iterations, as this combination appears to result in very close to optimal performance for the cases considered. Due to the insensitivity of the performance with respect to these values, no further attempt will be made to optimize these values on a problem-by-problem basis.

Tables 5.4 and 5.3 show the effect of using a coarse angular approximation (here an S_2 angular quadrature) for the smoothing iterations rather than using the full S_N equations for the Davidson method applied to the two problems described in Chapter 6. As would be expected, the coarse angle smoothing is slightly less effective in terms of the outer iteration counts, although the coarse angle approximation still results in rapid convergence. However, due to the significantly lower computational cost of the preconditioner it actually requires much less time to reach convergence for the coarse

Table 5.2: Davidson/MGE timings in seconds (iteration counts) with varying pre- and post-smoothing steps, C5G7 problem.

	ν_1				
ν_2	0	1	2	3	4
1	292.1(34)	321.0(36)	372.1(40)	354.0(37)	453.9(45)
2	211.0(23)	199.1(21)	196.2(20)	213.0(21)	219.5(21)
3	177.2(18)	172.6(17)	178.2(17)	184.0(17)	189.4(17)
4	178.8(17)	173.6(16)	167.6(15)	172.6(15)	177.6(15)
5	177.6(16)	182.7(16)	176.1(15)	181.0(15)	173.6(14)
6	176.8(15)	181.7(15)	186.6(15)	178.8(14)	183.4(14)

angle approach. The speed up becomes more significant as the angular order of the problem is increased, resulting in a reduction in computational cost of nearly a factor of 4 for an S_{16} quadrature.

It is actually rather fortunate that the convergence behavior of the MGE method is so insensitive to problem parameters. It makes it much easier to achieve fast and reliable performance for a wide range of problems and minimizes the amount of knowledge that the code user must have concerning the internal workings of the solver, potentially broadening the prospective user base for such a method.

Table 5.3: Davidson/MGE performance with and without coarse angle smoothing, HTTR problem.

		S_N Smoothing		S_2 Smoothing	
S_N Order	Angles	Iters	Time	Iters	Time
4	12	9	129.4	14	84.5
8	40	9	421.6	14	156.9
12	84	9	904.2	13	262.4
16	144	9	1682.8	13	440.8

Table 5.4: Davidson/MGE performance with and without coarse angle smoothing, C5G7 problem.

		S_N Smoothing		S_2 Smoothing	
S_N Order	Angles	Iters	Time	Iters	Time
4	12	13	248.1	16	127.4
8	40	13	805.1	16	235.7
12	84	13	1700.3	16	408.6
16	144	13	2886.9	16	640.8

Chapter 6

Numerical Results

In order to study the behavior of different eigensolvers for the k-eigenvalue problem, we now consider their performance in a realistic environment. The radiation transport solver NEWT [38] has been selected as the test bed for these numerical experiments. NEWT is a 2-D solver readily available within the SCALE [2] package developed at Oak Ridge National Laboratory. It is based on the extended step characteristics method, a member of the class of slice balance methods that were mentioned in 2.2.3. The version of NEWT that has been used is that distributed with SCALE version 6.0, the latest available version at the time of this work.

A variety of eigensolvers have been implemented to identify their relative strengths and weaknesses. In the distributed copy of the code, the only available eigensolver is the power method. Coarse mesh finite difference (CMFD) acceleration of the power method iterates is available. These methods have been left unchanged and are executed with default parameters except as otherwise specified. Three additional solvers have been implemented in NEWT, namely the Arnoldi method, Rayleigh quotient iteration, and a Davidson solver. The Arnoldi and Davidson solvers require solution of a linear system at each iteration that is performed using a full GMRES solver with the

MGE method as a preconditioner. The Davidson solver directly uses the MGE preconditioner for expansion of subspaces. All numerical experiments were performed on a laptop computer with a 2.6 GHz Core 2 Duo processor (although only single-threaded execution was performed) with 4 GB of RAM.

6.1 Test Problems

In order to test the effectiveness of the various eigensolvers, two sample problems are proposed. The first test problem consists of a single block from the high temperature test reactor (HTTR), a Japanese research reactor. The HTTR block has a hexagonal outer geometry containing 37 fuel pins placed in a hexagonal array, as shown in Figure 6.1. The fuel pins consist of 6.3% enriched uranium oxide rods surrounded by a graphite sleeve, helium coolant, and a matrix of graphite. Three burnable poison rods are placed at three of the vertices of the array. The use of white boundary conditions on all sides simulates an infinite array of such assemblies. The base configuration for the numerical model was selected to have a 108×96 spatial grid, 16 group cross sections, an S_8 angular quadrature and P_1 scattering.

The dominant eigenvalue for this base case is 1.019063 with a dominance ratio of approximately 0.94. The dominant eigenvector for several energy groups is illustrated in Figure 6.1. Because the CMFD implementation available in NEWT with SCALE 6.0 allows only for rectangular problem boundaries, it was not possible to apply the CMFD acceleration technique directly to the standard HTTR geometry (it is worth noting that the version of NEWT to be distributed with SCALE 6.1 does allow hexagonal external boundaries [62]). Instead, an equivalent geometry was created that contains the same domain rearranged to form a rectangular outer boundary, as shown in Figure 6.1. This formulation exactly preserves the volume of the original geometry; mesh parameters were selected to produce a problem size that

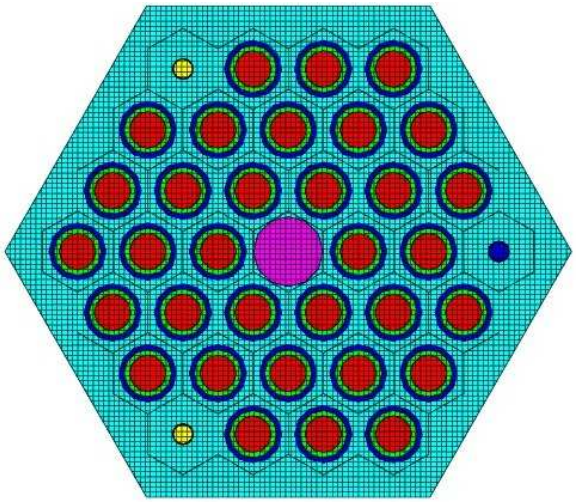


Figure 6.1: HTTR base configuration.

matched the original problem as closely as possible.

The second test problem is based on the well-known C5G7 MOX 2-D benchmark problem [86], though with select modifications. The geometry, shown in Figure 6.1, consists of a 2×2 array of fuel assemblies, each of which contains a 17×17 array of fuel pins. A 21.42 cm water reflector surrounds the fuel assemblies on the bottom and left sides. The fuel pins each have a diameter of 0.54 cm and a center-to-center spacing of 1.26 cm. The fuel pins in the bottom left and upper right assemblies contain a standard 4.0% enriched uranium oxide fuel (shown in green); the upper left and lower right assemblies have a mixed oxide fuel (shown in blue). The moderator is natural water. No cladding is explicitly modeled, rather a volume weighting of stainless steel is included in the fuel mixture, consistent with the benchmark specification. The geometry is modeled with vacuum boundaries on the bottom and left sides and reflecting boundaries on the top and right, providing the equivalent of a 4×4 arrangement of fuel assemblies with a water reflector on all sides.

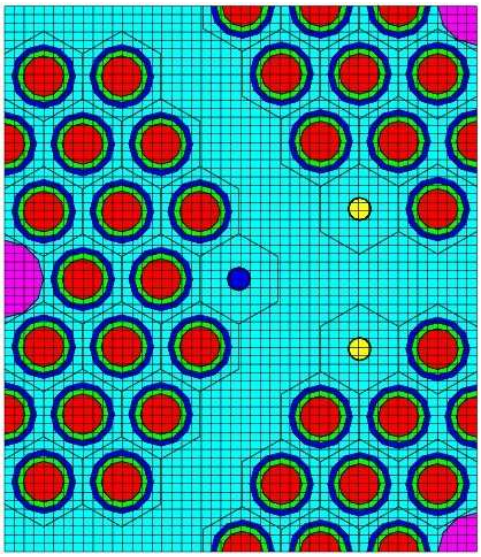
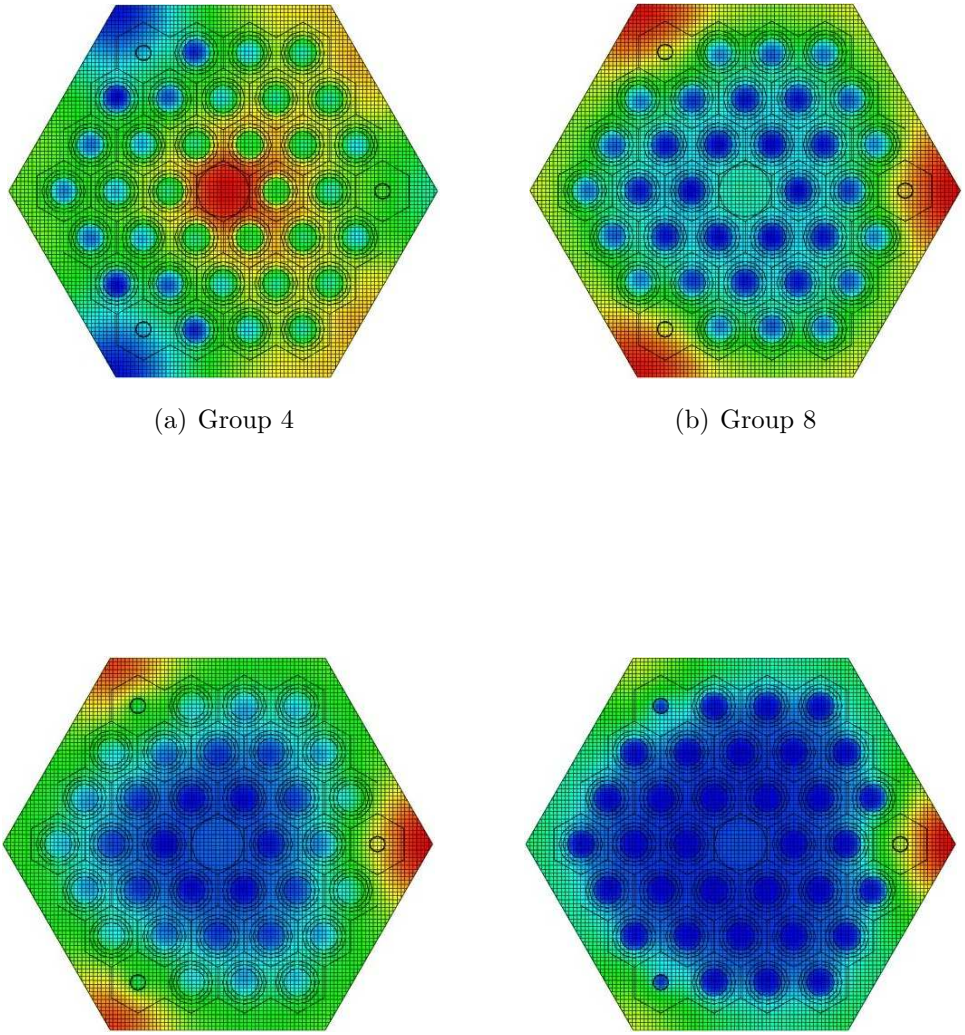


Figure 6.2: HTTR CMFD equivalent configuration.

Although the original specification provided 7-group cross sections for all materials, it was decided that using cross sections generated through the SCALE package would more closely replicate standard reactor physics calculations and also allow for the energy group structure to be varied. For the base configuration, each fuel pin cell is split into computational cells using a 2×2 Cartesian grid, creating 8 cells per pin and a 102×102 Cartesian mesh ($\Delta_x = \Delta_y = 0.62$ cm) is used to determine the computational cells in the outer moderator region for a total of 15028 spatial cells in the geometry. A 16 group cross section set is used, along with an S_8 level symmetric angular quadrature (containing 40 angles) and P_1 scattering in all materials. Studies on the effects of varying each of these parameters can be found in Section 6.2.

The dominant eigenvalue for this configuration is 1.140080 with a dominance ratio of approximately 0.97. The spatial distribution of the eigenvector corresponding to the dominant eigenvalue is displayed in Figure 6.1 for



(a) Group 4

(b) Group 8

(c) Group 12

(d) Group 16

Figure 6.3: Dominant eigenvector in selected energy groups for the HTTR problem.

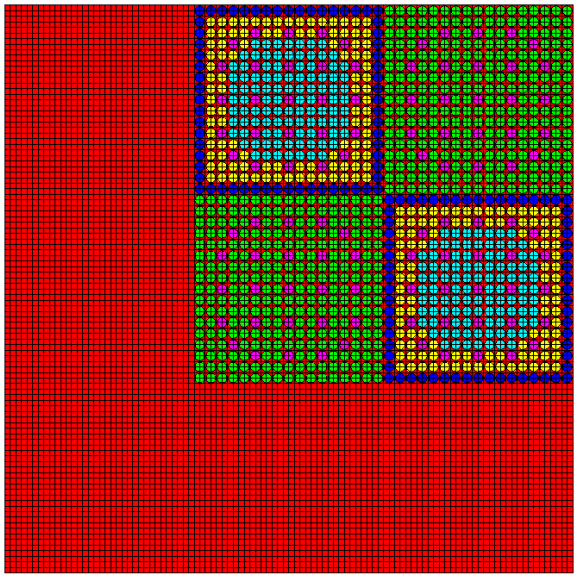
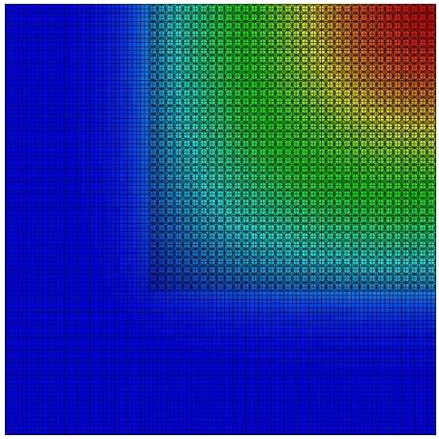


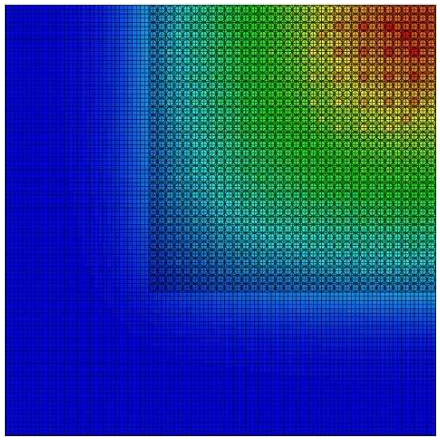
Figure 6.4: C5G7 base configuration.

several energy groups.

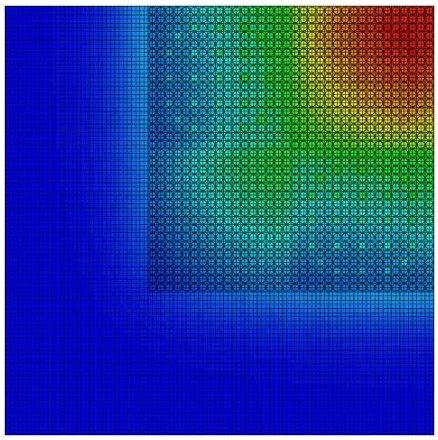
The performance of several eigensolvers for the two test problems is shown in Table 6.1. Particularly in the C5G7 problem, the difficulties associated with the power method are clearly evident. CMFD provides a very large improvement for both problems, although we note that the coarse mesh parameter has been selected to provide optimal results and such a value would generally not be known a priori. The Arnoldi method offers a fairly significant improvement relative to the power method, although it is unable to compete with the rapid convergence of CMFD. Rayleigh Quotient Iteration displays remarkable performance with respect to the outer iteration counts, converging in only 2 iterations for the HTTR problem, due to its quadratic convergence. The large computational cost per iteration reduces the overall performance although it is still fairly competitive with CMFD. The top performer for both problems is the Davidson method with MGE



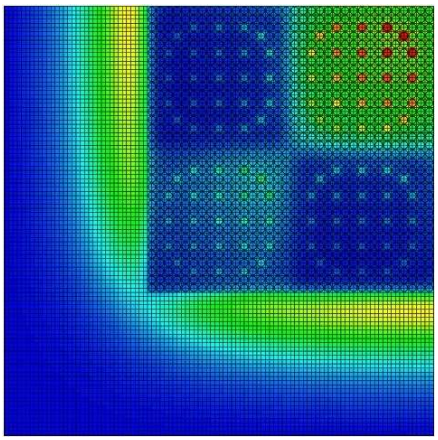
(a) Group 4



(b) Group 8



(c) Group 12



(d) Group 16

Figure 6.5: Dominant eigenvector in selected energy groups for the C5G7 problem.

Table 6.1: Eigensolver convergence and timings (seconds) for base cases.

	HTTR		C5G7	
Solver	Iters	Time (s)	Iters	Time (s)
Power	103	1752	320	4494
CMFD	14	184	11	248
Arnoldi	5	640	10	1375
RQI	2	244	4	675
Dav/MGE	14	157	16	236

preconditioning, edging out CMFD in both cases. This achievement is particularly impressive considering that the Davidson approach does not rely on any problem-dependent parameters to achieve its peak performance. Such favorable performance suggests that this method deserves much more consideration as a candidate for large-scale k-eigenvalue problems.

6.2 Parametric Studies

In order to determine the viability of each of the eigensolvers for use on large scale problems, we now investigate the effect of refining the problem with respect to each of the fundamental problem parameters: space, angle, energy, and scattering order. Ideally, the number of iterations required for convergence should be insensitive to changes in these parameters. A robust method will display a linear variation in computational time with respect to changes in the number of space, angle, and energy unknowns and exhibit only slow growth with respect to increasing the scattering order. Results for each of the methods will be presented with the exception of the power method. This is due to the exceedingly long run times for the power method

Table 6.2: Problem size for spatial refinement.

	HTTR		C5G7	
Mesh	Cells	DOF	Cells	DOF
1	3909	187632	10693	622064
2	11169	536112	15028	851904
3	22217	1066416	32657	1763376
4	37827	1815696	50864	2702592

and the fact that it is unlikely to be competitive for any problem.

6.2.1 Spatial Refinement

We begin by considering the effect of refining the spatial resolution of the problem. For each problem we consider four different levels of mesh refinement to illustrate the convergence behavior. With the HTTR problem, grids range from 54×48 for the coarsest level to 216×192 for the finest. For the C5G7 problem, the base Cartesian mesh consists of a 51×51 grid for the coarsest mesh up to a 204×204 grid at the finest level. The number of spatial cells and total degrees of freedom for each problem are shown in Table 6.2 and the behavior of the computed eigenvalue in Table 6.3.

The iteration counts and timings for each method are given in Tables 6.5 and 6.4. For both problems, all of the methods display a complete insensitivity in the number of iterations required for convergence with respect to spatial refinement. The timings follow the results of the base configuration cases, with CMFD and the Davidson method showing the smallest times to solution and the Arnoldi method displaying the poorest performance in all cases. The scaling for all of the methods is approximately linear with the

Table 6.3: Eigenvalue convergence for spatial refinement.

Mesh	HTTR	C5G7
1	1.019473	1.139948
2	1.019063	1.140080
3	1.018913	1.141882
4	1.018797	1.142127

number of spatial unknowns and the scaling between the two finest levels is nearly exactly so. It should again be noted that CMFD is used here with the optimal selection of the coarse mesh parameter that would not be known beforehand (except through the experience and intuition of the user). To reveal a bit more about the true nature of CMFD, Tables 6.7 and 6.6 show the performance at each spatial level for a number of different coarse mesh parameters, (CMFD(k) indicates that a coarse mesh cell is k times as large as the base mesh in each coordinate direction). These tables display the difficulty with using CMFD in practice: poor selection of the mesh parameter leads to poor convergence or even divergence. For the HTTR problem, on the finest grids we see that selecting the coarse mesh to be too fine leads to divergence. For the C5G7 problem the reverse is true in that selecting the coarse mesh too coarse leads to divergence. Also selecting the coarse mesh parameter too fine results in an increased computational time due to the amount of time spent solving the diffusion problem at each iteration. Such difficulties are the reason that many production radiation transport codes do not offer CMFD (or other nonlinear acceleration techniques that generally suffer from similar difficulties), and those that do generally do not provide it as the default solver.

Table 6.4: Performance for spatial refinement, HTTR.

	CMFD		Arnoldi		RQI		Dav/MGE	
Grid	Iters	Time	Iters	Time	Iters	Time	Iters	Time
54×48	14	61	5	206	2	79	13	49
108×96	14	184	5	640	2	244	14	157
162×144	14	419	5	1376	2	523	14	337
216×192	14	764	5	2432	2	910	14	583

Table 6.5: Performance for spatial refinement, C5G7.

	CMFD		Arnoldi		RQI		Dav/MGE	
Grid	Iters	Time	Iters	Time	Iters	Time	Iters	Time
1	14	227	10	966	4	480	16	171
2	11	248	10	1375	4	675	16	236
3	11	568	10	3314	4	1566	16	553
4	11	900	10	5058	4	2398	16	843

Table 6.6: CMFD performance for spatial refinement, HTTR.

	CMFD1		CMFD2		CMFD3		CMFD4	
Grid	Iters	Time	Iters	Time	Iters	Time	Iters	Time
54×48	14	69	14	62	14	61	14	61
108×96	14	537	14	195	14	186	14	184
162×144	–	–	14	481	14	419	14	433
216×192	–	–	14	1068	14	791	15	764

Table 6.7: CMFD performance for spatial refinement, C5G7.

	CMFD(1)		CMFD(2)		CMFD(3)		CMFD(4)	
Grid	Iters	Time	Iters	Time	Iters	Time	Iters	Time
1	14	227	–	–	–	–	–	–
2	11	472	11	248	–	–	–	–
3	11	1416	11	656	11	568	11	576
4	11	2928	11	1122	11	955	11	900

6.2.2 Angular Refinement

Next we look at the effect of altering the order of the angular quadrature. The impact of such refinement on the dominant eigenvalue is displayed in Table 6.8 and the impact on the performance in Tables 6.9 and 6.10. As with spatial refinement, the iteration counts are not affected by changes in the quadrature order. It is interesting to observe in these results, however, that the time required to converge scales approximately linearly with the number of angles for CMFD but actually scales slower than linear for the other methods. This behavior occurs due to the coarse angular smoothing in the MGE preconditioner that is used in these solvers, so the cost of the preconditioner is independent of the angular order. Because part of the computational cost (the matrix vector products) is scaling linearly with the number of angles and part of the cost is remaining constant, the net result is something less than linear. Of course if the number of angles became large enough then the cost of the preconditioner would become very small compared to the matrix vector products and the behavior would appear linear from that point forward. This favorable scaling with respect to angular refinement is a very valuable feature for a transport solver, as most problems of interest in the

Table 6.8: Eigenvalue convergence for angular refinement.

S_N Order	Angles	HTTR	C5G7
4	12	1.019515	1.139674
8	40	1.019063	1.140080
12	84	1.019258	1.140111
16	144	1.019310	1.140119

Table 6.9: Performance for angle refinement, HTTR.

	CMFD		Arnoldi		RQI		Dav/MGE	
S_N Order	Iters	Time	Iters	Time	Iters	Time	Iters	Time
4	15	60	5	418	2	168	14	85
8	14	184	5	640	2	244	14	157
12	14	384	5	1018	2	694	13	262
16	14	661	5	1602	3	1121	13	441

field require very high angular orders to appropriately resolve the physics.

6.2.3 Energy Refinement

The effect of varying the number of energy groups in the problems is shown in Table 6.11 and the resulting performance for the eigensolvers in Tables 6.12 and 6.13. As with space and angle refinement, the convergence behavior with respect to increasing the number of energy groups is quite robust for all of the methods. Interestingly, there is a slight reduction in the outer iteration count in the Davidson method for the C5G7 problem and a slight increase for the HTTR problem, although the differences in both cases are

Table 6.10: Performance for angle refinement, C5G7.

	CMFD		Arnoldi		RQI		Dav/MGE	
S_N Order	Iters	Time	Iters	Time	Iters	Time	Iters	Time
4	11	967	10	897	4	427	16	127
8	11	248	10	1375	4	675	16	236
12	11	495	10	2142	4	1068	16	409
16	11	818	10	3215	4	1620	16	641

Table 6.11: Convergence for energy refinement.

Groups	HTTR	C5G7
16	1.019063	1.140080
32	1.017169	1.138705
64	1.017979	1.139998
128	1.019108	1.140836

fairly small. The asterisks on the finest energy group structures indicate that the maximum amount of memory was reached and the program had to resort to virtual memory (Linux swap space) and thus the computational times are possibly slightly higher than might otherwise be observed. Except for this memory issue, the Davidson method continues to outperform all of the other methods, including CMFD.

6.2.4 Scattering Order Refinement

Given all of the previous observations, it should come as no surprise that scattering order refinement has little to no impact on the convergence be-

Table 6.12: Performance for energy refinement, HTTR.

	CMFD		Arnoldi		RQI		Dav/MGE	
Groups	Iters	Time	Iters	Time	Iters	Time	Iters	Time
16	14	184	5	640	2	244	14	157
32	15	381	5	1331	2	528	14	328
64	15	769	5	2939	2	1016	13	650
128	15	1632	5	7476*	3	5190*	15	1771*

Table 6.13: Performance for energy refinement, C5G7.

	CMFD		Arnoldi		RQI		Dav/MGE	
Groups	Iters	Time	Iters	Time	Iters	Time	Iters	Time
16	11	248	10	1375	4	675	16	236
32	11	497	10	2758	4	1422	16	488
64	11	999	10	5527	4	2754	15	967
128	13	2166	10	10828	4	5900	14	1737

Table 6.14: Convergence for P_N order refinement.

P_N Order	HTTR	C5G7
0	1.020627	1.179741
1	1.019063	1.140080
2	1.019428	1.140691
3	1.019322	1.140613

havior of the eigensolvers, as we see in Tables 6.15 and 6.16. However, this study displays a slightly different behavior than the others in that CMFD seems to exhibit better scaling in terms of the timings than the other methods and appears to be slightly more efficient than the Davidson method at higher scattering orders for the C5G7 problem. The reason for this behavior is that CMFD performs very few operations on the spherical harmonic moment vectors and thus displays only a very small growth in run time with increasing scattering order. The other methods, however, perform a larger portion of their work on vectors containing the spherical harmonic moments (since operations such as orthogonalization and projections involve vectors containing these moments) and thus the timings show more of an impact.

6.2.5 Subspace Dimensions

As is evidenced by the energy refinement study, the large memory requirements can be a significant drawback to the use of subspace type methods and the Davidson method in particular. To illustrate that this does not present a true limitation of the method, we now look at the effect of changing the maximum allowable subspace size and thus forcing the Davidson method to periodically restart. The restarting technique that we use is equivalent to the thick restarting process of Stathopoulos that is described in Section 3.2.6. We

Table 6.15: Performance for P_N order refinement, HTTR.

	CMFD		Arnoldi		RQI		Dav/MGE	
P_N Order	Iters	Time	Iters	Time	Iters	Time	Iters	Time
0	14	190	5	611	2	225	13	135
1	14	184	5	640	2	244	14	157
2	15	204	5	722	2	284	14	175
3	15	214	5	828	2	321	15	210

Table 6.16: Performance for P_N order refinement, C5G7.

	CMFD		Arnoldi		RQI		Dav/MGE	
P_N Order	Iters	Time	Iters	Time	Iters	Time	Iters	Time
0	12	227	12	1472	4	633	18	243
1	11	248	10	1375	4	675	16	236
2	11	261	10	1610	4	777	17	283
3	11	275	10	1934	4	891	16	320

Table 6.17: Davidson performance for subspace size variation, HTTR.

	Restart Dimension					
Max Dim.	1	2	4	6	8	10
2	25	–	–	–	–	–
4	20	16	–	–	–	–
6	16	16	15	–	–	–
8	15	16	16	16	–	–
10	15	15	15	15	15	–
12	14	14	14	14	14	14

also vary the number of vectors that are retained after a restart. Increasing the restart dimension will increase the amount of 'good' information that is retained in the new subspace but if too many vectors are retained then the method may have to restart too many times, hindering convergence.

Tables 6.17 and 6.18 demonstrate that it is not necessary to have a very large subspace to achieve a rate of convergence nearly the same as that of the unrestarted method (recall the unrestarted approach required 14 iterations for convergence on the HTTR problem and 16 for the C5G7). Although a maximum subspace size of 2 results in significant degradation, even allowing only 4 basis vectors still results in very favorable performance and with 6 basis vectors the iteration counts are almost the same as their unrestarted counterparts. Although there seems to be some benefit in retaining additional vectors upon restarting, the impact of this selection remains fairly small.

These results would seem to alleviate one of the primary concerns associated with the use of the Davidson method, namely that the memory requirements are simply too steep for a practical method. While having to store 3 sets of basis vectors that each contain 20 vectors may be a prohibitive memory cost

Table 6.18: Davidson performance for subspace size variation, C5G7.

	Restart Dimension					
Max Dim.	1	2	4	6	8	10
2	42	–	–	–	–	–
4	23	21	–	–	–	–
6	18	18	18	–	–	–
8	18	18	17	17	–	–
10	18	17	17	16	17	–
12	18	17	17	17	16	16

for large problems (at least for certain computer architectures), reducing the size of the bases to 6 or 8 vectors each may certainly be feasible.

6.3 Linear Solvers

Although the primary goal of this work is the solution of the k-eigenvalue problem, because many eigensolvers require solving a linear system with the transport equation at each iteration we feel it is relevant to briefly discuss the strategies for solving such linear systems. Table 6.19 shows the performance of three solvers applied to the same problems used for the eigenvalue studies. Here the right hand side is taken to be $\mathbf{x} = B\mathbf{e}$, where $\mathbf{e}$ is a vector of all ones, approximating the type of right hand side that would be present within an outer eigenproblem. The block Gauss-Seidel implementation is intended to be representative of the standard linear solver that is typically available in transport solvers. The diagonal blocks of the multigroup problem are solved using GMRES with no preconditioning and a maximum of 4 iterations per group. The other methods shown are GMRES applied directly to the multigroup problem with no preconditioning and GMRES with MGE as a preconditioner. This last method is the technique that was used to solve the linear systems in the Arnoldi and RQI methods for the eigensolver results. We also note that an attempt was made to use the transport two-grid method to accelerate the block Gauss-Seidel iterations but it proved to be unstable for these problems.

The block Gauss-Seidel method behaves quite differently for the two problems, with convergence occurring quite rapidly for the HTTR problem and much slower for the C5G7. However, even on the HTTR problem the time required for convergence is much larger for block Gauss-Seidel than for either of the GMRES approaches. This happens because each block Gauss-Seidel iteration requires performing a number of inner iterations, making each Gauss-Seidel iteration much more expensive than a GMRES step. The impact of preconditioning is evident in both problems, reducing the iteration count by approximately a factor of 4 and the total computational effort by

Table 6.19: Comparison of linear solver performance for base cases.

	HTTR		C5G7	
Solver	Iters	Time (s)	Iters	Time (s)
Block G-S	13	479	69	1963
GMRES	59	221	60	298
GMRES/MGE	18	141	15	155

roughly a factor of 2. Detailed parametric studies comparable to those found for the eigensolvers can be found in Appendix A.

As discussed previously, the memory costs associated with subspace methods may become unacceptably large under some circumstances and it is therefore important to consider the impact of restarting such methods. Tables 6.20 and 6.21 demonstrate this behavior for GMRES(m) with a range of restart lengths with and without preconditioning. We consider the solution of linear systems involving both the unshifted matrix A as well as the shifted matrix $A - \mu B$. The former is representative of an inner iteration with the Arnoldi method and the latter roughly approximates the inner linear solve at an intermediate stage of Rayleigh quotient iteration. We display here only the iteration counts, illustrating the scaling of each problem with respect to these parameters. As would generally be expected, the performance of GMRES tends to degrade as the maximum subspace dimension is reduced. For the unshifted C5G7 problem this degradation is quite significant, especially at the smallest restart lengths. The HTTR behavior is slightly more favorable, although it is still evident that better performance is achieved when the subspace can be made large. Preconditioning significantly reduces the impact of restarting and for both problems robust performance is observed for a subspace as small as 5.

Table 6.20: Linear solver performance for $(A - \mu B)$ for subspace variation, HTTR. Dashes indicate stagnation of the method.

	GMRES(m)		GMRES(m)/MGE	
m	$\mu = 0$	$\mu = 0.99\lambda_{\min}$	$\mu = 0$	$\mu = 0.99\lambda_{\min}$
5	77	–	20	–
10	83	–	20	60
25	81	–	18	21
50	62	350	18	21

The effect of shifting the linear system reinforces the importance of preconditioning when a restarted GMRES is used. The unpreconditioned GMRES is unable to converge if the maximum subspace size is smaller than 50, whereas the preconditioned approach is able to reach a solution (albeit with significantly degraded performance) for a subspace size of only 10.

The reason for the stagnation of the unpreconditioned GMRES can be demonstrated by observing the convergence behavior as the shift parameter is increased. The result, shown in Figure 6.6 for the C5G7 problem, reveals the difficulty. When no shift is applied, the convergence is very consistent for the entire iteration. As the shift parameter becomes close to the true eigenvalue of the problem, however, the convergence curves develop a plateau region where little reduction in the residual norm occurs. The length of this plateau increases as the shift parameter is increased, although the ultimate rate of convergence is largely unaffected. The difficulty therefore occurs when the method is restarted before the end of the plateau is reached. Upon restarting, the convergence curve immediately enters another plateau and is never able to overcome this obstacle and make meaningful progress towards a solution. Preconditioning dramatically increases the rate of convergence,

Table 6.21: Linear solver performance for $(A - \mu B)$ for subspace variation, C5G7. Dashes indicate stagnation of the method.

	GMRES(m)		GMRES(m)/MGE	
m	$\mu = 0$	$\mu = 0.99\lambda_{\min}$	$\mu = 0$	$\mu = 0.99\lambda_{\min}$
5	108	–	15	–
10	84	–	16	70
25	69	–	15	24
50	65	387	15	24

largely circumventing the issues related to the convergence plateaus. For very large problems where a subspace size of several hundred vectors is not feasible, the MGE preconditioner may not only speed up computations but also allow some problems to be solved that would not be possible with other strategies.

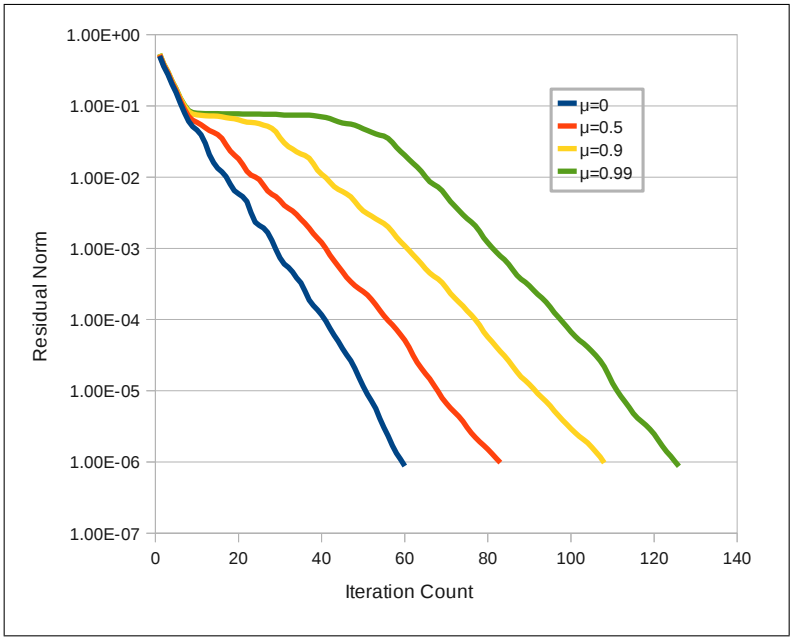


Figure 6.6: GMRES convergence with fission contribution.

Chapter 7

Conclusion

Although numerical solution of the k-eigenvalue problem has been the subject of significant research over the past several decades, there exist no methods that are truly satisfactory for use with difficult problems of interest in the nuclear engineering community. Standard techniques are based upon a nested iteration structure that hampers their effectiveness and generally exhibit poor convergence behavior for difficult problems. Approaches to accelerate this slow convergence can be extremely effective for certain problems but suffer from a general lack of stability, with performance frequently depending quite heavily on problem parameters.

In this work, we have investigated the performance of a variety of numerical solvers applied to the k-eigenvalue formulation of the neutron transport equation. In particular, the application of a generalized Davidson technique has been studied in detail. Although the Davidson method has been used in fields such as computational chemistry, to our knowledge this is the first instance of its application to this problem. The primary advantage of the Davidson framework is that it allows the problem to be treated directly as a generalized eigenvalue problem rather than first converting it to a standard eigenvalue problem as most commonly used methods require. Such a

conversion requires inverting (in practice, solving linear systems with) a full multigroup transport matrix, introducing a nested iteration structure that plagues nearly all current solvers. This process can be very computationally expensive and ultimately limit the efficiency that can be achieved with a given method. The Davidson method, on the other hand, simply requires the action of a preconditioner that may be an inexpensive approximation to a matrix inversion and in this respect can be viewed as an inexact Arnoldi method. However, unlike the situation with a true Arnoldi method, the inexactness of this preconditioner has no impact on the accuracy of any computed eigenvalues or eigenvectors.

We have also developed a preconditioner that is intended to be used with the Davidson method for the solution of the k-eigenvalue problem or as a preconditioner to a Krylov method for solving linear systems involving the transport matrices. Because one of the most problematic aspects of solving problems in nuclear reactor analysis is the presence of energy upscattering, we have developed a strategy with the goal of efficiently handling such problems. Based upon two level upscatter acceleration methods in the transport literature, the MGE approach extends these ideas to a true multigrid method with respect to the energy variable. This new method has the advantage that it is able to effectively damp all error modes present in a problem, unlike the two-grid methods that are only effective on a single mode. This feature allows MGE to be used without the need for a block Gauss-Seidel iteration, thus eliminating a full level of iteration that must be present in other solvers. Additionally, combining the MGE approach with either the Davidson eigenvalue solver or a Krylov method circumvents the stability issues associated with the two-grid (and other) methods.

The approach that we have selected for the MGE preconditioner is a standard multigrid V-cycle with a small number (3-4) of smoothing iterations at each level. Several possible smoothers have been proposed, with the lead-

ing candidates being damped Richardson or modified Gauss-Seidel methods. Infinite medium analysis and numerical experiments guide the selection of the relaxation parameter with the fortuitous observation that the optimal selection is largely independent of the particular problem under consideration and completely independent of discretization parameters. A sizable reduction in the computational effort per iteration can be achieved by using a coarse angular approximation to the original transport operator with only a small increase in the number of outer iterations required for convergence.

The methods described have been implemented and numerical tests performed both for the solution of the k-eigenvalue problem as well as the linear (fixed point) problem within NEWT, a 2-D neutron transport code developed at Oak Ridge National Laboratory. The linear solver tests indicate that MGE can be a very effective preconditioner when used with a Krylov subspace method (in our case, GMRES). The impact of the preconditioning is particularly evident when restarted GMRES is used or when the transport problem is shifted to include a component of the fission matrix in addition to standard scattering. In fact, MGE-preconditioned GMRES appears to be the only available method that is capable of solving such shifted problems when limitations are placed on the maximum allowable subspace dimension.

The use of MGE in combination with the Davidson eigensolver also proves to be a very effective strategy, consistently outperforming the standard power method, Arnoldi's method and Rayleigh quotient iteration across a variety of problems. The method is even competitive with (and usually superior to) the nonlinear coarse mesh finite difference acceleration available in NEWT. Although the gains relative to CMFD are minor for some problems, it is important to recall some of the issues pertaining to such nonlinear methods. In particular, CMFD is hindered by a lack of robustness that leads to poor convergence or even divergence if the size of the coarse mesh is not chosen appropriately. The optimal mesh parameter is problem dependent and usu-

ally must be selected based on the experience and intuition of the user. This issue is significant enough that CMFD is frequently excluded from radiation transport codes and in fact is not used as the default eigensolver in NEWT for precisely this reason. As an additional difficulty, CMFD relies on an underlying Cartesian grid structure in order to obtain its coarse spatial mesh. Such a Cartesian structure will typically not be available in unstructured mesh solvers and may not even be an option in structured mesh methods if the spatial mesh varies in size across different regions of the problem. Although we feel that more testing (and likely a 3-D implementation) is necessary before the method could be widely adopted, the favorable properties shown in this work provide a very strong indication that the Davidson/MGE approach will compete for a place among the best eigensolvers available to the transport community.

Appendix A

Linear Solver Parametric Studies

Table A.1: Linear solver performance for angle refinement, C5G7.

	GMRES		GMRES/MGE	
S_N Order	Iters	Time (s)	Iters	Time (s)
4	60	101	15	101
8	60	298	15	155
12	60	617	15	240
16	60	1035	16	377

Table A.2: Linear solver performance for angle refinement, HTTR.

	GMRES		GMRES/MGE	
S_N Order	Iters	Time (s)	Iters	Time (s)
4	58	73	18	92
8	59	221	18	141
12	59	464	18	220
16	59	874	18	345

Table A.3: Linear solver performance for energy refinement, C5G7.

	GMRES		GMRES/MGE	
Groups	Iters	Time (s)	Iters	Time (s)
16	60	298	15	155
32	63	631	14	305
64	66	1415*	12	573
128	–	–	11	1235

Table A.4: Linear solver performance for energy refinement, HTTR.

	GMRES		GMRES/MGE	
Groups	Iters	Time (s)	Iters	Time (s)
16	59	221	18	141
32	71	544	17	283
64	87	1387	17	619
128	–	–	19	1679

Table A.5: Linear solver performance for spatial refinement, C5G7.

	GMRES		GMRES/MGE	
Grid	Iters	Time (s)	Iters	Time (s)
1	57	201	14	107
2	60	298	15	155
3	61	709	16	387
4	63	1144	16	582

Table A.6: Linear solver performance for spatial refinement, HTTR.

	GMRES		GMRES/MGE	
Grid	Iters	Time (s)	Iters	Time (s)
1	56	70	18	46
2	59	221	18	141
3	60	484	18	300
4	61	854	18	520

Table A.7: Linear solver performance for P_N order refinement, C5G7.

	GMRES		GMRES/MGE	
P_N Order	Iters	Time (s)	Iters	Time (s)
0	56	255	16	149
1	60	298	15	155
2	59	327	16	185
3	59	360	17	225

Table A.8: Linear solver performance for P_N order refinement, HTTR.

	GMRES		GMRES/MGE	
P_N Order	Iters	Time (s)	Iters	Time (s)
0	59	206	18	127
1	59	221	18	141
2	58	243	18	160
3	59	267	19	191

Bibliography

[1] *ENDF/B-VII.0: Next generation evaluated nuclear data library for nuclear science and technology*, 2006. Nuclear Data Sheets 107.

[2] *SCALE: A modular code system for performing standardized computer analyses for licensing evaluation*, 2009. ORNL/TM-2005/39, Version 6, Vols. I-III.

[3] B. T. Adams and J. E. Morel, *A two-grid acceleration scheme for the multigroup Sn equations with neutron upscattering*, Nuclear Science and Engineering, 115 (1993), p. 253.

[4] M. Adams, *Subcell balance methods for radiative transfer on arbitrary grids*, Transport Theory and Statistical Physics, 26 (1997), p. 385.

[5] M. L. Adams and E. W. Larsen, *Fast iterative methods for discrete-ordinates particle transport calculations*, Progress in Nuclear Energy, 40 (2002), p. 3.

[6] R. E. Alcouffe, *A stable diffusion synthetic acceleration method for neutron transport iterations*, Transactions of the American Nuclear Society, 23 (1976), p. 203.

[7] ——, *Diffusion synthetic acceleration for the diamond-differenced discrete-ordinates equations*, Nuclear Science and Engineering, 64 (1977), p. 344.

[8] E. J. Allen and R. M. Berry, *The inverse power method for calculation of multiplication factors*, Annals of Nuclear Energy, 29 (2002), p. 929.

[9] W. E. Arnoldi, *The principle of minimized iterations in the solution of matrix eigenvalue problems*, Quarterly of Applied Mathematics, 9 (1951), p. 17.

[10] S. F. Ashby, P. N. Brown, M. R. Dorr, and A. C. Hindmarsh, *A linear algebraic analysis of diffusion synthetic acceleration for the Boltzmann transport equation*, SIAM Journal on Numerical Analysis, 32 (1995), p. 128.

[11] A. V. Averin and A. M. Voloschenko, *Consistent P_1 acceleration method for outer iterations*, Transport Theory and Statistical Physics, 23 (1994), p. 701.

[12] Y. Y. Azmy, *Iterative convergence acceleration of neutral particle transport methods via adjacent-cell preconditioners*, Journal of Computational Physics, 152 (1999), p. 359.

[13] T. Bailey and R. Falgout, *Analysis of massively parallel discrete ordinates transport sweep algorithms with collisions*, in International Conference on Mathematics, Computational Methods, and Reactor Physics, Saratoga Springs, NY, 2009, American Nuclear Society.

[14] R. S. Baker and R. E. Alcouffe, *Parallel 3-D S_n performance for DANTSYS/MPI on the Cray T3D*, in Joint International Conference on Mathematical Methods and Supercomputing for Nuclear Applications, Saratoga Springs, New York, October 1997, American Nuclear Society.

[15] R. S. Baker and K. R. Koch, *An S_n algorithm for the massively parallel CM-200 computer*, Nuclear Science and Engineering, 128 (1998), p. 312.

[16] M. Benzi, *Preconditioning techniques for large linear systems: a survey*, Journal of Computational Physics, 182 (2002), p. 418.

[17] M. Borysiewicz and J. Mika, *Existence and uniqueness of the solution to the critical problem in the multigroup neutron-transport theory*, Transport Theory and Statistical Physics, 2 (1972), p. 243.

[18] A. Bouras and V. Frayssé, *A relaxation strategy for inexact matrix-vector products for Krylov methods*, Tech. Rep. TR/PA/00/15, CERFACS, 2000.

[19] P. N. Brown and Y. Saad, *Convergence of nonlinear Newton-Krylov algorithms*, SIAM Journal of Optimization, 4 (1994), p. 297.

[20] L. Cao, H. Wu, and Y. Zheng, *Solution of neutron transport equation using Daubechies' wavelet expansion in the angular discretization*, Nuclear Engineering and Design, 238 (2008), p. 2292.

[21] B. G. Carlson, *Method of characteristics and other improvements in solution methods for the transport equation*, Nuclear Science and Engineering, 61 (1976), p. 408.

[22] B. G. Carlson and K. D. Lathrop, *Transport theory – the method of discrete ordinates*, in Computing Methods in Reactor Physics, New York, 1968, Gordon and Breach.

[23] G. R. Cefus and E. W. Larsen, *Stability analysis of coarse-mesh rebalance*, Nuclear Science and Engineering, 105 (1990), p. 31.

[24] C. Cercignani, *The Boltzmann Equation and Its Applications*, Springer-Verlag, 1988.

[25] B. Chang, *The conjugate gradient method solves the neutron transport equation h-optimally*, Numerical Linear Algebra with Applications, 14 (2007), p. 751.

[26] B. Chang, T. Manteuffel, S. McCormick, J. Ruge, and B. Sheehan, *Spatial multigrid for isotropic neutron transport*, SIAM Journal on Scientific Computing, 29 (2007), p. 1900.

[27] F. Chatelin and W. L. Miranker, *Acceleration by aggregation of successive approximation methods*, Linear Algebra and its Applications, 43 (1982), p. 17.

[28] ——, *Aggregation/disaggregation for eigenvalue problems*, SIAM Journal on Numerical Analysis, 21 (1984), p. 567.

[29] N. Z. Cho and C. J. Park, *A comparison of coarse mesh rebalance and coarse mesh finite difference accelerations for the neutron transport calculations*, in Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew, Gatlinburg, TN, April 2003.

[30] K. Clarno, V. de Almeida, E. d'Azevedo, C. de Oliveira, and S. Hamilton, *GNES-R: global nuclear energy simulator for reactors task 1: High-fidelity neutron transport*, in Advances in Nuclear Analysis and Simulation (PHYSOR 2006), Vancouver, B.C., September 2006.

[31] K. T. Clarno, *Implementation of generalized coarse-mesh rebalance in NEWTRNX for acceleration of parallel block-Jacobi transport*, Transactions of the American Nuclear Society, 97 (2007), p. 498.

[32] G. G. M. Coppa, G. Lapenta, and P. Ravetto, *Angular finite element techniques in neutron transport*, Annals of Nuclear Energy, 17 (1990), p. 363.

[33] M. Crouzeix, B. Philippe, and M. Sadkane, *The Davidson method*, SIAM Journal on Scientific Computing, 15 (1994), p. 62.

[34] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices*, Journal of Computational Physics, 17 (1975), p. 87.

[35] G. G. Davidson, T. M. Evans, R. N. Slaybaugh, and C. G. Baker, *Massively parallel solutions to the k-eigenvalue problem*, in 2010 American Nuclear Society Winter Meeting, Las Vegas, NV, November 2010.

[36] C. R. E. de Oliveira, *An arbitrary geometry finite element method for multigroup neutron transport with anisotropic scattering*, Progress in Nuclear Energy, 18 (1986), p. 227.

[37] C. R. E. de Oliveira and A. Goddard, *EVENT: A multidimensional finite-element spherical harmonics radiation transport code*, in International Seminar on 3-D Deterministic Radiation Transport Codes, Paris, France, December 1996, Organization for Economic Cooperation and Development.

[38] M. D. DeHart, *NEWT: A new transport algorithm for two-dimensional discrete ordinates analysis in non-orthogonal geometries*, 2006. NUREG/CR-0200, Rev. 8 (ORNL/NUREG/CSD-2/R8), Vols. I-III.

[39] M. D. DeHart, R. E. Pevey, and T. A. Parish, *An extended step characteristic method for solving the transport equation in general geometries*, Nuclear Science and Engineering, 118 (1994), p. 79.

[40] S. Douglass, *Generalized energy condensation theory*, Master's thesis, Georgia Institute of Technology, 2007.

[41] 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 (2010), p. 1.

[42] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, and K. T. Clarno, *Denovo – a new three-dimensional parallel discrete ordinates code in SCALE*, Nuclear Technology, 171 (2010), p. 171.

[43] V. Faber and T. A. Manteuffel, *A look at transport theory from the point of view of linear algebra*, in Conference on Transport Theory, Invariant Imbedding and Integral Equations, vol. 20, 1988.

[44] D. R. Fokkema, G. L. G. Sleijpen, and H. A. Van der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils*, SIAM Journal on Scientific Computing, 20 (1998), p. 94.

[45] D. Gill, *Newton-Krylov Methods for the Solution of the k-Eigenvalue Problem in Multigroup Neutronics Calculations*, PhD thesis, Penn State University, 2009.

[46] V. Y. Gol'din, *A quasi-diffusion method for solving the kinetic equation*, USSR Computational Mathematics and Mathematical Physics, 4 (1967), p. 136. English translation of original 1964 Russian paper.

[47] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.

[48] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[49] R. E. Grove, *The slice balance approach (SBA): A characteristic-based, multiple balance SN approach on unstructured polyhedral meshes*, Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications, Avignon, France (2005).

[50] A. Gupta and R. S. Modak, *Krylov sub-space methods for k-eigenvalue problem in 3-D neutron transport*, Annals of Nuclear Energy, 31 (2004), p. 2113.

[51] B. Guthrie, J. P. Holloway, and B. W. Patton, *GMRES as a multi-step transport sweep accelerator*, Transport Theory and Statistical Physics, 28 (1999), p. 83.

[52] M. J. Halsall, *CACTUS: A Characteristics Solution to the Neutron Transport Equations in Complicated Geometries*, Atomic Energy Establishment, Winfrith, Dorchester, Dorest, UK, 1980. AEEW-R 1291.

[53] S. Hamilton, M. Benzi, and J. Warsa, *Negative flux fixups in discontinuous finite element SN transport*, in 2009 International Conference on Advances in Mathematics, Computational Methods, and Reactor Physics, Saratoga Springs, NY, May 2009.

[54] S. P. Hamilton, *A time-dependent slice balance method for high-fidelity radiation transport computations*, Master's thesis, Georgia Institute of Technology, 2007.

[55] D. L. Harrar II, *On the Davidson and Jacobi-Davidson methods for large-scale eigenvalue problems*, Tech. Rep. MRR99-047, Centre for Mathematics and its Applications, Australian National University, 1999.

[56] M. Haviv, *Aggregation/disaggregation methods for computing the stationary distribution of a Markov chain*, SIAM Journal on Numerical Analysis, 4 (1987), p. 952.

[57] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradient for solving linear systems*, Journal of Research of the National Bureau of Standards, 49 (1952), p. 409.

[58] S. G. Hong and N. Z. Cho, *CRX: A code for rectangular and hexagonal lattices based on the method of characteristics*, Annals of Nuclear Energy, 25 (1998), p. 547.

[59] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1990.

[60] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics, Philadelphia, 1995.

[61] D. Kershaw and J. Harte, *2D deterministic radiation transport with the discontinuous finite element method*, tech. rep., UCRL-ID115525, Lawrence Livermore National Lab., CA, 1993.

[62] K.-S. Kim and M. D. DeHart, *Unstructured partial- and net-current based coarse mesh finite difference acceleration applied to the extended step characteristics method in NEWT*, Annals of Nuclear Energy, 38 (2011), p. 527.

[63] D. Knoll and D. Keyes, *Jacobian-free Newton-Krylov methods: A survey of approaches and applications*, Journal of Computational Physics, 193 (2004), p. 357.

[64] H. J. Kopp, *Synthetic method solution of the transport equation*, Nuclear Science and Engineering, 17 (1963), p. 65.

[65] V. S. Kumar, M. V. Marathe, S. Parthasarathy, A. Srinivasan, and S. Zust, *Provable algorithms for parallel generalized sweep scheduling*, Journal of Parallel and Distributed Computing, 66 (2006), p. 821.

[66] E. W. Larsen, *Unconditionally stable diffusion-acceleration of the transport equation*, Transport Theory and Statistical Physics, 11 (1982), p. 29.

[67] ——, *Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete ordinates equations. Part 1: Theory*, Nuclear Science and Engineering, 82 (1982), p. 47.

[68] ——, *Unconditionally stable diffusion-synthetic acceleration methods for the slab geometry discrete ordinates equations. Part 2: Numerical results*, Nuclear Science and Engineering, 82 (1982), p. 64.

[69] K. D. Lathrop, *Spatial differencing of the transport equation: Positivity vs. accuracy*, Journal of Computational Physics, 4 (1969), p. 475.

[70] D. Lee and T. J. Downar, *Convergence analysis of the nonlinear coarse mesh finite difference method*, in Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew, Gatlinburg, TN, April 2003, American Nuclear Society.

[71] R. B. Lehoucq, *Implicitly restarted Arnoldi methods and subspace iteration*, SIAM Journal on Matrix Analysis and Applications, 23 (2001), p. 551.

[72] E. E. Lewis and W. F. Miller, Jr., *Computational Methods of Neutron Transport*, John Wiley & Sons, New York, 1984.

[73] I. Marek and P. Mayer, *Convergence analysis of an iterative aggregation/disaggregation method for computing stationary probability vectors of stochastic matrices*, Numerical Linear Algebra with Applications, 5 (1998), p. 253.

[74] W. R. Martin, C. E. Yehnert, L. Lorence, and J. J. Duderstadt, *Phase-space finite element methods applied to the first-order form of the transport equation*, Annals of Nuclear Energy, 8 (1981), p. 633.

[75] J. Mika, *Existence and uniqueness of the solution to the critical problem in neutron-transport theory*, Studia Mathematica, 37 (1971), p. 213.

[76] W. L. Miranker and V. Y. Pan, *Methods of aggregation*, Linear Algebra and its Applications, 29 (1980), p. 231.

[77] R. S. Modak and A. Gupta, *New applications of Orthomin(1) algorithm for k-eigenvalue problem in reactor physics*, Annals of Nuclear Energy, 33 (2006), p. 538.

[78] M. Mordant, *Phase-space finite elements encoded in zephyr for X-Y and R-Z transport calculations*, Progress in Nuclear Energy, 18 (1986), p. 27.

[79] J. E. Morel, B. T. Adams, T. Noh, J. M. McGhee, T. M. Evans, and T. J. Urbatsch, *Spatial discretizations for self-adjoint forms of the radiative transfer equations*, Journal of Computational Physics, 214 (2006), p. 12.

[80] J. E. Morel and T. A. Manteuffel, *An angular multigrid acceleration technique for s_n equations with highly forward-peaked scattering*, Nuclear Science and Engineering, 107 (1991), p. 330.

[81] J. E. Morel and J. M. McGhee, *A self-adjoint angular flux equation*, Nuclear Science and Engineering, 132 (1999), p. 312.

[82] R. B. Morgan and D. S. Scott, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices.*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), p. 817.

[83] Y. Notay, *Convergence analysis of inexact Rayleigh quotient iteration*, SIAM Journal on Matrix Analysis and Applications, 24 (2003), p. 627.

[84] P. Nowak and M. K. Nemanic, *Radiation transport calculations on unstructured grids using a spatially decomposed and threaded algorithm*, in American Nuclear Society Conference on Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications, 1999.

[85] Oak Ridge National Laboratory, *Consortium for the Advanced Simulation of Light Water Reactors*. Fact Sheet, 2010.

[86] OECD/NEA, *Benchmark on Deterministic Transport Calculations Without Spatial Homogenisation: A 2-D/3-D MOX Fuel Assembly Benchmark*. NEA/NSC/DOC(2005)16.

[87] S. Oliveira and Y. Deng, *Preconditioned Krylov subspace methods for transport equations*, Progress in Nuclear Energy, 33 (1998), p. 155.

[88] J. Olsen, P. Jorgensen, and J. Simons, *Passing the one-billion limit in full configuration-interaction (FCI) calculations*, Chemical Physics Letters, 169 (1990), p. 463.

[89] H. Park and D. Knoll, *Development and application of nonlinear methods in computational nuclear engineering*, in Eleventh Copper

Mountain Conference on Iterative Methods, Copper Mountain, CO, April 2010.

[90] B. W. Patton and J. P. Holloway, *Application of preconditioned GMRES to the numerical solution of the neutron transport equation*, Annals of Nuclear Energy, 29 (2002), p. 109.

[91] S. D. Pautz, *An algorithm for parallel S_N sweeps on unstructured meshes*, Nuclear Science and Engineering, 140 (2002), p. 111.

[92] G. Peters and J. H. Wilkinson, *Inverse iteration, ill-conditioned equations and Newton's method*, SIAM Review, 21 (1979), p. 339.

[93] S. Plimpton, B. Hendrickson, S. Burns, and W. M. III, *Parallel algorithms for radiation transport on unstructured grids*, in ACM/IEEE 2000 Conference Supercomputing, 2000, p. 25.

[94] S. J. Plimpton, B. Hendrickson, S. P. Burns, W. McLendon, and L. Rauchwerger, *Parallel Sn sweeps on unstructured grids: Algorithms for prioritization, grid partitioning, and cycle detection*, Nuclear Science and Engineering, 150 (2005), p. 267.

[95] W. H. Reed, *The effectiveness of acceleration techniques for iterative methods in transport theory*, Nuclear Science and Engineering, 45 (1971), p. 245.

[96] W. H. Reed and T. R. Hill, *Triangular mesh methods for the neutron transport equation*, Los Alamos Report LA-UR-73-479, (1973).

[97] M. Rosa, J. S. Warsa, and T. M. Kelley, *Fourier analysis of cell-wise block-Jacobi splitting in two-dimensional geometry*, in International Conference on Advances in Mathematics, Computational Methods and Reactor Physics, Saratoga Springs, NY, May 2009.

[98] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, 1992.

[99] ——, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM Journal on Scientific Computing, 14 (1993), p. 461.

[100] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems.*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), p. 856.

[101] M. Sadkane, *Block-Arnoldi and Davidson methods for unsymmetric large eigenvalue problems*, Numerische Mathematik, 64 (1993), p. 195.

[102] F. Scheben and I. G. Graham, *Iterative methods for neutron transport eigenvalue problems*, SIAM Journal on Scientific Computing. submitted.

[103] P. Silvennoinen, *A self-adjoint form of the linear transport equation*, Journal of Mathematical Analysis and Applications, 43 (1973), p. 529.

[104] V. Simoncini, *Variable accuracy of matrix-vector products in projection methods for eigencomputation*, SIAM Journal on Numerical Analysis, 43 (2005), p. 1155.

[105] V. Simoncini and L. Eldén, *Inexact Rayleigh quotient-type methods for eigenvalue computations*, BIT Numerical Mathematics, 42 (2002), p. 159.

[106] V. Simoncini and D. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM Journal on Scientific Computing, 25 (2004), p. 454.

[107] G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. Van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT Numerical Mathematics, 36 (1996), p. 595.

[108] G. L. G. Sleijpen and H. A. Van Der Vorst, *The Jacobi-Davidson method for eigenvalue problems and its relation with accelerated inexact Newton schemes*, IMACS Annals on Computing and Applied Mathematics, 3 (1996), p. 377.

[109] G. L. G. Sleijpen and H. A. Van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM Review, 42 (2000), p. 267.

[110] G. L. G. Sleijpen, H. A. Van der Vorst, and E. Meijerink, *Efficient expansion of subspaces in the Jacobi-Davidson method for standard and generalized eigenproblems*, Electronic Transactions on Numerical Analysis, 7 (1998), p. 75.

[111] K. S. Smith and J. D. Rhodes, *CASMO-4 characteristics methods for two-dimensional PWR and BWR core simulations*, Transactions of the American Nuclear Society, 83 (2000), p. 294.

[112] D. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), p. 357.

[113] W. M. Stacey, *Nuclear Reactor Physics*, John Wiley & Sons, New York, 2007.

[114] A. Stathopoulos, Y. Saad, and K. Wu, *Dynamic thick restarting of the Davidson, and the implicitly restarted Arnoldi methods*, SIAM Journal of Scientific Computing, 19 (1998), p. 227.

[115] W. J. Stewart, ed., *Numerical Solution of Markov Chains*, Marcel Dekker, 1991.

[116] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, San Diego, 2001.

[117] S. Van Criekingen, F. Nataf, and P. Have, *PARAFISH : A parallel FE-PN neutron transport solver based on domain decomposition*, Annals of Nuclear Energy, 38 (2011), p. 145.

[118] T. Van Noorden and J. Rommes, *Computing a partial generalized real Schur form using the Jacobi-Davidson method*, Numerical Linear Algebra with Applications, 14 (2007), p. 197.

[119] W. F. Walters and T. A. Wareing, *An accurate, strictly-positive, nonlinear characteristic scheme for the discrete-ordinate equations*, Transport Theory and Statistical Physics, 25 (1996), p. 197.

[120] T. A. Wareing, *An exponential discontinuous scheme for discrete-ordinate calculations in Cartesian geometries*, in Joint international conference on mathematical methods and supercomputing in nuclear applications, Saratoga Springs, NY, October 1997.

[121] T. A. Wareing, E. W. Larsen, and M. L. Adams, *Diffusion accelerated discontinuous finite element schemes for the SN equations in slab and X-Y geometries*, in International Topical Meeting on Advances in Mathematics, Computations and Reactor Physics, Pittsburg, PA, April 1991, American Nuclear Society.

[122] T. A. Wareing, J. M. McGhee, J. E. Morel, and S. D. Pautz, *Discontinuous finite element S_N methods on 3-D unstructured grids*, in Mathematics and Computations, Madrid, Spain, September 1999.

[123] J. S. Warsa, *A continuous finite element-based, discontinuous finite element method for SN transport*, Nuclear Science and Engineering, 160 (2008), p. 1.

[124] J. S. Warsa, M. Benzi, T. A. Wareing, and J. E. Morel, *Preconditioning a mixed discontinuous finite element method for radiation diffusion*, Numerical Linear Algebra with Applications, 11 (2004), p. 795.

[125] J. S. Warsa, T. A. Wareing, and J. E. Morel, *Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional SN calculations in problems with material discontinuities*, Nuclear Science and Engineering, 147 (2004), p. 218.

[126] J. S. Warsa, T. A. Wareing, J. E. Morel, J. M. McGhee, and R. B. Lehoucq, *Krylov subspace iterations for deterministic k-eigenvalue calculations*, Nuclear Science and Engineering, 147 (2004), p. 26.

[127] R. Wienands and W. Joppich, *Practical Fourier Analysis for Multigrid Methods*, Chapman & Hall, 2005.

[128] W. Wieselquist, *The Quasidiffusion Method for Transport Problems on Unstructured Meshes*, PhD thesis, North Carolina State University, 2009.

[129] A. Yamamoto, *Generalized coarse-mesh rebalance method for acceleration of neutron transport calculations*, Nuclear Science and Engineering, 151 (2005), p. 274.

[130] W. Yang, H. Wu, Y. Zheng, and L. Cao, *Application wavelets scaling function expansion method in resonance self-shielding calculation*, Annals of Nuclear Energy, 37 (2010), p. 653.

[131] Z. Zhong, T. J. Downar, Y. Xu, M. D. DeHart, and K. T. Clarno, *Implementation of two-level coarse-mesh finite difference acceleration in an arbitrary geometry, two-dimensional discrete ordinates transport method*, Nuclear Science and Engineering, 158 (2008), p. 289.

[132] Y. Zhou, *Studies on Jacobi-Davidson, Rayleigh quotient iteration, inverse iteration generalized Davidson and Newton updates*, Numerical Linear Algebra with Applications, 13 (2006), p. 621.