Explicitly Restarted Arnoldi's Method for Monte Carlo Nuclear Criticality Calculations

by

Jeremy Lloyd Conlin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in The University of Michigan
2009

Doctoral Committee:
Professor James Paul Holloway, Chair
Professor Edward W. Larsen
Professor William R. Martin
Associate Professor Martin Strauss

To: Annie, Brigham, Lily, and Emma

Here is the black book I promised.

Acknowledgments

I have been fortunate during my time at the University of Michigan to have been associated with many remarkable, talented, and brilliant people. I have been educated by the best Nuclear Engineering professors. I have been peers with very bright students who have assisted me throughout my endeavours.

First is my advisor and friend Professor James Paul Holloway. You have been an example to me in all aspects of life. You have been very patient while helping me and many others who have struggled to do what was expected. My work has been improved because you have never been satisified with "good enough", but have demanded work of the highest quality. I have learned that if I was stuck on a problem, all I needed to do was make you sufficiently interested that you couldn't put it down until it was solved. You have learned to not become sufficiently interested so that I would be forced to solve the problem on my own. I have enjoyed teaching under you and learning while watching you teach. I have watched you in your interactions with others and have learned a greater respect for people as individuals, *not* as professors, janitors, secretaries, watiresses, graduate students, or deans.

Professor Bill Martin has been a great resource to me during my time at the University of Michigan. You were my first advisor and made it much easier for me to go through graduate school with a young and growing family. Through your connections I was able to have a fulfilling summer internship at Los Alamos National Laboratory which eventually led to my employment after finishing school. I am grateful for your confidence in my abilities to let me try many times to pass the

candidacy exam; I finally did it! During times when you were not my advisor, your door was always open (on the rare occasion when you were in town) and you were always willing to help.

Professor Larsen is the greatest teacher for advanced nuclear reactor theory. Your careful, step-by-step approach makes learning these difficult concepts possible. Your seemingly infinite knowledge and experience of reactor theory never overshadowed your desire to help students learn.

I'm grateful to Professor Martin Strauss who agreed to be on my PhD thesis committee without knowing me or anyone else in the Nuclear Engineering department. You are a brave man.

I have associated with many students with abilities far greater than my own. Their assistance on homework assignments, studying for the candidacy exam, and programming have been invaluable. David Griesheimer, Greg Davidson, Troy Becker, Jesse Cheatham, Scott Kiff, Seth Johnson, and Allan Wollaber; you have made it possible for me to make it through.

I would not have made it to graduate school (or anywhere else in life) without my parents Scott and Geri Conlin. It is your emphasis on the importance of a good education that led me to pursue a college education and eventually graduate school. Certainly your influence goes far beyond my educational pursuits. Mom, I'm glad we enjoyed (suffered?) through graduate school together. Now there will be two doctors in the house.

As important as my professors and fellow students have been at school, my family has been a greater support than all. Annie, Brigham, Lily, and Emma, you have made everything I do worthwhile. Regardless of how hard my day at school has been or how unimportant I feel, when I came home you were always there with a smile, and a hug to show me I was loved and always needed. Your endless supply of tickle bugs has brought music to my ears.

Trisha, my beautiful wife. You have been an incalculable strength to me. Your ability to take care of all the other aspects of my life has allowed me to succeed in my educational and professional endeavors. Your patience has helped me to become the man I should be. Your love has been a comfort and a joy. I'm glad we have enjoyed life together for ten years. I'm sure the next million will be even greater.

Finally I wish to express gratitude to my Savior, Jesus Christ. "I know that I am nothing; as to my strength I am weak; therefore I will not boast of myself, but I will boast of my God, for in his strength I can do all things".

Table of Contents

List of Figures

Figure

List of Tables

Abstract

Explicitly Restarted Arnoldi's Method for Monte Carlo Nuclear Criticality Calculations

by

Jeremy Lloyd Conlin

Chair: James Paul Holloway

A Monte Carlo implementation of explicitly restarted Arnoldi's method is developed for estimating eigenvalues and eigenvectors of the transport-fission operator in the Boltzmann transport equation. Arnoldi's method is an improvement over the power method which has been used for decades. Arnoldi's method can estimate multiple eigenvalues by orthogonalising the resulting fission sources from the application of the transport-fission operator. As part of implementing Arnoldi's method, a solution to the physically impossible—but mathematically real—negative fission sources is developed. The fission source is discretized using a first order accurate spatial approximation to allow for orthogonalization and normalization of the fission source required for Arnoldi's method. The eigenvalue estimates from Arnoldi's method are compared with published results for homogeneous, one-dimensional geometries, and it is found that the eigenvalue and eigenvector estimates are accurate within statistical uncertainty.

The discretization of the fission sources creates an error in the eigenvalue estimates. A second order accurate spatial approximation is created to reduce the error in eigenvalue estimates. An inexact application of the transport-fission operator is also investigated to reduce the computational expense of estimating the eigenvalues and eigenvectors.

The convergence of the fission source and eigenvalue in Arnoldi's method is analysed and compared with the power method. Arnoldi's method is superior to the power method for convergence of the fission source and eigenvalue because both converge nearly instantly for Arnoldi's method while the power method may require hundreds

of iterations to converge. This is shown using both homogeneous and heterogeneous one-dimensional geometries with dominance ratios close to 1.

Chapter 1

Introduction

For decades [see 15, 13, 18, 19], the power method has been the technique of choice for calculating eigenvalues and eigenvectors of the transport-fission operator of the Boltzmann transport equation in Monte Carlo particle transport applications. The power method is useful for estimating the fundamental or largest eigenvalue and associated eigenvector. The fundamental eigenmode describes the reactivity of a nuclear reactor operating in steady-state conditions.

The higher-order eigenmodes of the transport-fission operator are necessary to calculate the space and time dependent reactivity where there are significant changes in the flux [30]. When performing time and space dependent reactor kinetics, the shape of the neutron flux is approximated by an expansion with the eigenfunctions. The difference between the eigenvalues defines how quickly the contributions from the higher order eigenmodes decay away [1]. Estimating the higher-order eigenmodes with computer simulation provides the reactor designer with a tool to predict the behavior of the reactor.

Recently, some work has been done to estimate higher-order eigenmodes using the power method [3]. This dissertation will explore a heretofore unexplored approach to estimating both the fundamental and higher-order eigenmodes of the transport-fission operator: Arnoldi's method [2]. Arnoldi's method can estimate multiple eigenvalues

of a linear operator, but is particularly useful when the linear operator is large, sparse, or has no explicit form.

1.1 Particle Transport

The one-speed Boltzmann transport equation describing neutron transport for criticality problems,

$$\boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma_t \psi(\mathbf{r}, \boldsymbol{\Omega}) = \frac{\Sigma_s}{4\pi} \int \psi(\mathbf{r}, \boldsymbol{\Omega}') \ \mathrm{d}\Omega' + \frac{1}{k} \frac{\nu \Sigma_f}{4\pi} \int \psi(\mathbf{r}, \boldsymbol{\Omega}') \ \mathrm{d}\Omega', \tag{1.1}$$

can be written in operator form

$$(\mathbf{L} + \mathbf{C} - \mathbf{S})\psi = \frac{1}{k}\mathbf{F}\psi \tag{1.2a}$$

or

$$\mathbf{T}\psi = \frac{1}{k}\mathbf{F}\psi, \tag{1.2b}$$

where $\mathbf{L}, \mathbf{C}$, and $\mathbf{S}$ are the leakage, collision, and scattering operators respectively, and $\mathbf{F}$ is the fission multiplication operator; $\mathbf{T} = \mathbf{L} + \mathbf{C} - \mathbf{S}$ is the transport-collision operator. The left-hand side of Equation (1.2b) represents the neutron loss and scattering mechanisms, and the right-hand side represents the neutron gain mechanism.

If we define

$$v \equiv \mathbf{F}\psi, \tag{1.3a}$$

as the fission source and define

$$\mathcal{A} \equiv \mathbf{F}\,\mathbf{T}^{-1} \tag{1.3b}$$

as the transport-fission operator and manipulate Equation (1.2b), we find

$$\mathcal{A}v = kv. \tag{1.4}$$

This is a standard eigenvalue problem with eigenvalue k and eigenvector v for the transport-fission operator, $\mathcal{A}$. The eigenvalue, k, is just the multiplication or criticality factor and the eigenvector, v, is the fission source.

The application of $\mathcal{A}$ in Monte Carlo calculations is conceptually simple. Particles are sampled from the fission source and transported in the medium until they initiate a fission reaction. Sampling and transporting particles is repeated many times and the fission sites are stored, creating a new fission source.

1.2 Krylov Methods

There is a class of techniques which approximate eigenvalues and eigenvectors from a subspace of the form

$$\mathcal{K}_m(\mathcal{A}, v) \equiv \text{span}\left\{v, \mathcal{A}v, \mathcal{A}^2v, \ldots, \mathcal{A}^{m-1}v\right\}; \tag{1.5}$$

$\mathcal{K}_m(\mathcal{A}, v)$ is called a *Krylov subspace* [see 22, chapter VI]. The basis vectors of a Krylov subspace are calculated by repeated application of a linear operator $\mathcal{A}$ on an initial starting vector v.

The basis vectors defining a Krylov subspace are constructed iteratively. The linear operator $\mathcal{A}$ is applied to the starting vector v, getting

$$v_1 = \mathcal{A}v.$$

In later iterations, rather than applying $\mathcal{A}$ multiple times, the operator is applied to the result of the previous iteration

$$v_i = \mathcal{A}v_{i-1}, \qquad \text{for } i = 1, 2, \ldots, m-1, \tag{1.6}$$

with $v_0 = v$, the starting vector. The vectors $\{v_i\}_{i=1}^{m-1}$ are the basis vectors for the Krylov subspace. Equation (1.5) can thus be written as

$$\mathcal{K}_m(\mathcal{A}, v) \equiv \text{span}\left\{v_0, v_1, \ldots, v_{m-1}\right\} \tag{1.7}$$

It is important to note that the linear operator used to construct a Krylov subspace does not need to be known explicitly; rather, the repeated application of the operator on a vector must be known. This characteristic is sometimes referred to as a *matrix-free method* and makes Krylov subspace methods particularly attractive to Monte Carlo criticality applications, where an explicit form for the transport-fission operator does not exist. To illustrate how Krylov subspace methods can estimate the eigenvalues and eigenvectors of $\mathcal{A}$, the power method—a straightforward implementation of a Krylov subspace method—will first be described.

1.3 Power Method

The power method is ideally suited for calculating eigenvalues and eigenvectors of $\mathcal{A}$ with Monte Carlo particle transport. The process of sampling and transporting

particles can be performed iteratively. To do this we begin with an initial estimate of the fundamental eigenvalue and fission source, $\left(k_0, v_0\right)$, and apply the linear operator, $\mathcal{A}$. The resulting fission source can then be sampled for the next iteration.

To illustrate this iterative procedure we rearrange Equation (1.4) and add subscripts obtaining

$$v_{j+1} = \frac{1}{k_j}\mathcal{A}v_j. \tag{1.8}$$

Neutron positions are sampled from v_j and transported by the application of $\mathcal{A}$, forming a new fission source v_{j+1}. The fission source v_{j+1} is a new estimate of the fundamental eigenvector. A new estimate of the fundamental eigenvalue can be calculated as

$$k_{j+1} = k_j \frac{\int \mathcal{A}v_j}{\int v_j} = k_j \frac{\int v_{j+1}}{\int v_j}. \tag{1.9}$$

The integral of the fission source $\int v_j$ is the fission rate. In Monte Carlo calculations it is the number of fission sites in the fission source, v_j.

If $\mathcal{A}$ has a dominant eigenvalue λ_1, that is if $\lambda_1 > \lambda_2 \geq \lambda_3 \cdots$ then the eigenpair $\left(k_j, v_j\right)$ approaches the fundamental eigenpair (λ_1 and associated eigenvector) as j becomes large. To see how this happens let us write the initial estimate of the fundamental eigenvector as a linear combination of the eigenvectors of $\mathcal{A}$

$$v_0 = c_1 y_1 + c_2 y_2 + \cdots + c_n y_n, \tag{1.10}$$

where $y_1, \ldots, y_n$ are the eigenvectors of $\mathcal{A}$ and $c_1, \ldots, c_n$ are scalar constants. When we apply $\mathcal{A}$ to v_0, we get

$$\begin{aligned} \mathcal{A}v_0 &= c_1\mathcal{A}y_1 + c_2\mathcal{A}y_2 + \cdots + c_n\mathcal{A}y_n \\ &= c_1\lambda_1 y_1 + c_2\lambda_2 y_2 + \cdots + c_n\lambda_n y_n. \end{aligned} \tag{1.11}$$

After j applications of $\mathcal{A}$ (iterations), Equation (1.11) becomes

$$\begin{aligned}\mathcal{A}^j v_0 &= c_1\mathcal{A}^j y_1 + c_2\mathcal{A}^j y_2 + \cdots + c_n\mathcal{A}^j y_n \\ &= c_1\lambda_1^j y_1 + c_2\lambda_2^j y_2 + \cdots + c_n\lambda_n^j y_n. \end{aligned} \tag{1.12}$$

We can factor the dominant or fundamental eigenvalue λ_1 out of Equation (1.12) to obtain

$$\mathcal{A}^j v_0 = \lambda_1^j \left[c_1 y_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^j y_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^j y_n \right]. \tag{1.13}$$

Note that every multiple of an eigenvector is also an eigenvector—the magnitude does not matter. We can simplify Equation (1.13) by letting $v_j = \mathcal{A}^j v_0/\lambda_1^j$. Using this in Equation (1.13) we have

$$v_j = c_1 y_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^j y_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^j y_n. \tag{1.14}$$

Since λ_1 is larger than the other eigenvalues, the factors $\left(\lambda_i/\lambda_1\right)^j$ decrease as j increases and Equation (1.14) becomes

$$v_j = c_1 y_1 + \mathcal{O}\left(\frac{\lambda_2}{\lambda_1}\right)^j. \tag{1.15}$$

The power method generally requires many iterations before it has converged to the fundamental eigenvalue. We can see from Equation (1.15) that v_j approaches the fundamental eigenvector as $\left(\lambda_2/\lambda_1\right)^j$ goes to zero; the ratio λ_2/λ_1 is called the dominance ratio.

In Monte Carlo particle transport, while the eigenvalue is converging to the fundamental eigenvalue the results of the iterations are discarded. These iterations are called *inactive* iterations. Once the power method has converged to the fundamental eigenvalue, *active* iterations are begun. The eigenvalue estimate calculated during

each active iteration is stored. We can calculate the mean and standard deviation of the active eigenvalue estimates. The standard deviation can be used to estimate the statistical uncertainty in the estimate of the mean.

One major problem with the power method is the slow rate of convergence to the fundamental eigenvalue. When $\lambda_2 \approx \lambda_1$—as it often is for an optically thick system—then the dominance ratio is large (close to 1) and the power method converges slowly. When the dominance ratio is large, many inactive iterations must be performed and discarded. It is clear that needing fewer inactive iterations is computationally more efficient than when more inactive iterations are required. Unfortunately for the power method the number of inactive cycles required is dependent on the dominance ratio of the problem.

1.4 Alternatives to the Power Method

Krylov subspace alternatives to the power method exist, including the Lanczos and Arnoldi's method of minimized iterations[2]. The Lanczos method is Arnoldi's method applied to hermitian matrices. Arnoldi's method was chosen as the focus of this dissertation because it can estimate multiple eigenpairs of any linear operator.

Like the power method, Arnoldi's method only needs to know the application of the linear operator on a vector. The operator is still applied iteratively, but the resulting vectors are orthogonalized against all previously calculated basis vectors. Arnoldi's method can calculate multiple eigenmodes with little additional computational expense than is required for estimating the fundamental eigenmode.

This dissertation explores using Arnoldi's method instead of the power method for estimating eigenvalues and eigenvectors of the transport-fission operator $\mathcal{A}$. In Chapter 2, the basic Arnoldi's method is presented and implementation in a Monte Carlo particle transport code is descrived. Some new techniques used to transport un-

physical, but nevertheless important, negative sources are also defined. In Chapter 3, some challenges associated with the spatial discretization required for orthogonalization of the Krylov subspace vectors is discussed. In Chapter 4, a strategy used to reduce the computational expense of Arnoldi's method is shown. In Chapter 5, the convergence of the fission source and the accuracy of the estimated uncertainty using Arnoldi's method is discussed, both issues of current interest in the Monte Carlo particle transport community. Finally, in Chapter 6, this work is summarized and introduces research and enhancements to Monte Carlo Arnoldi's method that needs to be investigated.

Chapter 2

Arnoldi's Method for Monte Carlo Particle Transport

As described in the introduction, the power method has been the method of choice for calculating eigenvalues and eigenvectors for Monte Carlo criticality applications. The power method is suited well for this type of calculation because of its simple, matrix-free implementation.

The power method, however, has a few drawbacks. The rate of convergence to the fundamental eigenvalue—determined by the dominance ratio of the first higher-order eigenvalue to the fundamental eigenvalue—can be too small. It is also limited to calculating one eigenvalue at a time. Other Krylov subspace methods exist that have some of the same positive benefits of the power method, but also address the negative aspects that make the power method slow and limited. Chief among these is Arnoldi's method.

Arnoldi's method [2] is just one such method from the class of Krylov subspace methods. While its implementation is not as straightforward as with the power method, an explicit form of the transport-fission operator remains unnecessary; we only need to know how to apply the operator to a fission source. In this sense, Arnoldi's method is as equally suited to Monte Carlo criticality applications as the power method, but it has never been studied in this application. In this disserta-

tion, the first Monte Carlo application of Arnoldi's method for particle transport is demonstrated.

Arnoldi's method has advantages over the power method that may prove to make the difficulty in its implementation a small matter. One benefit Arnoldi's method brings to Monte Carlo particle transport is the ability to calculate multiple eigenvalues and eigenvectors with minimal extra computational expense over what is required to calculate the fundamental mode. In addition, the fission source converges faster in Arnoldi's method than in the power method, reducing the number of inactive iterations required. In this chapter the basic Arnoldi method and the Monte Carlo implementation for particle transport is introduced. The ability of Monte Carlo Arnoldi's method to calculate multiple eigenpairs of the fission-transport operator is also demonstrated. Arnoldi's superior performance in converging the eigenvalue and the fission source is shown in Chapter 5.

2.1 Arnoldi's Method

Arnoldi's method generates a Krylov subspace similar to Equation (1.7) except at each iteration the basis vectors $\{v_i\}_{i=1}^{m}$ are orthogonalized against all the previously calculated Arnoldi vectors and normalized. The basis vectors that form the Krylov subspace are called Arnoldi vectors [see 31, pp. 435-438].

An Arnoldi process begins with a normalized vector

$$v_1 = \frac{v}{\|v\|_2}, \tag{2.1}$$

where

$$\|v\|_2 = \left(\sum_{i=1}^{n} |v_i|^2 \right)^{1/2} \tag{2.2}$$

is the Euclidean norm. We then apply the linear operator to v_1 and orthogonalize the result against v_1, yielding

$$\tilde{v}_2 = \mathcal{A}v_1 - \langle \mathcal{A}v_1, v_1 \rangle v_1 = \mathcal{A}v_1 - h_{1,1}v_1. \tag{2.3}$$

where

$$h_{jm} = \langle \mathcal{A}v_m, v_j \rangle \tag{2.4}$$

is the inner product between the vectors v_m and v_j

$$\int v_m(x) v_j(x) \, \mathrm{d}x.$$

Then we normalize $\tilde{v}_2$

$$v_2 = \frac{\tilde{v}_2}{\|\tilde{v}_2\|_2} = \frac{\tilde{v}_2}{h_{2,1}}. \tag{2.5}$$

This process continues iteratively; at the m-th iteration we have

$$\tilde{v}_{m+1} = \mathcal{A}v_m - \sum_{j=1}^{m} h_{jm} v_j \qquad \text{(Orthogonalization)} \tag{2.6a}$$

$$v_{m+1} = \frac{\tilde{v}_{m+1}}{h_{m,m+1}} \qquad \text{(Normalization)}. \tag{2.6b}$$

The process of orthogonalization and normalization involves the calculation of the values h_{jm}. These values are the elements of an upper-Hessenberg matrix, $H_{m+1,m}$. (An upper Hessenberg matrix is upper triangular except that the first subdiagonal is non-zero.) We can write the m-th Arnoldi iteration in matrix form as

$$\mathcal{A}V_m = V_{m+1} H_{m+1,m}, \tag{2.7}$$

where the columns of V_m are the Arnoldi vectors and the elements of $H_{m+1,m}$ are the results of the inner products of Arnoldi vectors as described in Equation (2.4).

If we separate the last column of V_{m+1} and the last row of $H_{m+1,m}$ we obtain from Equation (2.7)

$$\mathcal{A}V_m = V_m H_m + v_{m+1} h_{m+1,m} e_m^T \tag{2.8}$$

where e_m is the m-th standard basis vector, v_{m+1} is the Arnoldi vector calculated during the m-th Arnoldi iteration and $h_{m+1,m}$ is the normalization factor for the new Arnoldi vector. Thus we see that at the m-th Arnoldi iteration we add a row and a column to $V_m H_m$.

Equation (2.8) is called the *Arnoldi factorization* and is an important equation in further analysis of Arnoldi's method.

2.1.1 Finding Ritz Pairs from Arnoldi Factorization

The Arnoldi process generates an upper-Hessenberg matrix H_m which is the projection of $\mathcal{A}$ onto the Krylov subspace defined by the Arnoldi vectors. Since H_m is generated after a small number of iterations its size is small and we can find its eigenvalues and eigenvectors with relative ease. The eigenpairs of H_m, $\left(\mu_i, x_i\right)$ can be used to find Ritz pairs—approximate eigenpairs—of $\mathcal{A}$. To see this multiply the Arnoldi factorization, Equation (2.8) on the right by an eigenvector of H_m, x_i

$$\begin{aligned}
\mathcal{A}V_m x_i &= V_m \left(H_m x_i\right) + v_{m+1} h_{m+1,m} e_m^T x_i \\
\mathcal{A}V_m x_i &= V_m \left(\mu_i x_i\right) + v_{m+1} h_{m+1,m} e_m^T x_i \\
\mathcal{A}y_i &= \mu_i y_i + v_{m+1} h_{m+1,m} e_m^T x_i,
\end{aligned} \tag{2.9}$$

where $y_i = V_m x_i$. $\left(\mu_i, y_i\right)$ is a Ritz pair of $\mathcal{A}$ or an approximation to an eigenpair of $\mathcal{A}$. The Ritz vector y_i is just the product of the Arnoldi vectors and an eigenvector of H_m.

The residual is defined as

$$r_m \equiv \mathcal{A}y - \mu y. \tag{2.10}$$

We can see from Equation (2.9) that the magnitude of the residual is just the magnitude of the last element of the eigenvector x_i times $h_{m+1,m}$

$$|r_m| = \left\|\mathcal{A}y - \mu y\right\| = \left|h_{m+1,m}\right| |e_m x_i| \,. \tag{2.11}$$

It is easy to see that if the residual is zero, then the Ritz pair is an eigenpair of $\mathcal{A}$. The residual is therefore a good indication—but not a guarantee—of the quality of the eigenpair approximation; in fact a small residual guarantees that $\big(\mu, x\big)$ is an exact eigenpair of a matrix close to $\mathcal{A}$ [see 31].

2.1.2 Explicitly Restarted Arnoldi

As Arnoldi's method proceeds, each iteration adds one Arnoldi vector, and the size of Krylov subspace expands. The increase in the number of Arnoldi vectors and the size of the Krylov subspace is problematic. First, the memory requirements increase and second, it is computationally more expensive because there are more vectors that the newest Arnoldi vector must be orthogonalized against.

Arnoldi's method begins with an estimate of the fundamental eigenvector After a few Arnoldi iterations we have a better estimate of the fundamental eigenvector than what we started with. We can therefore restart Arnoldi's method using the better estimate of the eigenvector as the starting vector for the Arnoldi process. The idea of starting Arnoldi's method using the results of several previous iterations of Arnoldi's method is known as Restarted Arnoldi's method (RAM) and was first proposed by Saad [21].

Arnoldi's method can be restarted repeatedly, each time starting with an improved starting vector that is a linear combination of the estimates of the desired eigenvectors of $\mathcal{A}$. Each sequence of several iterations and a calculation of the Ritz pairs of $\mathcal{A}$ is called a *restart*. Restarting Arnoldi's method saves computational expense by

reducing the number of Arnoldi vectors that we must orthogonalize against and reduces memory requirements as fewer Arnoldi vectors must be stored. Each iteration adds one basis vector to the Krylov subspace so the size of the Krylov subspace is the same as the number Arnoldi vectors. The size, m, of the Krylov subspace is also the size of the upper-Hessenberg matrix $H_m \in \mathbb{R}^{m \times m}$ and is therefore the number of eigenpairs that can be estimated in one restart. Although not necessary, the number of iterations in each restart is generally the same.

The estimated eigenvectors of $\mathcal{A}$ form a basis that we can use in a linear expansion to represent a vector,

$$\hat{v} = c_1 y_1 + c_2 y_2 + \cdots + c_n y_n, \tag{2.12}$$

where the y_i's are the estimated eigenvectors of the linear operator $\mathcal{A}$ and the c_i's are some expansion coefficients. If the j eigenvalues largest in magnitude are desired, then we would like to have our initial vector be

$$\hat{v} = c_1 y_1 + \cdots + c_j y_j + 0\, y_{j+1} + \cdots + 0\, y_n; \tag{2.13}$$

i.e., we want to suppress any information from the undesired portion of the spectrum of $\mathcal{A}$.

The initial vector will most likely contain significant components of all the eigenvectors. We can reduce the components from eigenvectors from the undesired region of the spectrum by forcing $c_{j+1} = \cdots = c_n = 0$. The initial vector then becomes

$$\hat{v} = c_1 y_1 + \cdots + c_j y_j. \tag{2.14}$$

The value of the coefficients $c_1, c_2, \ldots, c_j$ are not important and are conveniently chosen to be 1. When Arnoldi's method is restarted using $\hat{v}$ from Equation (2.14) as the initial vector for a new Arnoldi restart, it is an explicit restart and will be

referred to as *explicitly* restarted Arnoldi's method (ERAM). It is implemented by only summing the vectors $x_1 + \cdots + x_j$ and ignoring the other eigenvectors.

2.2 Monte Carlo Implementation of Explicitly Restarted Arnoldi's Method

Now that the basic restarted Arnoldi's method has been described, we can proceed to show how it can be implemented in a Monte Carlo particle transport application. First we note that eigenvalue estimates are made at the end of every restart. Eigenvalue estimates could have been made at every iteration, but that takes additional computational expense. A choice was made to calculate and store the eigenvalue estimates at the end of every restart to reduce this cost. The mean and standard deviation of these stored eigenvalue estimates can be calculated. In this sense, an Arnoldi restart is similar to a power method iteration.

In conjunction with the eigenvalues, the eigenvectors of the transport-fission operator are estimated at the end of each restart. The eigenvectors (y_i) are normalized such that

$$\int y_i(x)^2 \, \mathrm{d}x = 1 \tag{2.15}$$

and stored. The mean of the eigenvectors can also be calculated, but we must be careful that we sum appropriate values of the eigenvectors. For example, suppose the fundamental eigenvector estimate from one restart has the opposite sign as the fundamental eigenvector estimate from another restart. (Eigenvectors are unique only up to a multiplicative constant so both estimates are valid.) Prior to adding a new eigenvector estimate when calculating the mean, the dot product of the new eigenvector estimate with the previous eigenvector estimate is calculated. If the dot product is negative, the new eigenvector is multiplied by -1. This preserves the normalization given in Equation (2.15).

The application of the linear operator in Equation (2.6a) was described in Section 1.1. In brief, neutrons are sampled from the fission source v_k, transported and the position of the neutrons at the time they initiate fission is stored, creating a new fission source, $\tilde{v}_{k+1}$.

2.2.1 Negative Sources

When using the power method to calculate the fundamental eigenmode for criticality calculations, we are guaranteed that the solution is everywhere positive and the vectors that are sampled from at each iteration are also everywhere positive. With Arnoldi's method we are not so fortunate. The process of orthogonalization guarantees that some of the Arnoldi vectors will be partially negative. This presents two challenges to Monte Carlo particle transport: first, a negative source means a negative flux which is not physical and second, to sample from a source it must be assumed to be everywhere positive.

To sample from a distribution, it must be everywhere positive and integrate to 1. If $p(x)$ is a probability distribution function, the quantity $p(x)\,\mathrm{d}x$ is the probability of choosing a point in $\mathrm{d}x$ about x. For a fission source $v(x)$ which may have negative regions it is first normalized such that

$$\int \left|v(x)\right| \mathrm{d}x = q \tag{2.16a}$$

$$p(x) = \frac{\left|v(x)\right|}{q}. \tag{2.16b}$$

With this normalization, the quantity $p(x)\,\mathrm{d}x$ is the probability of choosing a point in $\mathrm{d}x$ about x. A neutron position x_s is sampled from $p(x)$ and is given an initial weight of

$$\omega = \frac{v(x_s)}{\left|v(x_s)\right|}, \tag{2.17}$$

or, alternatively

$$\omega = \begin{cases} 1, & v(x_s) > 0 \\ -1, & v(x_s) < 1. \end{cases} \tag{2.18}$$

Neutrons sampled where $v(x_s) < 0$ reduce the tally where they score; neutrons sampled where $v(x_s) > 0$ contribute positively to the tally where they score. A neutron is never sampled where $v(x_s) = 0$ because there the probability of choosing this point is exactly zero.

Once a particle has been sampled, Monte Carlo transport proceeds as usual, with the neutron scoring $\omega \left(\nu\Sigma_f / \Sigma_T \right)$ in the proper bin at each collision. If non-analog Monte Carlo is being done, particle weight is reduced at each collision and Russian Roulette is played if the absolute value of the weight becomes small. Giving a neutron a signed weight does not interfere with any variance reduction or tallying techniques.

At alternative approach to applying $\mathcal{A}$ to $v(x)$ this way is to separate $v(x)$ into its positive and negative parts, $v^{(+)}(x)$ and $v^{(-)}(x)$ respectively, and apply the transport-fission operator to each part independently

$$\mathcal{A}v(x) = \mathcal{A}v^{(+)}(x) - \mathcal{A}\left|v^{(-)}(x)\right|. \tag{2.19}$$

The first approach does this directly by assigning the weights as described. The first approach is also superior because it samples the positive and negative parts proportionately to the magnitude of the positive or negative part; the second approach would use the same number of particles to apply the linear operator for both the positive and negative part, even if one part is significantly larger than the other.

Once the sampling of a fission source, transporting particles, and creation of a new fission source is completed the new fission source is normalized per source particle

and multiplied by integral of the absolute value of the previous source

$$v_{j+1}(x) = \mathcal{A}v_j(x)\frac{1}{N}\int |v_j(x)|\,\mathrm{d}x, \tag{2.20}$$

where $v_j(x)$ is the source we sample from and $\mathcal{A}v(x)$ is the new source created after sampling from $v_j(x)$ and transporting. This scaling is similar to the scaling performed in the power method shown in Equation (1.8) and Equation (1.9). After the vector has been properly scaled, Arnoldi's method continues with orthogonalization and normalization of the newest Arnoldi vector.

2.2.2 Spatial Discretization

The orthogonalization and normalization of the basis vectors in Arnoldi's method requires taking the inner product of two vectors

$$\langle v_j, v_k \rangle = \int v_j(x)v_k(x)\,\mathrm{d}x. \tag{2.21}$$

In Monte Carlo Arnoldi's method we must take the inner product of two fission sources. To do this, the fission source is represented as a linear combination of piecewise constant functions

$$v_\Pi(x) = \sum_{b=1}^{B} a_b \Pi_b(x), \tag{2.22}$$

where B is the number of spatial bins and

$$\Pi_b(x) = \begin{cases} \left(\frac{1}{\Delta x_b}\right)^{1/2}, & x_b \leq x < x_{b+1} \\ 0, & \text{otherwise,} \end{cases} \tag{2.23}$$

where $\Delta x_b = \big(x_{b+1} - x_b\big)$ is the width of bin b. The term $a_b\sqrt{\Delta x_b}$ is the number of fission neutrons generated in the spatial bin b, $x \in \big[x_b, x_{b+1}\big)$. Note that, to sample from $v_\Pi(x)$, we first sample a bin and then sample uniformly within the bin to determine the position of the particle.

Representing the fission source with a piecewise-constant-in-space approximation, the elements of the Arnoldi vectors are just the expansion coefficients

$$v_\Pi = \big[a_1, a_2, \ldots, a_B\big]^T, \tag{2.24}$$

and the inner product between two fission sources is defined as

$$\langle v_\Pi^{(j)}, v_\Pi^{(k)} \rangle = \sum_{b=1}^{B} a_b^{(j)} a_b^{(k)}, \tag{2.25}$$

where $a_b^{(j)}$ and $a_b^{(k)}$ are the expansion coefficients from the fission sources $v_\Pi^{(j)}$ and $v_\Pi^{(k)}$ respectively.

Applying $\mathcal{A}$ to a vector of coefficients $\{a_b\}_{b=1}^B$ simply requires sampling the piecewise constant source function $v_\Pi(x)$ in Equation (2.22), transporting these neutrons until they cause another fission, and tallying the resulting fission neutrons over the bins [see 7]. This generates a truncation error.

With this representation of the fission source, the sampling techniques described in Section 2.2.1 can be used and the inner product between two fission sources have been defined. To estimate the mean eigenvector, we can simply calculate the mean value of the coefficient in each bin.

2.3 Numerical Results

Everything necessary for a Monte Carlo application of explicitly restarted Arnoldi's method has been described. To demonstrate how Monte Carlo Arnoldi's method can

calculate multiple eigenvalues and eigenvectors of the transport-fission operatorm a few simulations of a homogeneous, semi-infinite slab of multiplying material are shown. The simulations shown here were chosen to match results published by [12, 20] and [9]. The cross sections are: $\nu\Sigma_f = 1.0$, $\Sigma_a = 0.2$, $\Sigma_s = 0.8$ with $\Sigma_t = 1.0$, thus the mean free path for this geometry is $1/\Sigma_t = 1.0$. I will show the results of slabs with width 0.2, 2.0 or 20 mfp.

For each slab width I have run two simulations; one simulation using Arnoldi's method and the other simulation using the power method for comparison. In every simulation 10^5 particles are tracked in an iteration. The power method has 250 inactive and 1000 active iterations while Arnoldi's method has 25 inactive and 100 active restarts with 10 iterations in each restart. Therefore the Krylov subspace size is 10. The number of inactive and active iterations in each method is the same. The total number of particles tracked in each simulation is also the same. The slab is discretized into 50 spatial bins for the 0.2 mfp problem and 75 spatial bins for the 2.0 and 20 mfp problems.

The results of these simulations are shown in Table 2.1, along with the dominance ratio (DR) for each of the three slab widths. The fundamental eigenvalue estimates are shown for both the power method and Arnoldi's as well as the first and second harmonic eigenvalue estimates from Arnoldi's method. The published results from [12, 20] and [9] are given as the reference. Almost all the eigenvalue estimates are within one standard deviation of the reference solution. The only exceptions are the second harmonic estimates for the 2.0 and 20 mfp thick problems and they are both within two standard deviations of the reference solution.

The figure of merit (FOM) and computational time for each simulation is given in Table 2.2. The figure of merit is a measure of the efficiency of a Monte Carlo calculation. The variance of a Monte Carlo eigenvalue calculation goes as $\sqrt{1/N}$, where N is the number of eigenvalue estimates. The computational expense should

Width (mfp)	Method	Eigenvalue	Standard Deviation	Reference	Error
0.2	Power	0.329979	6.3×10^{-5}	0.330000	2.1×10^{-5}
	Arnoldi	0.33008	1.8×10^{-4}	0.33000	8.3×10^{-5}
DR = 0.23997		0.07911	1.5×10^{-4}	0.07919	7.6×10^{-5}
		0.04493	1.6×10^{-4}	0.04499	5.8×10^{-5}
2.0	Power	2.09593	2.7×10^{-4}	2.09599	6.0×10^{-5}
	Arnoldi	2.09652	6.9×10^{-4}	2.09599	5.3×10^{-4}
DR = 0.4015		0.84183	5.8×10^{-4}	0.84150	3.3×10^{-4}
		0.48279	4.5×10^{-4}	0.48230	4.9×10^{-4}
20	Power	4.82734	6.3×10^{-4}	4.82780	4.6×10^{-4}
	Arnoldi	4.8290	1.5×10^{-3}	4.8278	1.2×10^{-3}
DR = 0.9079		4.3827	1.4×10^{-3}	4.3831	4.2×10^{-4}
		3.8152	1.4×10^{-3}	3.8174	2.2×10^{-3}

Table 2.1: Eigenvalue estimates from power method and Arnoldi's method for slab geometries of width 0.2, 2.0 and 20 mfp. Reference values taken from [12], and [9].

be directly proportional to the number of eigenvalue estimates. The figure of merit is therefore defined to be

$$\text{FOM} \equiv \frac{1}{\sigma^2 T} \tag{2.26}$$

where σ^2 is the variance and T is the time required to perform the Monte Carlo calculation.

Width (mfp)	Method	Fundamental Eigenvalue	Standard Deviation	FOM	Time (sec)
0.2	Power	0.329979	6.3×10^{-5}	1.7×10^{6}	149.0
	Arnoldi	0.33008	1.8×10^{-4}	3.3×10^{5}	95.3
2.0	Power	2.09593	2.7×10^{-4}	5.5×10^{4}	258.1
	Arnoldi	2.09652	6.9×10^{-4}	9.8×10^{3}	212.5
20	Power	4.82734	6.3×10^{-4}	5.4×10^{3}	463.0
	Arnoldi	4.8290	1.5×10^{-3}	1.1×10^{3}	378.5

Table 2.2: Eigenvalue estimates, figure of merit for fundamental eigenvalue from the power method and Arnoldi's method for slab geometries of width 0.2, 2.0, and 20 mfp.

We see that the figure of merit is always larger for the power method than for Arnoldi's method, but that the Arnoldi simulations run faster. The figure of merit for the power method is larger than the figure of merit for Arnoldi's method by a factor of 5 while the computational time for the power method is 1.6 times larger than Arnoldi's method for the 0.2 mfp simulation and 1.2 times larger than Arnoldi's method for the 2.0 and 20 mfp simulations.

The power method has many (10 times for these simulations) more eigenvalue estimates because it calculates an estimate after every iteration while Arnoldi's method calculates an estimate after each restart consisting of many iterations. Thus for the same number of particles tracked (computational expense, T) the power method has many more eigenvalue estimates and therefore its variance is smaller, and the FOM is larger.

We can calculate the spread of the eigenvalue estimates from each method. The spread is the root mean sqared difference of the eigenvalue estimates, x_i, from $\overline{x}$, the mean eigenvalue estimate;

$$s \equiv \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}.$$

This should not be confused with the *population* standard deviation

$$\sigma \equiv \sqrt{\frac{1}{N-1}\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2\right)}$$

which is called simply the standard deviation in this dissertation. The spread of the fundamental eigenvalue estimates from the active iterations/restarts for the three slab widths is shown in Table 2.3. We can see that even though the standard deviation of the mean is larger in Arnoldi's method than in the power method, the spread of the eigenvalue estimates is smaller in Arnold's method than in the power method. From

this we can conclude that the estimate of the standard deviation (and thus the figure of merit) is dominated by the number of eigenvalue estimates.

The figure of merit isn't a complete comparison between these two methods. It only measures the efficiency of estimating one eigenvalue, but in Arnoldi's method we have estimates of the first three eigenvalues at no additional cost. Furthermore, because we are electing to compare multiple eigenvalues, we have fewer eigenvalue estimates to average together in Arnoldi's method.

	0.2 mfp	2.0 mfp	20 mfp
Power	0.0020	0.0084	0.0201
Arnoldi	0.0018	0.0069	0.0153

Table 2.3: Spread of active fundamental eigenvalue estimates from the power method and Arnoldi's method for slab geometries of width 0.2, 2.0, and 20 mfp.

Graphical results for the 20 mfp thick slab are shown in figures 2.1–2.2. In Figure 2.1 we see the eigenvalue convergence for the fundamental and first two harmonic eigenvalue estimates. Both inactive and active iterations are shown; the active iterations are the running average of the active eigenvalue estimates. We can see that the spread of the estimates of the fundamental eigenvalue from Arnoldi's method is smaller than the estimates from the power method. Black lines are drawn indicating the reference values published in [12] and [9]. It appears that all three eigenvalue estimates agree with the reference solution. However we know from Table 2.1 that the estimate of the second harmonic is just outside of one standard deviation.

Figure 2.2 shows the fundamental eigenvector estimate from the power method and Arnoldi's method as well as a reference solution from an S_N code. We see that both the power method and Arnoldi's method accurately estimate the fundamental eigenmode. Figure 2.3 shows the fundamental eigenvector and the first two harmonics all calculated by Arnoldi's method. The higher-order eigenmodes of a semi-infinite

slab are similar to the higher modes of the cosine function as expected [see 10, pg. 173].

The results of the 0.2 and 2.0 mfp simulations are shown in Figures 2.4a–2.5b. The results are similar to what we have seen for the 20 mfp problem.

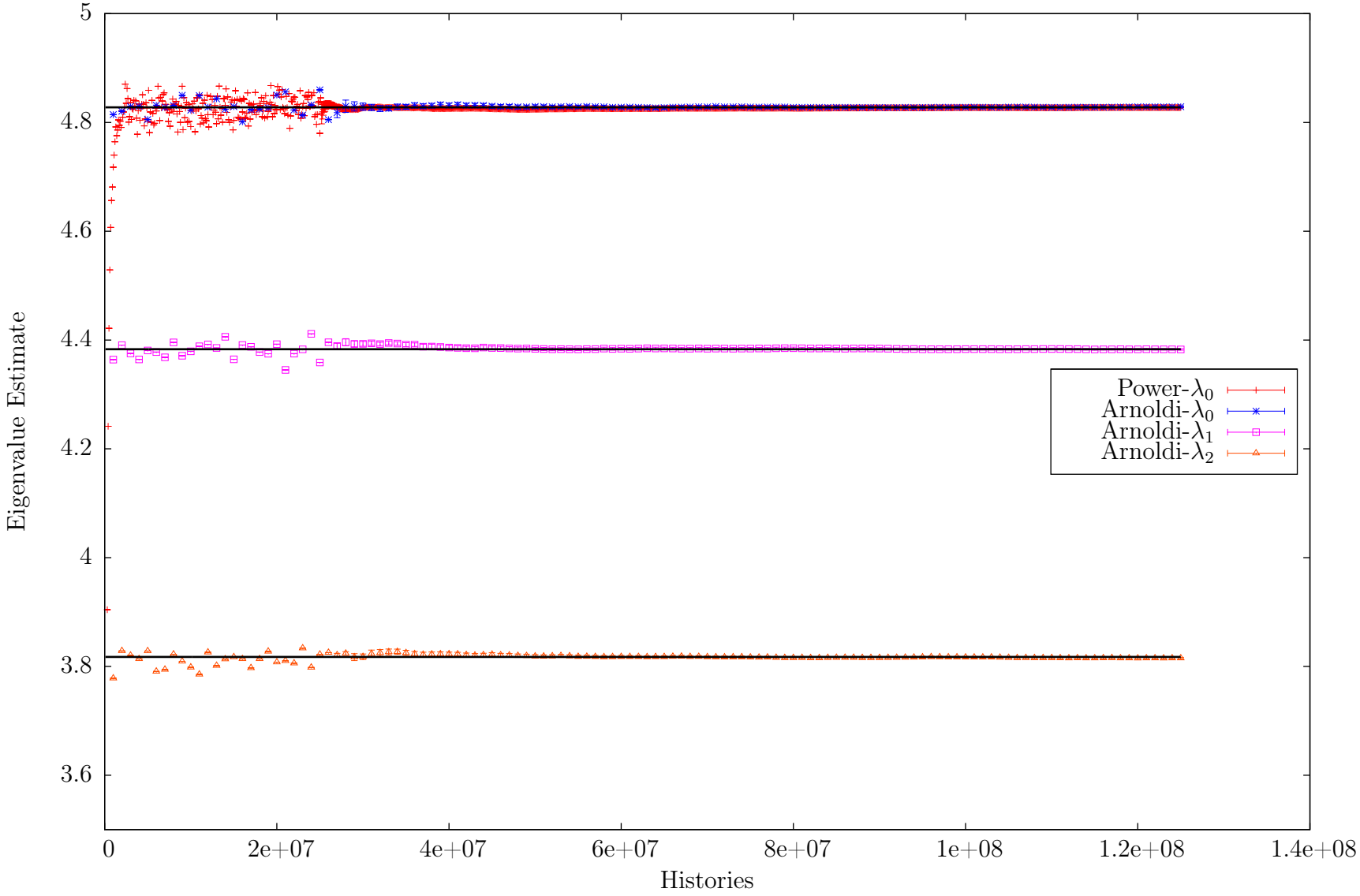


Figure 2.1: Eigenvalue estimates for the power method and Arnoldi's method for the 20 mfp thick slab. The heavy lines indicate the reference eigenvalues.

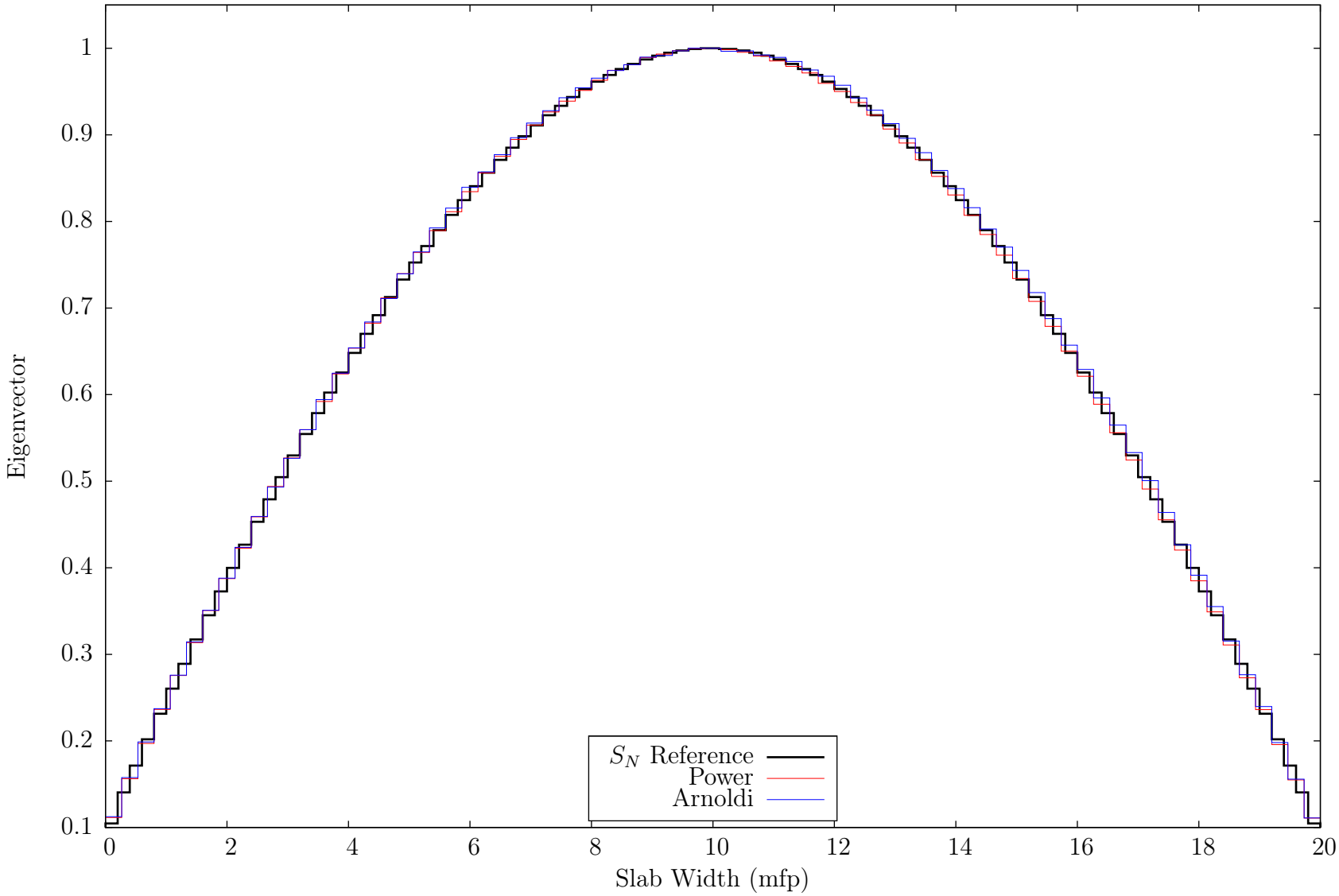


Figure 2.2: Fundamental eigenvector estimates from the power method and Arnoldi's method for the 20 mfp thick slab. The heavy line shows the S_N solution.

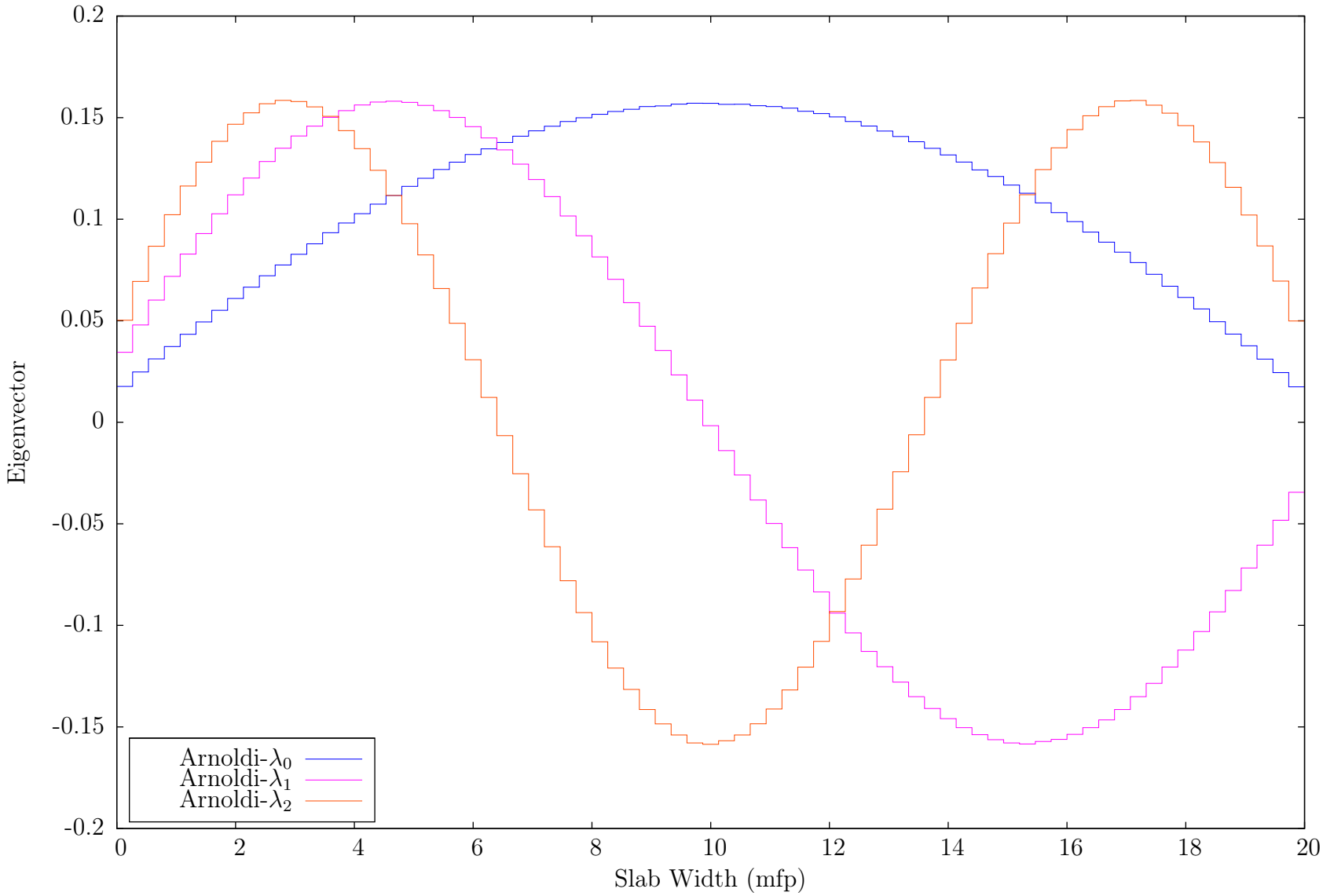
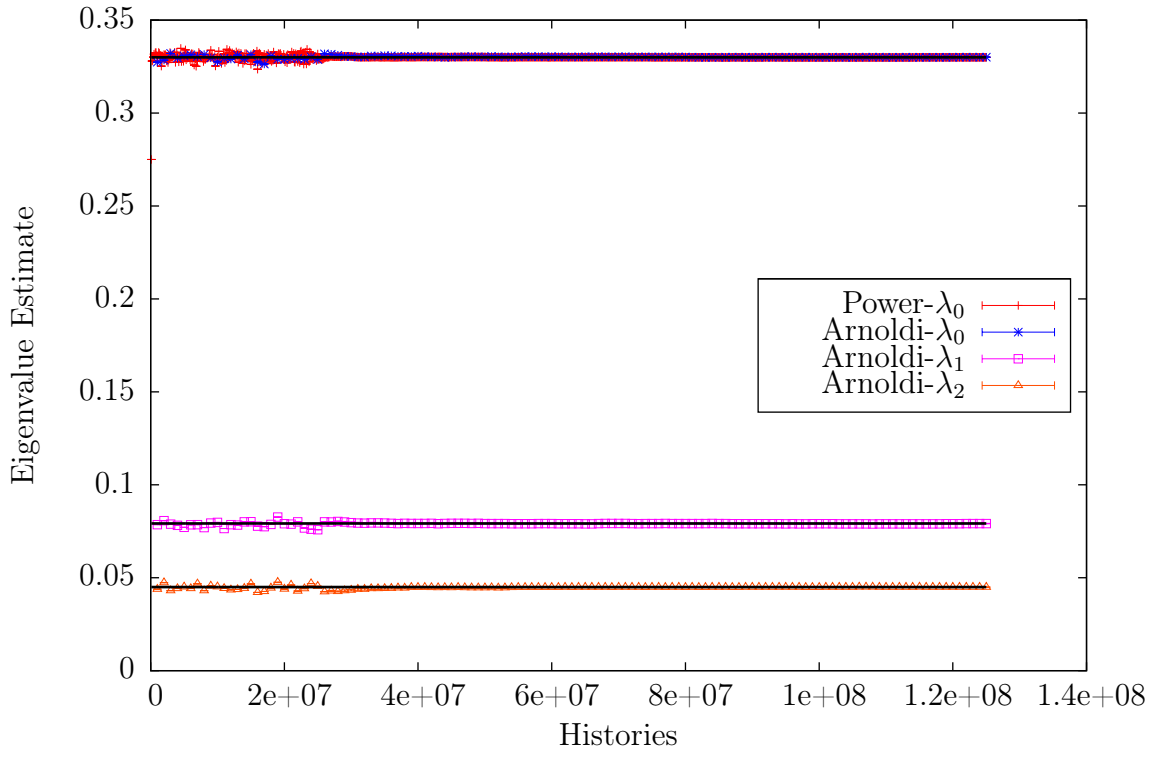
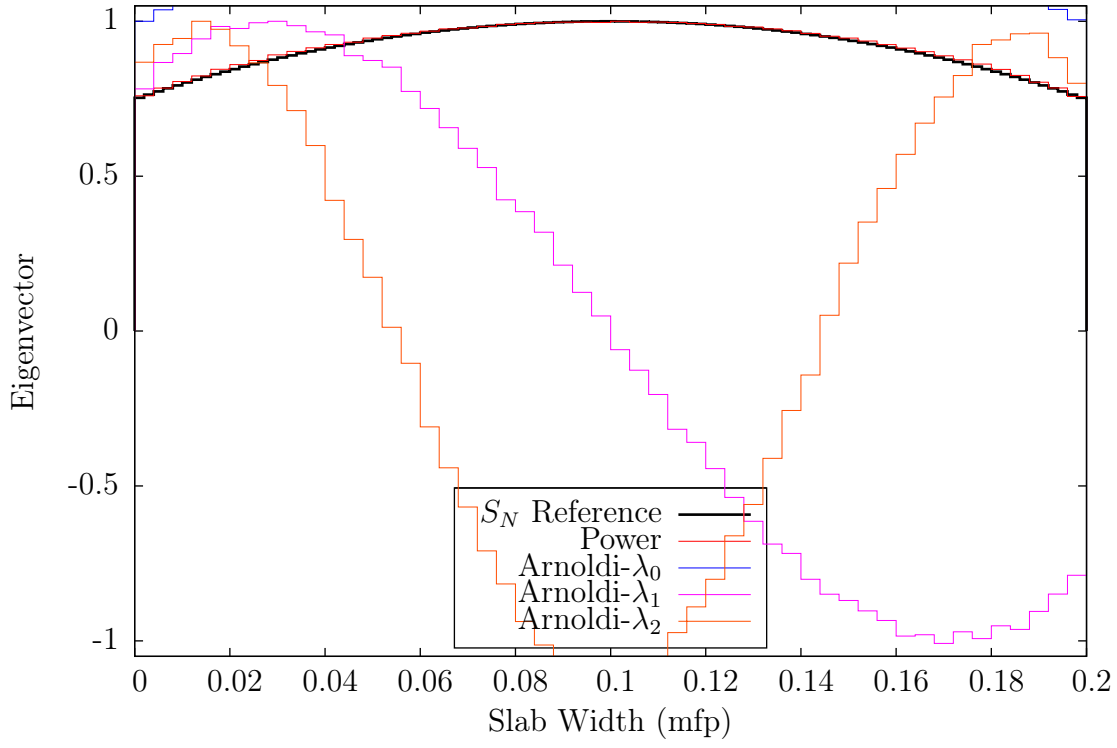


Figure 2.3: Fundamental and first and second harmonic eigenvector estimates from Arnoldi's method for the 20 mfp thick slab.

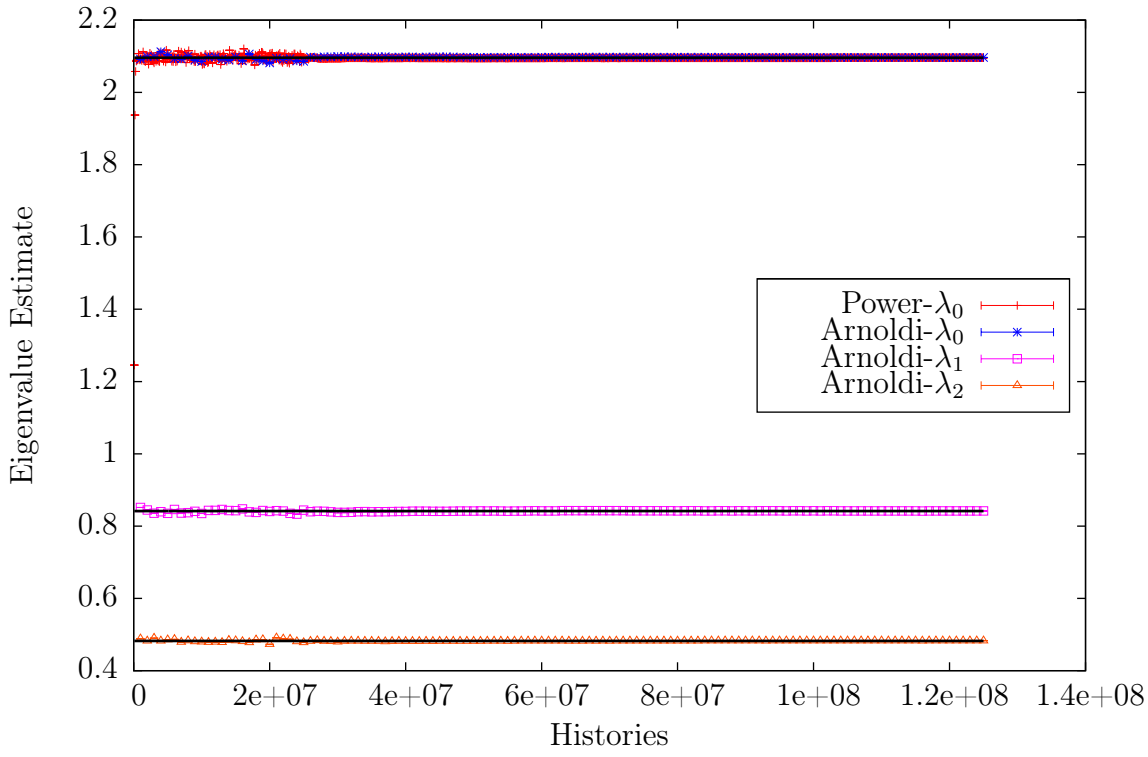


(a) Eigenvalue estimates

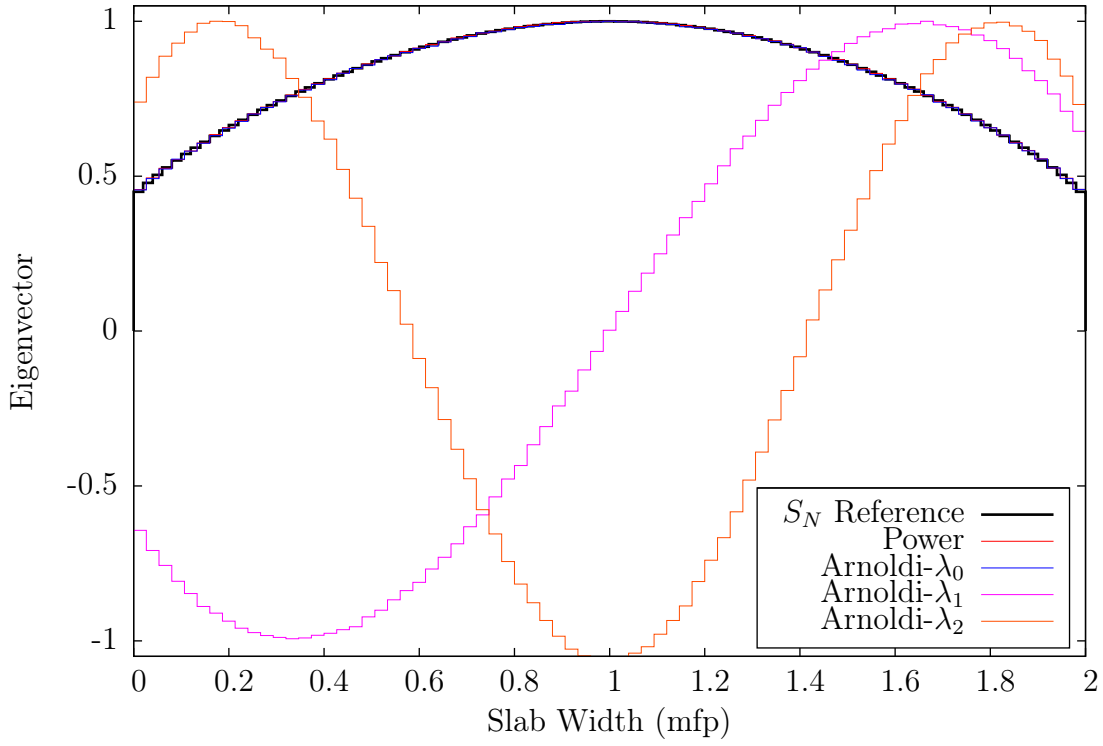


(b) Eigenvector Estimates

Figure 2.4: Eigenvalue and eigenvector estimates from power method and Arnoldi's method for the 0.2 mfp thick slab. Heavy lines show reference solution from [12] and [9].



(a) Eigenvalue estimates



(b) Eigenvector estimates

Figure 2.5: Eigenvalue and eigenvector estimates from power method and Arnoldi's method for the 2.0 mfp thick slab. Heavy lines show reference solution from [12] and [9].

2.3.1 Discretization Error

One of the benefits of Monte Carlo particle transport is the ability to use exact geometry without discretization. This is true for the transport of particles in the power method, but the fission source must be discretized for tallying. Arnoldi's method, on the other hand, must have a discretized source in order to orthogonalize the Arnoldi vectors, as described in Section 2.2.2.

The discretization of the fission source can lead to an error in the estimated eigenvalue if an insufficient number of spatial bins are used to represent the fission source. To illustrate this effect a series of simulations is shown using the same slab of multiplying material with varying number of spatial bins. The slab here is exactly the same as for the 20 mfp problem shown earlier ($\nu\Sigma_f = 1.0,\ \Sigma_a = 0.2,\ \Sigma_s = 0.8$ with $\Sigma_t = 1.0$). This time 10^5 histories are tracked per iteration with 50 inactive restarts and 500 active restarts. The increase in the number of histories and restarts is to reduce the statistical uncertainty to ensure the error can be seen outside the noise. This simulation was performed eleven times varying the number of spatial discretization bins from 10 to 150.

The results of these simulations are shown in Table 2.4 for the fundamental eigenvalue. We can see that the uncertainty in the eigenvalue estimate (standard deviation) is relatively independent of the number of spatial bins. The error in the eigenvalue estimate is the absolute value of the difference between the eigenvalue estimate and the reference solution. We see that the error in the eigenvalue estimate is larger than the statistical uncertainty for bin widths ≤ 0.5 mfp thick. For bin widths greater than 0.5 mfp thick the error is less than the statistical uncertainty.

The data from Table 2.4 is shown graphically in Figure 2.6. The error in the eigenvalue estimate for the first two harmonics are also shown in Figure 2.6. The error in the eigenvalue estimate is denoted marked as $\mathcal{B}$ for each of the eigenvalue

estimates. Best fit lines are drawn through the points on the graph. We see that there is a very good linear fit to these data points.

# of Bins	Bin Width (mfp)	Eigenvalue	Uncertainty	Error	FOM
10	2.00	4.8003	6.6×10^{-4}	2.7×10^{-2}	831.2
25	0.80	4.8224	6.8×10^{-4}	5.3×10^{-3}	773.2
40	0.50	4.8251	6.3×10^{-4}	2.6×10^{-3}	872.0
50	0.40	4.8273	6.5×10^{-4}	4.2×10^{-4}	829.0
60	0.33	4.8258	6.9×10^{-4}	2.0×10^{-3}	704.3
75	0.27	4.8275	6.7×10^{-4}	2.4×10^{-4}	753.0
90	0.22	4.8277	6.7×10^{-4}	4.2×10^{-5}	746.1
105	0.19	4.8277	6.9×10^{-4}	5.2×10^{-5}	698.3
120	0.17	4.8282	6.5×10^{-4}	4.1×10^{-4}	767.8
135	0.15	4.8274	7.0×10^{-4}	3.1×10^{-4}	656.9
150	0.13	4.8285	6.4×10^{-4}	7.5×10^{-4}	792.0

Table 2.4: Eigenvalue estimates of the fundamental eigenvalue from Arnoldi's method and the error in the estimate. The error is the difference between the estimate and the reference value ($\lambda_0 = 4.8278$, see [12] and [9]).

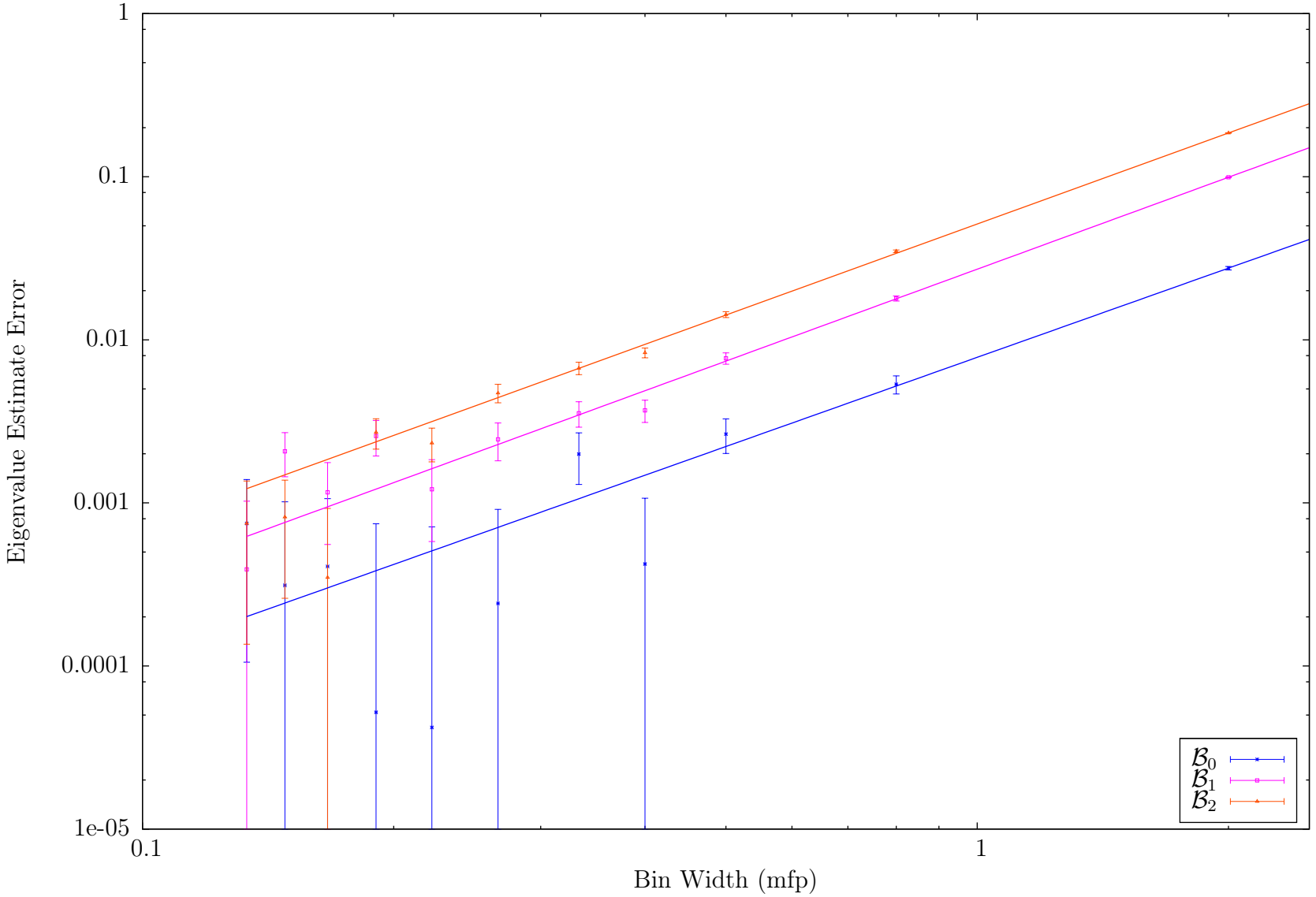


Figure 2.6: Discretization error for Arnoldi's method. The error is the difference between the eigenvalue estimate from Arnoldi's method and the reference value given in Table 2.1.

2.4 Variance

One of the problems with Monte Carlo particle transport is the underestimation of the variance of the mean eigenvalue estimate. This topic has received considerable attention lately [6]. In this section I will investigate how this issue manifests itself in Arnoldi's method.

The process of using the fission source calculated in a previous iteration as the source of neutrons for the current iteration causes the uncertainty in the eigenvalue (or some other tally) to be too small. The mean $\overline{X}$ and standard deviation $\sigma_{\overline{X}}$ for some tally X are calculated as

$$\overline{X} = \frac{1}{N}\sum_{n=1}^{N} X_n, \tag{2.27a}$$

$$\sigma_{\overline{X}} = \left[\frac{1}{N-1}\left(\frac{1}{N}\sum_{n=1}^{N} X_n^2\right) - \overline{X}^2\right]^{1/2}, \tag{2.27b}$$

where X_n is one estimate of the tally and N is the number of estimates. Equations (2.27) assume that each estimate is independent of all the others. Because of the procedure of using previous sources to generate the next source, the sources are correlated. Kiedrowski and Brown [16] explain it best, "If the concentration of fission neutrons at a location within a cycle [iteration] is statistically high, the concentration of fission neutrons in the next cycle is likely to be higher than average as well. The same applies if the concentration is statistically low. This implies a positive correlation between the fission source distributions."

Using Equation (2.27b) to estimate the standard deviation with correlated estimates causes the calculated standard deviation to be too small [see 6]. A standard deviation that is too small would give immoderate confidence in the eigenvalue.

2.4.1 Numerical Results

While there is no immediate way to reduce or eliminate the correlation between iterations, we can calculate the true standard deviation by running many independent, identical simulations and compute the mean and standard deviation of all of these. This can then be compared to the standard deviation of an individual run.

For this calculation a 50 mfp thick, homogeneous slab is used with cross sections: $\nu\Sigma_f = 1.0,\ \Sigma_a = 0.2,$ and $\Sigma_s = 0.8;\ \Sigma_t = 1.0$. The fundamental eigenvalue for this geometry is 0.997520. Both the power method and Arnoldi's method are run so as to compare the results. In Arnoldi's method, 100 inactive and 100 active restarts are used with 25 iterations per restart. For the power method 2500 inactive and 2500 active iterations are used. Both methods track 500,000 particles per iteration. Each method has 100 independent simulations.

The results of this study are shown in Table 2.5. I show the mean of the eigenvalue estimate from the 100 simulations, the mean of the standard deviations from the simulations and the true standard deviation. The true standard deviation is the standard deviation of eigenvalue estimates from all the simulations, while the mean reported standard deviation is the mean of the reported standard deviations from the simulations.

Method	Mean Eigenvalue	Mean Reported Standard Deviation	True Standard Deviation	Percent Difference
Power	0.99752	2.4×10^{-5}	2.7×10^{-5}	-11.1
Arnoldi	0.9974	1.1×10^{-4}	9.7×10^{-5}	13.4

Table 2.5: Mean eigenvalue estimate of fundamental eigenvalue from Arnoldi's method and the power method from 100 independent simulations. The mean reported standard deviation is the mean of the standard deviation from the 100 independent simulations. The true standard deviation is the standard deviation of the eigenvalue estimates from the 100 independent simulations. The difference is (Reported-True)/True.

We see from these results that both Arnoldi's method and the power method report a standard deviation that is different from the true standard deviation by about

10%. The problem is that the power method underpredicts the standard deviation. For Arnoldi's method we can have confidence that—at least for this problem—the reported standard deviation is larger than true standard deviation.

2.5 Summary

In this chapter the basic explicitly restarted Arnoldi's method for Monte Carlo particle transport has been described. It has been shown that Monte Carlo Arnoldi's method estimates the fundamental eigenvalue as well as first two higher-order eigenmodes within statistical uncertainty of published results; the eigenvectors are similar to cosine functions as they are expected to be.

Arnoldi's method suffers from two problems; the discretization of the fission source causes an error in the estimate of the eigenvalue, and a smaller figure of merit than the power method. These issues are addressed in Chapter 3 and Chapter 4 respectively.

Chapter 3

Spatial Discretization

In Chapter 2 necessity of spatially discretizing the fission source in order to take the inner product was introduced. The inner product is necessary for orthogonalizing and normalizing the Arnoldi vectors. The discretization causes an error in the eigenvalue estimate if the discretization is too coarse.

The effect of discretization was demonstrated in Section 2.3 and in Figure 2.6 the error in the eigenvalue estimate is shown as a function of the spatial bin width. We see from this figure the importance of having a sufficient number of spatial bins to eliminate discretization errors. Using a large number of spatial bins will remove the error in the eigenvalue estimate associated with discretization, but can increase the computational expense of sampling from and scoring in a fission source, as well as taking the inner product of two fission sources if too many bins are used.

When sampling from a discretized fission source, or tallying in a discretized fission source, the bin to sample/tally must be determined. The time required to find the appropriate bin is proportional to the number of spatial bins used, so as the number of spatial bins increases, the time required for sampling or tallying increases and the figure of merit decreases.

When choosing a discretization strategy, it is important to find the smallest number of spatial bins that will reduce the eigenvalue error due to spatial discretization

smaller than the statistical uncertainty. We must be careful not to use too many spatial bins or the efficiency will suffer.

In Figure 2.6 we see that the slopes of the linear best fit approximations have values ranging from 1.8 to 1.9. This indicates there is a nearly quadratic or second-order relationship between the spatial discretization and the error in the eigenvalue estimate. In this chapter I will demonstrate a higher order accurate approximation to the spatial discretization and show how it reduces the error caused by the discretization of the fission source. This idea is based on work performed by Griesheimer [14] on Functional Expansion Tallies.

3.1 Second-Order Accurate Approximation—Linear in Space

The spatial approximation used in Chapter 2 is a first-order accurate approximation, i.e. constant in space. The fission source was approximated as

$$v_\Pi(x) = \sum_{b=1}^{B} a_b \Pi_b(x), \tag{3.1}$$

where B is the number of spatial bins and

$$\Pi_b(x) = \begin{cases} \left(\frac{1}{x_{b+1}-x_b}\right)^{1/2}, & x_b \leq x < x_{b+1} \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

The fission source, $v_\Pi(x)$ is represented in Arnoldi's method as a vector of the form

$$v_\Pi = \left[a_1, a_2, \ldots, a_B\right]^T. \tag{3.3}$$

A spatial discretization of order 2 is not very different from the first-order accurate approximation given in Equation (3.1) and Equation (3.2). We approximate the fission source as a linear combination of functions

$$v_{\mathcal{L}}(x) = \sum_{b=1}^{B} \mathcal{L}_b(x, \alpha_b, \beta_b), \tag{3.4}$$

where B is the number of spatial bins. The fission source in each bin is approximated by a linear function, $\mathcal{L}_b(x, \alpha_b, \beta_b)$, over the range of the bin:

$$\mathcal{L}_b(x, \alpha_b, \beta_b) = \begin{cases} \alpha_b + \beta_b x, & x_b \leq x < x_{b+1} \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

In this second-order accurate approximation the term $\beta_b x$ is included which preserves some of the spatial information ignored in a first-order accurate approximation. Similarly to the first-order accurate approximation, the Arnoldi vector representation of $v_{\mathcal{L}}$ is a vector of the form

$$v_{\mathcal{L}} = \left[\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_B\right]^T. \tag{3.6}$$

The inner product between two piecewise linear in space fission sources is defined to be

$$\langle v_{\Pi}^{(j)}, v_{\Pi}^{(k)} \rangle = \sum_{b=1}^{B} \left(\alpha_b^{(j)} \alpha_b^{(k)} + \beta_b^{(j)} \beta_b^{(k)} \right). \tag{3.7}$$

3.1.1 Sampling

The integral

$$q_b = \int \left| \mathcal{L}_b(x, \alpha_b, \beta_b) \right| \mathrm{d}x \tag{3.8}$$

represents the rate of fission neutrons generated in the range $\left[x_b, x_{b+1}\right)$ and its magnitude—relative to the integrals over every other bin—is the probability of sampling a neutron from that bin. We can create a normalized, discrete distribution $p(x) = \left\{p_b\right\}_{b=1}^{B}$ where $p_b = q_b/Q$ and $Q = \sum_{b=1}^{B} q_b$ is the total source strength. We can sample a bin from $p(x)$ as we did for a first-order fission source. Once a bin has been chosen, the position of the neutron is sampled from within the bin with the distribution function defined as

$$p_b(x) = \frac{1}{q_b} \left|\mathcal{L}_b(x, \alpha_b, \beta_b)\right| \tag{3.9}$$

where q_b is from Equation (3.8). Once the position of the neutron has been sampled we give it a weight just as in Section 2.2.1

$$\omega = \begin{cases} 1 & v(x_s) > 0 \\ -1 & v(x_s) < 1. \end{cases} \tag{3.10}$$

3.1.2 Determining the Expansion Coefficients α and β

With the fission source being approximated by the function in Equation (3.5) we turn our attention to determining the expansion coefficients α and β. To do this we must evaluate two integrals, something that is is well suited for Monte Carlo methods. We first define the midpoint of bin b

$$x_{b,\text{mid}} = \frac{x_{b+1} + x_b}{2}. \tag{3.11}$$

Taking the zeroth and first spatial moments over the bin

$$\int_{x_b}^{x_{b+1}} \mathcal{L}_b(x, \alpha_b, \beta_b)\,\mathrm{d}x = \frac{1}{2}\left(x_{b+1} - x_b\right)\left[2\alpha_b + \beta_b\left(x_{b+1} + x_b\right)\right] \tag{3.12a}$$

$$\int_{x_b}^{x_{b+1}} \left(x - x_{b,\mathrm{mid}}\right) \mathcal{L}_b(x, \alpha_b, \beta_b)\,\mathrm{d}x = \frac{\beta_b}{12}\left(x_{b+1} - x_b\right)^3 \tag{3.12b}$$

gives two equations for α_b and β_b. The left-hand side of Equation (3.12) can be evaluated via Monte Carlo

$$\int_{x_b}^{x_{b+1}} \mathcal{L}_b(x, \alpha_b, \beta_b)\,\mathrm{d}x = \frac{1}{N}\sum_{i=1}^{N} \omega_i \tag{3.13a}$$

$$\int_{x_b}^{x_{b+1}} \left(x_i - x_{b,\mathrm{mid}}\right) \mathcal{L}_b(x, \alpha_b, \beta_b)\,\mathrm{d}x = \frac{1}{N}\sum_{i=1}^{N} \left(x - x_{b,\mathrm{mid}}\right) \omega_i \tag{3.13b}$$

where N is the number of source particles and ω_i and x_i are the weight and position of the particle that induces fission in bin b. Note that ω_i can be negative. By equating Equation (3.12a) with Equation (3.13a) and Equation (3.12b) with Equation (3.13b) we can obtain expressions for α_b and β_b

$$\alpha_b = \frac{1}{x_{b+1} - x_b}\frac{1}{N}\sum_{i=1}^{N} \omega_i - \frac{\beta_b}{2}\left(x_{b+1} + x_b\right) \tag{3.14a}$$

$$\beta_b = \frac{12}{\left(x_{b+1} - x_b\right)^3}\frac{1}{N}\sum_{i=1}^{N} \left(x_i - x_{b,\mathrm{mid}}\right) \omega_i \tag{3.14b}$$

For Monte Carlo particle transport this means every time a fission is caused in bin b the tallies ω_i and $\left(x_i - x_{b,\mathrm{mid}}\right)\omega_i$ are recorded. At the end of the iteration, when the sampling and tallying are finished, the fission source is normalized similarly to the constant in space approximated source (Equation (2.20))

$$v_{\mathcal{L}}^{(j+1)}(x) = \mathcal{A} v_{\mathcal{L}}^{(j)}(x) \int \left|v_{\mathcal{L}}^{(j)}(x)\right| \mathrm{d}x. \tag{3.15}$$

Notice the difference between Equation (3.15) and Equation (2.20) is the term $1/N$ is missing here. The source is still scaled by the number of source particles in Equation (3.14a) and Equation (3.14b).

3.2 Numerical Results

To demonstrate the difference between a first and second-order accurate approximation to the fission source a series of simulations, similar to those given in Section 2.3.1 has been performed. In Section 2.3.1 the effect of the coarseness of the spatial discretization on the bias of the eigenvalue estimate was shown by performing the same calculation but varying the number of spatial bins. Here, the same set of calculations is performed, a second-order accurate, linear in space approximation is used to the fission source. The results will be compared with the results from Section 2.3.1.

The problem is a 20 mfp thick semi-infinite homogeneous slab of multiplying material with cross sections $\nu\Sigma_f = 1.0,\ \Sigma_a = 0.2,$ and $\Sigma_s = 0.8;\ \Sigma_t = 1.0.$ In each iteration 10^5 histories are tracked, 10 iterations per restart with 50 inactive restarts and 500 active restarts. The number of spatial bins range from 10 to 150. Both first and second order approximations start with a uniform source across the entire slab. First-order accurate results are denoted with a subscript Π and second-order accurate results are denoted with a subscript $\mathcal{L}$.

On the following pages I present tables showing the numerical results of the bias and uncertainty of the estimated eigenvalues. Tables 3.1a and 3.1b show the results for the fundamental eigenvalue for the first and second-order accurate approximations, respectively. In each of these tables is the eigenvalue estimate (λ), the standard deviation (σ), the error in the eigenvalue estimate ($\mathcal{B}$) and the figure of merit (FOM). The error is the absolute value of the difference between the estimated eigenvalue and the published reference values from [12] and [9].

# Bins	Bin Width (mfp)	λ_{Π}	σ_{Π}	$\mathcal{B}_{\Pi}$	FOM (Π)
10	2.00	4.8003	6.6×10^{-4}	2.7×10^{-2}	831.2
25	0.80	4.8224	6.8×10^{-4}	5.3×10^{-3}	773.2
40	0.50	4.8251	6.3×10^{-4}	2.6×10^{-3}	872.0
50	0.40	4.8273	6.5×10^{-4}	4.2×10^{-4}	829.0
60	0.33	4.8258	6.9×10^{-4}	2.0×10^{-3}	704.3
75	0.27	4.8275	6.7×10^{-4}	2.4×10^{-4}	753.0
90	0.22	4.8277	6.7×10^{-4}	4.2×10^{-5}	746.1
105	0.19	4.8277	6.9×10^{-4}	5.2×10^{-5}	698.3
120	0.17	4.8282	6.5×10^{-4}	4.1×10^{-4}	767.8
135	0.15	4.8274	7.0×10^{-4}	3.1×10^{-4}	656.9
150	0.13	4.8285	6.4×10^{-4}	7.5×10^{-4}	792.0

(a) First-order accurate (Π) spatial discretization

# Bins	Bin Width (mfp)	$\lambda_{\mathcal{L}}$	$\sigma_{\mathcal{L}}$	$\mathcal{B}_{\mathcal{L}}$	FOM ($\mathcal{L}$)
10	2.00	4.8302	1.2×10^{-3}	2.5×10^{-3}	259.5
25	0.80	4.8280	4.4×10^{-4}	2.1×10^{-4}	1765.7
40	0.50	4.8283	4.5×10^{-4}	5.7×10^{-4}	1675.6
50	0.40	4.8277	4.5×10^{-4}	6.2×10^{-5}	1470.6
60	0.33	4.8283	3.9×10^{-4}	5.4×10^{-4}	2171.6
75	0.27	4.8276	4.0×10^{-4}	1.5×10^{-4}	2069.3
90	0.22	4.8275	3.8×10^{-4}	2.0×10^{-4}	2251.3
105	0.19	4.8259	5.9×10^{-4}	1.8×10^{-3}	877.1
120	0.17	4.8172	1.3×10^{-3}	1.1×10^{-2}	189.4
135	0.15	4.8110	2.1×10^{-3}	1.7×10^{-2}	66.2
150	0.13	4.8102	2.0×10^{-3}	1.8×10^{-2}	73.0

(b) Second-order accurate ($\mathcal{L}$) spatial discretization

Table 3.1: Error ($\mathcal{B}$) in the fundamental eigenvalue estimate (λ) for first-order accurate (a) and second-order accurate (b) discretization as a function of the bin width. Figure of merit is also given for first and second-order accurate spatial discretizations. 1E5 particles were tracked in each iteration.

The purpose of going to a second-order accurate accurate approximation to the fission source is to reduce the error in the eigenvalue estimate associated with discretizing the fission source. We can see from tables 3.1a and 3.1b that for bin widths between 0.4 and 2.0 mfp thick the error in the eigenvalue estimate from the second-order accurate approximation is an order of magnitude smaller than the error from the first-order accurate approximation. Thus, moving to a second-order accurate approximation to the fission source can greatly reduce the error.

The second-order accurate approximation has a smaller statistical uncertainty than the first-order accurate approximation for bin widths between 0.8 and 0.22 mfp and is fairly independent of the bin width in this range. The uncertainty for the first-order accurate approximation appears to be independent of the size of the bin width over the whole range of bin widths. The largest and three smallest bin widths from the second-order accurate approximation however have large errors in the eigenvalue estimate and large statistical uncertainties. For thin bins, the number of neutrons that are born in those bins is small, causing the Monte Carlo noise to dominate the results. The behavior of the linear-in-space approximation in the 2 mfp wide bins is a mystery.

The figure of merit is 2-3 times larger for the second-order accurate approximation than the first-order for bin widths between 0.8 and 0.22 mfp. For very small bin widths or for the largest bin width, the figure of merit is smaller for the second-order accurate approximation than the first-order. The figure of merit as a function of bin width is shown graphically in Figure 3.1.

In the first-order accurate approximation, the shape of the eigenvectors were limited by the flat approximation in each bin. Since the second-order accurate approximation is a linear approximation in each bin, we expect the second-order accurate approximation to have an improved eigenvector than the first-order accurate approximation. In Figure 3.2 the eigenvectors from the second-order accurate approximation

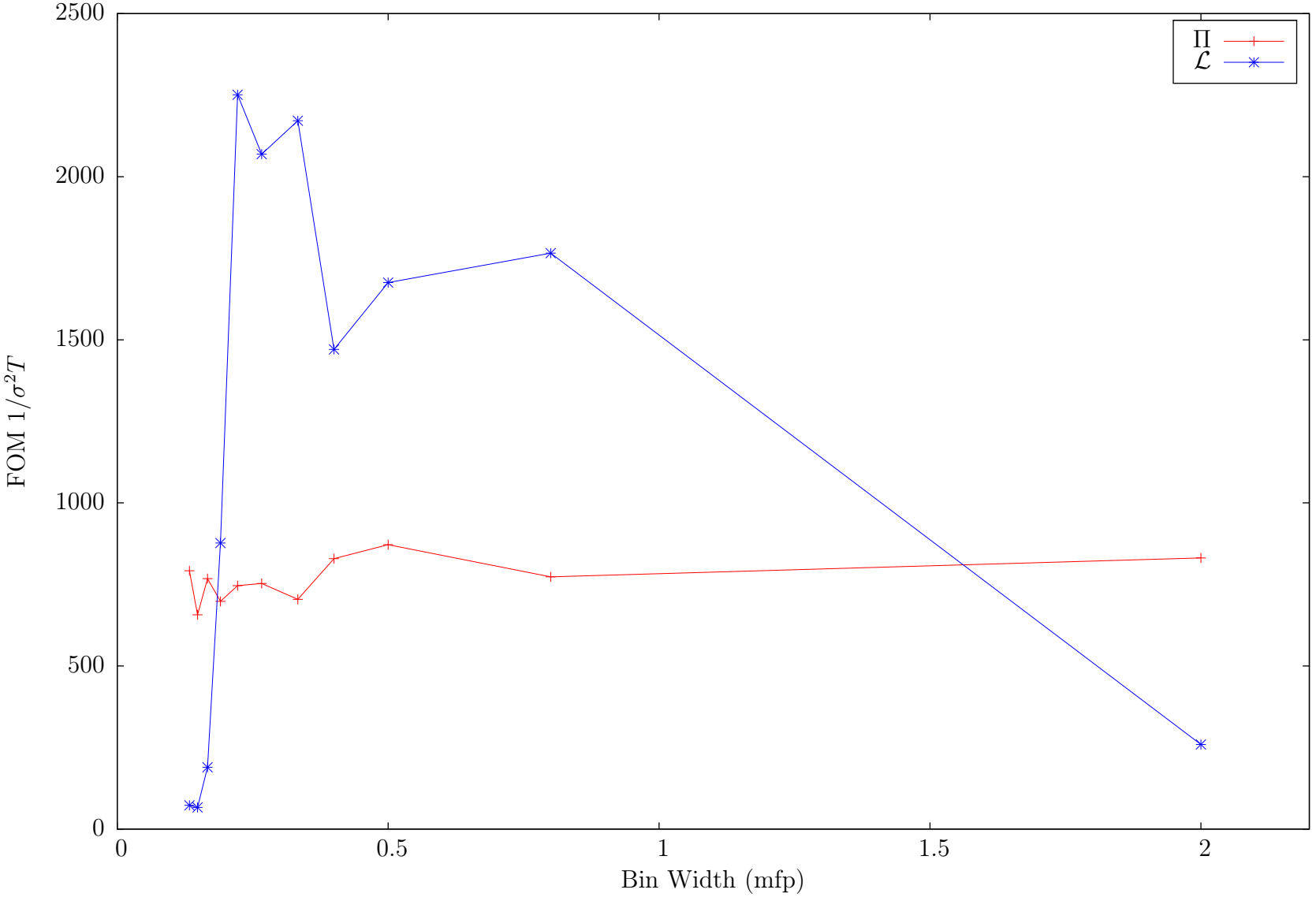


Figure 3.1: Figure of merit as a function of bin width for a slab of width 20 mfp and tracking 1E5 particles per iteration. Included are results from a first-order (Π) and second-order ($\mathcal{L}$) approximation to the fission source.

with 60 spatial bins are shown. In Figure 3.3, the same eigenvectors are plotted along with the eigenvectors from the first-order accurate approximation to show they have the same shape. From these figures we see that the second-order accurate approximation is a smooth approximation to the fission source and appears continuous across the bin boundaries.

For small bin widths the second-order accurate approximation seems to have difficulties due to too few particles scoring in a bin. If more particles were tracked in an iteration we expect more particles to score in each bin and the statistical uncertainty to improve. The above simulations have been repeated, but have tracked 1E6 particles in each iteration—ten times more particles—to improve the statistics in each simulation. The results for the first-order accurate approximation are given in Table 3.2a and the results for the second-order accurate approximation are given in Table 3.2b.

Tracking more particles in an iteration seems to have made a big improvement for the second-order accurate approximation with very small bin widths. Now all the results, except the large bin width of 2 mfp, estimates the fundamental eigenvalue within statistical uncertainty. Tracking ten times more particles per iteration has improved the poor results from the simulation with small bins and has reduced the standard deviation by approximately one-third, as we expect for the second-order accurate approximation. The standard deviation was also reduced for the first-order accurate approximation simulations. The error in the eigenvalue estimate is also significantly smaller for the second-order accurate approximation but remains essentially unchanged for the first-order accurate approximation.

In Figure 3.4 the figure of merit is shown for the first and second-order spatial approximations, where 1 million particles are tracked for each iteration. We see that when tracking more particles in an iteration, the figure of merit improves for second-order accurate approximations with small bin widths. This is because now we have

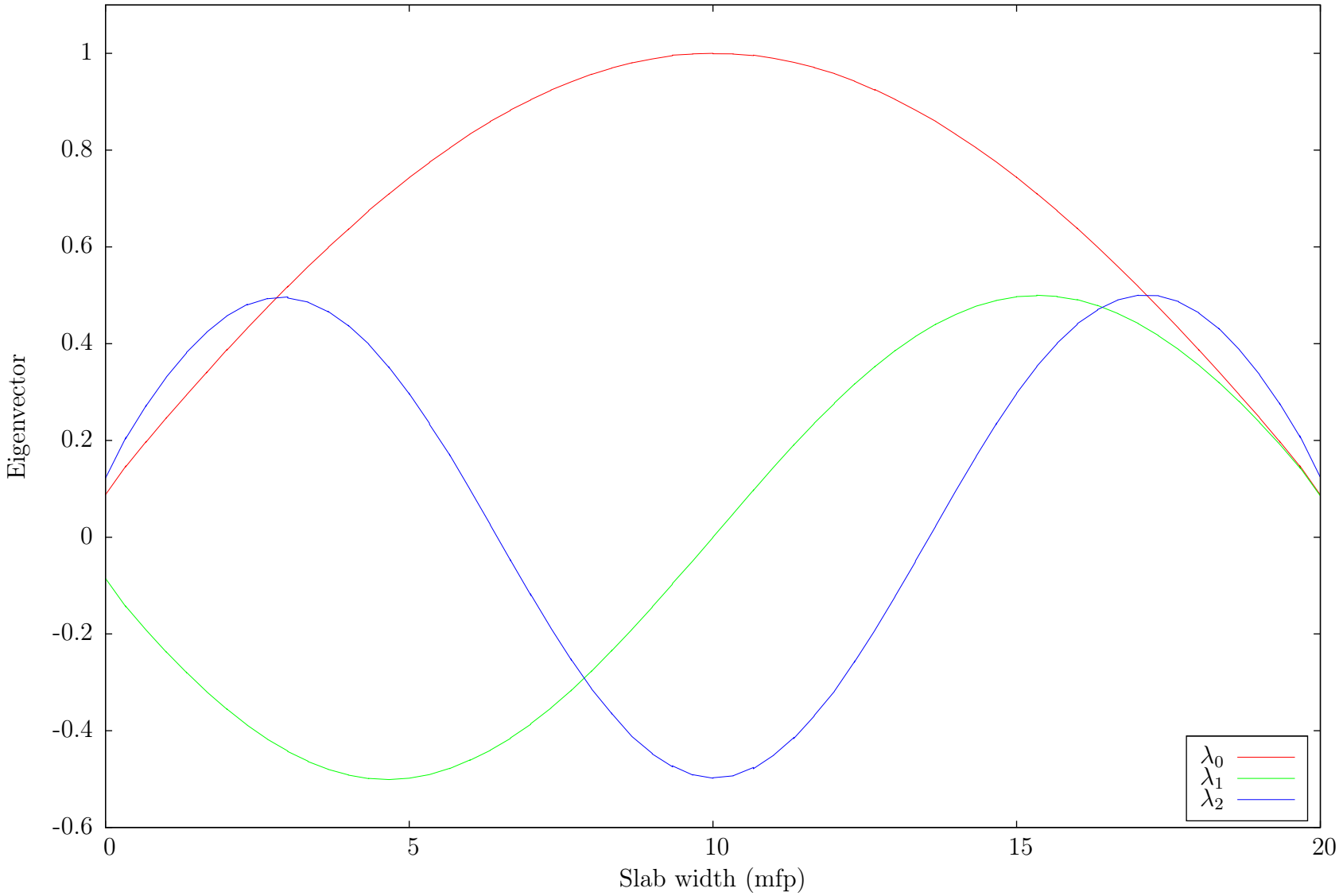


Figure 3.2: Estimates of the fundamental and first two harmonic eigenvectors, tracking 1E5 particles per iteration and, using 60 spatial bins with a second-order accurate approximation to the fission source.

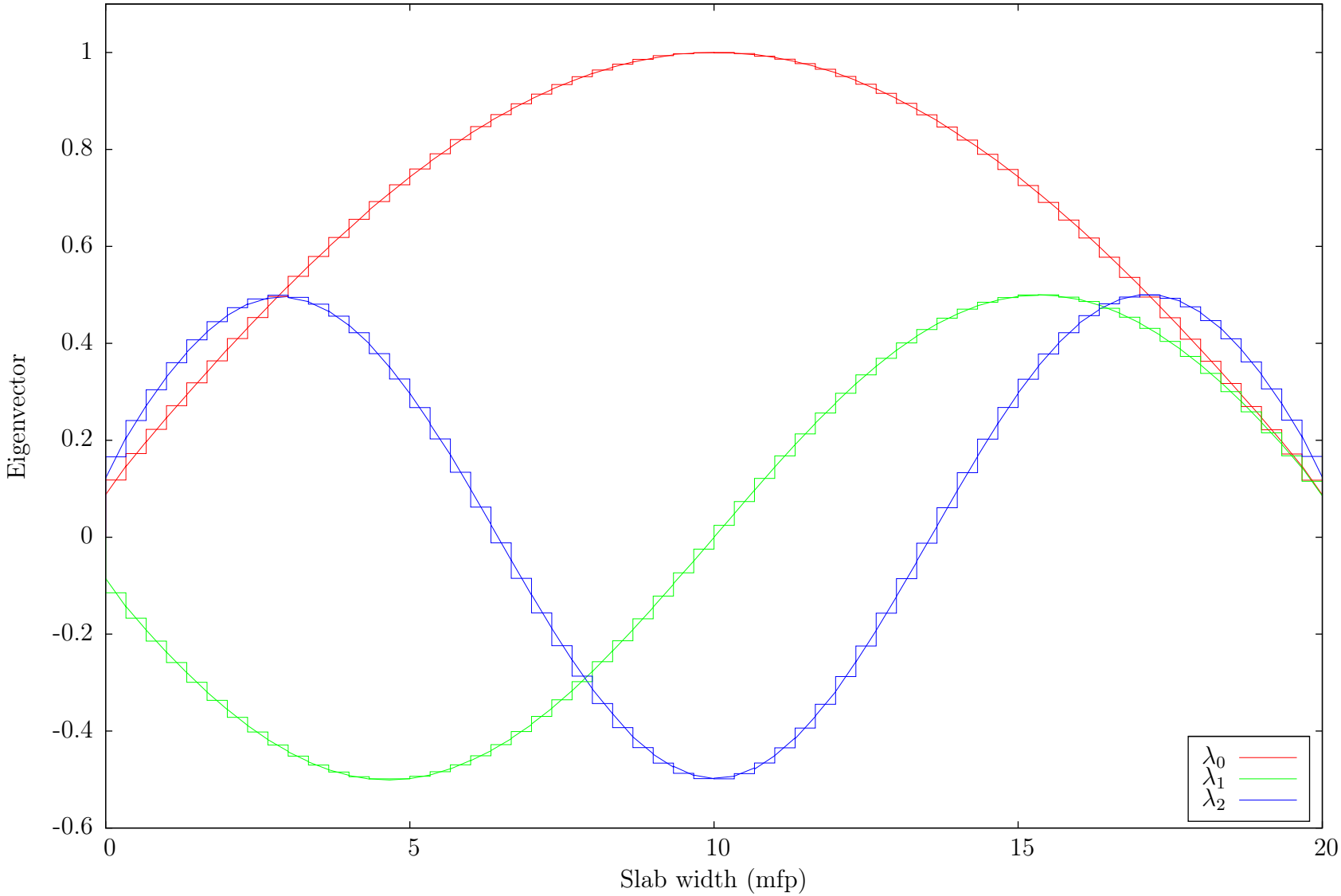


Figure 3.3: Estimates of the fundamental and first two harmonic eigenvectors tracking 1E5 particles per iteration and using 60 spatial bins. The sawtooth shaped curves use a first-order accurate approximation while the smooth curves use a second-order accurate approximation to the fission source.

an accurate estimate of the eigenvalue as well as a reduced standard deviation. With the exception of simulations with very large bin widths, the figure of merit for second-order accurate approximations are two times larger than that for first-order accurate approximations. It should be noted that the linear-in-space approximation is not only more efficient than the power method, but it is in fact computing 3 eigenpairs compared to the power methods single eigenvector.

The anomaly in these results is the second-order accurate approximation with a bin width of 2.0 mfp. The error in the eigenvalue estimate is smaller than the error for the same problem but with a first-order accurate approximation, but the eigenvalue estimate is not within statistical uncertainty regardless of the number of particles tracked. In addition, the error in the eigenvalue estimate is larger than for any other bin width.

# Bins	Bin Width (mfp)	λ_{Π}	σ_{Π}	$\mathcal{B}_{\Pi}$	FOM (Π)
10	2.00	4.8003	2.1×10^{-4}	2.7×10^{-2}	838.5
25	0.80	4.8232	2.2×10^{-4}	4.5×10^{-3}	751.1
40	0.50	4.8258	2.1×10^{-4}	1.9×10^{-3}	832.9
50	0.40	4.8267	2.1×10^{-4}	1.1×10^{-3}	796.1
60	0.33	4.8270	2.1×10^{-4}	7.3×10^{-4}	751.0
75	0.27	4.8269	2.1×10^{-4}	8.3×10^{-4}	787.5
90	0.22	4.8275	2.2×10^{-4}	2.7×10^{-4}	694.3
105	0.19	4.8274	2.1×10^{-4}	3.4×10^{-4}	768.8
120	0.17	4.8276	2.0×10^{-4}	1.6×10^{-4}	846.8
135	0.15	4.8275	2.0×10^{-4}	2.6×10^{-4}	793.9
150	0.13	4.8280	2.1×10^{-4}	2.7×10^{-4}	746.8

(a) First-order (Π) spatial discretization

# Bins	Bin Width (mfp)	$\lambda_{\mathcal{L}}$	$\sigma_{\mathcal{L}}$	$\mathcal{B}_{\mathcal{L}}$	FOM ($\mathcal{L}$)
10	2.00	4.8329	4.1×10^{-3}	5.1×10^{-3}	2.0
25	0.80	4.8274	1.5×10^{-4}	3.1×10^{-4}	1607.8
40	0.50	4.8278	1.4×10^{-4}	6.8×10^{-5}	1742.0
50	0.40	4.8276	1.5×10^{-4}	1.7×10^{-4}	1512.8
60	0.33	4.8276	1.3×10^{-4}	1.1×10^{-4}	1832.8
75	0.27	4.8277	1.4×10^{-4}	5.2×10^{-5}	1692.6
90	0.22	4.8278	1.4×10^{-4}	7.8×10^{-5}	1730.6
105	0.19	4.8279	1.4×10^{-4}	1.1×10^{-4}	1701.9
120	0.17	4.8278	1.3×10^{-4}	4.8×10^{-5}	1985.2
135	0.15	4.8279	1.3×10^{-4}	1.9×10^{-4}	1757.3
150	0.13	4.8277	1.2×10^{-4}	6.2×10^{-5}	2015.0

(b) Second-order ($\mathcal{L}$) spatial discretization

Table 3.2: Error ($\mathcal{B}$) in the fundamental eigenvalue estimate (λ) for first-order (a) and second-order (b) discretization as a function of the bin width. Figure of merit is also given for first and second-order spatial discretizations. 1E6 particles were tracked in each iteration.

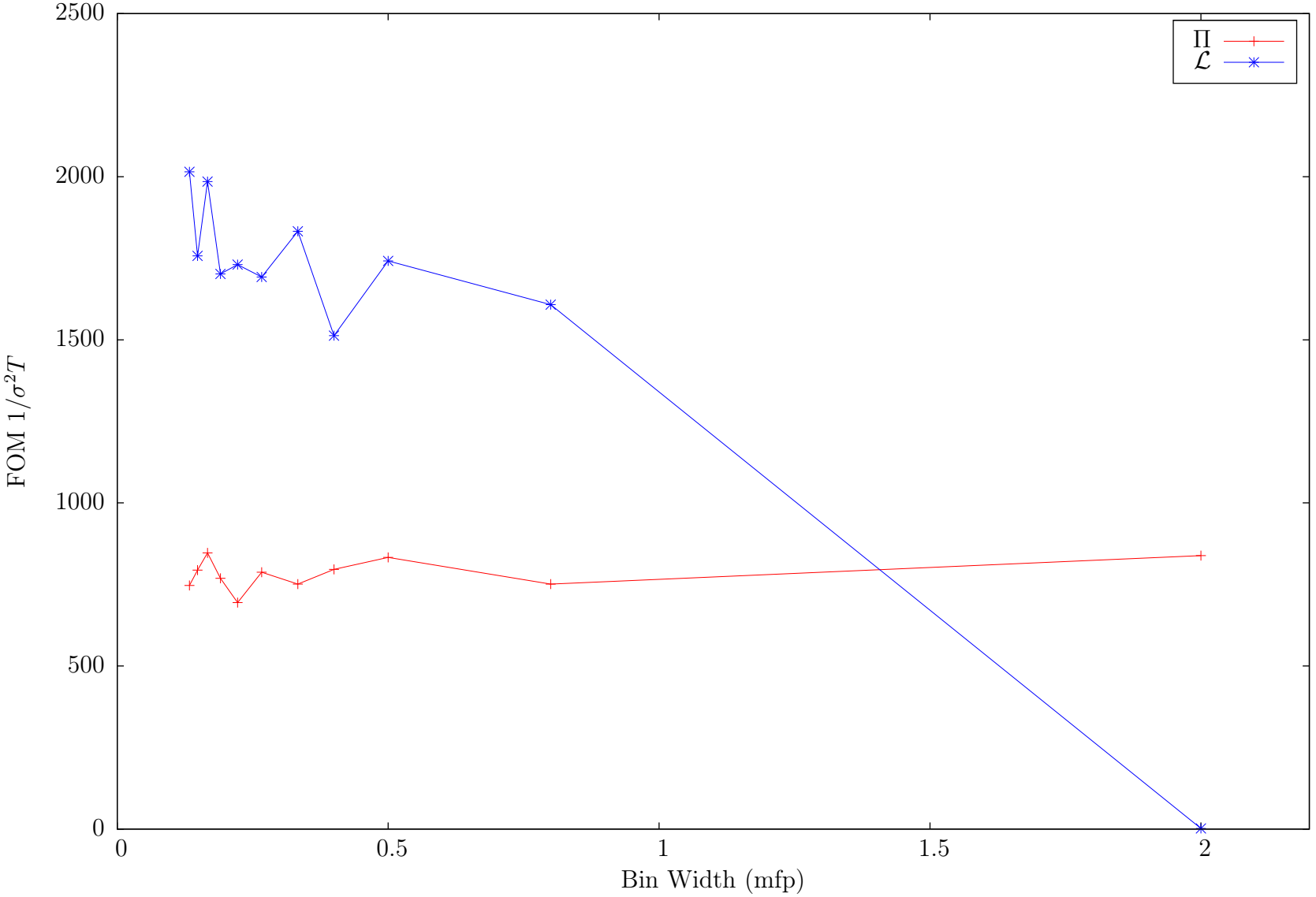


Figure 3.4: Figure of merit as a function of bin width for a slab of width 20 mfp and tracking 1E6 particles per iteration. Included are results from a first-order (Π) and second-order ($\mathcal{L}$) approximation to the fission source.

3.3 Summary

In this chapter first and second-order accurate approximations to the fission source have been investigated. These were demonstrated on a semi-infinite, homogeneous slab of multiplying material of width 20 mfp. Both techniques have an error associated with the discretization. The power method does not suffer from this kind of error. It has been shown that for a sufficient number of spatial bins the error in the eigenvalue estimate is smaller than the standard deviation of the eigenvalue estimates.

Using a second-order accurate approximation for the fission source is a significant improvement over the first-order accurate approximation; the standard deviation is smaller and the error in the eigenvalue estimate is an order of magnitude smaller when a sufficient number of particles is used. The standard deviation becomes smaller for both first and second-order accurate approximations as the number of particles tracked in an iteration increases. The error in the eigenvalue estimate is not significantly improved for the first-order accurate approximation as the number of particles tracked increases, while the error decreases for the second-order accurate approximation.

The second-order accurate approximation has much better approximation to the eigenvectors. Because the approximation is linear in space, the eigenvector approximation is smoother in the second-order accurate approximation than in the first-order accurate approximation. The results shown in this chapter show that the eigenvectors are nearly continuous across bin boundaries for the second-order accurate approximation.

Both the first and second-order accurate approximations have errors in the eigenvalue estimate that are less than the statistical uncertainty for bin widths in the 0.5–0.2 mfp range. If the spatial discretization is too coarse, too much information is lost; if the discretization is too fine, the Monte Carlo transport is too noisy within each bin to give good results. In both extreme cases an error eigenvalue estimates

arises. In the second-order accurate approximation, the figure of merit significantly decreases in these extreme case because the variance of the mean of the eigenvalue estimates is larger.

Using a moderate number of spatial bins can give excellent results. When tracking more particles per iteration, a finer discretization can be used. Further work needs to be performed to identify—if possible—a general rule for the granularity of the discretization.

Chapter 4

Relaxed Arnoldi

In Monte Carlo particle transport the figure of merit (FOM) is a measure of efficiency of a particular algorithm or simulation. FOM was given in Equation (2.26) and again here for clarity

$$\text{FOM} \equiv \frac{1}{\sigma_\lambda^2 T}. \tag{4.1}$$

The variance of an estimated eigenvalue λ in a Monte Carlo approach is

$$\sigma_\lambda^2 = \frac{1}{N}\sigma^2, \tag{4.2}$$

where N is the number of estimates of the eigenvalue and σ^2 is the variance of the distribution of the estimates. A larger FOM indicates a more efficient calculation, i.e. the variance is smaller for a given amount of computer time or number of particles tracked. To increase the FOM, one must increase N while keeping T as small as possible which, in turn, will decrease σ.

In this thesis the focus has been on estimating the eigenvalues of the fission-transport operator $\mathcal{A}$. In Chapter 2, I compared the figure of merit for Arnoldi's method to that produced by the power method for estimating the fundamental eigenvalue. The power method had a FOM that was much larger than that of Arnoldi's method, even though both methods tracked the same number of particles in each iteration.

The power method calculates an estimate of the fundamental eigenvalues at the end of every iteration, while in Arnoldi's method it was chosen to calculate an eigenvalue estimate at the end of an Arnoldi restart. The simulations used in Chapter 2 had ten iterations in each Arnoldi restart. Even though the power method and Arnoldi's method used the same number of particles to apply the linear operator at each iteration, the power method had ten times more eigenvalue estimates than Arnoldi's method—for the same number of particles tracked. The FOM shown in Table 2.1 is nearly exactly ten times larger for the power method than for Arnoldi's method. This indicates that the variance is driven primarily by the number of eigenvalue estimates.

4.1 Relaxed Arnoldi's Method

Recently [29, 25, 24, 11] there has been some interest in Krylov subspace methods where the application of the linear operator is performed inexactly; for example when the application of the operator $\mathcal{A}$ is an iterative process and the iterations are terminated before the calculation converges. By terminating early, computation time is saved at the expense of the precision of the application of $\mathcal{A}$.

A group of researchers [4] discovered experimentally that under certain circumstances the inexact application of the linear operator has little or no effect upon the convergence of the eigenpair calculation. They discovered that, as Arnoldi's method proceeds and the size of the Krylov subspace increases, the precision to which the matrix-vector product is calculated can be decreased, or relaxed. Their results have been further investigated leading to more formal theories and proofs (see [5] and [23]).

When a linear operator is applied inexactly we can obtain the inexact vector $\hat{v}_k$ by computing

$$\hat{v}_k = \left(\mathcal{A} + \mathbf{\Delta}\mathcal{A}_k\right) \hat{v}_{k-1}, \tag{4.3}$$

where $\mathbf{\Delta}\mathcal{A}_k$ is some perturbation matrix. Of course, when the linear operator is applied inexactly, we are no longer forming a true Krylov subspace as shown in Equation (1.5) but have rather

$$\hat{\mathcal{K}} = \text{span}\left\{v_0, \hat{v}_1, \hat{v}_2, \ldots, \hat{v}_{m-1}\right\}, \tag{4.4}$$

where $\hat{v}_k$ is given in Equation (4.3) and is orthogonalized against the previously calculated Arnoldi vectors $\hat{V}_k = \left[v_0\, \hat{v}_1\, \ldots\, \hat{v}_{k-1}\right]$.

The basic idea presented by Bouras and Frayssé is to reduce the precision to which the linear operator is applied in an iteration of Arnoldi's method, but limit the size of $\mathbf{\Delta}\mathcal{A}_k$ to some fraction of $\mathcal{A}$. To show how this is done, let η be the final tolerance required and let α_k be a scalar defined by

$$\alpha_k = \frac{1}{\min\left(\left\|r_{k-1}\right\|, 1\right)}, \tag{4.5}$$

where $\left\|r_{k-1}\right\|$ is the magnitude of the residual as given in Equation (2.11). The limit to $\mathbf{\Delta}\mathcal{A}_k$'s size is defined as

$$\left\|\mathbf{\Delta}\mathcal{A}_k\right\| = \varepsilon_k \left\|\mathcal{A}\right\| \tag{4.6}$$

where

$$\varepsilon_k = \min\left(\alpha_k \eta, 1\right). \tag{4.7}$$

.

If we put together Equation (4.5), Equation (4.6), and Equation (4.7) we see that as the residual decreases the size of $\mathbf{\Delta}\mathcal{A}_k$ is allowed to become larger—the precision to which the linear operator $\mathcal{A}$ is applied is *relaxed*.

Relaxed Monte Carlo Application of Fission-Transport Operator

In Monte Carlo particle transport the magnitude of the perturbation matrix, $\|\boldsymbol{\Delta}\mathcal{A}\|$ is proportional to $1/\sqrt{N}$ where N is the number of particles tracked when applying the fission-transport operator. Relaxing the application of the fission-transport operator means simply reducing the number of particles used in applying the operator

$$N_k = f\left(\|r_{k-1}\|\right) N_0, \tag{4.8}$$

where N_k is the number of particles to be used in iteration k and N_0 is the number of particles tracked when not relaxed. The scalar $f\left(\|r_{k-1}\|\right)$ is the fraction of the number of particles N_0 that should be tracked in iteration k.

In Monte Carlo particle transport the parameter $\eta < 1$ is some input value defining when relaxing can occur. We want to relax the application of $\mathcal{A}$ when the residual is smaller than η and to track N_0 particles when the residual is larger than η. To do this we define

$$\varepsilon_k = \begin{cases} \eta / \|r_{k-1}\| & \|r_{k-1}\| < \eta \\ 1 & \|r_{k-1}\| \geq \eta. \end{cases} \tag{4.9}$$

and insert into Equation (4.6). We note that

$$\|\boldsymbol{\Delta}\mathcal{A}\| = \frac{\boldsymbol{C}}{\sqrt{N_k}}$$

where $\boldsymbol{C}$ is just some constant of proportionality, we can obtain an equation for N_k,

$$N_k = \frac{1}{\varepsilon_k^2}\left(\frac{\boldsymbol{C}}{\|\mathcal{A}\|}\right)^2. \tag{4.10}$$

To determine the fraction of particles to be tracked in iteration k we equate the right-hand sides of Equation (4.10) and Equation (4.8),

$$f\left(\left\|r_{k-1}\right\|\right) N_0 = \frac{1}{\varepsilon_k^2}\left(\frac{\boldsymbol{C}}{\|\mathcal{A}\|}\right)^2. \tag{4.11}$$

The term $\left(\boldsymbol{C}/\|\mathcal{A}\|\right)^2$ is just some constant which we conveniently choose to be N_0. Therefore the fraction we are seeking must be

$$f\left(\left\|r_{k-1}\right\|\right) = \frac{1}{\varepsilon_k^2} = \begin{cases} \left(\left\|r_{k-1}\right\|/\eta\right)^2 & \left\|r_{k-1}\right\| < \eta \\ 1 & \left\|r_{k-1}\right\| \geq \eta. \end{cases} \tag{4.12}$$

So if the residual is less than η the number of particles tracked in an iteration is reduced, but the number of particles tracked in an iteration will never be more than N_0. From Equation (4.8) we then have

$$N_k = \begin{cases} \left(\left\|r_{k-1}\right\|/\eta\right)^2 N_0 & \left\|r_{k-1}\right\| < \eta \\ N_0 & \left\|r_{k-1}\right\| \geq \eta. \end{cases} \tag{4.13}$$

This method of relaxation is not the only strategy that can be used. Here we have a quadratic function. A less aggressive strategy would be a linear function where

$$f\left(\left\|r_{k-1}\right\|\right) = \frac{1}{\varepsilon_k} = \begin{cases} \left\|r_{k-1}\right\|/\eta & \left\|r_{k-1}\right\| < \eta \\ 1 & \left\|r_{k-1}\right\| \geq \eta. \end{cases} \tag{4.14}$$

What is important is that as the residual decreases, relaxation increases and fewer particles are tracked in that iteration.

With the number of particles used in an iteration defined by Equation (4.13) we have a procedure for relaxing the precision to which the fission-transport operator is applied; as the residual decreases—the estimate of the eigenvalue improves—fewer particles are tracked in an iteration. We expect this relaxation to have no negative effect on the convergence of the eigenvalue estimate and therefore have a positive effect on the figure of merit as the computational expense required to apply the linear operator is decreased with the decrease in the number of particles tracked.

4.2 Numerical Results

To demonstrate the effect of relaxing Arnoldi's method, a similar problem to those already displayed in this thesis is shown. I will keep the same cross sections ($\nu\Sigma_f = 1.0$, $\Sigma_a = 0.2$, and $\Sigma_s = 0.8$; $\Sigma_t = 1.0$) but the width of the slab will be 50 mfp. This is a more difficult problem than the thinner slabs because the dominance ratio is 0.9924, which is larger than for the thinner geometries previously used.

For these simulations the more aggressive relaxation shown in Equation (4.13) is used and the relaxation parameter, η, is varied from 1E-8 to 1.0. As η increases, the number of particles tracked in an iteration/restart decreases. To keep the total number of particles the same among all simulations, the number of active restarts was increased when necessary. (The number of inactive restarts is 100 for all the simulations. Relaxation is done during inactive restarts as well as active restarts.) This means that for larger values of η there will be more eigenvalue estimates and a lower uncertainty in the eigenvalue estimate. When the uncertainty decreases the figure of merit increases.

If what Bouras and Frayssé suggest for relaxed Arnoldi is valid for Monte Carlo criticality calculations, we expect to see the estimates of the eigenvalue unaffected by relaxation. In addition, we expect the figure of merit to increase when relaxation

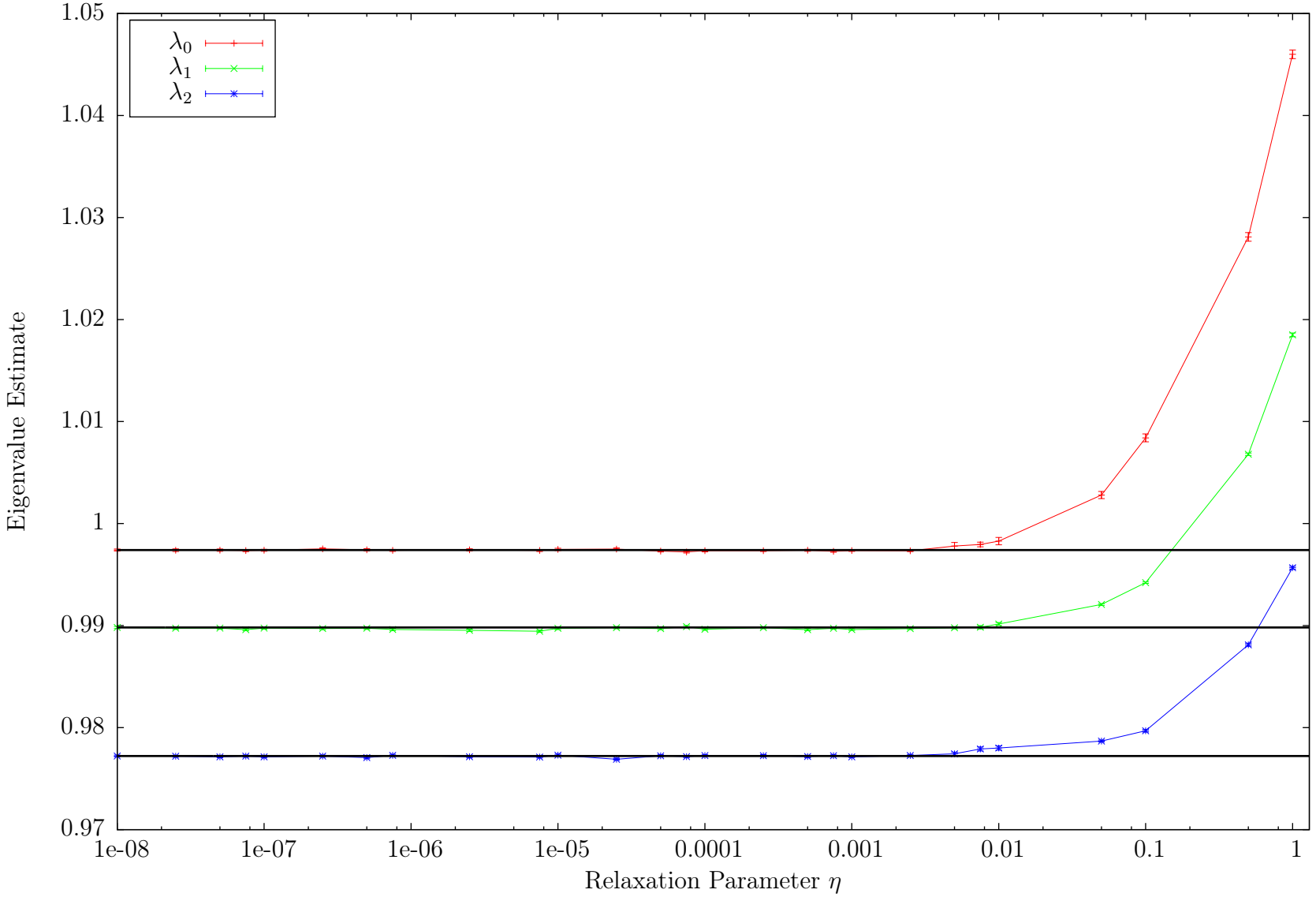


Figure 4.1: Eigenvalue estimates for the fundamental and first two harmonics for varying values of the relaxation parameter η. The number of particles tracked in a non-relaxed iteration is 5E5. The heavy lines are the reference eigenvalues from [12] and [9].

increases (i.e. η becomes larger) because fewer particles are being tracked in an iteration and therefore less time required to calculate an eigenvalue estimate. This doesn't hold for every value of η; eventually it will become so large the application of $\mathcal{A}$ is so imprecise—too relaxed—the eigenvalue estimate is just wrong.

The eigenvalue estimates as a function of relaxation parameter are graphed in Figure 4.1. The heavy lines are the eigenvalue estimates when no relaxation is used. We see that the eigenvalue estimates all fall within one standard deviation of the estimate calculated without any relaxation (shown in the dark heavy lines) until the relaxation becomes too great at around $\eta = 0.005$.

In Figure 4.2 the fundamental eigenvalue estimates are plotted along with the FOM for varying values of η. The dashed line in the Figure 4.2 is the figure of merit when Arnoldi's method is not relaxed. It appears that no relaxation ($\eta = 0.0$) is better than a little bit of relaxation ($\eta =$ small). There are a few values for η which show about a 50% increase in the figure of merit. However the range of η where there is an improvement over no relaxation is small. Also, at the same value of η where the eigenvalue estimate starts to diverge from the true value we see the figure of merit decrease by 1–2 orders of magnitude.

In Table 4.1 the eigenvalue estimates, standard deviation and the figure of merit are shown for these simulations. For $\eta \leq 0.0025$ the standard deviation is constant, $\sim 1 \times 10^{-4}$. When $\eta > 0.0025$ the eigenvalue estimates diverge and the standard deviation increases. This is what causes the figure of merit to drop.

Bouras and Frayssé [4] suggest that the first few Arnoldi vectors need to be known with "full accuracy" and that Arnoldi vectors calculated in later iterations can be relaxed. Given that Monte Carlo Arnoldi can never really apply $\mathcal{A}$ with full precision it is important to know how accurately the first Arnoldi vectors really need to be applied. In the next set of results, an attempt to come closer to full accuracy for the initial applications of the fission-transport operator by repeating the simulations

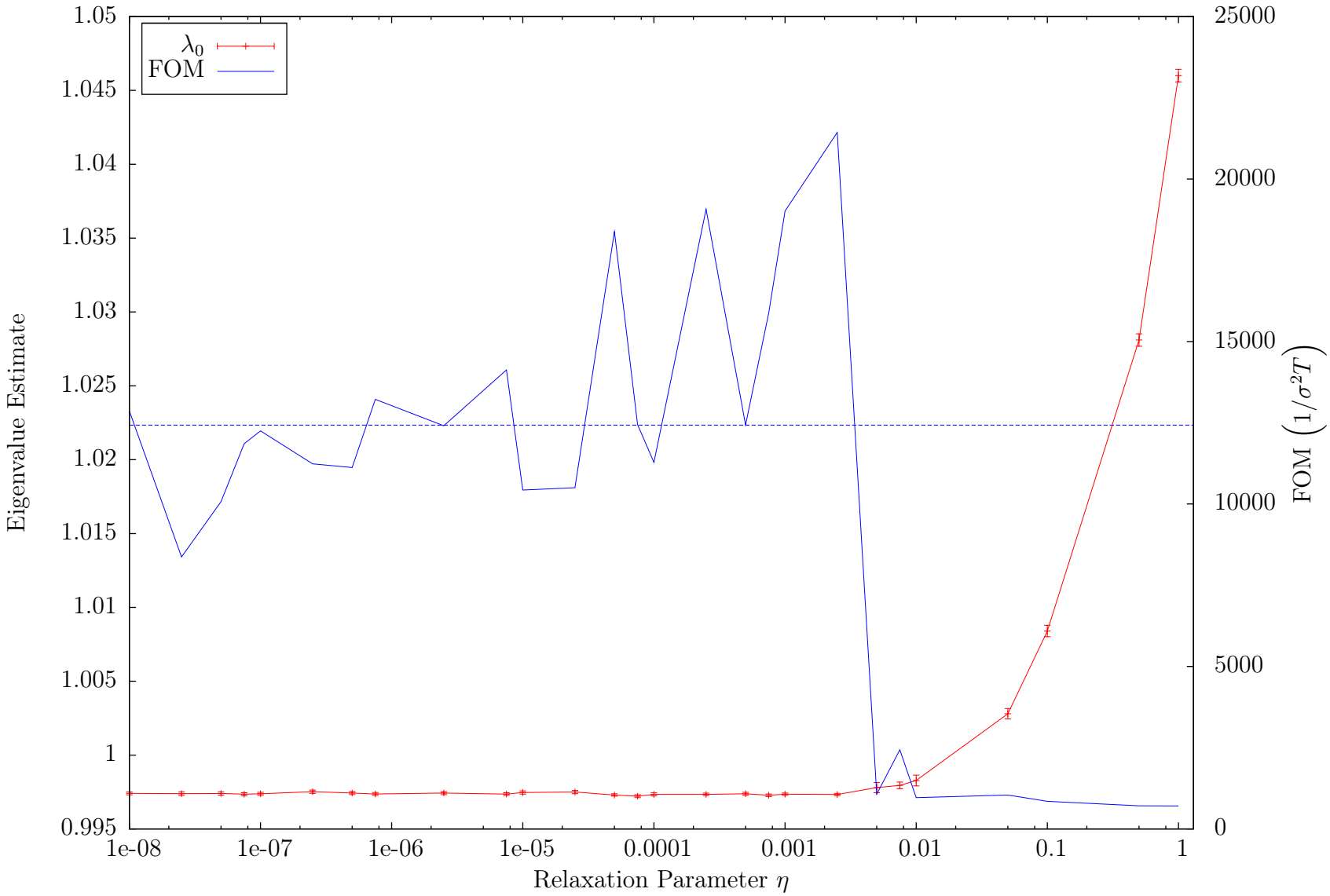


Figure 4.2: Fundamental eigenvalue estimate and figure of merit for varying values of the relaxation parameter η. The dashed line is the value of the figure of merit when there is no relaxation. The number of particles tracked in a non-relaxed iteration is 5E5.

η	Active Restarts	Total # Particles	λ	FOM	Time (s)
0	100	2.500e+09	$0.9973 \pm 1.0\times10^{-4}$	12427.9	7776.7
1E-08	100	2.500e+09	$0.9974 \pm 1.0\times10^{-4}$	12861.4	7741.5
2.5E-08	100	2.500e+09	$0.9974 \pm 1.2\times10^{-4}$	8371.5	7732.8
5E-08	100	2.500e+09	$0.9974 \pm 1.1\times10^{-4}$	10072.6	7734.8
7.5E-08	100	2.500e+09	$0.9974 \pm 1.0\times10^{-4}$	11854.2	7809.2
1E-07	100	2.500e+09	$0.9974 \pm 1.0\times10^{-4}$	12249.2	7727.9
2.5E-07	100	2.500e+09	$0.9975 \pm 1.1\times10^{-4}$	11235.8	7734.2
5E-07	100	2.500e+09	$0.9974 \pm 1.1\times10^{-4}$	11123.8	7757.4
7.5E-07	100	2.500e+09	$0.9974 \pm 9.8\times10^{-5}$	13221.0	7835.0
2.5E-06	100	2.500e+09	$0.9974 \pm 1.0\times10^{-4}$	12407.1	7753.4
7.5E-06	100	2.499e+09	$0.9974 \pm 9.5\times10^{-5}$	14125.4	7817.1
1E-05	100	2.500e+09	$0.9975 \pm 1.1\times10^{-4}$	10430.4	7741.5
2.5E-05	105	2.555e+09	$0.9975 \pm 1.1\times10^{-4}$	10501.4	7910.3
5E-05	112	2.621e+09	$0.9973 \pm 8.2\times10^{-5}$	18382.2	8122.4
7.5E-05	109	2.565e+09	$0.9972 \pm 1.0\times10^{-4}$	12444.6	8006.8
0.0001	110	2.544e+09	$0.9973 \pm 1.1\times10^{-4}$	11280.2	7871.7
0.00025	124	2.541e+09	$0.9973 \pm 8.2\times10^{-5}$	19072.5	7853.4
0.0005	140	2.554e+09	$0.9974 \pm 1.0\times10^{-4}$	12433.1	7909.4
0.00075	150	2.562e+09	$0.9973 \pm 8.9\times10^{-5}$	15879.3	7945.2
0.001	160	2.585e+09	$0.9973 \pm 8.1\times10^{-5}$	19017.0	8010.1
0.0025	187	2.551e+09	$0.9973 \pm 7.7\times10^{-5}$	21433.5	7899.2
0.005	220	2.498e+09	$0.9978 \pm 3.5\times10^{-4}$	1069.1	7754.7
0.0075	270	2.559e+09	$0.9980 \pm 2.3\times10^{-4}$	2437.4	7938.2
0.01	316	2.537e+09	$0.9983 \pm 3.6\times10^{-4}$	962.5	7879.3
0.05	1250	2.476e+09	$1.0028 \pm 3.5\times10^{-4}$	1042.8	7724.8
0.1	1800	2.520e+09	$1.0084 \pm 3.9\times10^{-4}$	851.8	7872.8
0.5	2300	2.506e+09	$1.0281 \pm 4.2\times10^{-4}$	710.5	7854.4
1	2350	2.502e+09	$1.0460 \pm 4.2\times10^{-4}$	710.3	7824.0

Table 4.1: Eigenvalue estimates for fundamental eigenvalue, figure of merit, and time for a relaxed Arnoldi simulation of a 20mfp thick slab. Also shown is the number of active restarts and the total number of particles tracked in the simulation. 5E5 particles were tracked in each non-relaxed iteration.

but increasing the number of particles tracked in a non-relaxed iteration to 5E6—ten times more than in the previous simulations.

In Figure 4.3 and in Table 4.2 the results of these simulations are shown. We see a similar pattern for these as we did with the simulations tracking 5E5 particles per iteration; the eigenvalue estimates are unaffected by relaxation until the relaxation parameter becomes too large at which point the eigenvalue estimate diverges.

These simulations have been repeated using 1E5 and 5E4 particles per iteration. The data is not given here in tables, but it has been plotted. In Figure 4.5 I show the effect of relaxation on the eigenvalue estimates similar to Figure 4.1 and Figure 4.3, but have plotted for the different numbers of particles per iteration each with a different line dashing. Plotting these together helps to identify the effect of running more or fewer particles per iteration. We see that all cases can accurately estimate the eigenvalues for small values of η, but as η becomes large the eigenvalue estimate will diverge. The value of η at the point where the eigenvalue estimate diverges ($\eta \approx 0.005$) depends upon the number of particles tracked in an iteration; the greater the number of particles tracked, the larger η can be and still obtain a good estimate of the eigenvalue.

In Figure 4.6 the figures of merit for calculating the fundamental eigenvalue for the four cases of different particles tracked per iteration have been plotted. We see what we have already seen previously, that at roughly the same value of η where the eigenvalue estimate begins to diverge, the figure of merit becomes small. The highest figure of merit occurs for values of η just smaller than that for which the eigenvalue estimate diverges. This is not too surprising; as η increases, the number of particles tracked in an iteration decreases and the computational time decreases. Eventually the increase in variance due to poor estimates of the eigenvalue become the dominate factor in the figure of merit.

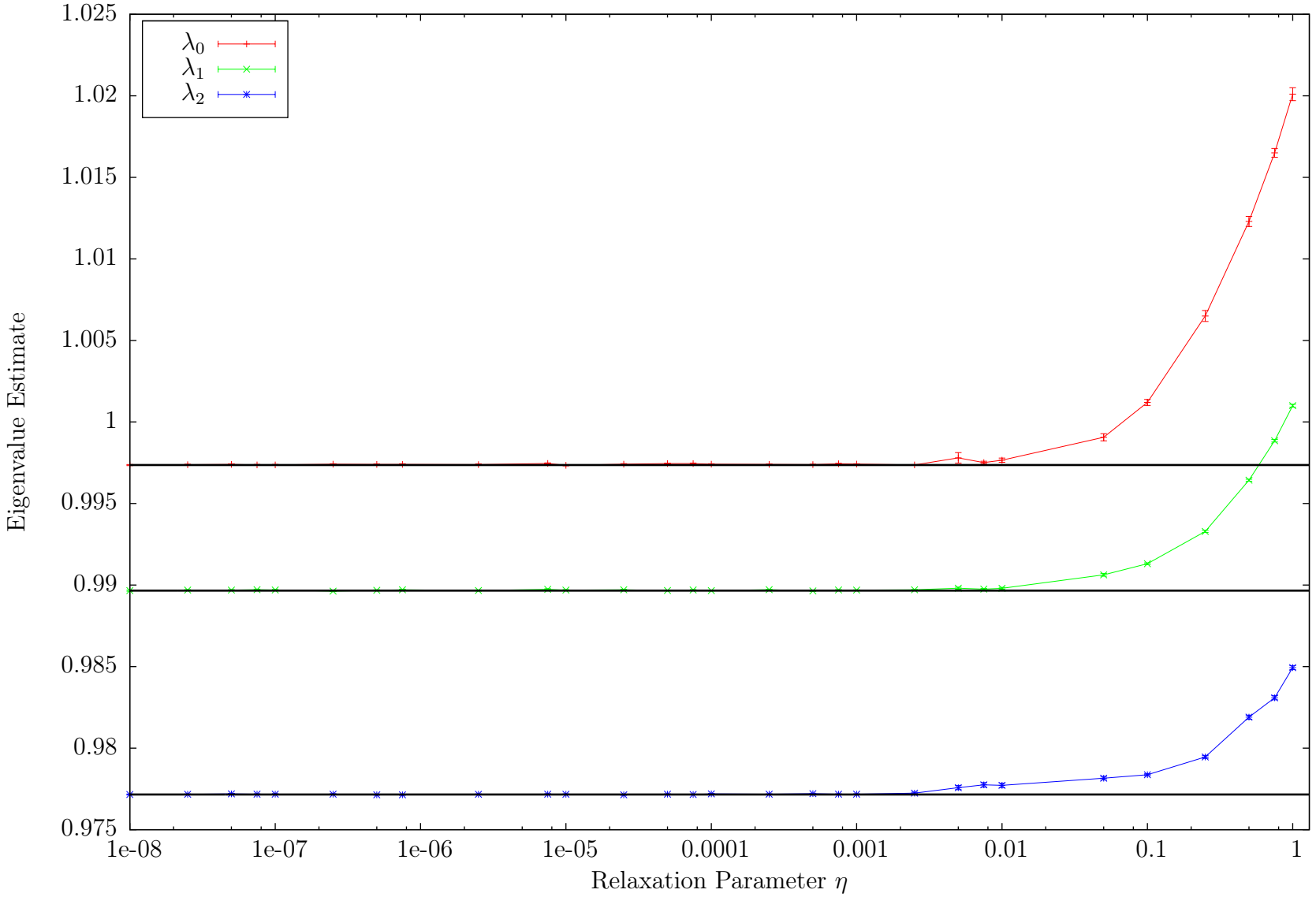


Figure 4.3: Eigenvalue estimates for the fundamental and first two harmonics for varying values of the relaxation parameter η. The number of particles tracked in a non-relaxed iteration is 5E6. The heavy lines are the reference eigenvalues from [12] and [9].

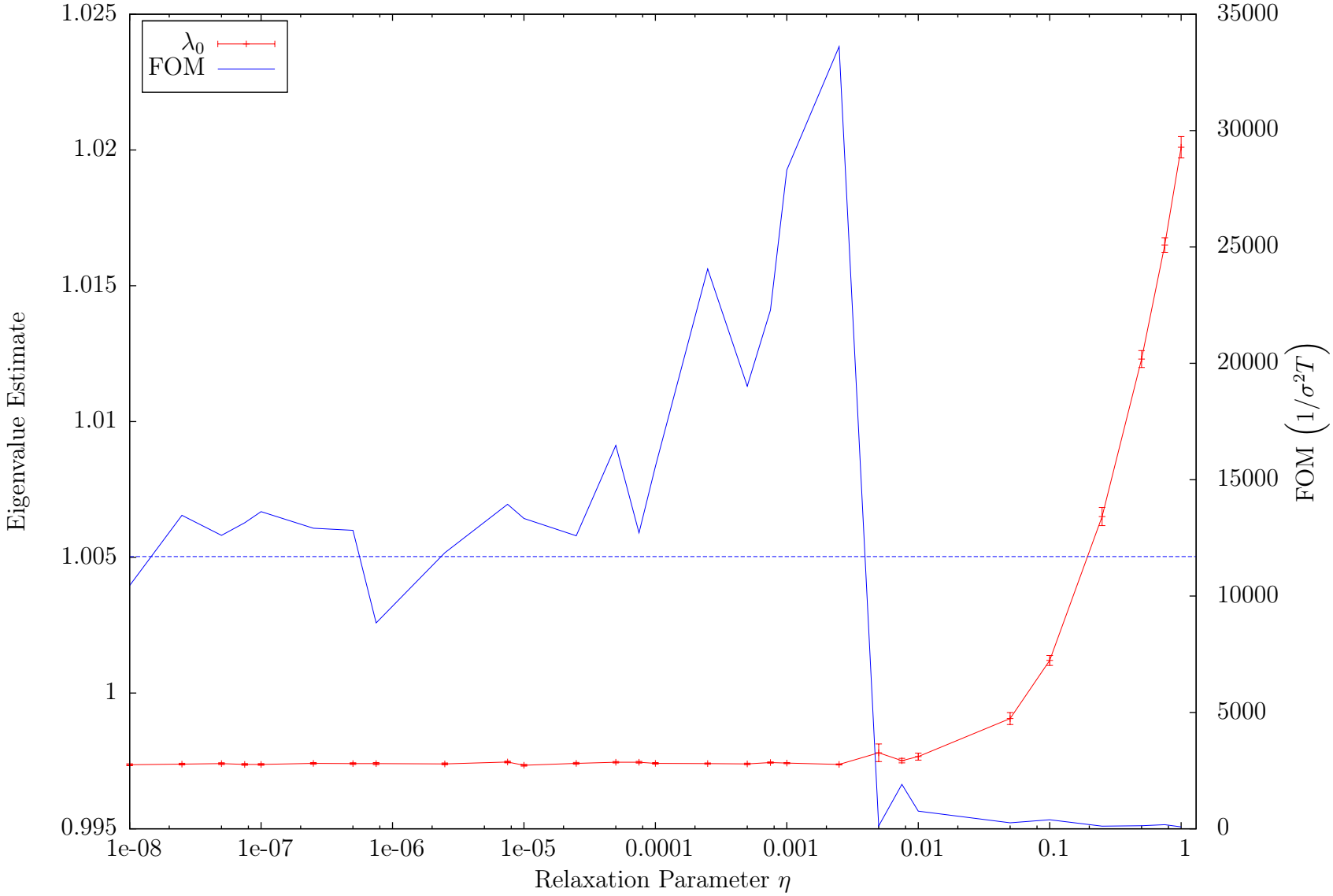


Figure 4.4: Fundamental eigenvalue estimate and figure of merit for varying values of the relaxation parameter η. The dashed line is the value of the figure of merit when there is no relaxation. The number of particles tracked in a non-relaxed iteration is 5E6.

η	Active Restarts	Total # Particles	λ	FOM	Time (s)
0	100	2.500e+10	$0.9974 \pm 3.3\times10^{-5}$	11697.8	78476.2
1E-08	100	2.500e+10	$0.9974 \pm 3.5\times10^{-5}$	10452.8	78779.5
2.5E-08	100	2.500e+10	$0.9974 \pm 3.1\times10^{-5}$	13470.2	78307.7
5E-08	100	2.500e+10	$0.9974 \pm 3.2\times10^{-5}$	12606.0	77916.5
7.5E-08	100	2.500e+10	$0.9974 \pm 3.1\times10^{-5}$	13150.2	78104.0
1E-07	100	2.500e+10	$0.9974 \pm 3.1\times10^{-5}$	13628.3	78784.8
2.5E-07	100	2.500e+10	$0.9974 \pm 3.1\times10^{-5}$	12913.8	78219.4
5E-07	100	2.500e+10	$0.9974 \pm 3.2\times10^{-5}$	12823.6	77835.0
7.5E-07	100	2.500e+10	$0.9974 \pm 3.8\times10^{-5}$	8847.1	78099.9
2.5E-06	100	2.500e+10	$0.9974 \pm 3.3\times10^{-5}$	11857.0	78717.5
7.5E-06	100	2.497e+10	$0.9975 \pm 3.0\times10^{-5}$	13944.0	78135.7
1E-05	100	2.492e+10	$0.9973 \pm 3.1\times10^{-5}$	13334.8	77898.4
2.5E-05	100	2.452e+10	$0.9974 \pm 3.2\times10^{-5}$	12592.5	76967.0
5E-05	118	2.544e+10	$0.9975 \pm 2.8\times10^{-5}$	16472.7	79677.3
7.5E-05	125	2.550e+10	$0.9974 \pm 3.1\times10^{-5}$	12717.2	79890.8
0.0001	129	2.544e+10	$0.9974 \pm 2.9\times10^{-5}$	15565.5	78825.1
0.00025	153	2.560e+10	$0.9974 \pm 2.3\times10^{-5}$	24056.5	79514.8
0.0005	172	2.556e+10	$0.9974 \pm 2.6\times10^{-5}$	19015.5	79022.0
0.00075	186	2.548e+10	$0.9974 \pm 2.4\times10^{-5}$	22290.5	78981.0
0.001	194	2.515e+10	$0.9974 \pm 2.1\times10^{-5}$	28311.4	78137.4
0.0025	237	2.523e+10	$0.9974 \pm 2.0\times10^{-5}$	33610.1	77832.7
0.005	272	2.534e+10	$0.9978 \pm 3.2\times10^{-4}$	121.7	78369.7
0.0075	307	2.470e+10	$0.9975 \pm 8.3\times10^{-5}$	1906.6	76614.9
0.01	395	2.525e+10	$0.9977 \pm 1.3\times10^{-4}$	758.7	78486.7
0.05	1571	2.515e+10	$0.9991 \pm 2.2\times10^{-4}$	256.2	81338.9
0.1	2024	2.487e+10	$1.0012 \pm 1.8\times10^{-4}$	389.9	77344.4
0.25	2291	2.497e+10	$1.0065 \pm 3.3\times10^{-4}$	113.5	78881.1
0.5	2361	2.504e+10	$1.0123 \pm 3.1\times10^{-4}$	132.7	78254.0
0.75	2377	2.503e+10	$1.0165 \pm 2.7\times10^{-4}$	175.2	78251.1
1	2385	2.503e+10	$1.0201 \pm 3.9\times10^{-4}$	81.1	79328.2

Table 4.2: Eigenvalue estimates for fundamental eigenvalue, figure of merit, and time for a relaxed Arnoldi simulation of a 20mfp thick slab. Also shown is the number of active restarts and the total number of particles tracked in the simulation. In non-relaxed iterations, 5E6 particles were tracked.

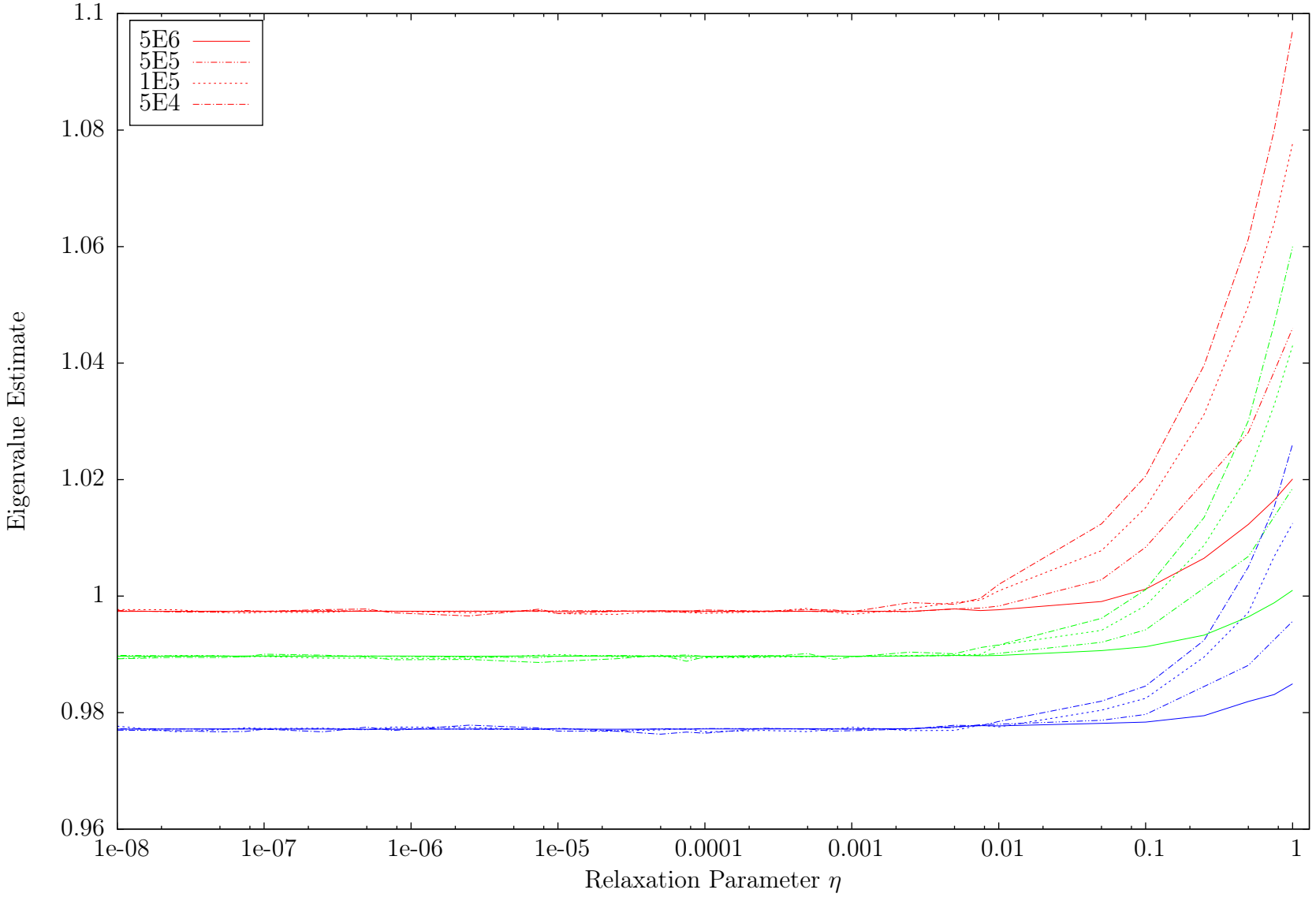


Figure 4.5: Eigenvalue estimates for the fundamental and first two harmonics for varying values of the relaxation parameter η. The different curves indicate the number of particles tracked in a non-relaxed iteration.

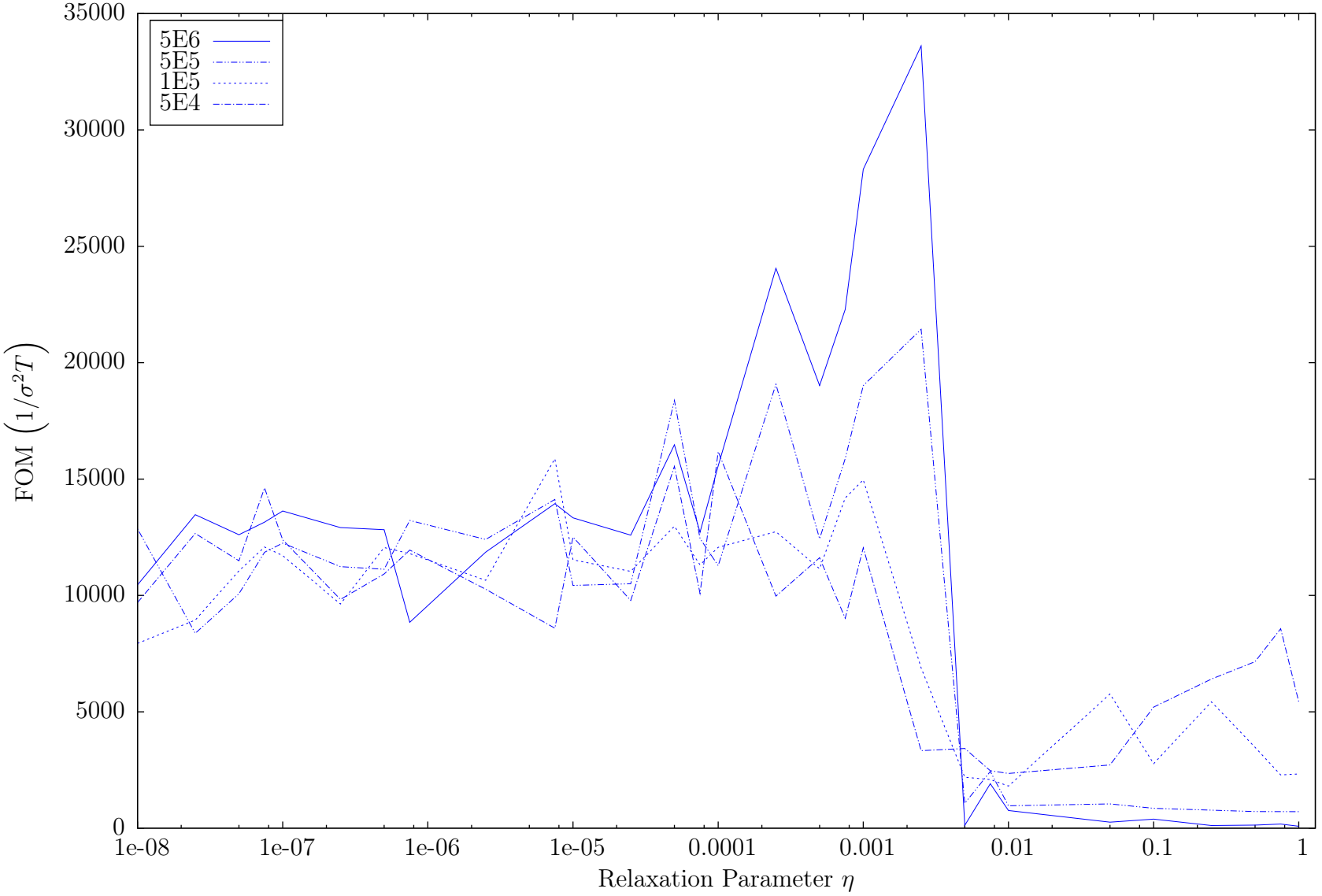


Figure 4.6: Figure of merit for varying values of the relaxation parameter η. The different curves indicate the number of particles tracked in a non-relaxed iteration.

4.2.1 Linear relaxation

With Equation (4.14) it was mentioned that different relaxation strategies could be employed. All the results shown so far have used a quadratic relaxation as given in Equation (4.13). To demonstrate the effect using a different relaxation strategy, I have performed simulations using a linear relaxation strategy for the same problems that have already been shown using the quadratic relaxation. (See also Conlin and Holloway [8].) In this section is shown the results of this strategy and a brief comparison between the two methods.

Any relaxation strategy begins to relax at the same time, whenever the residual is less than η. A quadratic strategy will reduce the number of particles tracked in an iteration more aggressively than will a linear strategy. Figure 4.7 shows the eigenvalue estimates for both linear and quadratic relaxation strategies. (The data for the linear strategy is given in Table 4.3.) The quadratic results are shown with the solid, colored lines while the linear results used the dashed lines. We see that for the quadratic strategy for relaxation the eigenvalue estimates begin to diverge from the true eigenvalue for smaller values of η as compared to the linear strategy.

Figure 4.8 shows the figure of merit for both strategies. The dashed lines show the figure of merit for an Arnoldi calculation with no relaxation. Both strategies demonstrate similar behavior in that a small amount of relaxation is actually worse than no relaxation at all. The range of values of η for which relaxation may benefit the calculation is larger for the linear strategy.

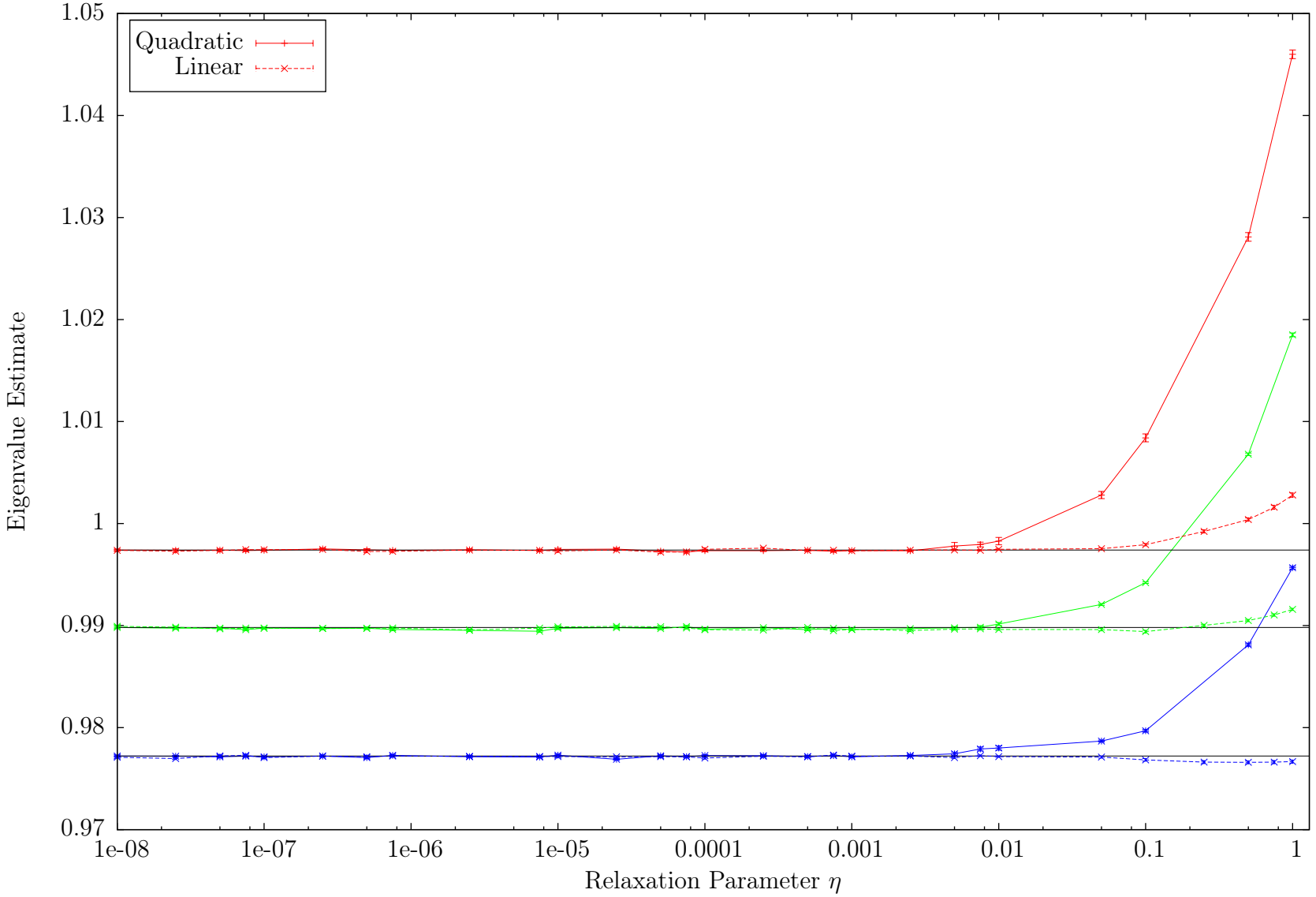


Figure 4.7: Eigenvalue estimates for the fundamental and first two harmonics for varying values of the relaxation parameter η. The number of particles tracked in a non-relaxed iteration is 5E5. The solid lines show a quadratic approach to relaxing and the dashed lines show a linear approach. The black lines are the reference eigenvalues from [12] and [9].

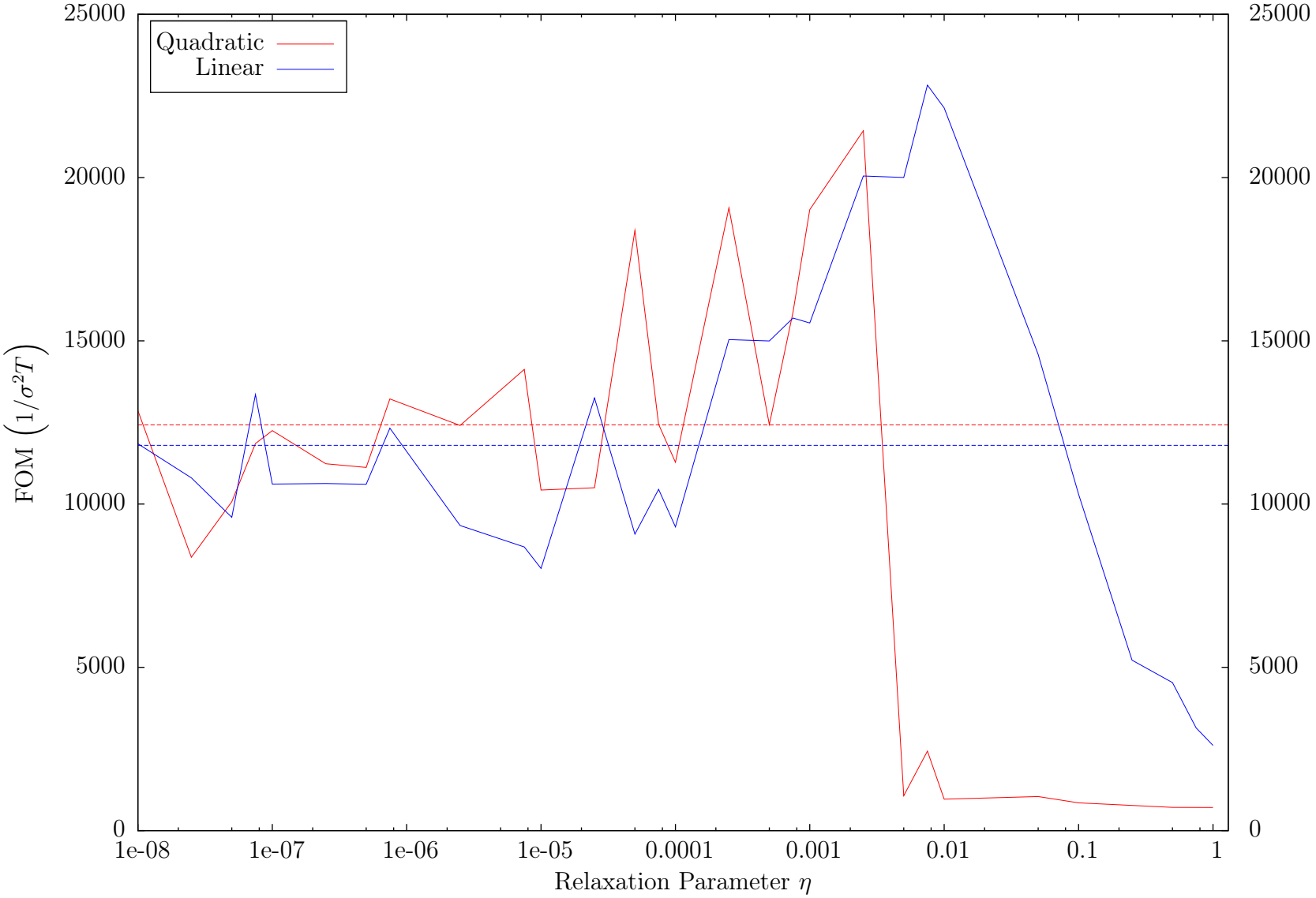


Figure 4.8: Fundamental eigenvalue estimate and figure of merit for varying values of the relaxation parameter η. The red lines show a quadratic approach to relaxing and the blue lines show a linear approach. The dashed lines is the value of the figure of merit when there is no relaxation. The number of particles tracked in a non-relaxed iteration is 5E5.

η	Active Restarts	Total # Particles	λ	FOM	Time (s)
0	100	2.500e+09	$0.9975 \pm 1.0\times10^{-4}$	11799.5	7806.2
1E-08	100	2.500e+09	$0.9974 \pm 1.0\times10^{-4}$	11845.8	7795.1
2.5E-08	100	2.500e+09	$0.9973 \pm 1.1\times10^{-4}$	10801.6	7812.4
5E-08	100	2.500e+09	$0.9974 \pm 1.2\times10^{-4}$	9595.2	7817.4
7.5E-08	100	2.500e+09	$0.9974 \pm 9.8\times10^{-5}$	13357.0	7790.3
1E-07	100	2.500e+09	$0.9974 \pm 1.1\times10^{-4}$	10614.9	7796.8
2.5E-07	100	2.500e+09	$0.9975 \pm 1.1\times10^{-4}$	10629.2	7793.8
5E-07	100	2.500e+09	$0.9973 \pm 1.1\times10^{-4}$	10606.7	7818.0
7.5E-07	100	2.500e+09	$0.9973 \pm 1.0\times10^{-4}$	12324.6	7917.3
2.5E-06	100	2.500e+09	$0.9974 \pm 1.2\times10^{-4}$	9345.4	7809.8
7.5E-06	100	2.500e+09	$0.9974 \pm 1.2\times10^{-4}$	8686.7	7801.4
1E-05	100	2.500e+09	$0.9973 \pm 1.3\times10^{-4}$	8029.7	7803.3
2.5E-05	100	2.497e+09	$0.9974 \pm 9.8\times10^{-5}$	13247.9	7800.2
5E-05	100	2.484e+09	$0.9972 \pm 1.2\times10^{-4}$	9080.7	7764.0
7.5E-05	100	2.471e+09	$0.9972 \pm 1.1\times10^{-4}$	10451.4	7706.1
0.0001	100	2.437e+09	$0.9975 \pm 1.2\times10^{-4}$	9302.1	7596.6
0.00025	114	2.493e+09	$0.9976 \pm 9.2\times10^{-5}$	15038.1	7785.9
0.0005	124	2.481e+09	$0.9974 \pm 9.3\times10^{-5}$	14997.3	7766.4
0.00075	133	2.505e+09	$0.9974 \pm 9.0\times10^{-5}$	15698.0	7889.1
0.001	140	2.491e+09	$0.9973 \pm 9.1\times10^{-5}$	15544.8	7781.8
0.0025	166	2.492e+09	$0.9974 \pm 8.0\times10^{-5}$	20047.8	7783.3
0.005	194	2.472e+09	$0.9974 \pm 8.0\times10^{-5}$	20003.4	7728.1
0.0075	224	2.475e+09	$0.9974 \pm 7.5\times10^{-5}$	22830.9	7746.0
0.01	262	2.501e+09	$0.9975 \pm 7.6\times10^{-5}$	22131.4	7840.6
0.05	817	2.520e+09	$0.9975 \pm 9.3\times10^{-5}$	14582.1	7871.5
0.1	1173	2.483e+09	$0.9979 \pm 1.1\times10^{-4}$	10302.1	7832.0
0.25	1649	2.503e+09	$0.9992 \pm 1.6\times10^{-4}$	5219.9	7918.0
0.5	1917	2.506e+09	$1.0004 \pm 1.7\times10^{-4}$	4533.4	7921.6
0.75	2037	2.515e+09	$1.0016 \pm 2.0\times10^{-4}$	3143.8	7944.6
1	2090	2.496e+09	$1.0028 \pm 2.2\times10^{-4}$	2609.6	7953.4

Table 4.3: Eigenvalue estimates for fundamental eigenvalue, figure of merit, and time for a relaxed Arnoldi simulation of a 20mfp thick slab. The relaxation for this simulation is linear. Also shown is the number of active restarts and the total number of particles tracked in the simulation. 5E5 particles were tracked in each non-relaxed iteration.

4.3 Summary

Relaxing the precision to which a linear operator is applied to a vector is an interesting topic. Bouras and Frayssè have shown—as well as many others citing their work—that relaxing Arnoldi's method can reduce the computational expense of Arnoldi's method while maintaining the convergence properties.

We have also seen that relaxation can reduce computational expense for Monte Carlo Arnoldi's method. This allows us to perform more restarts and calculate more eigenvalue estimates for a relaxed Arnoldi's method in the same amount of time required for a non-relaxed Arnoldi's method. The problem however is that it can be difficult to determine the optimal value for the relaxation parameter η. For the problems shown here, the range of values that gives better results than a non-relaxed Arnoldi calculation is small. If a poor value of η is chosen, it can cause the calculation to be slow or even wrong.

We have seen that for a variety of particles tracked in an iteration, the eigenvalue estimate still diverges and the figure of merit plummets when η becomes too large, but with more particles tracked η can be larger than with fewer particles tracked without any problems in calculating the eigenvalue estimate. The figure of merit can be larger when tracking more particles. Even so the range of good values of η is relatively small.

Chapter 5

Fission Source Convergence

Monte Carlo criticality calculations have come a long way since their beginning in the 1950s. Along with the increase in the speed and complexity of computers the problems for which Monte Carlo techniques have been used have increased. Problems that were prohibitively expensive in the past have now become routine. Some problems still remain, however, with Monte Carlo criticality calculations. Two of those problems are fission source convergence and the underestimation of the variance of the mean eigenvalue [see 6]. In this chapter we will discuss how a Monte Carlo application of Arnoldi's method improves the source convergence as compared with the traditional power method.

5.1 Shannon Entropy and Fission Source Convergence

In Section 1.3 it was observed that a Monte Carlo code must discard initial iterations as the eigenvalue estimate converges to the fundamental eigenmode. The number of iterations required for convergence of the power method to the fundamental depends on the dominance ratio (λ_1/λ_0) for the problem. Also of concern is the convergence of the fission source. To obtain good estimates of the fission source—or eigenfunction of

the fission-transport operator—the fission source should be converged before tallying begins.

Knowing how many cycles to discard can be a challenge. One could monitor the eigenvalue estimates to see if they have converged and begin active cycles once convergence has been achieved. Brown [6] suggests that when the dominance ratio is close to 1, the fundamental eigenvalue will converge sooner than the fission source distribution. We must, therefore, monitor *both* the convergence of the eigenvalue as well as the convergence of the fission source. It is clear that convergence in fewer iterations is desired because fewer iterations will have to be discarded and tallying can begin sooner.

One problem for which source convergence in the power method is problematic is a geometry with multiplying regions separated by large region of a highly-absorbing material. This is because of the particles born in one multiplying region only a few, rare, neutrons would ever transport through the absorbing region to cause fission in the other multiplying region. In the power method, the only way in which particles can move from the initial source to a new location is through the scattering that occurs in the application of the fission-transport operator. The fission source for the next iteration contains only the points at which neutrons induced fission in the previous iteration.

When restarting, Arnoldi's method uses all of the available information about the operator applied to a source to estimate an eigenvector; the power method only uses information from the latest application of the operator to estimate the fundamental eigenvector. Using all available information makes it possible for the fission source to be converged more quickly.

To monitor the convergence of the fission source Ueki and Brown [27] have used information theory to calculate the Shannon entropy, H, of the fission source. The

Shannon entropy of a spatially discretized fission source $S(x)$ is defined as

$$H(S(x)) \equiv -\sum_{b=1}^{B} S_b \log(S_b), \tag{5.1}$$

where B is the number of spatial bins and S_b is the number of fission neutrons in bin b. As the calculation proceeds, the Shannon entropy will converge to some value. The actual value of the Shannon entropy is not important in determining the fission source convergence. What is important is that it converges to some value.

To calculate the Shannon entropy when using the power method the fission source must be discretized. This discretization is most easily performed using a first order—or histogram—approximation as described in Equations (3.1), (3.2) and (3.3). The discretization of the fission source will not bias the estimate of the eigenvalue in the power method as it does for Arnoldi's method. The power method samples from the non-discretized source and only uses the discretized fission source for tallying bins and calculating the Shannon entropy.

In Arnoldi's method, we can't calculate the Shannon entropy of the Arnoldi vectors. We must, therefore, calculate the Ritz vectors of $\mathcal{A}$ and use the fundamental Ritz vector as $S(x)$ in Equation (5.1). With the first order approximation to the fission source

$$S_b = \int_{x_b}^{x_{b+1}} \left|\Pi_b(x)\right| \mathrm{d}x. \tag{5.2}$$

This is equivalent to calculating the Shannon entropy for the Power method. When using a second order approximation,

$$S_b = \int_{x_b}^{x_{b+1}} \left|\mathcal{L}_b(x, \alpha_b, \beta_b)\right| \mathrm{d}x. \tag{5.3}$$

In both Equation (5.2) and Equation (5.3) S_b is the number of neutrons that have tallied in bin b.

5.2 Numerical Results

Homogeneous Slab

To demonstrate the eigenvalue and fission source convergence both the power method and Arnoldi's method have been run on a homogeneous slab, 50 mfp thick with $\nu\Sigma_f = 1.0$, $\Sigma_a = 0.2$, and $\Sigma_s = 0.8$; $\Sigma_t = 1.0$. This geometry is the same as in Chapter 4. Arnoldi's method has 25 iterations per restart and both methods track 5E5 particles per iteration. The dominance ratio for this problem as estimated by Arnoldi's method is 0.992201. For this problem I have used a second order approximation to the fission source as described in Chapter 3.

In Figure 5.1a I show the eigenvalue estimates (red) and Shannon entropy convergence (blue) from the power method. The solid black line shows the reference value of 0.997520 from a deterministic S32 code with 200 equally-spaced regions. The dashed black line shows the entropy at the end of the simulation. Figure 5.1b shows the eigenvalue estimate and Shannon entropy convergence as calculated by Arnoldi's method. In this figure, the dashed black line is the mean value of the Shannon entropy.

From Figure 5.1a and Figure 5.1b, we see that Arnoldi's method converges immediately for both the eigenvalue estimate and the fission source. The power method, however, requires approximately 250 iterations to converge. We do see something different in these figures that we did not see before. In Figures 5.4a–5.3b, it appears that the spread in the eigenvalue estimates was approximately the same between Arnoldi's method and the power method. For the homogeneous slab, however, the spread in the eigenvalue estimates from the power method is 0.00052 while the spread from Arnoldi's method is approximately 0.0012—the eigenvalue estimates from Arnoldi's method are less noisy than the estimates from the power method.

Thus we see that Arnoldi's method is superior to the power method for eigenvalue and fission source convergence, both converge immediately. The eigenvalue estimates from Arnoldi's method are also less noisy that those from the power method.

Heterogeneous Slab

We now turn to a heterogeneous geometry used in Kornreich and Parsons [17] and Ueki and Brown [28]. Two problems are studied in these papers. The first problem is a symmetrical, five-region slab with vacuum boundary conditions. The slab widths (left-to-right) are 1.0, 1.0, 5.0,1.0, 1.0 cm thick. The second problem is identical except the right most fuel slab has a width of 1.01 cm. The materials are of three types (left-to-right): fuel, scatterer, absorber, scatterer, fuel. The cross sections for these materials are

Fuel $\Sigma_t = 1.0$, $\Sigma_s = 0.8$, $\Sigma_\gamma = 0.1$, $\Sigma_f = 0.1$, $\nu = 3.0$,

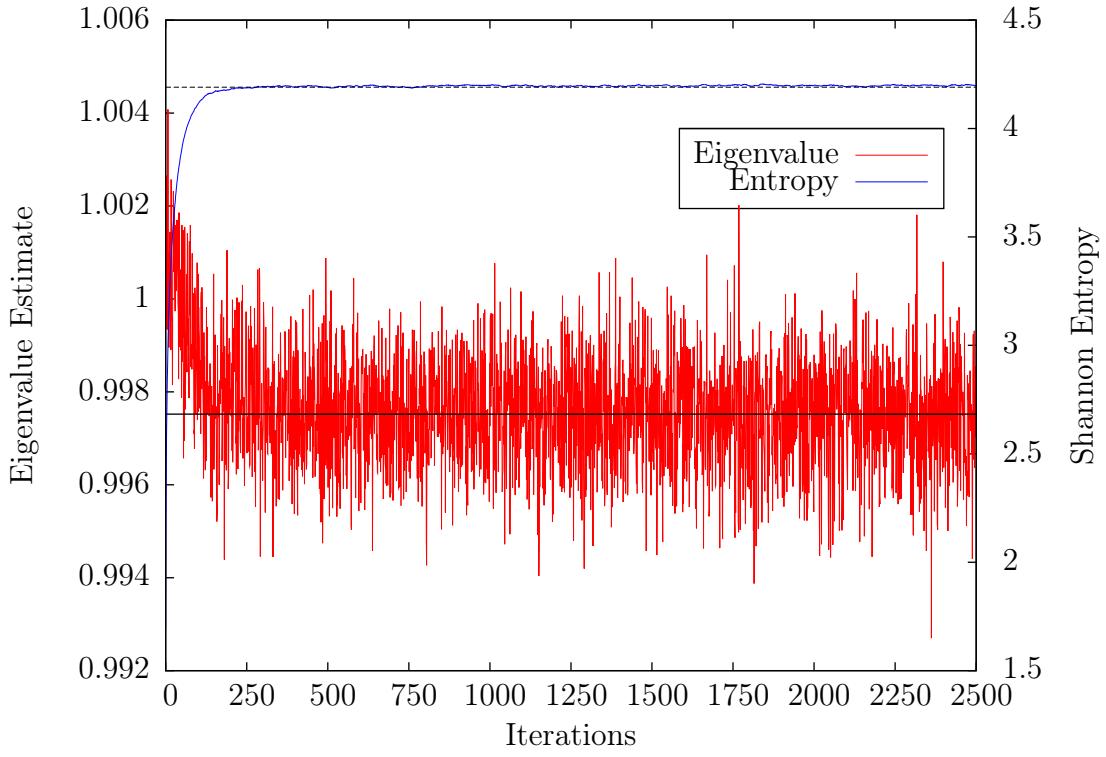
Scatterer $\Sigma_t = 1.0$, $\Sigma_s = 0.8$, $\Sigma_a = 0.2$,

Absorber $\Sigma_t = 1.0$, $\Sigma_s = 0.1$, $\Sigma_a = 0.9$.

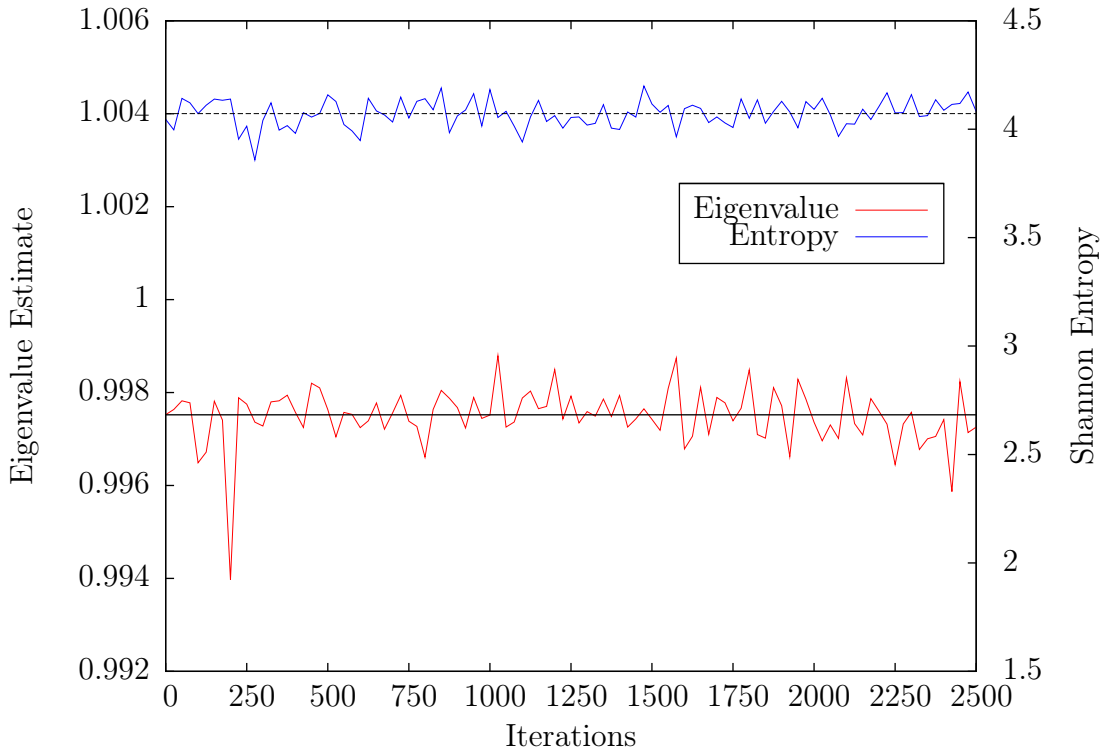
A graphical description of this geometry is shown in Figure 5.2.

This geometry is particularly difficult for source convergence. The initial source for this problem lies entirely in the leftmost bin. With a large highly absorbing slab in the center, it is difficult to move particles from the left to the right side and therefore it will take many histories for the fission source to converge. The dominance ratio for the symmetric problem is 0.999566 and it is 0.992504 for the asymmetric problem. For these simulations, 1E5 particles are tracked in each iteration; Arnoldi's method uses 10 iterations per restart and calculated 2 eigenvalues.

The results for the asymmetric problem will be shown first; with a smaller dominance ratio it should be easier to converge the fission source and eigenvalue. Figure 5.3a shows the convergence of the eigenvalue estimates and fission source as cal-



(a) Power method.



(b) Arnoldi's method.

Figure 5.1: Convergence of eigenvalue estimate and Shannon entropy for 50mfp thick, homogeneous slab. Solid black line is the S_N eigenvalue solution for the fundamental eigenvalue ($\lambda_0 = 0.997520$).

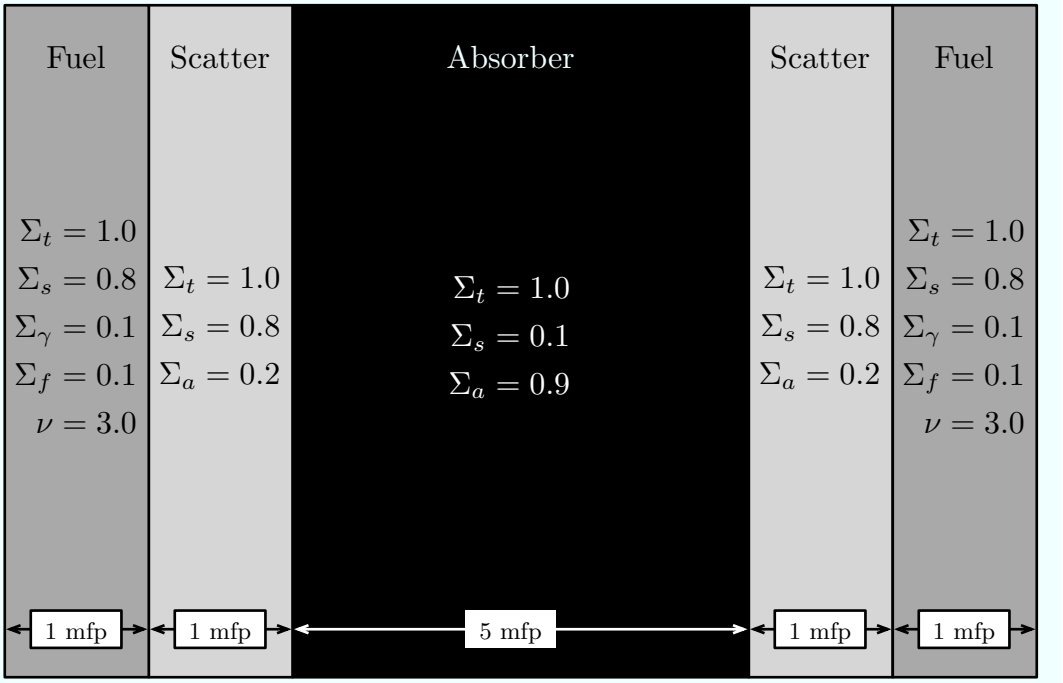
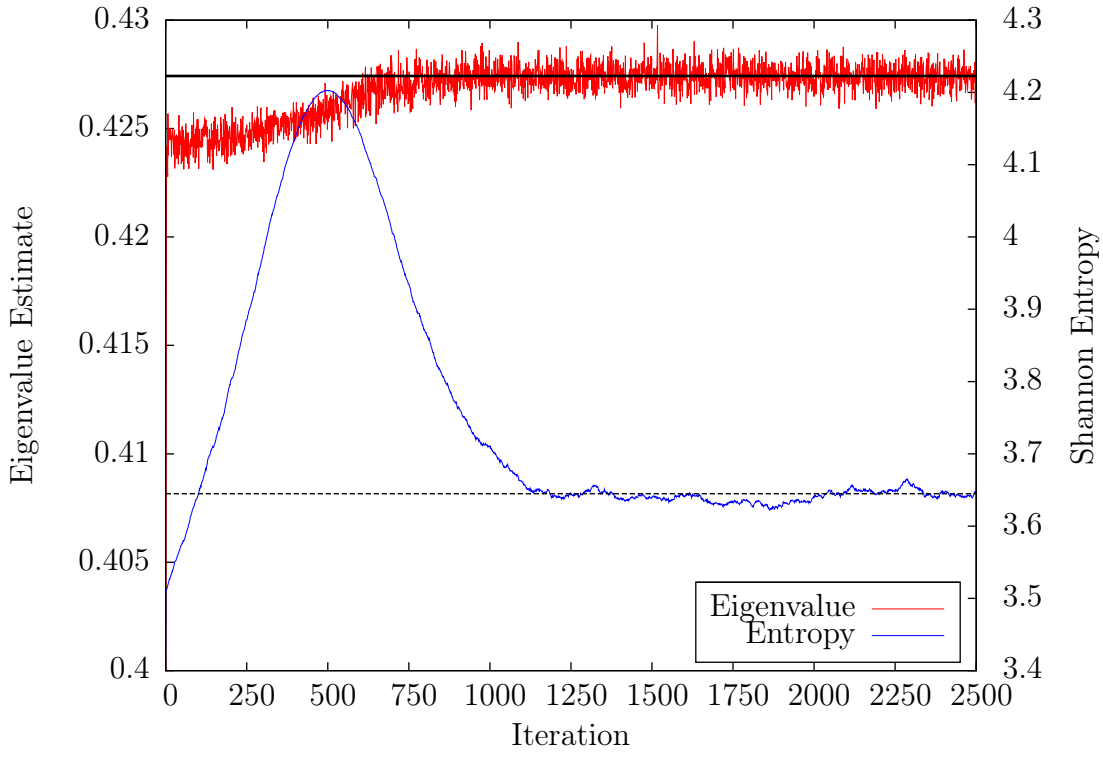


Figure 5.2: Diagram of heterogeneous slab geometry.

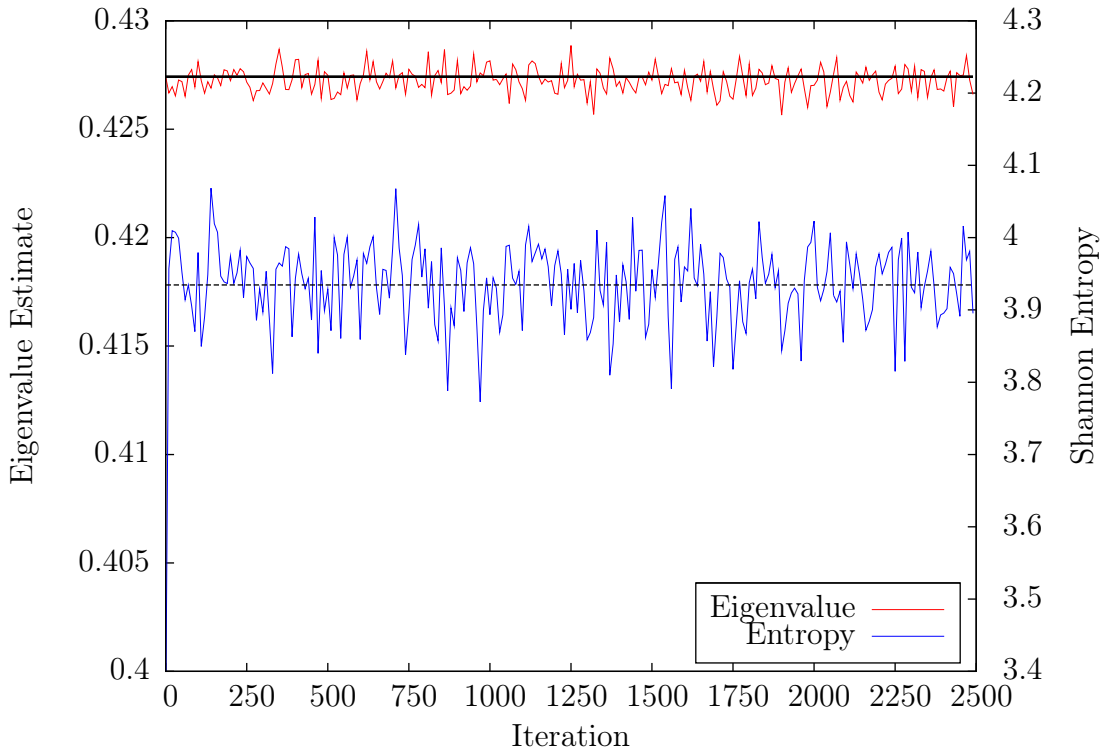
culated by the power method. In Figure 5.3b are shown the eigenvalue estimates and fission source convergence for Arnoldi's method. The solid black line again shows the reference eigenvalue from Kornreich and Parsons [17] which is 0.427425. The dashed line shows the Shannon entropy at the end of the simulation for the power method and the mean of the Shannon entropy for Arnoldi's method.

The differences in convergence between Arnoldi's method and the power method are easy to see by comparing Figure 5.3a and Figure 5.3b. We see that Arnoldi's method has converged both the eigenvalue and fission source almost immediately, just like with the homogeneous slab. The power method however requires 800–900 iterations before the eigenvalue estimates have converged and approximately 1200 iterations before the fission source converges.

The symmetric problem shows a different convergence for the power method as displayed in Figure 5.4a. Figure 5.4b shows convergence of the eigenvalue estimates and fission source as calculated by Arnoldi's method. The solid black lines show the published eigenvalue [17], 0.424316. The dashed black line for the power method



(a) Power method.



(b) Arnoldi's method.

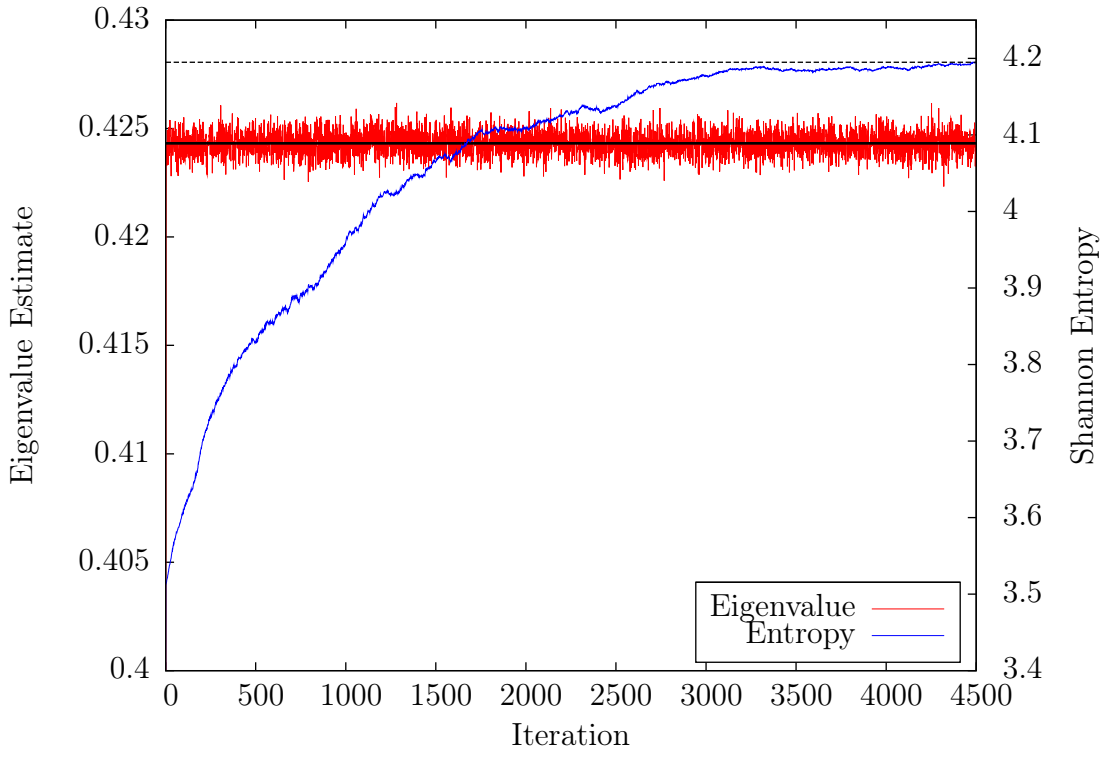
Figure 5.3: Convergence of eigenvalue estimate and Shannon entropy for asymmetric geometry. Solid black line is the fundamental eigenvalue ($\lambda_0 = 0.427425$ [17]); dashed black line is the entropy at the end of the simulation for the power method and mean entropy for Arnoldi's method.

shows the final value of the Shannon entropy. The dashed black line for Arnoldi's method shows the mean value of the Shannon entropy.

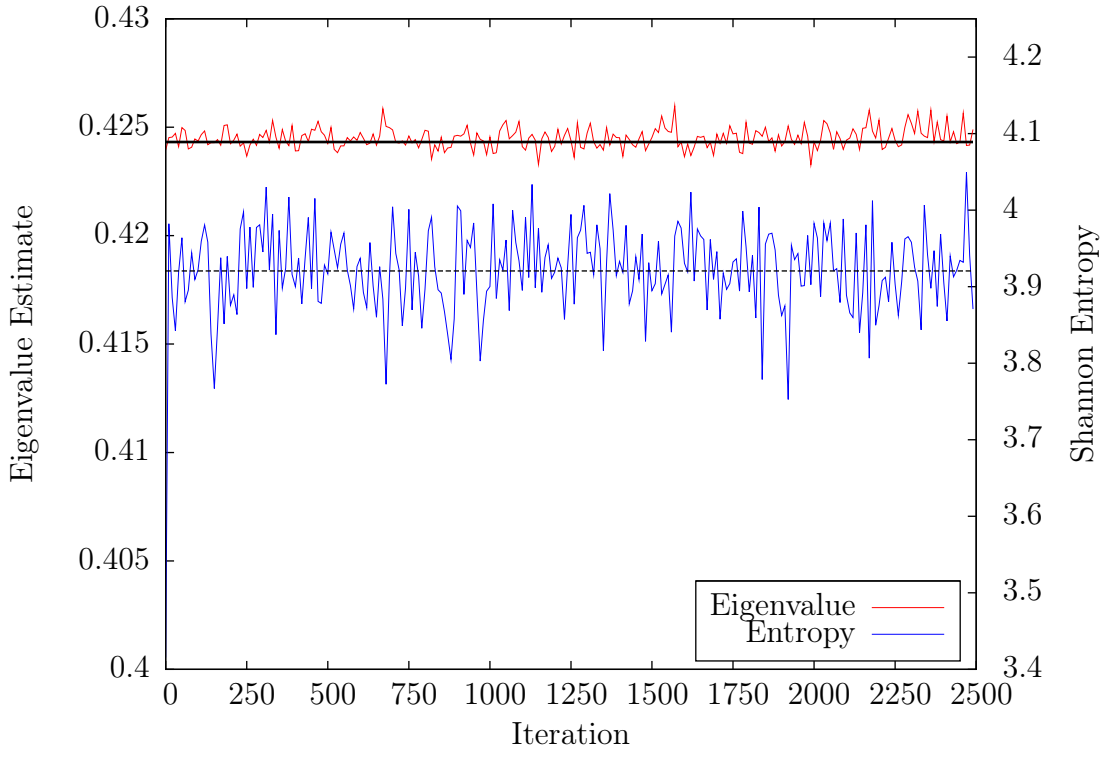
In the power method we see that the eigenvalue estimate has converged almost immediately, but the fission source takes much longer to converge. The power method ran for an extra 2000 iterations and it looks like the fission source might be converged after 4500 iterations, but it looks like the Shannon entropy may still be trending upwards and therefore still has not converged. In Arnoldi's method both the eigenvalue estimate and the fission source from Arnoldi's method have converged almost immediately.

In Figure 5.5 is shown the fundamental eigenvector estimates of the symmetric, heterogeneous problem from the power method and Arnoldi's method. These eigenvectors are the mean of the last 1250 power iterations or 125 Arnoldi restarts. We can see that height of the eigenvector in the fuel regions is not the same, but we know that they must be because the slab is symmetric. Not enough particles are being transported into the right fuel region causing the eigenvector estimate to be smaller in that region.

In Figure 5.6 is shown the fundamental eigenvector for the same symmetric, heterogeneous problem, but this time 1E7 particles were tracked in each iteration. The results for the previous problem are included for comparison. We can see that when more particles are used, more particles reach the right fuel region and the eigenvector estimate becomes larger. We expect that if enough particles are tracked, the eigenvector will become symmetric.



(a) Power Method



(b) Arnoldi's Method

Figure 5.4: Convergence of eigenvalue estimate and Shannon entropy for symmetric geometry. Solid black line is the fundamental eigenvalue ($\lambda_0 = 0.424316$ [17]); dashed black line is the entropy at the end of the simulation for the power method and mean entropy for Arnoldi's method.

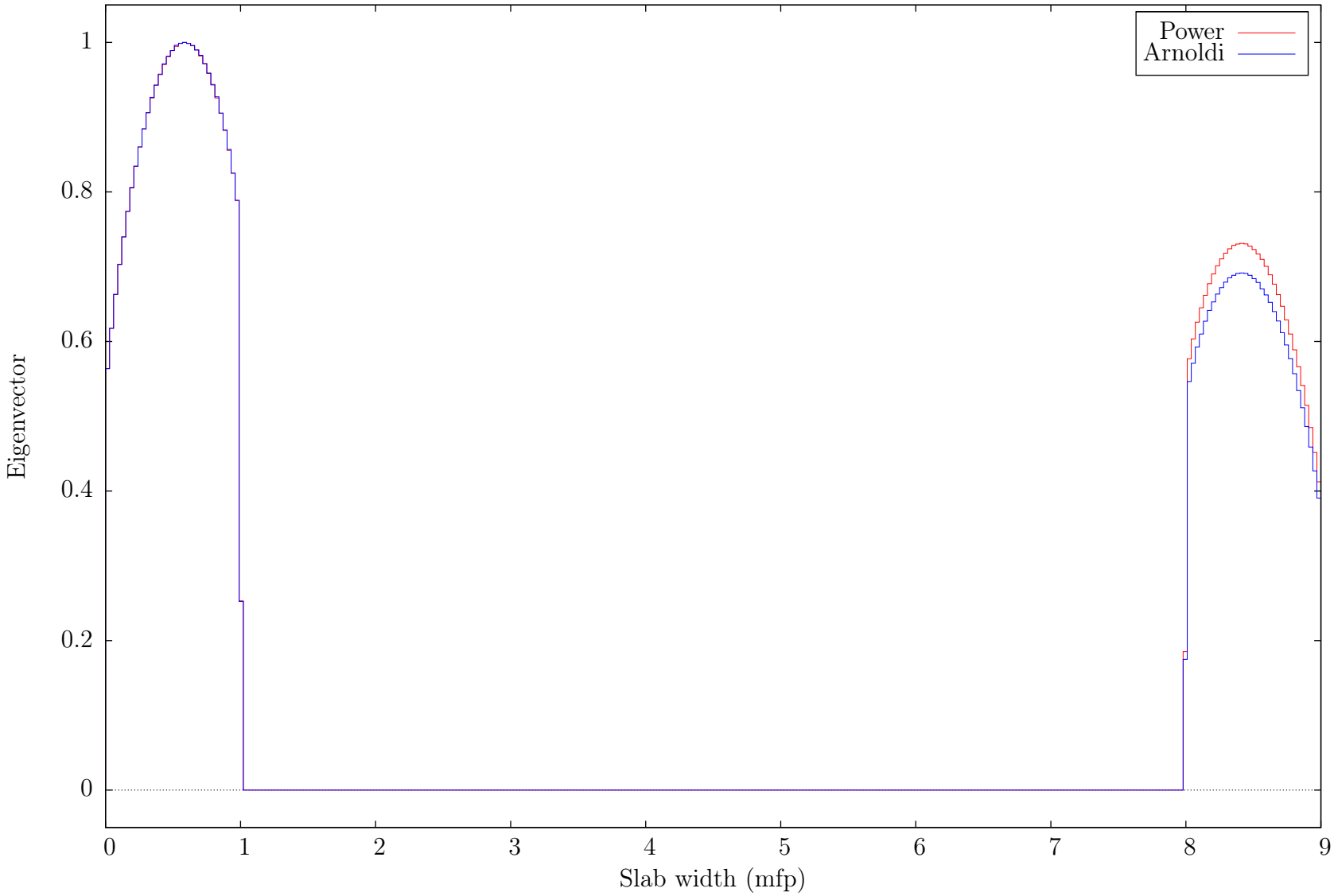


Figure 5.5: Fundamental eigenvector estimates for the symmetric, heterogeneous problem from the power method and Arnoldi's method.

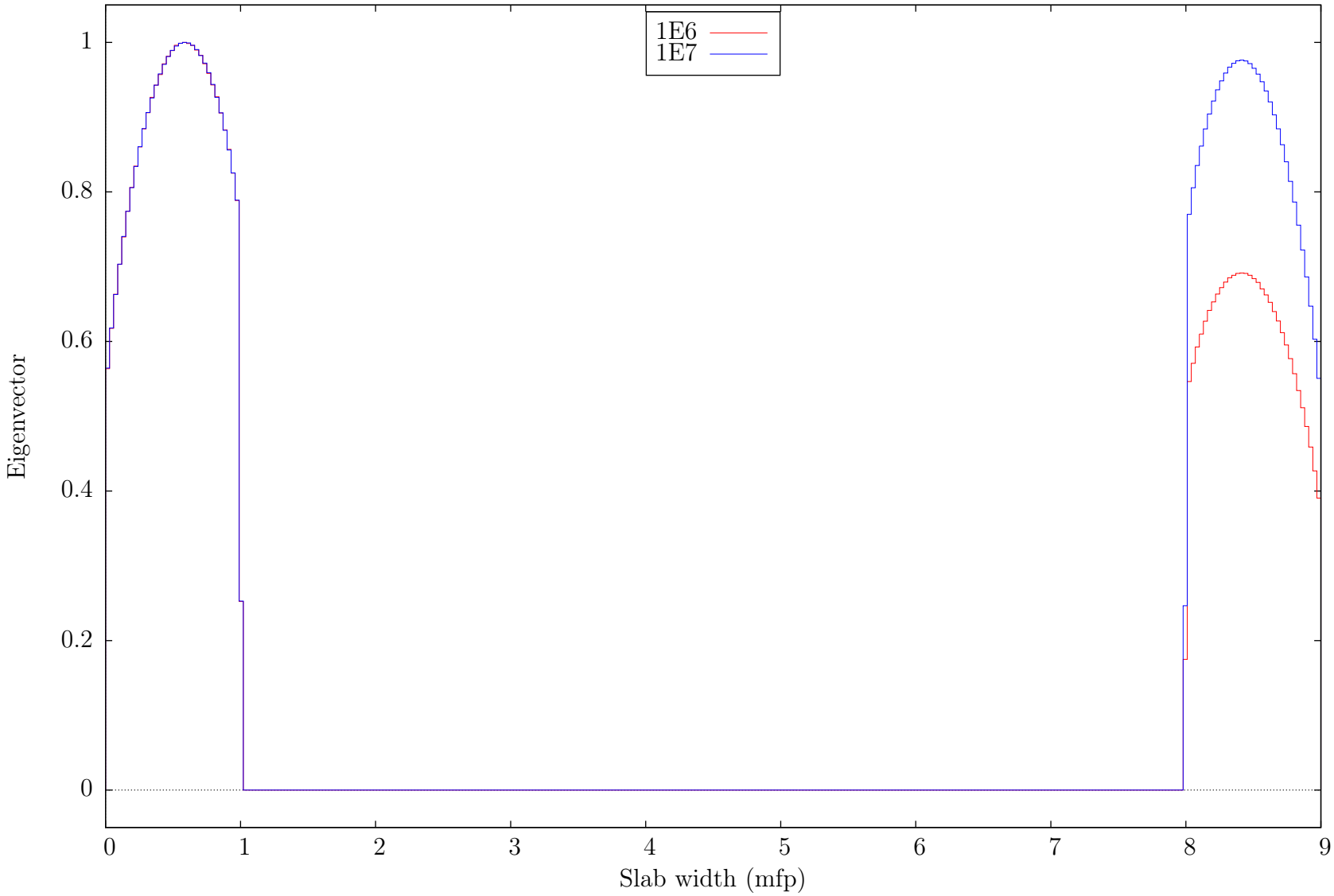


Figure 5.6: Fundamental eigenvector estimates for the symmetric, heterogeneous problem from Arnoldi's method using 1E6 and 1E7 particles per iteration.

5.3 Summary

When doing Monte Carlo particle transport, it is important that the eigenvalue estimates and fission source have converged before we begin tallying. If convergence has not been achieved then estimates of the eigenvalue (or some other tally) which are wrong will be used thus affecting the final result. Using the Shannon entropy of the fission source, it is easy to monitor the convergence of both the eigenvalue estimate and fission source. Brown [6] suggests that the number of inactive iterations can be estimated by running the simulation with a small number of particles tracked in an iteration and see how many iterations are required for convergence.

Regardless of how the number of necessary inactive iterations is determined, the inactive iterations must still be performed; for Arnoldi's method only one inactive restart is required. Because of its rapid convergence, Arnoldi's method can save computational expense because it does not require the many hundreds of inactive iterations to converge like the power method. Arnoldi's method converges immediately for the three problems shown in this chapter and does not appear to need any inactive iterations.

Chapter 6

Conclusions

In this dissertation I have developed the first Monte Carlo implementation of Arnoldi's method for neutron transport. This implementation uses explicitly restarted Arnoldi's method to estimate multiple eigenvalues of the transport-fission operator of the Boltzmann transport equation (1.1).

Using Arnoldi's method for estimating eigenvalues is a new technique in the Monte Carlo particle transport field; traditionally, the power method has been used. Arnoldi's method has been used in the numerical analysis community for many years to estimate multiple eigenvalues and eigenvectors of a linear operator, but this is the first time it has been used with a Monte Carlo application of the linear operator.

I have demonstrated the ability to use Arnoldi's method to estimate up to three eigenvalues of the transport-fission operator for a variety of homogeneous and heterogeneous one-dimensional problems. The eigenvalue estimates have been compared to and agree with published results and independent deterministic calculations within statistical uncertainty. The eigenvectors have also been estimated and compared with deterministic calculations; again the results from Arnoldi's method are in harmony with the deterministic calculations. In some situations, Arnoldi's estimated eigenvectors are improvements over the power method estimated eigenvectors.

Arnoldi's method can be used to calculate more than three eigenmodes. Calculating additional higher-order eigenmodes will require additional iterations in a restart,

and it may be necessary to track more particles in an iteration or use a finer spatial discretization.

Arnoldi's method requires the fission source to be discretized. The simplest way to discretize the fission source is to use a constant in space or first-order accurate spatial approximation of the fission source which can, unfortunately, cause an error in the eigenvalue estimate if too few spatial bins are used to discretize the source. I have implemented a second order accurate approximation to the fission source and have used it to reduce the error in the eigenvalue calculation by an order of magnitude for the same number of particles tracked. The eigenvector estimates from the second order accurate approximation are a great improvement over the first order accurate approximation. Rather than a jagged, step-wise approximation to the eigenvector, the second-order accurate approximation is a smooth and nearly continuous function across bins. In addition, the figure of merit for the second-order accurate approximation is 2–3 times larger than the first order accurate approximation.

I have investigated relaxing the precision to which the transport-fission operator is applied at every Arnoldi iteration. Studies have shown that relaxing Arnoldi's method has no effect on the convergence to the correct eigenvalues. Relaxing Arnoldi's method in a Monte Carlo particle transport application simply involves tracking fewer particles in an iteration than was initially specified; tracking fewer particles is less computationally expensive and reduces the overall time for the simulation.

Relaxing Arnoldi's method for Monte Carlo criticality applications can save on computation time. Relaxing too much, however, can cause the eigenvalue estimates to be incorrect. In addition, relaxing Arnoldi's method only a little can cause the figure of merit to be smaller than for a non-relaxed Arnoldi's method. Relaxing Arnoldi's method turns out to be not a sufficient improvement to make it worthwhile to use in practice.

Two important topics currently being investigated in the Monte Carlo particle transport community are the underestimation of the variance of the mean eigenvalue estimate and the convergence of the fission source. The power method *under*estimates the variance because it ignores the correlation between power method iterations. My implementation of Arnoldi's method also ignores the inter-iteration correlation, but the reported variance appears to be more conservative. In one of the problems discussed in this dissertation, I have shown that Arnoldi's method *over*estimates the uncertainty in the eigenvalue estimate by approximately 10%.

The power method can take a long time to converge the fission source—especially for problems with a large dominance ratio. In Monte Carlo criticality calculations, both the eigenvalue estimate and the fission source must be converged before tallying begins. If convergence is slow, more iterations must be discarded and computation wasted. I have shown that Arnoldi's method is superior to the power method in converging both the eigenvalue estimate and the fission source. Arnoldi's method appears to converge both the eigenvalue estimate and the fission source immediately, while the power method can require several hundreds of iterations.

6.1 Future Work

Work on Monte Carlo Arnoldi's method for criticality calculations is far from complete. This dissertation represents the first work performed in this field. Some of the many topics that still need to be explored are described next.

6.1.1 Implicit Restarts

One of the most intriguing ways that restarted Arnoldi's method could be improved is by implementing *implicit* restarts. Implicit restarts were developed by Sorensen [26] as a way to restart Arnoldi's method with an improved starting vector, and at

the same time reduce the computational expense of the algorithm and increase the stability of maintaining orthogonality between Arnoldi vectors.

Implicitly restarted Arnoldi's method (IRAM), as it is called, performs iterations of the shifted QR algorithm on the upper Hessenberg matrix, H_m, in the Arnoldi factorization in exchange of Arnoldi iterations. It can be shown that performing these shifted QR iterations allows Arnoldi's method to jump into the middle of the next restart, skipping several iterations. IRAM is mathematically equivalent to picking a vector for the beginning of a new restart as a linear combination of the eigenvectors associated with the desired region of the spectrum of $\mathcal{A}$ as we did in explicitly restarted Arnoldi's method. The full derivation and proof of implicitly restarted Arnoldi's method is given in Appendix A.

In Arnoldi's method, applying the linear operator $\mathcal{A}$ is the most computationally expensive part, using greater than 80% of the computer cycles in a given iteration. Since the number of iterations in an Arnoldi restart is small, performing QR iterations on H_m will be inexpensive. Trading computationally expensive applications of $\mathcal{A}$ for inexpensive QR iterations on H_m should significantly reduce the computational expense of Arnoldi's method.

6.1.2 Calculating Eigenvalue Estimates at Every Iteration

In Chapter 2 it was suggested that an Arnoldi restart could be treated similarly to the power method; that is, at the end of an Arnoldi restart an estimate for the eigenvalues are calculated and stored. In the power method, an eigenvalue estimate is calculated at every iteration; since multiple iterations make up one Arnoldi restart, the power method has many more eigenvalue estimates than does Arnoldi's method for the same number of iterations and number of particles tracked.

The variance of the mean of the eigenvalue estimates goes as one over the square-root of the number of eigenvalue estimates. Thus, the power method has an advantage

over Arnoldi's method in that it has more eigenvalue estimates—the variance for the mean eigenvalue from the power method will almost certainly be smaller.

There is no reason why an eigenvalue estimate could not be calculated at every Arnoldi iteration instead of just at the end of a restart. Of course, after the first iteration we could only estimate the fundamental eigenvalue; we would have to wait for additional iterations to estimate higher order eigenvalues. It would be slightly more computationally expensive, but the decrease in the variance may be worth the extra expense. In all of the calculations in which Arnoldi's method was directly compared to the power method, the figures of merit from the power method calculations were larger than the figures of merit for Arnoldi's method even though both methods took approximately the same amount of computational time. The figure of merit is larger for the power method because the variance is smaller. Calculating an eigenvalue estimate at every iteration would increase the number of estimates in Arnoldi's method, but is more computationally expensive.

A preliminary test has been performed to see how estimating the eigenvalue at each iteration might work. I have repeated the 20 mfp simulation in Chapter 2 from Arnoldi's method, but instead of estimating three eigenvalues only the fundamental eigenvalue is estimated and only two iterations are done per restart. In each iteration 1E5 particles are tracked. The number of restarts are 125 inactive and 500 active. The total number of particles tracked and the total number of iterations is the same for this simulation and the power method and Arnoldi's method from Chapter 2.

The results of this simulation are given in Table 6.1. The results for the new simulation are given first and denoted with a star. The eigenvalue estimate is within statistical uncertainty of the reference value of $\lambda_0 = 4.82780$. We can see that the standard deviation of this simulation is identical to the power method simulation from Chapter 2, but that the figure of merit is larger.

	λ_0	σ	FOM
Arnoldi*	4.82806	6.3×10^{-4}	6.6×10^{3}
Power	4.82734	6.3×10^{-4}	5.4×10^{3}
Arnoldi	4.8290	1.5×10^{-3}	1.1×10^{3}

Table 6.1: Eigenvalue estimate and figure of merit for Arnoldi's method (Arnoldi*) with just 2 iterations per restart and only saving the fundamental eigenmode. Also included are results from Table 2.1 for comparison. (Reference $\lambda_0 = 4.82780$.)

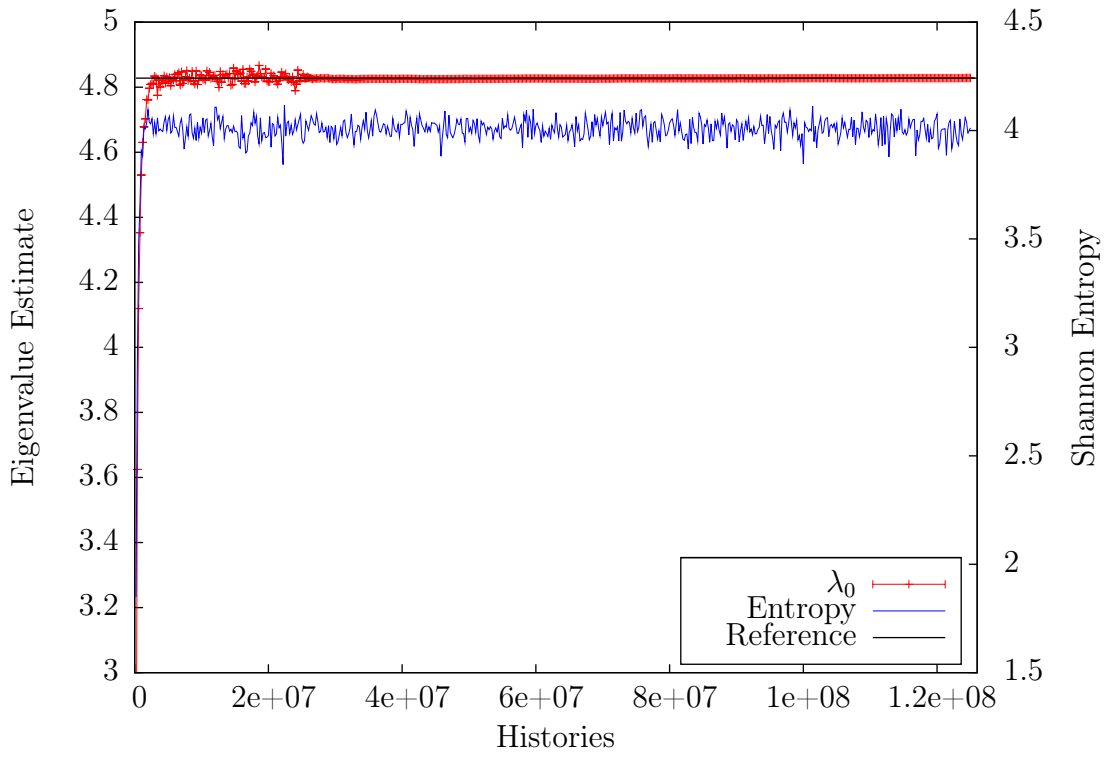
Figure 6.1a shows the eigenvalue estimate convergence as well as the Shannon Entropy. In this particular simulation, Arnoldi's method does not converge the eigenvalue estimate or the Shannon entropy immediately as we have seen previously, but it still only requires a few restarts to converge. Figure 6.1b shows the estimated fundamental eigenvector along with the reference solution. Again we see that Arnoldi's method can accurately estimate the fundamental eigenvector.

These preliminary results show two important things. First, if we are interested in just one eigenmode we should use a smaller Krylov subspace (fewer iterations per restart) and have more active restarts; this reduces the variance and increases the figure of merit. Second, we see that computing an eigenvalue estimate more frequently can reduce the variance for the mean eigenvalue.

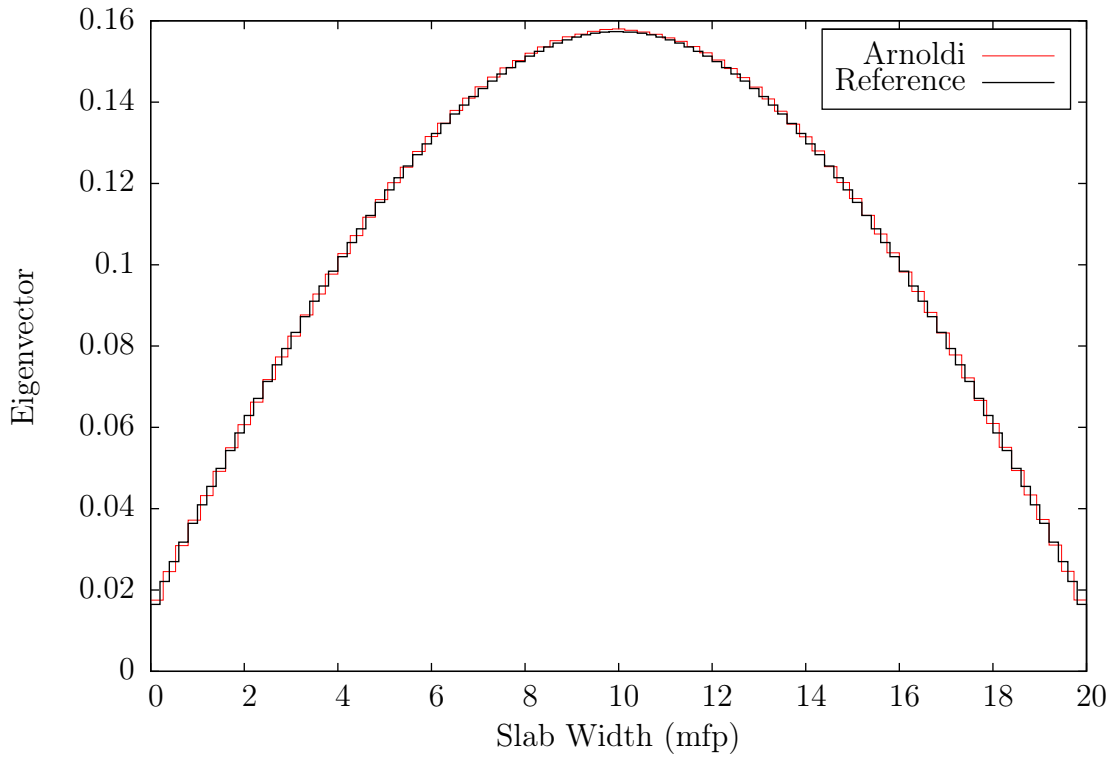
6.1.3 Condensing Arnoldi's Method

One suggestion that has been made[1] for Arnoldi's method is to eliminate restarts and just do a few highly accurate iterations. We have seen that Arnoldi's method does not need many inactive restarts to converge the fission source. The only remaining reason to use many restarts is to obtain an estimation of the statistical uncertainty of the eigenvalue estimate. However we also know that the estimate of the uncertainty is wrong.

[1] very determinedly, in fact



(a) Eigenvalue estimate and Shannon Entropy



(b) Eigenvector

Figure 6.1: Preliminary calculation of eigenvalue estimates, eigenvector, and Shanon entropy from an Arnoldi's method with just 2 iterations per restart and only saving the fundamental eigenmode. The black line is the reference solution.

It has been proposed that rather than performing many Arnoldi restarts, it may be beneficial to track many more particles during just one or two Arnoldi restarts with sufficient Krylov subspace size. The application of $\mathcal{A}$ would be performed much more accurately and the eigenvalue estimates would be much better.

With only one or two Arnoldi restarts the statistical uncertainty could not be calculated as described in this dissertation; the variance of just two estimates isn't helpful. In principle the statistical uncertainty could be calculated by propagating the statistical error through the iterations of Arnoldi's method. This would be a profound change in how statistical uncertainties are calculated in Monte Carlo eigenvalue computations, no one has done this before.

A preliminary simulation has been done using this idea. We return to the 20 mfp slab geometry introduced in Chapter 2. We will use the same total number of particles, but will put all of them into one inactive restart and one active restart; each restart has 10 iterations with 6.25×10^6 particles tracked in each iteration.

The eigenvalues from this simulation are given in Table 6.2 and the estimated eigenvectors in Figure 6.2; the "Condensed" Arnoldi is Arnoldi's method with just two restarts, but many particles tracked in each iteration. We see that the eigenvalue estimates are not exactly the same as the reference solution but that the eigenvector is an excellent estimate as with the regular Arnoldi's method. However we do not have a way yet of estimating the statistical uncertainty of these values so we do not know if the eigenvalue estimates are within statistical uncertainty of the reference solution.

	λ_0	λ_1	λ_2
Condensed Arnoldi	4.8309	4.3824	3.8131
Reference	4.8278	4.3831	3.8174

Table 6.2: Eigenvalue estimates for 20 mfp thick slab geometry from a condensed Arnoldi's method and Reference eigenvalues from [12], and [9].

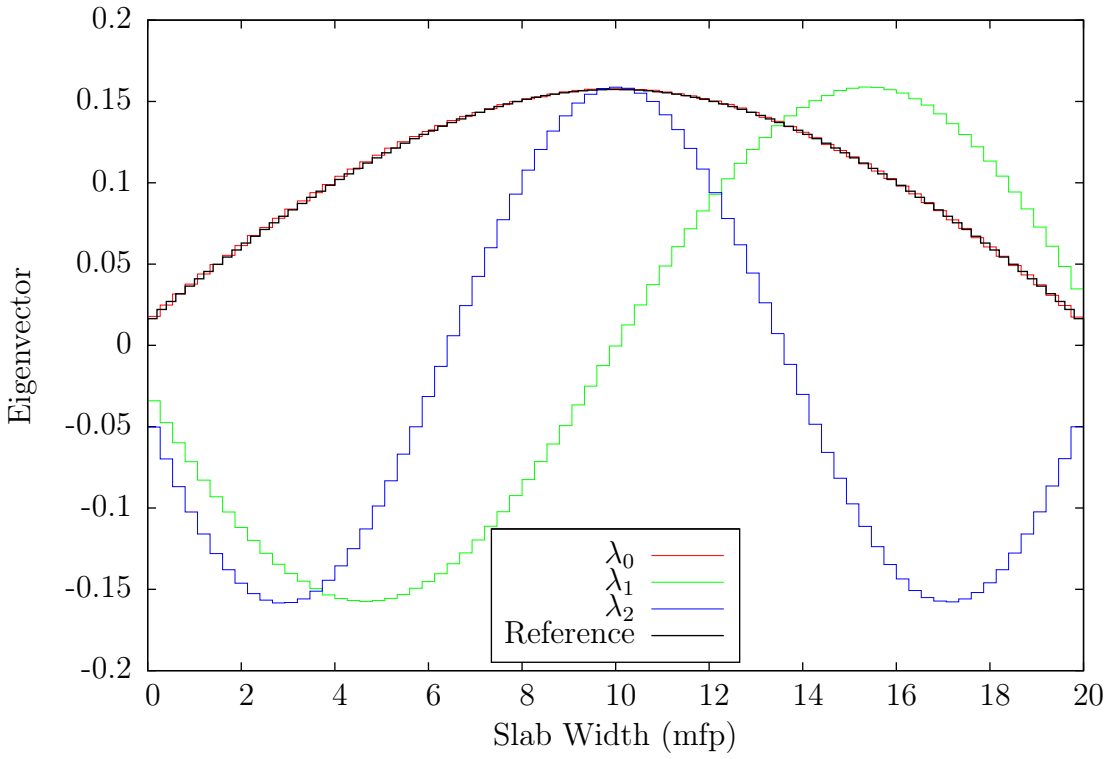


Figure 6.2: Preliminary calculation of eigenvectors from condensed Arnoldi.

6.1.4 Multi-dimensional and Real-world Problems

All of the simulations demonstrated in this dissertation have been one-dimensional. Restricting a prolem to one-dimension is sufficient for a proof-of-concept but it certainly does not represent a real-world problem. As of yet, there have been no attempts at using Arnoldi's method in a three-dimensional, production code such as MCNP; any estimates of how Arnoldi's method may operate in three-dimensions is just speculation.

Most likely the biggest issue when moving to three-dimensions is the length of the Arnoldi vectors. It is expected that the number of discretization bins necessary for three dimensions would be at least N^3 where N is the number of bins for a one-dimensional problem. Increasing the size of the Arnoldi vectors by this magnitude would increase the computational expense of orthogonalizing the Arnoldi vectors as well as the expense of sampling and scoring in and from a three-dimensional source.

The size of the Krylov subspace (number of iterations in a restart) necessary for a three-dimensional problem would seem to be greater than for a one-dimensional problem. In this dissertation, the more difficult problems (i.e. problems with larger dominance ratios) required additional iterations for an accurate eigenvalue estimate. A three-dimensional problem would almost certainly require more iterations than a one-dimensional problem and would therefore require more time.

Arnoldi's method, while having been used extensively in the numerical analysis community has yet to be incorporated in Monte Carlo particle transport algorithms. This dissertation represents the first work in this area. Arnoldi's method has many promising qualities; further investigation will determine the ability of Arnoldi's method to be used in production Monte Carlo codes.

Appendix A

Implicitly Restarted Arnoldi's Method

A good restart vector is one where the undesired region of the spectrum of our linear operator is suppressed while the desired region is enhanced. This is done by zeroing out the Ritz vectors associated with the Ritz values from the undesired region of the spectrum. This is done explicitly in equation (2.14) and repeated here for clarity;

$$\hat{v} = c_1 y_1 + \cdots + c_j y_j + 0\, y_{j+1} + \cdots + 0\, y_n; \tag{A.1}$$

In addition to saving computational expense the improved restarts *implicitly* restart Arnoldi's method with an improved restart vector and is thus called an implicit restart or implicitly restarted Arnoldi's method (IRAM). To see how this is done, a brief discussion of the QR algorithm will first be presented and then IRAM will be shown.

QR Algorithm

A matrix $A \in \mathbb{R}^{n \times n}$can be decomposed into two matrices

$$A = QR \tag{A.2}$$

where $Q, R \in \mathbb{C}^{n\times n}$ with Q unitary (orthonormal columns) and R upper triangular. The QR decomposition (equation (A.2)) is unique [see 31, Chapter 3, pg. 204] for nonsingular matrices and R with positive main diagonal entries. The QR factors can be recombined in reverse order to form a new matrix

$$\hat{A} = RQ. \tag{A.3}$$

The decomposition of a matrix and the recombination of the factors is typically performed iteratively. A simple change of notation makes this illustration clear

$$A_{n-1} = Q_n R_n \tag{A.4a}$$

$$A_n = R_n Q_n. \tag{A.4b}$$

Using this notation $A_0 = A$. The decomposition of A_{n-1} and formation of A_n constitutes one QR iteration.

We can see from equation (A.4a) that $R_n = Q_n^* A_{n-1}$. Substituting this in equation (A.4b) we obtain an alternative form for A_n

$$A_n = Q_n^* A_{n-1} Q_n. \tag{A.5}$$

Shifted QR Iteration

The QR iteration can be shifted by subtracting the identity matrix multiplied by some scalar shift. The factors in the shifted QR iteration can be recombined similarly to the non-shifted counterpart

$$A_{n-1} - \nu_n I = Q_n R_n \tag{A.6a}$$

$$A_n = R_n Q_n + \nu_n I \tag{A.6b}$$

where Q_n and R_n are the same as in the non-shifted QR iteration, I is the identity matrix, and ν_n is the shift being applied. Similarly to the non-shifted QR iteration, we can form A_n alternatively. To see this first note from equation (A.6a) that

$$R_n = Q_n^* A_{n-1} - \nu_n Q_n^*.$$

Substituting this into equation (A.6b) we obtain

$$\begin{aligned} A_n &= \left[Q_n^* A_{n-1} - \nu_n Q_n^*\right] Q_n + \nu_n I \\ &= Q_n^* A_{n-1} Q_n - \nu_n Q_n^* Q_n + \nu_n I \\ &= Q_n^* A_{n-1} Q_n - \nu_n I + \nu_n I \\ A_n &= Q_n^* A_{n-1} Q_n \end{aligned} \tag{A.7}$$

Equation (A.7) generalizes the alternative form of A_m from equation (A.5) to the shifted QR algorithm.

Now that the basic QR algorithm has been given we will proceed to to develop some identities that will be of use to us in analyzing implicit restarts for Arnoldi's method.

Lemma A.0.1. *Let Q_n, A_n be defined by equations* (A.6) *and let*

$$\hat{Q}_n \equiv Q_1 Q_2 \cdots Q_n, \tag{A.8}$$

then

$$A_n = \hat{Q}_n^* A \hat{Q}_n \tag{A.9}$$

$$\hat{Q}_n A_n = A \hat{Q}_n. \tag{A.10}$$

Proof. For $n = 1$, we know from equation (A.6)

$$A_1 = Q_1^* A_0 Q_1 = \hat{Q}_1^* A \hat{Q}_1. \tag{A.11}$$

For $n = 2$, after a second iteration, we have

$$\begin{aligned} A_2 &= Q_2^* A_1 Q_2 \\ &= Q_2^* \left(Q_1^* A Q_1 \right) Q_2 \\ &= \hat{Q}_2^* A \hat{Q}_2 \end{aligned} \tag{A.12}$$

where $\hat{Q}_2 = Q_1 Q_2$. We can prove this in general by induction on n:

$$\begin{aligned} A_n &= Q_n^* A_{n-1} Q_n \\ &= Q_n^* \left(\hat{Q}_{n-1}^* A_{n-2} \hat{Q}_{n-1} \right) Q_n \\ &= Q_n^* Q_{n-1}^* \cdots Q_1 A Q_1 \cdots Q_{n-1} Q_n \\ &= \hat{Q}_n^* A \hat{Q}_n \end{aligned} \tag{A.13}$$

where $\hat{Q}_n = Q_1 Q_2 \cdots Q_n$, as defined in equation (A.8). Equation (A.10) immediately follows from equation (A.9) since $\hat{Q}_n$ is a unitary matrix. □

Lemma A.0.2. *Let Q_n, A_n be defined by equation* (A.6a) *and let $\{\nu_1, \ldots, \nu_m\}$ be the shifts, then*

$$\left(A - \nu_n I \right) \hat{Q}_n = \hat{Q}_n \left(A_n - \nu_n I \right), \tag{A.14}$$

where $\hat{Q}_n \equiv Q_1 Q_2 \cdots Q_n$.

Proof. Using equation (A.10) this becomes trivial:

$$\begin{aligned}\left(A-\nu_n I\right)\hat{Q}_n &= A\hat{Q}_n - \nu_n\hat{Q}_n \\ &= \hat{Q}_n A_n - \nu_n\hat{Q}_n \\ \left(A-\nu_n I\right)\hat{Q}_n &= \hat{Q}_n\left(A_n-\nu_n I\right). \end{aligned} \tag{A.15}$$

□

With these two lemmas we now proceed to the theorem that is important to showing how IRAM restarts with the ideal restart vector.

Theorem A.0.3. *Let $\hat{Q}_n$ be defined as in equation* (A.8) *and*

$$\hat{R}_n \equiv R_n R_{n-1}\cdots R_1 \tag{A.16}$$

and using $\left\{\nu_1,\nu_2,\ldots\nu_n\right\}$ as the shifts of a shifted QR algorithm, then

$$\boldsymbol{p}_j(A) = \hat{Q}_j\hat{R}_j, \tag{A.17}$$

where $\boldsymbol{p}_j$ is a polynomial of degree j with zeros $\nu_1,\ldots,\nu_j$

$$\boldsymbol{p}_j(z) = \left(z-\nu_1 I\right)\left(z-\nu_2 I\right)\cdots\left(z-\nu_j I\right). \tag{A.18}$$

Proof. For $n=1$ equation (A.17) is true by definition of the shifted QR decomposition given in equation (A.6a)

$$\left(A-\nu_1 I\right) = Q_1R_1 = \hat{Q}_1\hat{R}_1. \tag{A.19}$$

In general we can prove this by induction

$$\begin{aligned}
\left(A-\nu_n I\right)\left[\left(A-\nu_{n-1}I\right)\cdots\left(A-\nu_1 I\right)\right] &= \left(A-\nu_n I\right)\left[\hat{Q}_{n-1}\hat{R}_{n-1}\right] \\
&= \left[A\hat{Q}_{n-1}-\nu_n\hat{Q}_{n-1}\right]\hat{R}_{n-1} \\
&= \left[\hat{Q}_{n-1}A_{n-1}-\nu_n\hat{Q}_{n-1}\right]\hat{R}_{n-1} \\
&= \hat{Q}_{n-1}\left(A_{n-1}-\nu_n I\right)\hat{R}_{n-1} \\
&= \hat{Q}_{n-1}\left(Q_n R_n\right)\hat{R}_{n-1} \\
\left(A-\nu_n I\right)\left[\left(A-\nu_{n-1}I\right)\cdots\left(A-\nu_1 I\right)\right] &= \hat{Q}_n\hat{R}_n. \qquad \text{(A.20)}
\end{aligned}$$

□

Equation (A.17) shows that the product of the unitary ($\hat{Q}_n$) and upper triangular ($\hat{R}_n$) matrices from the QR algorithm are equivalent to a polynomial of A of degree n with the shifts as the zeros of the polynomial. This is an important point in IRAM.

Updating Arnoldi Factorization by Shifted QR Iterations

To show how IRAM uses the QR algorithm we begin with the Arnoldi factorization first introduced in equation (2.8)

$$AV_m = V_m H_m + v_{m+1}h_{m+1,m}e_m^T. \qquad \text{(A.21)}$$

Implicitly restarted Arnoldi's method performs shifted QR iterations on H_m using the eigenvalues estimates from the undesired region of the spectrum of A. For Monte Carlo reactor analysis the eigenvalues largest in magnitude are of particular interest however, the following treatment is independent of what region of the spectrum is desired.

We note that the Arnoldi factorization in equation (A.21) is shown after m Arnoldi iterations. For this discussion we assume that k eigenvalues are desired and $m = k+j$ iterations are performed in each restart where $k \sim j$. After m Arnoldi iterations we have m Ritz values—eigenvalue estimates—of A; k of them are the eigenvalues of interest and the other j values are used as shifts for the shifted QR algorithm.

IRAM performs j iterations of the shifted QR algorithm on H_m using the undesired Ritz values as described previously. After j iterations we obtain

$$\hat{H}_j = \hat{Q}_j^* H_m \hat{Q}_j \tag{A.22}$$

where $\hat{Q}_j$ is defined in equation (A.8). Because H_m is upper Hessenberg, we can show that Q_i is upper Hessenberg and $\hat{Q}_j$ is properly j-Hessenberg.

Theorem A.0.4. *Let j be a non-negative integer. A matrix H is called j-Hessenberg if $h_{rc} = 0$ whenever $(r-c) > j$. An Hessenberg matrix is said to be properly j-Hessenberg if $h_{rc} \neq 0$ whenever $(r-c) = j$,*

$$h_{rc} = \begin{cases} 0 & r - c > j \\ x & \textit{otherwise.} \end{cases} \tag{A.23}$$

Then the product of a properly j-Hessenberg matrix (H_j) and a properly k-Hessenberg matrix H_k is properly $(j+k)$-Hessenberg (H_{j+k}).

Proof. The rth row of a properly j-Hessenberg matrix has $\max\left[r-j-1, 0\right]$ leading zeros. The cth column of a properly k-Hessenberg matrix has $\max\left[m-\left(c+k\right), 0\right]$ trailing zeros. The elements of H_{j+k}, h_{rc}, are zero if the sum of the leading zeros of the rth row of H_j is greater than or equal to the difference of the size of the matrix m and the trailing zeros of the cth column of H_k; $h_{rc} = 0$ if $\left(r-j-1\right) \geq m - \left[m-\left(c+k\right)\right]$.

Simplifying we obtain $\big(r-j-1\big) \geq \big(c+k\big)$ or equivalently

$$h_{rc} = \begin{cases} 0 & r-c > \big(j+k\big) \\ x & \text{otherwise.} \end{cases} \tag{A.24}$$

We see that equation (A.24) is equivalent to equation (A.23) for a $\big(j+k\big)$-Hessenberg matrix. □

When performing the QR decomposition on an upper Hessenberg matrix we obtain

$$H_m = Q_1 R_1 \tag{A.25}$$

with Q_1 and R_1 defined in equation (A.2). This can be re-written as

$$Q_1 = H_m R_1^{-1} \tag{A.26}$$

where we note that the inverse of an upper triangular matrix is an upper triangular matrix. An upper triangular matrix is properly 0-Hessenberg and H_m is 1-Hessenberg so we can apply Theorem A.0.4 on the preceding page to show that Q_1 is properly 1-Hessenberg.

Corollary A.0.5. *Let $\hat{Q}_n \in \mathbb{C}^{m\times m}$ be the combined unitary matrix resulting from n iterations of the QR iteration on an upper Hessenberg matrix, H. Then $\hat{Q}_n$ is a properly n-Hessenberg matrix and that the row vector $e_m^T \hat{Q}_n$ has $m-n-1$ leading zeros.*

Proof. We know that the unitary matrix from the QR decomposition of an upper Hessenberg matrix is also upper Hessenberg. $\hat{Q}_n$ is the product of n upper Hessenberg matrices, therefore $\hat{Q}_n$ is properly n-Hessenberg.

The elements of product $e_m^T \hat{Q}_n$ are non-zero when the element of the mth or last row of $\hat{Q}_n$ is also non-zero. The last row of $\hat{Q}_n$ has $m - n - 1$ leading zeros as given in Theorem A.0.4 on page 103. □

Let's return to the Arnoldi method. Solving equation (A.22) for H_m and substituting into equation (A.21) we obtain

$$AV_m = V_m \left(\hat{Q}_j \hat{H}_j \hat{Q}_j^* \right) + v_{m+1} h_{m+1,m} e_m^T. \tag{A.27}$$

Now operate on the right by $\hat{Q}_j$,

$$\begin{aligned} AV_m \hat{Q}_j &= V_m \hat{Q}_j \hat{H}_j \hat{Q}_j^* \hat{Q}_j + v_{m+1} h_{m+1,m} e_m^T \hat{Q}_j \\ A\hat{V}_m &= \hat{V}_m \hat{H}_m + v_{m+1} h_{m+1,m} e_m^T \hat{Q}_j, \end{aligned} \tag{A.28}$$

where

$$\hat{V}_m = V_m \hat{Q}_j. \tag{A.29}$$

The row vector $e_m^T \hat{Q}_j$ in equation (A.28) has $\left(m - j - 1\right)$ leading zeros according to Theorem A.0.5 on the preceding page. If we drop the first j columns of equation (A.28) we obtain

$$\begin{aligned} A\hat{V}_k &= \hat{V}_{k+1} \hat{H}_{k+1,k} + v_{m+1} h_{m+1,m} \beta e_k^T \\ &= \hat{V}_k \hat{H}_k + \left(\check{v}_{k+1} \check{h}_{k+1,k} + v_{m+1} h_{m+1,m} \beta \right) e_k^T \end{aligned} \tag{A.30}$$

where βe_k^T is the first $k+1$ columns of $e_m^T \hat{Q}_j$ and we have defined

$$\hat{v}_{k+1} = \gamma \left(\check{v}_{m+1} \check{h}_{m+1,m} + v_{m+1} h_{m+1,m} \beta \right) \tag{A.31}$$

where γ is chosen to normalize $\hat{v}_{k+1}$, $\|\hat{v}_{k+1}\|_2 = 1$. If we let $\hat{h}_{k+1,k} = 1/\gamma$ then equation (A.30) becomes

$$A\hat{V}_k = \hat{V}_k\hat{H}_k + \hat{v}_{k+1}\hat{h}_{k+1,k}e_k^T \tag{A.32}$$

which is exactly like equation (A.21) except we have added some hats on some symbols and replaced m with k. It can be shown [see 31] that the Arnoldi vectors contained as the columns of $\hat{V}_k$ are exactly those that would have been generated by explicitly starting with $\hat{v}_1$. Thus with IRAM we don't have have to start at the beginning of a restart, therefore we can jump right in at the kth iteration of the restart. After j additional steps we will have a Krylov subspace of size m, exactly as we would have had after m iterations in explicit Arnoldi.

Being able to jump into the middle of an Arnoldi restart can save considerable computational expense by reducing the number applications of the linear operator A required. (The application of the linear operator in Monte Carlo particle transport is responsible for $> 80\%$ of the computational runtime.) In short, IRAM exchanges Arnoldi iterations for shifted QR iterations. Shifted QR iterations are faster than Arnoldi iterations because H_m is small.

How Implicit Restarts Suppresses Unwanted Eigenvalue Information

We have just shown that implicit restarts can be computationally more efficient. Here we show that implicit restarts are mathematically equivalent to restarting with a linear combination of the Ritz vectors from the desired region of the spectrum of A [see 31, Chapter 5, pg. 456].

Theorem A.0.6. *Suppose we have the Arnoldi factorization*

$$AV_m = V_m H_m + v_{m+1} h_{m+1,m} e_m^T \tag{A.33}$$

and let $\boldsymbol{p}_j$ *be a polynomial of degree* $j < m$ *as shown in equation* (A.18)*. Then*

$$\boldsymbol{p}_j(A) V_m = V_m \boldsymbol{p}_j(H_m) + E_j, \tag{A.34}$$

where $E_j \in \mathbb{C}^{n \times m}$ *is identically zero, except in the last* j *columns.*

Proof. For $m = 1$ we can just apply a shift ν_1 to the Arnoldi factorization, equation (A.21),

$$\left(A - \nu_1 I\right) V_m = V_m \left(H_m - \nu_1 I\right) + E_1 \tag{A.35}$$

where $E_1 = h_{m+1,m} v_{m+1} e_m^T$. We can see that E_1 is zero except for the last column by the product $v_{m+1} e_m^T$. $\left(A - \nu_1 I\right)$ and $\left(H_m - \nu_1 I\right)$ are polynomials of A and H_m respectively with degree $j = 1$, both with root ν_1.

Assuming this theorem holds for polynomials of degree $j - 1$ (we have just shown this to be true for $j - 1 = 1$) we can now show it is valid for polynomials of degree j. We know:

$$\left(A - \nu_{j-1} I\right) \cdots \left(A - \nu_1 I\right) V_m = V_m \left(H_m - \nu_{j-1} I\right) \cdots \left(H_m - \nu_1 I\right) + E_{j-1} \tag{A.36}$$

operate on the left by $\left(A - \nu_j I\right)$

$$\begin{aligned} \left(A - \nu_j I\right) \left(A - \nu_{j-1} I\right) \cdots \left(A - \nu_1 I\right) V_m = \\ \left(A - \nu_j I\right) V_m \left(H_m - \nu_{j-1} I\right) \cdots \left(H_m - \nu_1 I\right) + \left(A - \nu_j I\right) E_{j-1}. \end{aligned} \tag{A.37}$$

Substituting ν_j for ν_1 in equation (A.35) we see that

$$\Big(A - \nu_j I\Big) V_m = V_m \Big(H_m - \nu_j I\Big) E_1 \tag{A.38}$$

which can be inserted into the previous equation

$$\begin{aligned}
&\Big(A - \nu_j I\Big) \cdots \big(A - \nu_1 I\big) V_m \\
&\quad = \left[\Big(A - \nu_j I\Big) V_m\right] \Big(H_m - \nu_{j-1} I\Big) \cdots \big(H_m - \nu_1 I\big) + \Big(A - \nu_j I\Big) E_{j-1} \\
&\quad = \left[V_m \Big(H_m - \nu_j I\Big) + E_1\right] \Big(H_m - \nu_{j-1} I\Big) \cdots \big(H_m - \nu_1 I\big) + \Big(A - \nu_j I\Big) E_{j-1} \\
&\quad = V_m \Big(H_m - \nu_j I\Big) \cdots \big(H_m - \nu_1 I\big) + E_1 \Big(H_m - \nu_{j-1} I\Big) \cdots \big(H_m - \nu_1 I\big) \\
&\qquad + \Big(A - \nu_j I\Big) E_{j-1} \\
&\quad = V_m \Big(H_m - \nu_j I\Big) \cdots \big(H_m - \nu_1 I\big) + E_j
\end{aligned} \tag{A.39}$$

where

$$E_j = E_1 \Big(H_m - \nu_{j-1} I\Big) \cdots \big(H_m - \nu_1 I\big) + \Big(A - \nu_j I\Big) E_{j-1}. \tag{A.40}$$

We can simplify equation (A.39) further and write

$$\boldsymbol{p}_j(A) V_m = V_m \boldsymbol{p}_j(H_m) + E_j, \tag{A.41}$$

where

$$\boldsymbol{p}_j(A) = \Big(A - \nu_j I\Big) \cdots \big(A - \nu_1 I\big)$$

is a polynomial of degree j on A and

$$\boldsymbol{p}_j(H_m) = \Big(H_m - \nu_j I\Big) \cdots \big(H_m - \nu_1 I\big)$$

is a polynomial of degree j on H_m. Equation (A.41) is exactly what we want to prove equation (A.34).

We know that the columns of E_{j-1} are zero except the last $j-1$ columns. We can see by inspection that the second term on the right hand side of equation (A.40) has the same structure. From Theorem A.0.4 on page 103 we know that the product $\Big(H_m - \nu_{j-1}I\Big)\cdots\big(H_m - \nu_1 I\big)$ is properly j-Hessenberg. Multiply this by E_1 on the left and we obtain a zero matrix except in the last j columns. The right hand side of equation (A.40) is therefore as we expected. □

Now that we have the necessary mathematical basis to understand what happens with an implicit Arnoldi restart we can investigate how IRAM generates it's starting vector. In IRAM the shifts $\nu_1, \nu_2, \ldots, \nu_j$ are chosen from the region of the eigenvalue spectrum that is to be suppressed. These shifts are then used in j iterations of the shifted QR algorithm on H_m resulting in $\hat{H}_j = \hat{Q}_j^* H_m \hat{Q}_j$ where $\hat{Q}_j$ is the combined unitary factor in the QR factorization

$$\boldsymbol{p}_j(H_m) = \hat{Q}_j \hat{R}_j \tag{A.42}$$

where j indicates the number of shifted QR iterations performed and $\boldsymbol{p}_j(z)$ is defined in equation (A.18). When we apply Theorem A.0.6 on page 106 and substitute equation (A.42) we obtain

$$\boldsymbol{p}_j(A)V_m = V_m \hat{Q}_j \hat{R}_m + E_j. \tag{A.43}$$

The Arnoldi factorization is uniquely defined by the starting vector and the linear operator. If we multiply equation (A.43) by e_1 we get

$$\boldsymbol{p}_j(A)V_m e_1 = V_m \hat{Q}_j \hat{R}_j e_1 + E_j e_1. \tag{A.44}$$

$E_j e_1 = 0$ because the first column of E_j is zero. Since $\hat{R}_j$ is upper triangular $\hat{R}_j e_1 = r_{11} e_1 = \alpha e_1$ and the first term on the right hand side is $\alpha V_m \hat{Q}_m = \alpha \hat{v}_1$. The left hand side of equation (A.44) is just a polynomial of degree j of A multiplied by v_1 the starting vector for this Arnoldi process. From equation (A.44) we can deduce

$$\hat{v}_1 = \frac{1}{\alpha} \boldsymbol{p}_j(A) v_1. \tag{A.45}$$

Let's return to the starting vector for an Arnoldi procedure. We can write the vector as a linear combination of a set of basis vectors

$$v_1 = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n, \tag{A.46}$$

where the x_i's are the basis vectors and the c_i's are some expansion coefficients. If we choose the eigenvectors of our linear operator A as our basis vectors then things become interesting with respect to IRAM. To see this, multiply v_1 by a polynomial of degree 1 of A

$$\begin{aligned} \boldsymbol{p}_1(A) v_1 &= \left(A - \nu_1 I\right) \left(c_1 x_1 + c_2 x_2 + \cdots + c_n x_n\right) \\ &= c_1 \left(A x_1 - \nu_1 x_1\right) + \cdots c_n \left(A x_n - \nu_1 x_n\right) \end{aligned} \tag{A.47}$$

where we remember the x_i's are the eigenvectors of A and that ν_1 is the first undesired eigenvalue, λ_{k+1}. Equation (A.47) becomes

$$\begin{aligned} \boldsymbol{p}_1(A) v_1 = c_1 \left(\lambda_1 x_1 - \nu_1 x_1\right) + \cdots \\ + c_{k+1} \underbrace{\left(\lambda_{k+1} x_{k+1} - \nu_1 x_{k+1}\right)}_{=0} + \\ \cdots + c_n \left(\lambda_n x_n - \nu_1 x_n\right) \end{aligned} \tag{A.48}$$

where the vector x_{k+1} has been completely eliminated.

If we include all the terms of the polynomial $\boldsymbol{p}_j(A)$ then all j eigenvectors from the undesired region of the spectrum are removed from our restart vector

$$\boldsymbol{p}_j(A)v_1 = c_1\left(\lambda_1 x_1 - \nu_1 x_1\right)\left(\lambda_1 x_1 - \nu_2 x_1\right)\cdots\Big(\lambda_1 x_1 - \nu_j x_1\Big) + \cdots \\ + c_k\left(\lambda_k x_k - \nu_k x_k\right)\cdots\left(\lambda_k x_k - \nu_k x_k\right) + 0. \quad \text{(A.49)}$$

We can plug this result into equation (A.45) and get our optimal restart vector as described in equation (A.1).

Thus we see that implicitly restarted Arnoldi's method is mathematically equivalent to explicitly restarted Arnoldi's method but can skip into the restart after k iterations. IRAM can save computational expense by trading expensive applications of the linear operator for cheap shifted QR iterations.

Bibliography

[1] Ziya Akcasu. personal communication.

[2] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9(1):17–29, 1951. ISSN 0033-569X.

[3] T. E. Booth. Computing the higher k-eigenfunctions by monte carlo power iteration: A conjecture. *NUCLEAR SCIENCE AND ENGINEERING*, 143(3): 291–300, Mar 2003. ISSN 0029-5639.

[4] A. Bouras and V. Frayssé. A relaxation strategy for the arnoldi method in eigenproblems. Technical report, CERFACS, 2000.

[5] A. Bouras and V. Frayssé. Inexact matrix-vector products in krylov methods for solving linear systems: A relaxation strategy. *SIAM Journal on Matrix Analysis and Applications*, 26(3):660–678, 2005. ISSN 0895-4798.

[6] Forrest B. Brown. A review of monte carlo criticality calculations–convergence, bias, statistics. In *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*, Saratoga Springs, NY, May 3-7 2009. American Nuclear Society.

[7] Jeremy L. Conlin and James P. Holloway. Arnoldi's method of minimized iterations for monte carlo criticality calculations. In *International Conference on*

Reactor Physics, Nuclear Power: A Sustainable Resource, Interlaken, Switzerland, September 14-19 2008.

[8] Jeremy Lloyd Conlin and James Paul Holloway. Relaxation scheme for monte carlo explicitly restarted arnoldi's method for criticality calculations. In *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*, Saratoga Springs, NY, May 3-7 2009. American Nuclear Society.

[9] E. B. Dahl and N. G. Sjostrand. Eigenvalue spectrum of multiplying slabs and spheres for monoenergetic neutrons with anisotropic scattering. *Nuclear Science and Engineering*, 69(1):114–125, Jan 1979.

[10] James J. Duderstadt and Louis J. Hamilton. *Nuclear Reactor Analysis*. John Wiley & Sons, 1976.

[11] Melina Annerose Freitag. *Inner-outer Iterative Methods for Eigenvalue Problems—Convergence and Preconditioning*. PhD thesis, University of Bath, September 2007.

[12] N. S. Garis. One-speed neutron-transport eigenvalues for reflected slabs and spheres. *Nuclear Science and Engineering*, 107(4):343–358, Jan 1991.

[13] W. Goad and R. Johnston. A monte carlo method for criticality problems. *Nuclear Science and Engineering*, 5:371–375, 1959.

[14] David Patrick Griesheimer. *Functional Expansion Tallies for Monte Carlo Simulations*. PhD thesis, University of Michigan, 2005.

[15] E. L. Kaplan. Monte carlo methods for equilibrium solutions in neutron multiplication. Technical Report UCRL-5275-T, Livermore National Laboratory, 1958.

[16] Brian C. Kiedrowski and Forrest B. Brown. An information theory based analysis of fission source correlation in monte carlo k-eigenvalue calculations. In *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*, Saratoga Springs, NY, May 3–7 2009. American Nuclear Society.

[17] D. E. Kornreich and D. K. Parsons. Semi-analytic results and insights into difficult source convergence problems. In *Trans. Am. Nucl. Soc.*, volume 87, pages 152–154, 2002.

[18] J. Lieberoth. A monte carlo technique to solve the static eigenvalue problem of the boltzmann transport equation. *Nukleonik*, 11:213, 1968.

[19] M. R. Mendelson. Monte carlo criticality calculations for thermal reactors. *Nuclear Science and Engineering*, 32:319–331, 1968.

[20] R. S. Modak, D. C. Sahny, and S. D. Paranjape. Evaluation of higher k-eigenvalues of the neutron-transport equation by the sn method. *Annals of Nuclear Energy*, 22(6):359–366, Jan 1995.

[21] Y. Saad. Variations on arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear Algebra and its Applications*, 34(DEC):269–295, 1980. ISSN 0024-3795.

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

[23] V. Simoncini. Variable accuracy of matrix-vector products in projection methods for eigencomputation. *SIAM Journal on Numerical Analysis*, 43(3):1155–1174, 2005. ISSN 0036-1429.

[24] V. Simoncini and D. B. Szyld. Theory of inexact krylov subspace methods and applications to scientific computing. *SIAM Journal on Scientific Computing*, 25 (2):454–477, 2003. ISSN 1064-8275.

[25] G. L. G. Sleijpen, J. van den Eshof, and M. B. van Gijzen. Restarted gmres with inexact matrix-vector products. *Numerical Analysis and its Applications*, 3401: 494–502, 2005. ISSN 0302-9743.

[26] Danny C Sorensen. Implicit application of polynomial filters in a k-step arnoldi method. *SIAM Journal on Matrix Analysis and Applications*, 13:357–385, Jan 1992.

[27] Taro Ueki and Forrest B. Brown. Informatics approach to stationarity diagnostics of the monte carlo fission source distribution. In *American Nuclear Society 2003 Winter Meeting*. American Nuclear Society, American Nuclear Society, November 16-20 2003.

[28] Taro Ueki and Forrest B. Brown. Stationarity modeling and informatics-based diagnostics in monte carlo criticality calculation. *Nuclear Science and Engineering*, 149(1):38–50, January 2005.

[29] J. van den Eshof and G. L. G. Sleijpen. Inexact krylov subspace methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):125–153, 2004. ISSN 0895-4798.

[30] Daryoosh Vashaee, Lobat Tayebi, and John Luxat. Reference model parameter identification of space-time dependent reactivity in a candu-phwr. *Annals of Nuclear Energy*, 35(2):228–237, 2 2008. URL `http://www.sciencedirect.com/science/article/B6V1R-4PG2M10-1/2/de9bc85ba9b250b`

[31] David S. Watkins. *Fundamentals of Matrix Computations*. Wiley-Interscience, New York, second edition, 2002.