ABSTRACT

Perturbed Arnoldi Method for Computing Multiple Eigenvalues

Thomas H. Gibson

Director: Ronald B. Morgan, Ph.D.

There are several known methods for computing eigenvalues of a large sparse nonsymmetric matrix. One of the most efficient methods is known as the Arnoldi method. The Arnoldi method is a Krylov subspace method that computes the eigenvalues of the projection of a matrix onto the Krylov subspace. In our investigation, we present both non-restarted and restarted Arnoldi methods and examine how round-off error helps find multiple eigenvalues. We introduce a new method that uses a diagonal matrix perturbation that separates multiple eigenvalues and improves performance. Our approach presents an alternative that avoids the need for a block method, or for relying on round-off error to introduce multiple copies of eigenvalues.

APPROVED BY DIRECTOR OF HONORS THESIS:

_____

Dr. Ronald B. Morgan, Department of Mathematics

APPROVED BY THE HONORS PROGRAM:

_____

Dr. Andrew Wisely, Director

DATE: _____

PERTURBED ARNOLDI METHOD FOR COMPUTING MULTIPLE EIGENVALUES

A Thesis Submitted to the Faculty of

Baylor University

In Partial Fulfillment of the Requirements for the

Honors Program

By

Thomas H. Gibson

Waco, Texas

May 2014

TABLE OF CONTENTS

ACKNOWLEDGMENTS

First and foremost, I wish to thank my research adviser, Dr. Ronald Morgan. He has been extremely supportive and influential since I first began working under him. He did an excellent job of introducing me to the field of numerical linear algebra and has been invaluable in fostering my interest in mathematical research. My work has constantly improved because of his demand for quality work. Without his guidance, I would not have been successful in my research endeavors. It has been an honor to learn and study under him.

I would like to extend my thanks to my thesis readers. Thank you to Dr. Robert Kirby. His support and guidance for graduate schools has been an incredible help to me. And thank you to Dr. Frank Mathis, with whom I have had the immense pleasure of learning real analysis. As an aspiring numerical analyst, it is a tremendous honor to have studied under both of them.

I am incredibly grateful to the entire mathematics department at Baylor for giving me a wonderful education. I would like to personally thank Dr. Lance Littlejohn. I credit my growing enthusiasm for mathematics to him, as he introduced me to the mathematics program here at Baylor University.

Beyond my studies, I would like honor my family for always being incredibly patient, understanding, and supportive of my aspirations. Thank you to T. Michael, and Melody Gibson. I also wish to thank my sister, Janel Gibson, who has put up with my obsession with mathematics for the past three years. Without their love and support, this work would not have been completed. Words alone cannot express how much I love and appreciate them. I hope this work makes all of you proud.

CHAPTER ONE

Introduction

The Importance of Eigenvalues

In physics and engineering, mechanical resonance is the tendency for a physical body, when subjected to forces or displacements, to respond at a greater amplitude when the frequency of its oscillations match the mechanical system's natural frequency. This frequency is often referred to as the resonant frequency, or natural frequency. Consequences of mechanical resonance may result in violent swaying and catastrophic failure of bridges, buildings, vehicles, and airplanes. On July 1, 1940, the Tacoma Narrows Bridge opened in the state of Washington. Four months after its opening, in a startling display, the bridge collapsed in a high wind. The dramatic, rhythmic twisting and distortion that resulted in the collapse can be explained through the analysis of the eigenvalues of the mechanical system [1, 2].

Structures exposed to aerodynamic forces are designed carefully within known parameters to avoid a phenomenon called aeroelastic flutter. Flutter deals with a mechanical systems dynamic response, which is usually a vibrational response. If a body is constructed without regard for aeroelastic flutter, then there may reach a point where the structure undergoes simple harmonic motion, and self-oscillation that leads to eventual failure. Mathematically, the eigenvalue problem naturally occurs in the vibration analysis of these mechanical systems. The eigenvalues are used to determine the natural frequencies of vibration, and the corresponding eigenvectors determine the mode of vibration. The Basse-Chane Bridge, more commonly known as Angers Bridge, was a suspension bridge in Angers, France, that suffered a similar fate to the Tacoma Bridge. The bridge collapsed on April 16, 1850, while a battalion of French soldiers were marching across is, killing more than 200 of the

soldiers. Many speculate that the soldiers marched in lock step too close to the natural frequency of the bridge, which resulted in the collapse [2]. While it is difficult to say with absolute certainty, theoretically under certain conditions, it is plausible for this to be the reason for the Angers Bridge collapse.

Eigenvalues and eigenvectors are crucial for understanding problems in quantum mechanics. Dynamical variables of a system such as particle position, momenta, and energy states are all represented by operators. A quantum system is described by a vector in an infinite dimensional function space. Each dynamical variable is associated with a Hermitian operator that acts on the state of the system whose eigenvalues correspond to the possible value of the variable. Physicists are extremely interested in effective computation of eigenvalues for reasons like this.

Principal component analysis (PCA) is a way of identifying patterns in large data, such as those encountered in data mining, econometrics, and image processing. It's a statistical procedure the uses an orthogonal transformation, such as an eigendecomposition of a symmetric matrix, to convert a set of observations of correlated variables into a set of linearly uncorrelated variables known as principal components. PCA is typically performed on a matrix known as the covariance matrix (correlation matrix) in which the eigenvectors correspond to the principal components. The eigenvalues are essential to represent the variance represented by the principal components [3, 4]. Its applications are numerous and extremely important for statisticians, computer scientists, and applied mathematicians.

In summary, eigenvalues and eigenvectors have many applications in both pure and applied mathematics. As such, mathematicians, scientists, and engineers are interested in effective computation of eigenvalues for their particular set of problems and research areas. However, very large matrices often arise from real-world problems. For example, problems that arise from discretizing partial differential equations often result in large matrices. Fortunately, these matrices are typically sparse,

but few computationally effective methods have been presented for finding the corresponding eigenvectors of eigenvalues of higher multiplicity. In this thesis, we will outline and introduce the reader to the Arnoldi method, as well as its variations. We also explore how numerical round-off error brings about multiple eigenvalues in various implementations of the Arnoldi method. We then present a perturbed approach to compare with the round-off error approach.

CHAPTER TWO

Preliminaries

We begin first by reviewing some basic definitions and theorems from linear algebra that are relevant to the topics discussed in this thesis. For all definitions and theorems, we follow and use terminology similar to that in most linear algebra texts such as in [3, 4, 5].

2.1 An Overview of Eigenvalues and Eigenvectors

Definition 2.1.1. *Let $A \in \mathbb{C}^{n\times n}$. Then a non-zero vector $z \in \mathbb{C}^n$ is called an eigenvector of A with a corresponding eigenvalue $\lambda \in \mathbb{C}$ if*

$$Az = \lambda z. \tag{2.1}$$

We call the mathematical pair of an eigenvector and its associated eigenvalue an eigenpair.

It is important to note that the eigenvectors of a matrix are precisely the vectors in $\mathbb{R}^n$ or $\mathbb{C}^n$ whose direction is preserved upon being multiplied by the matrix A. Although in the definition we recognize that, in general, eigenvalues and eigenvectors can be complex-valued, we will primarily be examining real-valued eigenpairs. For generality in our definitions, we will use the algebraic field of the complex numbers, $\mathbb{C}$.

It is often necessary to compute the eigenvalues of a given matrix. To do this, the most immediate method involves finding the roots of a polynomial.

Definition 2.1.2. *The characteristic polynomial of a matrix $A \in \mathbb{C}^{n\times n}$, denoted as $P_A(x)$ for some scalar $x \in \mathbb{C}$, is the polynomial of degree n defined by*

$$P_A(x) = det(A - xI), \tag{2.2}$$

where I is the $n \times n$ identity matrix.

If there is a scalar λ such that $P_A(\lambda) = 0$, then we also have that the matrix $A - \lambda I$ is singular, i.e. not invertible. Thus the homogenous system must have a non-trivial solution, z, such that

$$(A - \lambda I)z = 0,$$

or equivalently,

$$Az = \lambda z.$$

Therefore it is straightforward to see that any root to the characteristic polynomial of a square matrix A is an eigenvalue of A with an eigenvector z, since (2.1) is satisfied.

Naively, we can reduce the problem of finding the eigenvalues of a matrix to computing the roots of a polynomial equation. However there is a catastrophic problem that comes with this approach. The Abel-Ruffini theorem states that there is no general algebraic solution to polynomial equations of degree five or higher[1] . Therefore root-finding is an area of research in its own right and requires extensive knowledge of various iterative algorithms for finding roots.

In this thesis, we strictly examine eigenvalue computations via iteration rather than finding roots to a high-degree polynomial for precisely the reason mentioned above. We now quickly present the concept of matrix transformations, as this will be an essential topic for discussion in the Arnoldi method, and give definitions of special matrices.

Definition 2.1.3. *A matrix $A \in \mathbb{C}^{n \times n}$ is called normal if $A^H A = A A^H$, where A^H is the Hermitian transpose of A.*

[1] In the field of abstract algebra, this is also known as Abel's impossibility theorem. It is important to note that this does not assert that higher-degree polynomial equations have no solutions. In fact, the fundamental theorem of algebra guarantees that there is at least one.

Definition 2.1.4. *A matrix $A \in \mathbb{C}^{n\times n}$ is said to be unitary if $A^H A = I$, where I is the $n \times n$ identity matrix and A^H is the Hermitian transpose of A. In the case that A is a real matrix, we call A orthogonal.*

Definition 2.1.5. *Two matrices A and B in $\mathbb{C}^{n\times n}$ are said to be similar matrices if there exists a nonsingular matrix $P \in \mathbb{C}^{n\times n}$ such that*

$$A = PBP^{-1}. \tag{2.3}$$

The linear transformation $B \to A$ is called a similarity transformation. In the case that the matrix P is unitary, we say the two matrices A and B are unitarily similar.

Having two matrices that are similar can be a potentially helpful asset, as they share some special properties. Namely, similarity transformations preserve the eigenvalues of matrices. This is straightforward to show.

Proof. Assume that A and B are both $n \times n$ similar matrices. Now suppose (λ, z) is an eigenpair of A. Then since A and B are similar,

$$Az = \lambda z \implies PBP^{-1}z = \lambda z$$

Since P is a nonsingular matrix, apply P^{-1} to both sides and simplify to get,

$$P^{-1}PBP^{-1} = P^{-1}\lambda z$$
$$BP^{-1}z = \lambda P^{-1}z$$

Thus if we define $y = P^{-1}z$, we have an analogous form of (2.1). Thus λ is an eigenvalue for both A and B. Furthermore, since z is an eigenvector with respect to A, it is non-zero, and thus $y = P^{-1}z$ is non-zero. Therefore by definition, y is an eigenvector corresponding to λ with respect to the matrix B. □

This fact will become essential later on in this thesis, since we now have the tools to relate a potentially complicated matrix A, to a matrix B that could have a more

simple structure. We will now introduce some important terminology concerning the eigenvalues themselves. Since eigenvalues are the roots to a polynomial equations from (2.2), it is natural to discuss the multiplicity of eigenvalues since roots themselves have multiplicity.

Definition 2.1.6. *An eigenvalue λ of A has algebraic multiplicity p, if it is a root of multiplicity p of the characteristic polynomial of A. If an eigenvalue has algebraic multiplicity one, it is said to be a simple eigenvalue. An eigenvalue that is not simple is multiple.*

The geometric multiplicity μ of an eigenvalue λ of A is the maximum number of linearly independent eigenvectors associate with it. That is, the geometric multiplicity μ is the dimension of the eigenspace associated with λ, $Null(A - \lambda I)$.

Definition 2.1.7. *If an $n \times n$ matrix A has less than n linearly independent eigenvectors as a consequence of having eigenvalues of higher algebraic multiplicity than their geometric multiplicity, then the matrix is said to be deficient and the eigenvalue(s) responsible are degenerate eigenvalues.*

To conclude our overview of fundamental definitions and matrix properties, we end with an important theorem for matrices.

Theorem 2.1.8. *QR Factorization: Given an arbitrary matrix A, its QR factorization is a matrix decomposition of the form*

$$A = QR$$

where R is an upper triangular matrix and Q is an orthogonal matrix.

The QR factorization is a powerful tool used for both solving a $n \times n$ system of linear equations, as well as least squares problems. In this thesis, we will refer to it in a later chapter.

To summarize why computational methods for eigenvalues are essential, the characteristic polynomial is a powerful theoretical tool, but not a practical computational approach for modern day problems. Reasons for this include computing coefficients via matrix determinants, which is highly unstable in practice. Even if the characteristic polynomial were to be obtained by this approach, computing roots of a polynomial of degree larger than 4 cannot always be done directly; root finding requires an iterative solution process. In addition, finding roots of a polynomial can be an ill-conditioned problem.

In industrial sized matrix problems, techniques for finding eigenvalues and eigenvectors work by repeatedly refining a sequence of approximations to the desired eigenvector or eigenvalue. This is done through an iterative process and is terminated when the approximations reach a suitable accuracy, depending on the application. While there are many different types of iterative methods for computing eigenvalues, in this thesis, we restrict our attention to Krylov subspace methods. More specifically, we will study the Arnoldi method and its performance in computing eigenvalues of higher algebraic multiplicity.

CHAPTER THREE

Krylov Subpace Methods

In this chapter we will discuss iterative techniques for computing eigenvalues and eigenvectors. As mentioned previously, we can find eigenvalues by solving for roots to the characteristic equation of a given matrix. However, the Abel-Ruffini theorem suggests that this is not always possible algebraically for polynomials of degree five or higher. This suggests that we need an iterative process for finding roots of higher degree polynomials. In most practical applications, matrices are exceedingly large and finding eigenvalues by the characteristic polynomial method is impractical. Therefore we examine effective iterative methods for the eigenvalue problem, characterized by the use of the Krylov subspace.

3.1 The Krylov Subspace

Named after Russian applied mathematician and naval engineer, Alexei Nikolaevich Krylov, the Krylov subspace was named after his investigation of the Cayley-Hamilton theorem[1] and studying the characteristic equation of a matrix. There he exploited a consequence of the theorem, which states the inverse of a nonsingular matrix can be expressed as a linear combination of its powers. Krylov was interested in analyzing oscillations of mechanical systems in his 1931 paper, where he proposed a method for numerically approximating the minimal polynomial of a matrix [6]. He was one of the first computational scientists of his time that counted the work done by an algorithm as a number of separate numerical arithmetic operations [2] .

For large sparse matrices, which have most of the elements of the matrix as

[1] In linear algebra, the Cayley-Hamilton theorem simply states that the given square matrix A satisfies a root to its own characteristic polynomial. When over an algebraic field, the Cayley-Hamilton extends and is equivalent to saying that the minimal polynomial of the matrix divides its characteristic polynomial. Discussion on this can be found in [3].

[2] More information on Krylov can be found in the short biography by A. I. Balkashin, *Outstanding scientist and ship-builder Alexei Nikolaevich Krylov* (in Russian).

zeros, matrix-vector multiplication is a relatively quick and inexpensive process. Due to this nice property, we can store large matrices in a compact form and execute matrix-vector multiplication operations efficiently. This is an essential task for many iterative methods in the context of eigenvalue computations and solving systems of linear equations. A main ingredient in Krylov subspace methods is, as the name suggests, the *Krylov subspace*. We begin with a simple definition.

Definition 3.1.1. *Let $A \in \mathbb{R}^{n\times n}$ and $s \in \mathbb{R}^n$ with $s \neq 0$. Then we define the sequence of the form*

$$s, As, A^2s, A^3s, \ldots, A^{m-1}s \tag{3.1}$$

to be the Krylov sequence associated with A and s. The corresponding subspace

$$\mathcal{K}_m(A, s) = Span\{s, As, A^2s, A^3s, \ldots, A^{m-1}s\} \tag{3.2}$$

is the m-dimensional Krylov subspace associated with A and s.

This can be extended to the vector spaces $\mathbb{C}^{n\times n}$ and $\mathbb{C}^n$. If we choose to build a Krylov subspace, we first need a non-zero starting vector s. In most applications, knowledge of a "good" starting vector is unknown, thus typically a randomly generated vector would suffice. At the outset of a typical Krylov method, we simply take the vector s and apply a given matrix to it in each iteration. The overall dimension of the Krylov subspace is subjective to the application, and is typically decided based on what is necessary to get an appropriate result. Some applications take only a few iterations, while others may take hundreds or thousands. In a restarted method, which we will discuss in a later chapter, we attempt to choose a better starting vector after a certain amount of iterations to improve overall performance.

The Krylov subspace finds its applications in many different modern iterative methods for both solving linear systems of equations and eigenvalue computations. These methods are among the most widely used and efficient algorithms to date.

The best known Krylov subspace methods for linear systems and eigenvalues are the Arnoldi, Lanczos, conjugate gradient, bi-conjugate gradient stabilized (BiCGSTAB), minimal residual (MINRES), and generalized minimum residual (GMRES) methods [5, 7, 8, 10].

If our Krylov subspace has dimension m, then these methods will converge in m iterations, assuming exact arithmetic [10]. However, in the presence of numerical round-off error, this does not hold true; in fact, in practice m can be exceedingly large and the iterative process of building the Krylov subspace becomes too computationally expensive. In general, the convergence of Krylov subspace methods depends on the eigenvalue spectrum of the given matrix. For the purpose of this thesis, we will discuss the convergence of these methods in the context of the eigenvalue problem. Only slight modifications are needed to study the convergence of Krylov methods for linear systems [8, 10].

3.2 Convergence of Krylov Subspace Methods

We now consider the general framework of how, given an m-dimensional Krylov subspace, $\mathcal{K}_m(A, s)$, we can obtain good approximations to the eigenvectors of a matrix A. For practical purposes, we suppose that $m << n$ if A is an $n \times n$ square matrix. We first begin by establishing a characterization of $\mathcal{K}_m(A, s)$ in terms of polynomials.

By definition, we have that $\mathcal{K}_m(A, s)$ is the set of all linear combinations

$$c_0 s + c_1 A s + c_2 A^2 s + \ldots + c_{m-1} A^{m-1} s \tag{3.3}$$

Given any set of coefficients $c_0, c_1, \ldots, c_{m-1}$, we can construct a polynomial $q(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{m-1} x^{m-1}$ which is of degree at most $m - 1$. We then can extend this polynomial to be a matrix polynomial of degree $m - 1$. Now we can explicitly write, in a more compact form, the linear combination in (3.3) as $q(A)s$.

We summarize below with the following proposition.

Proposition 3.2.1. *Let $\mathcal{P}_{m-1}$ denote the set of all polynomials of degree at most $m-1$. That is, $\mathcal{P}_{m-1} = \{q(x) : deg(q) \leq m-1\}$. Then we now have*

$$\mathcal{K}_m(A, s) = \{q(A)s : q \in \mathcal{P}_{m-1}\} \tag{3.4}$$

Proof. This is easily shown by observing that the matrix polynomial is $q(A) = c_0 I + c_1 A + c_2 A^2 + \ldots c_{m-1} A^{m-1}$. Applying s, we get that $q(A)s$ is precisely all arbitrary linear combinations in (3.2). □

In analyzing whether or not $\mathcal{K}_m(A, s)$ contains a sufficiently good approximation to a given eigenvector of A, say some z_i, it is a matter of finding such a polynomial $q \in \mathcal{P}_{m-1}$ such that $q(A)s \approx z_i$. We now go through a simple argument to show this.

Let $A \in \mathbb{R}^{n \times n}$ and let us suppose that A has a linearly independent set of eigenvectors, $z_1, z_2, \ldots, z_n$, with associated eigenvalues, $\lambda_1, \lambda_2, \ldots, \lambda_n$. Since we have a complete set of linearly independent eigenvectors, they form a basis such that given a vector $s \in \mathbb{R}^n$ we may write

$$s = c_1 z_1 + c_2 z_2 + \ldots + c_n z_n$$

for some unknown coefficients $c_1, \ldots, c_n$. Then for any polynomial q we have

$$q(A)s = c_1 q(A) z_1 + c_2 q(A) z_2 + \ldots + c_n q(A) z_n,$$

and by the fact that $z_1, \ldots, z_n$ are eigenvectors of A, we have

$$q(A)s = c_1 q(\lambda_1) z_1 + c_2 q(\lambda_2) z_2 + \ldots + c_n q(\lambda_n) z_n. \tag{3.5}$$

If we wish to find a good approximation to a specific eigenvector of A, say, z_j, then we should look for a polynomial q such that $|q(\lambda_j)|$ is relatively large compared to

$|q(\lambda_1)|, |q(\lambda_2)|, \ldots, |q(\lambda_{j-1})|, |q(\lambda_{j+1})|, \ldots, |q(\lambda_n)|$. This is most easily accomplished when λ_j is well separated from the eigenvalue spectrum of A. If the eigenvalues in the spectrum are clustered together, with λ_j separated off at a distance, then a polynomial q with $degree(q) = j$ that has all of its roots, or zeros, in the cluster would fulfill the condition easily. Thus $\mathcal{K}_j(A, s)$ contains vectors that approximate z_j well.

The closer λ_j is to the rest of the eigenvalue spectrum, the more difficult it is to find a polynomial q with the desired properties. It is even more difficult when A contains eigenvalues with multiplicity greater than 1. The theoretical polynomials used to analyze the convergence of Krylov subspace methods are typically Chebyshev polynomials. More in-depth discussion on Chebyshev polynomials and this convergence analysis can be found in [7, 8, 10].

Mathematically, all Krylov subspace methods are based on projection-type methods. For example, instead of solving an eigenvalue problem of a potentially very large matrix, the idea is that we can approximate a select few of the eigenvalues of the matrix projected onto the Krylov subspace of smaller dimensions. What we have discussed above is that if the matrix given is unitarily diagonalizable, then the convergence behavior of various Krylov methods is determined by its eigenvalue spectrum. This isn't true in the case that the matrix is highly non-Normal. In the next section, we discuss how projecting onto the Krylov subspace and utilizing the Rayleigh-Ritz procedure allows us to approximate select eigenpairs of a matrix. This is the foundation of many Krylov subspace methods for eigenvalue computations and solving linear systems.

In the case for when our $n \times n$ matrix A does not have a set of n distinct eigenvalues, then the convergence analysis of Krylov subspace methods becomes more difficult to predict. We are dealing with multiple eigenvalues in this case. Most of the time, numerical round-off error will rectify the problem of finding multi-

ple eigenvalues by introducing a small amount of error that accumulates over time. Eventually, the error will be significant enough to produce a copy of another eigenvector corresponding to a multiple eigenvalue. We will discuss this in more detail and give some results demonstrating this.

CHAPTER FOUR

The Arnoldi Method

In this chapter, we now discuss an iterative method based on projecting onto the Krylov subspace for extracting a few eigenvalues and eigenvectors of a large matrix. The projection techniques are the foundation of many classes of eigenvalue algorithms avaliable, and have shown to be quite effective. The Arnoldi method was invented by American applied mathematician Walter Edwin Arnoldi in 1951. His paper, *The principle of minimized iterations in the solution of the eigenvalue problem*, is one of the most cited works in the field of numerical linear algebra [4, p. 653]. The Arnoldi method is an eigenvalue algorithm typically tailored towards large sparse matrices; it does not directly modify the elements of the matrix, but instead gains information by projecting onto a subspace. It is a combination of a subspace projection technique, called the Rayleigh Ritz procedure, and the Krylov subspace [9].

4.1 Rayleigh-Ritz Procedure

Now that we have established some foundational concepts and definitions for basic Krylov methods, we discuss the Rayleigh-Ritz procedure that is utilized in the Arnoldi method. We first present the Rayleigh-Ritz for the case of a general subspace.

The General Rayleigh Ritz Procedure

Let $A \in \mathbb{R}^{n\times n}$ be a general matrix. Then the following steps make up the Rayleigh Ritz procedure:

1. Let $\mathcal{S} \subset \mathbb{R}^n$ be a m-dimensional subspace.

2. Compute an orthonormal basis $\{v_i\}_{i=1,\dots,m}$, of the subspace $\mathcal{S}$. Define a ma-

trix $V = [v_1, v_2, \ldots, v_m]$. Then V is an orthonormal matrix whose columns span $\mathcal{S}$.

3. Compute the m by m matrix $H = V^T AV$.

4. Compute the eigenvalues θ_i of H and select k desired ones to keep. If desired, find the normalized eigenvectors g_i of H. The θ_i are approximate eigenvalues of A and are referred to as *Ritz values*. The approximate eigenvectors of A, $y_i = Vg_i$, are called *Ritz vectors*. The residual norms are defined to be $r_i = \| Ay_i - \theta y_i \|$, where $i \in [1, 2, \ldots k]$.

The particular subspace used by the Arnoldi method in the Rayleigh-Ritz procedure is the Krylov subspace [10]. We now break down the Rayleigh-Ritz procedure and examine how we can approximate the eigenpairs of a matrix.

Let $A \in \mathbb{R}^{n \times n}$ and suppose we are given a starting vector $s \in \mathbb{R}^n$ and we have generated an m-dimensional Krylov subspace $\mathcal{K}_{m-1}(A, s) \subset \mathbb{R}^n$. The Rayleigh-Ritz procedure seeks to find an approximate eigenpair (θ, y) with $y \in \mathcal{K}_{m-1}$. This approximate eigenpair is obtained by imposing an orthogonal projection of A onto the Krylov subspace, often called the *Galerkin projection*, or the *Galerkin condition* [7, 8, 10], which in mathematical notation is:

$$r \equiv Ay - \theta y \perp \mathcal{K}_{m-1}$$

or equivalently expressed in terms of inner products,

$$\langle r, v \rangle = \langle Ay - \theta y, v \rangle = 0, \forall v \in \mathcal{K}_{m-1}$$

Now we must first generate an orthonormal basis $\{v_1, v_2, \ldots, v_m\}$ of $\mathcal{K}_{m-1}$. For now, let us suppose we are given such a basis. Constuct $V = [v_1, v_2, \ldots, v_m]$, and let $y = Vg \in \mathcal{K}_{m-1}$, where $g \in \mathbb{R}^n$. Then after interpreting the Galerkin condition, we now have

$$\langle AVg - \theta Vg, v_j \rangle = 0, j \in [1, 2, \ldots, m]. \tag{4.1}$$

So we now have that the vector g, and the scalar θ must satisfy

$$AVg = \theta Vg$$
$$V^T(AVg) = V^T(\theta Vg)$$
$$V^T AVg = \theta(V^T Vg)$$

Notice that since V is a matrix whose columns v_i , $i = 1, 2, \ldots, m$, are orthonormal, then V itself is an orthogonal matrix whose columns span the Krylov subspace $\mathcal{K}_{m-1}$. A property of orthogonal matrices is that $V^T V = I$, the identity matrix, so we now have

$$V^T AVg = \theta g.$$

Define $H = V^T AV$ and we have

$$Hg = \theta g. \tag{4.2}$$

Therefore θ must be an eigenvalue of H and g is the corresponding eigenvector.

In most practical implementations of this procedure, A is a much larger matrix in comparison to H whose dimensions are $m \times m$. Both A and H are not similar in the sense of a similarity transformation, since $m \leq n$. However, Arnoldi showed that the sequence of Ritz values converge to the true eigenvalues of the original matrix [9]. Thus, up to m eigenvalues of A are the same eigenvalues of H. Therefore we have reduced a large $n \times n$ problem to a smaller, $m \times m$ problem, allowing for an easier computation to find the eigenpairs of H. The numerical solution of the $m \times m$ eigenvalue problem can be treated by standard library subroutines such as the MATLAB[1] functions found in LAPACK and LINPACK. The significance of the

[1] MATLAB is a technical computing environment for high-performance numeric computation and simulation. It is used extensively in both academia and industry. In this thesis, we will be using MATLAB as our programming environment.

formula $V^T AV = H$ lies in the fact that this can be interpreted as the orthogonal projection via a partial similarity transformation of A onto the Krylov subspace, with the columns of V serving as the basis vectors.

Now that we have fully presented the general Rayleigh-Ritz procedure, we are now able to present our implementation of the Arnoldi method for computing select eigenpairs of a given matrix A. We will use the Rayleigh-Ritz procedure with the Krylov subspace, as well as recall the Gram-Schmidt procedure, as our implementation uses the modified Gram-Schmidt algorithm. There are several ways to implement the Arnoldi method, but the idea is the same overall and primarily differ in the method of orthogonalization.

4.2 The Arnoldi Recurrence

We first present a definition that applies to the particular matrix generated in the Arnoldi iteration.

Definition 4.2.1. *An $n \times n$ matrix with $a_{i,j} = 0$ for $i > j + 1$ is called a Hessenberg matrix. This form is also commonly called upper-Hessenberg. The form of this matrix is*

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n-1} & a_{2,n} \\ 0 & a_{3,2} & \dots & a_{3,n-1} & a_{3,n} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

In the case that A is symmetric, then the structure becomes tridiagonal.

The Arnoldi method is an orthogonal projection technique for a general non-Hermitian matrix A. The method utilizes a partial unitary similarity transformation to reduce a large, dense matrix into Hessenburg form. Arnoldi's paper revealed that the eigenvalues of the Hessenberg matrix, obtained from a procedure called the

Arnoldi factorization, can provide accurate approximations to some eigenvalues of the original matrix A [9]. Below we present the basic implementation of the Arnoldi method. In our implementation, we use the modified Gram-Schmidt procedure, as it is well known that the classical method is numerically unstable [5, 7, 8, 10, 13].

The Basic Arnoldi Method with Modified Gram-Schmidt (MGS)

Let $\| \, . \, \|$ denote the standard Euclidean 2-norm.

1. *Initialization Step:* Choose a starting vector s, and let $v_1 = s/ \| \, s \, \|$.

2. *Generation of the Krylov subspace:* For $j = 1, 2, \ldots, m$ do:

$$w_j = Av_j,$$
$$h_{i,j} = w_j^T v_i, \; i = 1, 2, \ldots, j,$$
$$w_{j+1} = w_j - \textstyle\sum_{i=1}^{j} h_{i,j} v_i,$$
$$h_{j+1,j} = \| \, w_{j+1} \, \|,$$
$$v_{j+1} = w_{j+1}/h_{j+1,j}.$$

3. *Rayleigh Ritz Procedure:* For the m by m upper-Hessenberg matrix H, compute the Ritz values θ_i. Compute the Ritz vectors $z_i = V g_i$, where $V = [v_1, v_2, \ldots, v_m]$.

4. *Monitor the error:* Compute the residual norms of the approximate eigenpairs of A, $\| \, r_i \, \| = \| \, Ay_i - \theta_i y_i \, \|$, and check for convergence. This is done by terminating the method when the Krylov subspace has been built up to dimension m that is user specified at the start.

We will now present the Arnoldi factorization as presented in many advanced linear algebra texts. [8, 10, 11].

Theorem 4.2.2. *The Arnoldi Factorization: Let V_m be the $n \times m$ matrix with orthonormal columns $v_1, v_2, \ldots v_m$, $\bar{H}_m$ as the $(m+1) \times m$ Hessenberg matrix obtained*

from the Arnoldi method, and H_m be the matrix obtaining from $\bar{H}_m$ by deleting the final row. Then the following factorizations hold:

$$AV_m = V_m H_m + g_m e_m^T \tag{4.3}$$

$$= V_{m+1}\bar{H}_m, \tag{4.4}$$

$$V_m^T A V_m = H_m, \tag{4.5}$$

where $g_m = h_{m+1,m}v_{m+1}$ is a residual vector and $g_m e_m^T$ is a residual rank-one matrix.

Proof. We will prove this by appealing to the outline of the Arnoldi method presented above. Let $j \in [1, \ldots, m]$ and $\{v_i\}_{i=1,\ldots,j}$ be the orthonormal basis of $\mathcal{K}_j(A, s)$. Then by step 2 of the Arnolid method we have

$$w_{j+1} = Av_j - \textstyle\sum_{i=1}^{j} h_{i,j}v_i$$

where $h_{i,j} = w_j^T v_i$, $i = 1, 2, \ldots, j$. This is just the modified Gram-Schmidt procedure. Thus the work vector w_{j+1} is orthogonal to $\mathcal{K}_j(A, s)$. By the algorithm, we also have

$$h_{j+1,j} = \parallel w_{j+1} \parallel$$

and we define v_{j+1} which is orthogonal to $\mathcal{K}_j(A, s)$ by construction. Observe now that if we have $i > j + 1$, then $h_{i,j} = 0$ by the fact that $v_{j+1} = w_j / \parallel w_j \parallel$ was orthogonalized against all previous vectors. From step 2, we have

$$\begin{aligned} w_{j+1} &= Av_j - \sum_{i=1}^{j} h_{i,j}v_i \\ &= h_{j+1,j}v_{j+1}. \end{aligned}$$

Rearranging we get,

$$Av_j = h_{j+1,j}v_{j+1} + \textstyle\sum_{i=1}^{j} h_{i,j}v_i = \sum_{i=1}^{j+1} h_{i,j}v_i$$

Thus for $j \in [1, \ldots m]$, we rewrite the expression in matrix form as:

$$A[v_1, v_2, \ldots, v_m] = [v_1, v_2, \ldots, v_m, v_{m+1}]\bar{H}_m,$$

where $\bar{H}_m$ is the matrix

$$\bar{H}_m = \begin{bmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,m+1} & h_{2,m} \\ 0 & h_{3,2} & \ldots & h_{3,m-1} & h_{3,m} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \\ 0 & 0 & \ldots & 0 & h_{m+1,m} \end{bmatrix}$$

Thus, if we define H_m to be the partitioned matrix of $\bar{H}_m$ by removing the $m+1$ row, we have H_m is an $m \times m$ upper-Hessenberg matrix and we may write:

$$A[v_1, v_2, \ldots, v_m] = [v_1, v_2, \ldots, v_m]H_m + h_{m+1,m}v_{m+1}e_m^T,$$

where e_m is the m^{th} standard basis vector.

By setting $V_m = [v_1, v_2, \ldots, v_m]$ and $g_m = h_{m+1,m}v_{m+1}$, the factorizations proposed are achieved. □

This algorithm generates the Arnoldi factorization. Equation (4.2) is also commonly referred to as the *Arnoldi Recurrence* [10]. As stated earlier, the algorithm above utilized the modified Gram-Schmidt procedure. The Arnoldi method can be implemented with the classical Gram-Schmidt approach, and it would be mathematically equivalent to the one presented above, however this is only the case in exact arithmetic. In the presence of numerical round-off error, the Arnoldi method with the stabilized Gram-Schmidt is much more reliable.

In practice, the Arnoldi algorithm can only compute some of the approximate eigenvalues of a given matrix A. Since H_m is only $m \times m$, it has at most m approximations to the eigenvalues of A ($m < n$ if A is $n \times n$). If more approximations are desired, then it is necessary to increase the size of H_m and build up a larger Krylov

subspace. This is not always practical, depending on the size of the matrix. In the case that A is symmetric, then H_m will be tri-diagonal and the Arnoldi method reduces to what is known as the Lanczos method, using what is called the *Lanczos three-term recurrence* [10].

4.3 Numerical Experiments: Non-restarted Arnoldi with MGS

In theory, a Krylov subspace method will produce only a single eigenvector corresponding to a multiple eigenvalue. As stated by Morgan and Zeng in [14], one can compute eigenvectors of multiple eigenvalues by running a non-restarted or restarted Arnoldi method until round-off error introduces extra copies. This method is not ideal, because the eigenvalues appear one at a time. Also, it is difficult to predict how long we must run the Arnoldi method in case more copies appear. However, waiting for round-off error has the potential to work reasonably well.

Example 1. Consider as an example the matrix A that is the discretized two-dimensional finite difference Laplacian. This particular discretization is on the unit square with Dirichlet boundary conditions. The discretization size is $h = 1/51$ on each side, thus making the dimension of the matrix A to be $n = 2500$. The 10 smallest eigenvalues are 0.00759, then we have a double eigenvalue of 0.0190, a simple eigenvalue at 0.0303 and three sets of double eigenvalues of 0.0378, 0.0492, and 0.0642. The starting vector for the basic Arnoldi is generated random Normal(0,1). Figure 4.1 has a plot of the 10 smallest Ritz values at each iteration. By about iteration 170, all Ritz values begin to settle into place. Notice, however, there is a disturbance in the spectrum beginning at around iteration 260 as a new value appears and works its way down to join the second eigenvalue. This is the second copy of this particular double eigenvalue at 0.0190. At iteration 300, another new value joins with the second double eigenvalue at 0.0378. By iteration 350, the first four double eigenvalues have all been joined with its second copies. This example

illustrates that the Arnoldi with waiting for round-off error is successful at finding these double eigenvalues.

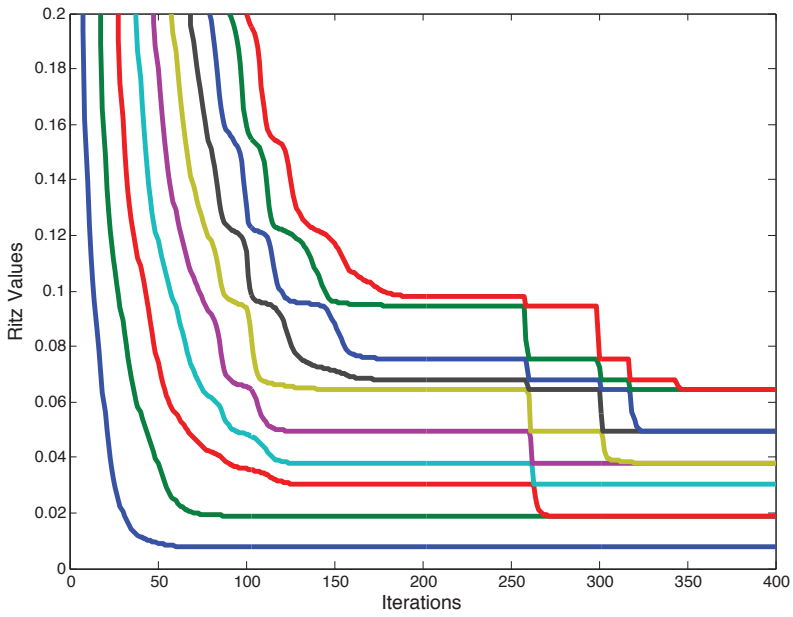

Figure 4.1: Non-restarted Arnoldi method with the 10 smallest Ritz values plotted at each iteration. The matrix is the 2-D Laplacian with $n = 2500$. By iteration 350, there are approximations to all 10 of the smallest eigenvalues, including the 4 double eigenvalues.

Continuing with this example further, we now examine how the smallest double eigenvalue converges. Figure 4.2 shows convergence of the first copy with the residual norm of the corresponding eigenvector against the iteration. At the same time, we also examine how round-off error brings about the second copy. Doing this requires a couple steps. First we find the vector in the eigenspace of the eigenvalue at 0.0190 that is orthogonal to the eigenvector of the first copy. Then we compute the eigenvector corresponding to the second copy. We then measure how much of this second eigenvector is present in the Krylov subspace and monitor it as the iterations proceed. The dashed line has the norm of the component of this

eigenvector projected onto the Krylov subspace. At the start, the component of the second eigenvector in the Krylov subspace has a value smaller than machine epsilon for MATLAB, which is roughly around 2.22×10^{-16}, and increases at approximately the same rate that the residual of the first copy is improving. From the figure, we can observe that the Arnoldi method is working on both copies equally, even though the second is submerged by the first for now. Once the component reaches around 1.0, the second copy appears and begins converging. The dashed-dotted line shows the convergence of the second copy. Notice the initial slow convergence of the first copy, then the more rapid convergence once approximations to nearby eigenpairs are accurate enough. The accurate eigenpairs are essentially deflated from the spectrum, allowing for faster convergence of the eigenvalues we are focusing on.

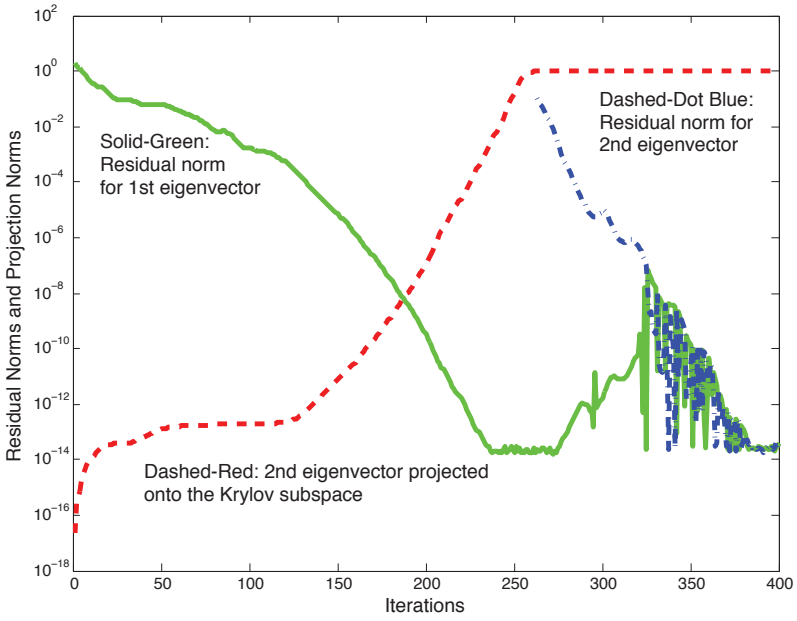

Figure 4.2: Non-restarted Arnoldi method with convergence shown for both eigenvectors corresponding with the multiple eigenvalue at 0.0190, plus the norm of the projection of the second eigenvector onto the Krylov subspace. The matrix is the 2-D Laplacian with $n = 2500$. This shows how the 2nd and 3rd eigenvalues converge.

Figure 4.3 gives similar graphs for the sixth set of double eigenvalues, which correspond to the 14th and 15th eigenvalues respectively. The convergence is slower initially due to more interference. This is due to the fact that, from the Krylov convergence analysis, the more interior eigenvalues are more computationally challenging. Again, we observe similar behavior as before. The second copy in the Krylov subspace roughly matches the convergence of the first copy. Once the component of the second eigenvector in the subspace has a component near 1.0, the second copy appears and converges faster than from its start.

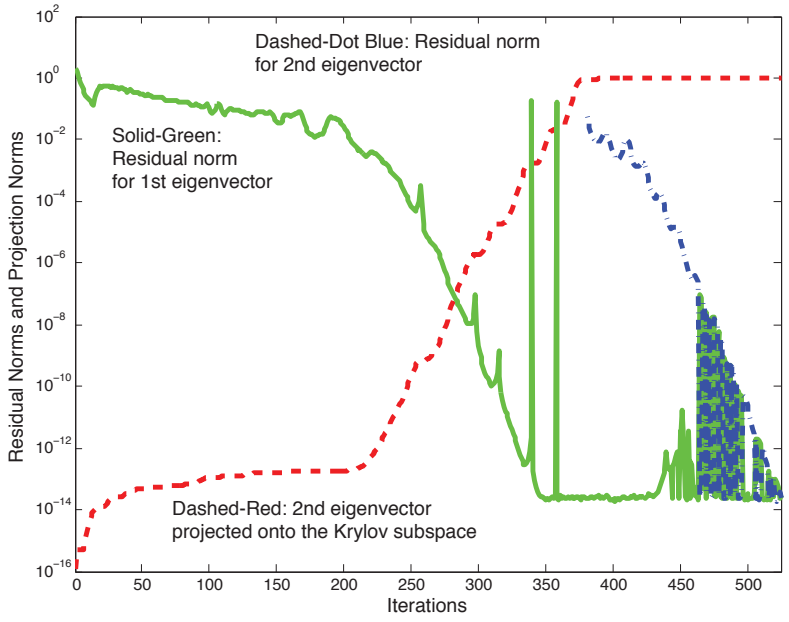

Figure 4.3: Non-restarted Arnoldi method with convergence shown for both eigenvectors corresponding with the 6th set of multiple eigenvalue plus the norm of the projection of the second eigenvector onto the Krylov subspace. The matrix is the 2-D Laplacian with $n = 2500$. This shows how the 14th and 15th eigenvalues converge

Example 2. We now consider a matrix from the three-dimensional discretized Laplacian on the unit cube with Dirichlet boundary conditions. The matrix is of size $n = 1000$. As with the previous example, we plot the 10 smallest eigenvalues.

This time, we have a simple eigenvalue at 0.2430, followed by three sets of triple eigenvalues at 0.4795, 0.7169, and 0.8523. Figure 4.4 shows the plot of the 10 smallest Ritz values at each iteration. Notice this time that at around iteration 80, we see the second copy of the first triple eigenvalue drop down, then the third at iteration 140. The behavior in the case with triple eigenvalues is similar to the double case; the multiple eigenvalues are found one at a time. If we wish to approximate all the triple eigenvalues of the 10 smallest ones, we will need to build up a larger Krylov subspace in order to give round-off error enough time to create the necessary copies.

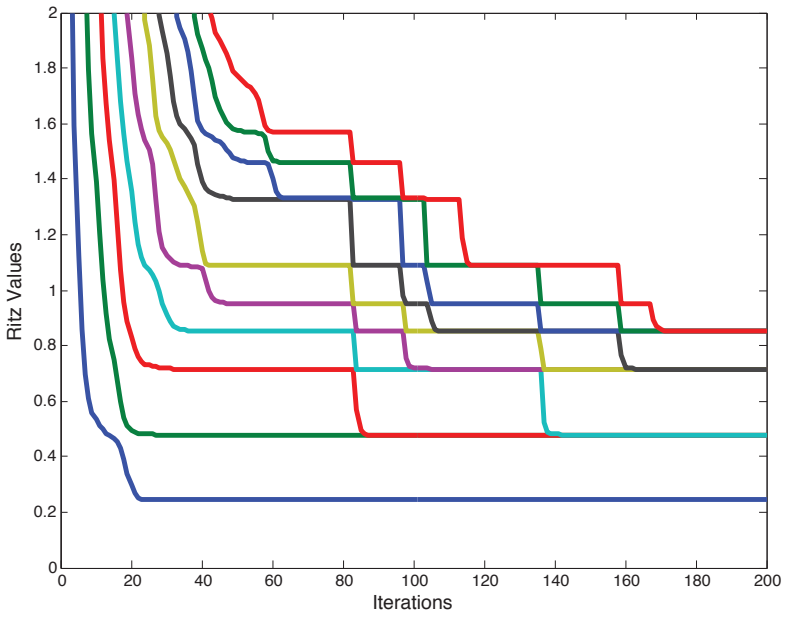

Figure 4.4: Non-restarted Arnoldi method with the 10 smallest Ritz values plotted at each iteration. The matrix is the 3-D Laplacian with $n = 1000$. By iteration 165, there are approximations to the first set of triple eigenvalues.

Figure 4.5 has a plot similar to that of Figure 4.2 and Figure 4.3, but for the smallest triple eigenvalue, which are the second through fourth eigenvalues. Notice here that, as demonstrated by Figure 4.4, the second and third copies of this eigen-

value develop one at a time. The component of the third eigenvector in the Krylov subspace, shown with the black dotted line, only starts to build up after the second copy has appeared.

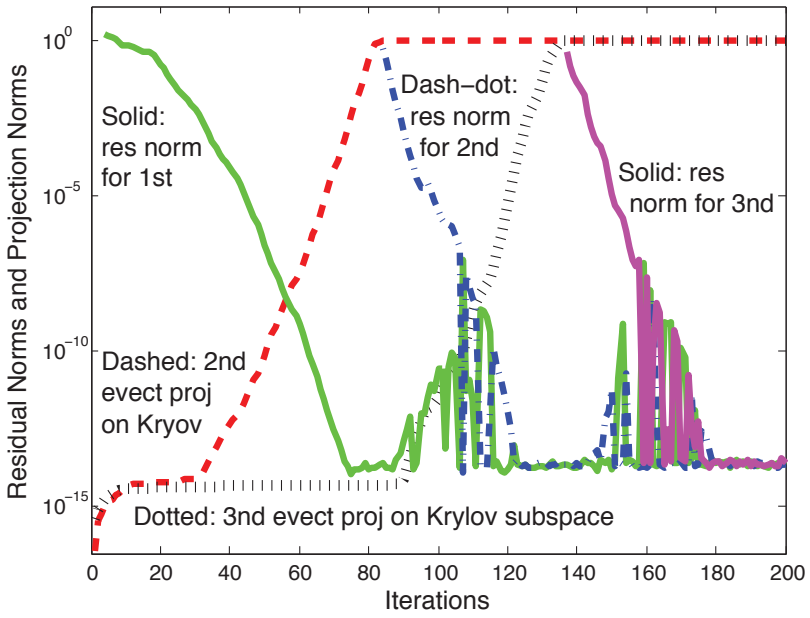

Figure 4.5: Non-restarted Arnoldi method with the convergence shown for both eigenvectors corresponding to the triple eigenvalue, plus the norms of the projection of the second and third eigenvectors onto the Krylov subspace. The matrix is the 3-D Laplacian with $n = 1000$. This shows how the 2nd, 3rd, and 4th eigenvalues converge.

We have seen in the previous examples how round-off error can help find multiple eigenvalues. However this is not the only option. As stated earlier, the multiple copies come down one at a time. It is possible to bring down multiple copies at once, but with the cost of some computational expense via methods that utilize multiple starting vectors. We now discuss the use of a block Krylov subspace and how it compares to the standard implementation of the non-restarted Arnoldi method.

CHAPTER FIVE

The Block Arnoldi Method

The block Arnoldi method [11] is a natural extension of the Arnoldi method. Generally speaking, block Arnoldi is an extension of the Krylov subspace generated as before, but with multiple starting vectors. We will first outline the general procedure of a block Arnoldi method, as well as extend the Arnoldi recurrence to a block analogue. Then we will discuss our implementation of block Arnoldi coupled with numerical examples of its performance in dealing with multiple eigenvalues.

5.1 The Block Arnoldi Process

The Block Arnoldi Procedure [10]

Let A be $n \times n$. Then the following steps will build the block Krylov subspace:

1. *Initialization Step:* Choose p orthonormal vectors $\{v_i\}_{i=1,\dots,p}$ and let V_1 be the unitary matrix of dimension $n \times p$ whose columns are the v_i.

2. *Building the Block Krylov subspace:* For $j = 1, 2, \dots, m$ do:

   $H_{i,j} = V_i^T A V_j$, $i = 1, 2, \dots, j$,

   $W_j = AV_j - \sum_{i=1}^{j} V_i H_{i,j}$,

   $W_j = V_{j+1} H_{j+1,j}$ is the QR factorization of W_j

Now let I_k be the $k \times k$ identity matrix, and we extend the Arnoldi recurrence to a block analogue.

Theorem 5.1.1. *The Block Arnoldi Factorization: Let $A \in \mathbb{R}^{n \times n}$. Define $U_m = [V_1, V_2, \dots, V_m]$, $H_m = (H_{i,j})_{1 \le i,j \le m}$, $H_{i,j} = 0$, if $i > j + 1$, where all elements are defined from the procedure above, and let E_m be the matrix of the last p columns of the identity I_{np}. Then,*

$$AU_m = U_m H_m + V_{m+1} H_{m+1,m} E_m^T,$$

or

$$AU_m = U_{m+1} \bar{H}_{m+1,m},$$

where $\bar{H}_{m+1,m}$ *is the* $(m+1)p \times mp$ *block upper Hessenberg matrix with an additional block row whose only nonzero block entry is* $H_{m+1,m}$*. Note that* H_m *is band-Hessenberg, in the sense that we have* $p-1$ *additional diagonals below the subdiagonal. This is the block analogue of the Arnoldi recurrence relation established in the Arnoldi factorization theorem from the previous chapter [10, 11].*

The proof for this is a generalization of the proof from theorem 4.2.1 and follows a similar argument [10, 12].

Proof. Notice that the block Arnold procedure utilizes the QR factorization, which involves Gram-Schmidt orthogonalization. Thus the resulting block matrices $U_m = [V_1, V_2, \ldots, V_m]$ have all their columns orthogonal and linearly independent, provided that the QR factorization yields no rank-deficient upper-triangular matrices. Define the generalized Krylov matrix subspace to be:

$$\mathbb{K}_m = Span\{V_1, AV_1, A^2V_1, \ldots, A^{m-1}V_1\}.$$

Now observe that, by the QR factorization in step 2, the matrices $V_1, V_2, V_3, \ldots, V_m$ form an orthonormal basis for $\mathbb{K}_m$. Now by step 2 of the procedure, H_m is the band-Hessenberg matrix whose form is given as:

$$
\begin{aligned}
H_m &= (H_{i,j})_{1\leq i,j\leq m} \\
&= (V_i^T A V_j)_{1\leq i,j\leq m} \\
&= U_m^T A U_m \\
&= \begin{bmatrix}
H_{1,1} & H_{1,2} & \dots & H_{1,m-1} & H_{1,m} \\
H_{2,1} & H_{2,2} & \dots & H_{2,m+1} & H_{2,m} \\
0 & H_{3,2} & \dots & H_{3,m-1} & H_{3,m} \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
0 & \dots & 0 & H_{m,m-1} & H_{m,m}
\end{bmatrix}.
\end{aligned}
$$

Notice that the submatrices below the main diagonal $H_{i,i-1}$, $2 \leq i \leq m$ are upper triangular matrices generated from the QR factorization. Thus H_m has the form as claimed. From the above block Arnoldi procedure, for $j \in [1, 2, \dots, m]$ we have:

$$
\begin{aligned}
W_j &= AV_j - \sum_{i=1}^{j} V_i H_{i,j} \\
&= V_{j+1} H_{j+1,j}.
\end{aligned}
$$

Rearranging we get,

$$AV_j = V_{j+1}H_{j+1,j} + \textstyle\sum_{i=1}^{j} V_i H_{i,j} = \sum_{i=1}^{j+1} V_i H_{i,j}$$

Thus we may rewrite as

$$AU_m = U_{m+1}\bar{H}_{m+1,m}$$

where $\bar{H}_{m+1,m}$ is the block matrix

$$\bar{H}_{m+1,m} = \begin{bmatrix} H_{1,1} & H_{1,2} & \dots & H_{1,m-1} & H_{1,m} \\ H_{2,1} & H_{2,2} & \dots & H_{2,m+1} & H_{2,m} \\ 0 & H_{3,2} & \dots & H_{3,m-1} & H_{3,m} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ & & & H_{m,m-1} & H_{m,m} \\ 0 & 0 & \dots & 0 & H_{m+1,m} \end{bmatrix}.$$

This is the form as claimed. And finally, we may simplify further to get the block recurrence

$$AU_m = U_m H_m + [0, \dots, 0, U_{m+1} H_{m+1,m}] = U_m H_m + U_{m+1} H_{m+1,m} E_m^T.$$

□

The implications of this theorem leaves an interesting investigation for those interested in modifying and constructing block Krylov methods that utilize the Arnoldi recurrence. Further reading and information on this can be found in [11, 12, 15]. It is important to note that the above procedure is simply theoretical, and in practice the algorithm must be modified in the presence of round-off error. We now present a modified algorithm developed by A. Ruhe, see reference [16].

The Block Arnoldi - Ruhe's variant

1. *Initialization Step:* Choose p orthonormal vectors $\{v_i\}_{i=1,\dots,p}$.

2. *Building the Block Krylov subspace:* For $j = p, p+1, \dots, m+p-1$ do:

   (a) Set $k = j - p + 1$

   (b) Compute $w = Av_k$

   (c) For $i = 1, 2, \dots, j$, do:

   $$h_{i,k} = w^T v_i$$

   $$w = w - h_{i,k} v_i$$

(d) Compute $h_{j+1,k} = \| w \|$ and $v_{j+1} = w/h_{j+1,k}$.

The advantage to this algorithm is its simplicity to implement. Also notice that for $p = 1$, this coincides with the standard non-restarted Arnoldi algorithm. We will use this implementation for our numerical experiments in the next section. The purpose of this is to demonstrate how the Block Arnoldi performs in the context of finding multiple eigenvalues.

5.2 Numerical Experiments: Block Method Approach

As stated before, the block Arnoldi generates a Krylov subspace for each of several starting vectors and puts the basis vectors together to form one large subspace. In exact arithmetic, block methods can find multiple eigenvalues of multiplicity up to the block size. However, there are disadvantages to using a block method; one being computational expense. In the following examples, we will run numerical experiments and compare to the non-restarted Arnoldi algorithm in the presence of round-off error.

Example 3. We use the matrix as in Example 1 and apply an non-restarted block Arnoldi to the matrix. Our MATLAB implementation follows Ruhe's outline of the block method and uses a block size of $p = 2$. Figure 5.1 has a plot similar to that in Figure 4.1, however, the multiple eigenvalues appear in unison rather than separately. It is important to note that in each iteration, since the block size is 2, we count two matrix-vector products to build up the Krylov subspace. Thus, the block Arnoldi demonstrating convergence of the first pair of double eigenvalues at around iteration 140 roughly corresponds to 280 iterations of the standard Arnoldi method. Therefore the two methods are roughly equivalent for the case when $n = 2500$, with the block method taking slightly longer. This is to be expected, since we are generating two Krylov subspaces contained in one larger subspace, the Krylov polynomials are of lesser degree than the polynomials generated by the standard

Arnoldi. This makes it more difficult for the Arnoldi to converge quickly, since we are limiting how large the degree of the polynomial can be. What is remarkably different about the block method approach is the fact that our eigenvalues are appearing together, rather than dropping down one by one as shown in Figure 4.1.

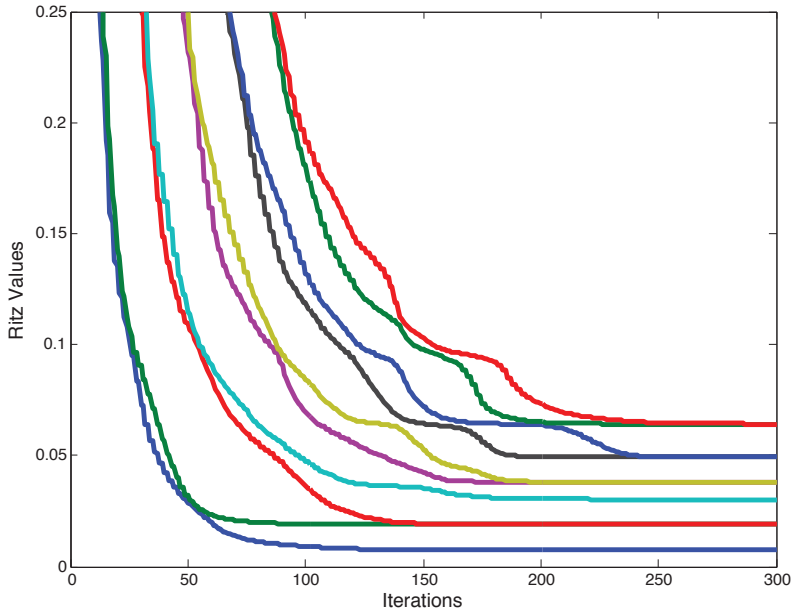

Figure 5.1: Non-restarted block Arnoldi method with the 10 smallest Ritz values plotted at each iteration. We have a block size of $p = 2$. The matrix is the 2-D Laplacian with $n = 2500$. By iteration 250, there are approximations to all 10 of the smallest eigenvalues, including the 4 double eigenvalues.

To further expand on the performance of the block Arnoldi for the double eigenvalue problem, we present a table comparing the performance of the block method with the standard Arnoldi for the 2-D Laplacian of various sizes. Table 5.1 displays the increasing sizes of the test matrix, and will list the iterations until the first double eigenvalue is found for the block method, $\mathcal{B}_n$, and the standard Arnoldi, $\mathcal{A}_n$. The iterations equivalent for the block method compared to non-restarted Arnoldi is also given for a more realistic comparison, denoted as $\mathcal{E}_n$. For relatively

small n, the block method and the standard non-restarted Arnoldi method perform roughly equivalent in the presence of round-off error for finding the first double eigenvalue. However, this changes as n increases. In fact, it consistently gets worse as n increases. While the rate at which the block Arnoldi worsens is somewhat variable, it is consistent in that its performance suffers in comparison as the matrix becomes large.

Table 5.1: Comparisons for 2-D Laplacian of various size n. Displays the direct comparisons between the block Arnoldi method with non-restarted Arnoldi. The iterations displayed indicates at what iteration the method will find the first double eigenvalue.

	Iterations		
Size (n)	$\mathcal{B}_n$	$\mathcal{E}_n$	$\mathcal{A}_n$
900	80	160	160
1600	105	210	210
2500	140	280	260
3600	180	360	310
4900	225	450	370
6400	265	530	415
8100	300	600	475
10000	340	680	525

Example 4. We use the matrix as in Example 2 with $n = 3375$ and apply the block Arnoldi method with a block size of $p = 3$. Figure 5.2 displays how the triple eigenvalues appear as the block Arnoldi method iterates. As before, the Ritz values converge together; since the block size is $p = 3$, the triple eigenvalues are appearing together and converge at around iteration 100 rather than separately like in Example 2. Since we have three starting vectors, we now are counting three matrix-vector products to build up the Krylov subspace. Thus, the block Arnoldi demonstrating convergence of the first triple eigenvalue at iteration 100 corresponds to around iteration 300 of non-restarted Arnoldi. Recall that in Example 2 all three

of the first triple eigenvalues of the same matrix are found at iteration 205. This is significantly worse in comparison. From a theoretical analysis of the convergence behavior of the Arnoldi, this makes sense since having a block size of $p = 3$ further restricts the size of the Krylov polynomial. Therefore it is reasonable to expect the block Arnoldi to suffer in performance even more so than in Example 3. In table 5.2 we present comparisons for the triple eigenvalue problem with the 3-D Laplacian of varying sizes. The results confirm our expectations. The notation is the same as previous; we compare the equivalent iteration count of the block Arnoldi with three starting vectors to the non-restarted Arnoldi method. The difference in this table is that we display the iteration when all three eigenvalues are found.

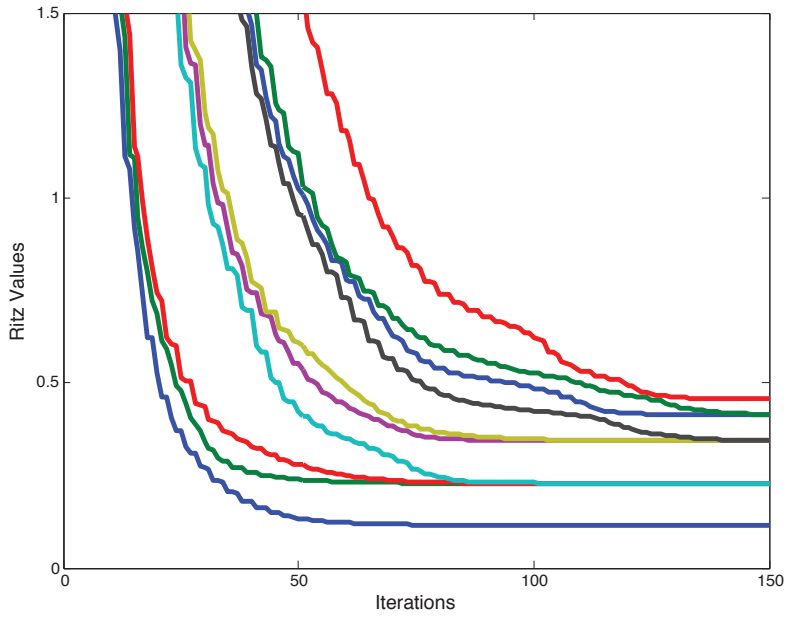

Figure 5.2: Non-restarted block Arnoldi method with the 10 smallest Ritz values plotted at each iteration. We have a block size of $p = 3$. The matrix is the 3-D Laplacian with $n = 3375$. By iteration 100, we approximate the first set of triple eigenvalues.

Table 5.2: Comparisons for 3-D Laplacian of various size n. Displays the direct comparisons between the block Arnoldi method with non-restarted Arnoldi. The iterations displayed indicates at what iteration the method will find the all three of the first triple eigenvalues.

	Iterations		
Size (n)	$\mathcal{B}_n$	$\mathcal{E}_n$	$\mathcal{A}_n$
1000	75	225	140
2197	90	270	180
3375	100	300	205
4913	120	360	230
6859	130	390	260
8000	145	435	280

We have seen with the previous results that the block Arnoldi method on its own is not much better than the standard non-restarted Arnoldi. What the block method exhibits, however, is an interesting contrast in how round-off error can bring about extra copies of multiple eigenvalues. In the case with the non-restarted Arnoldi, we witnessed the multiple copies appearing one at a time. With a block method, we have the opportunity to bring multiple copies down at once. For double eigenvalues, having two starting vectors allowed the block method to work on improving approximations to them simultaneously. This was further illustrated in the triple eigenvalue case with three starting vectors. However, the multiplicity of eigenvalues is not always known before implementing Arnoldi. The purpose of these experiments was to illustrate how multiple starting vectors can help find multiple eigenvalues. Taking into account the storage and computational costs of a block method, this raises the concern of efficiency and how we can continue to improve it. This method for computing eigenvalues and eigenvectors involves a numerical technique called restarting. Thus in the next chapter, we will discuss a restarted Arnoldi algorithm and see how it performs in computing multiple eigenvalues.

CHAPTER SIX

The Restarted Arnoldi Method

Sorensen's implicitly restarted Arnoldi (IRA) [17] is the standard method for computing eigenvalues and eigenvectors of a large nonsymmetric matrix. The Arnoldi method requires calculating the eigenvalues and eigenvectors of a Hessenberg matrix of order m at the cost of $\mathcal{O}(m^3)$ operations. Because of this, restarting is generally necessary. However, restarting slows down the convergence. In this chapter we briefly analyze how we restart the Arnoldi method.

As with the non-restarted Arnoldi method, the restarted Arnoldi will utilize both the Rayleigh-Ritz procedure and the Arnoldi recurrence. We now present an outline of the basic restarted Arnoldi.

The Basic Restarted Arnoldi Algorithm

1. *Initialization Step:* Choose a starting vector s, and let $v_1 = s/ \parallel s \parallel$.

2. *Generation of the Krylov subspace:* For $j = 1, 2, \ldots, m$ do:

$$w_j = Av_j,$$

$$h_{i,j} = w_j^T v_i,\ i = 1, 2, \ldots, j,$$

$$w_{j+1} = w_j - \sum_{i=1}^{j} h_{i,j} v_i,$$

$$h_{j+1,j} = \parallel w_{j+1} \parallel,$$

$$v_{j+1} = w_{j+1}/h_{j+1,j}.$$

3. *Rayleigh Ritz Procedure:* For the m by m upper-Hessenberg matrix H, compute the Ritz values θ_i. Compute the Ritz vectors $y_i = Vg_i$, where $V = [v_1, v_2, \ldots, v_m]$. Check residual for convergence.

4. *Restart:* Choose a new starting vector, possibly a Ritz vector or a linear combination of Ritz vectors. Normalize for v_1, and go to step 2 for the next cycle (we define a cycle to be each pass through the algorithm, in between restarts). The end of a restart will indicate the completion of one cycle of the restarted Arnoldi.

The Arnoldi recurrence from the previous chapters is

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T. \tag{6.1}$$

We will now derive a well-known formula for the residual norm. We will derive this using the Arnoldi recurrence:

$$\begin{aligned} r_i &\equiv Ay_i - \theta_i y_i \\ &= AV_m g_i - \theta_i V_m g_i \\ &= V_m H_m g_i + h_{m+1,m} v_{m+1} e_m^T g_i - \theta_i V_m g_i \\ &= V_m (H_m g_i - \theta_i g_i) + h_{m+1,m} v_{m+1} e_m^T g_i \\ &= h_{m+1,m} v_{m+1} e_m^T g_i \\ &= h_{m+1,m} e_m^T g_i v_{m+1}, \end{aligned}$$

since g_i is an eigenvector of H_m and $e_m^T g_i$ is a scalar. Thus by defining $\gamma_{mi} \equiv h_{m+1,m} e_m^T g_i$ we get

$$r_i = Ay_i - \theta_i y_i = \gamma_{mi} v_{m+1}, \tag{6.2}$$

and

$$\| r_i \| = \| Ay_i - \theta_i y_i \| = |\gamma_{mi}| \tag{6.3}$$

This gives a a quick calculation for computing residual norms and can relieve some of the computational expense when working with a large matrix problem.

As we have discussed earlier in this thesis, a higher-degree Krylov polynomial gives more accurate approximations to the eigenvalues of A. As what was observed in the block Arnoldi method, a problem with restarting is that we limit the size of the Krylov subspace and thus the degree of the Krylov polynomial. There are several ways of dealing with this problem. More discussion on this can be found in [10, 11, 13, 14, 17]. For the rest of this thesis, we use an implementation presented by Morgan and Zeng in [14] and give some numerical results.

6.1 Morgan's Restarted Arnoldi

Every cycle of Sorensen's implicitly restarted Arnoldi uses the subspace of the form

$$Span\{y_1, y_2, y_3, \ldots, y_k, r, Ar, A^2r, A^3r, \ldots, A^{m-k-1}r\} \tag{6.4}$$

where $\{y_1, y_2, \ldots, y_k\}$ are the Ritz vectors computed at the end of the previous cycle via the Rayleigh Ritz procedure, and r is the vector v_{m+1} from the previous cycle. Notice that from our derivation of equation (6.2), r is a multiple of the residual vectors for each of these Ritz vectors. This result can be found in [17]. In [13] Morgan shows that the subspace above is equivalent to

$$Span\{y_1, y_2, y_3, \ldots, y_k, Ay_i, A^2y_i, A^3y_i, \ldots, A^{m-k}y_i\} \tag{6.5}$$

for any i such that $1 \leq i \leq k$. So the subspace contains a Krylov subspace with each Ritz vector as a starting vector and potentially they all can be converging together. This is the basis for Morgan's implementation of a restarted Arnoldi. We present a simpler method that is mathematically equivalent to Sorensen's implicitly restarted Arnoldi [13]. This method is also used in [14] for the nonsymmetric case, and this is the version of the restarted Arnoldi that we will use. A harmonic Rayleigh-Ritz version is given in [14], but we only use the standard Rayleigh Ritz procedure, which

is equivalent to the procedure in Sorensen's method [17]. The algorithm is given below. At the restart, we form an orthonormal basis for $Span\{y_1, y_2, y_3, \ldots, y_k, r\}$ in the columns of V_{k+1}^{new}. We then build out to an orthonormal basis for the subspace in (6.4) with the standard Arnoldi iteration as the cycle proceeds.

Restarted Arnoldi: Arnoldi(m,k)

1. *Start and initial definitions:* Choose m, the maximum dimension of the Krylov subspace, and k, the number of approximate eigenvectors that we wish to recycle from one cycle to the next. Also, define $numev$, the number of desired eigenpairs we wish to approximate. It is standard in most restarted methods to keep more approximate eigenvectors than the desired number to assist the convergence of the algorithm. Thus, $k > numev$. Choose an initial starting vector v_1 with unit length.

2. *The Arnoldi process:* Apply the Arnoldi method from the current starting vector to generate the Arnoldi-like recurrence $AV_m = V_{m+1}H_{m+1,m}$. The current starting vector is either v_1 if it is the first cycle, or v_{k+1} from a previous cycle. Here $H_{m+1,m}$ is upper-Hessenberg for the first cycle. In the proceeding cycles, $H_{m+1,m}$ is no longer upper-Hessenberg, but possesses a structure that is upper-Hessenberg except for a full leading $k+1 \times k+1$ portion.

3. *Rayleigh Ritz procedure:* Compute the k appropriate eigenpairs (θ_i, g_i), with g_i normalized, of the square matrix $H_{m,m}$. The θ_i are the Ritz values. Compute the Ritz vectors $y_i = V_m g_i$.

4. *Check convergence:* Compute the residual norms using $\| r_i \| \equiv \| Ay_i - \theta_i y_i \| = h_{m+1,m}|g_i| = |\gamma_{mi}|$ and convergence can be checked. If all desired $numev$ eigenpairs have an acceptable residual norm, then stop. Otherwise continue.

The next step begins the restarting process.

5. *Orthonormalization of first k short vectors:* Orthonormalize g_i's, for $1 \leq i \leq k$, first separating into real and imaginary parts if vectors are complex. We use the components to form the real $m \times k$ matrix P_k. Both parts of the complex vectors need to be included. It may be necessary to reduce or augment k by 1.

6. *Form P:* Extend the columns of P_k, $\{p_1, p_2, p_3, \ldots, p_k\}$ to length $m+1$ by appending a zero vector to each, then set $p_{m+1} = e_{m+1}$, where e_{m+1} is the $(m+1)$st coordinate vector of length $m+1$. Let $P_{m+1,k+1}$ be the $m+1 \times k+1$ matrix with $\{p_i\}_{i=1}^{k+1}$ as the columns of the matrix.

7. *Form portions of the new H and V matrices using the old H and V:* Define $H_{k+1,k}^{new} = P_{k+1}^T H_{m+1,m} P_k$ and $V_{k+1}^{new} = V_{m+1} P_{m+1,k+1}$. Then let $H_{k+1,k} = H_{k+1,k}^{new}$ and $V_{k+1} = V_{k+1}^{new}$.

8. *Reorthogonalization of long $k+1$ vector:* Now orthogonalize v_{k+1} against the earlier columns of the new matrix V_{k+1}. Go to step 2.

We give the main expense of one cycle in this algorithm, considering only length n vector operations. The Arnoldi iteration requires approximately $m^2 - k^2$ vector operations for orthogonalization. In addition, there are also $m - k$ matrix-vector products per cycle at the cost of $(m-k)matvec$, where $matvec$ is the cost of multiplying a vector with the matrix A. Forming the basis for the Ritz vectors in V_{k+1}^{new} takes approximately km vector operations. Thus, the approximate total operation count is $m^2 + km - k^2 + (m-k)matvec$ [14].

Full reorthogonalization is generally necessary with Arnoldi methods. It is absolutely imperative to have an orthonormal basis. Without reorthogonalization, numerical round-off error can potentially remove the orthogonality of our basis,

and thus our algorithm can fail [18]. In our implementation, we use the modified Gram-Schmidt procedure. All experiments here do use this process in Step 2 of the algorithm. For Step 3 of this implementation of a restarted Arnoldi, we note that it is important to use the 'nobalance' option[1] in MATLAB and to avoid the balancing subroutine in LINPACK. This avoids large errors that can be introduced by the balancing option.

6.2 Numerical Experiments: Restarted Arnoldi

We now present some numerical experiments with the restarted Arnoldi in order to have a more realistic method. The results are fairly similar in both how extra copies of eigenvalues appear and in how long it takes for the restarted method to detect them.

Example 5. We use the matrix as in Example 1 and Example 3. This time we apply the restarted Arnoldi method, Arnoldi(35,15), presented in the previous section. We have $numev = 10$, so we are computing approximations to the first 10 eigenvalues of the 2-D Laplacian of size $n = 2500$. We are also restarting with 15 Ritz vectors from the previous iteration to help buffer the desired eigenvectors. Figure 6.1 has a plot similar to the previous plots in Figures 5.1 and 4.1. Notice the more rough appearance. The plot is different in the sense that each cycle, with the exception of the first cycle, contains approximately 20 iterations of the standard Arnoldi method. The first run has 35 iterations. Thus the figure shows the first double eigenvalue dropping at cycle 14, which corresponds to 295 iterations. The restarted Arnoldi is therefore slightly more expensive than the non-restarted Arnoldi method, but is generally a much more efficient algorithm. The expense comes from

[1] Balance option, specified as one two strings: 'balance', which enables a preliminary balancing step, or 'nobalance' which disables it. In most cases, the balancing step improves the conditioning of A to produce more accurate results. However, there are cases in which balancing produces incorrect results. For example, if A contains nonzero integers, as well as very small (near zero) values, then the balancing step might scale the small values to make them as significant as the integers and produce inaccurate results.

the fact that we are restricting the Krylov subspace dimension to not exceed 35. We can find the iteration count of Arnoldi(m,k) as a function of the number of cycles: $\mathcal{I}(c) = m + (m-k)(c-1)$, where c is the number of cycles.

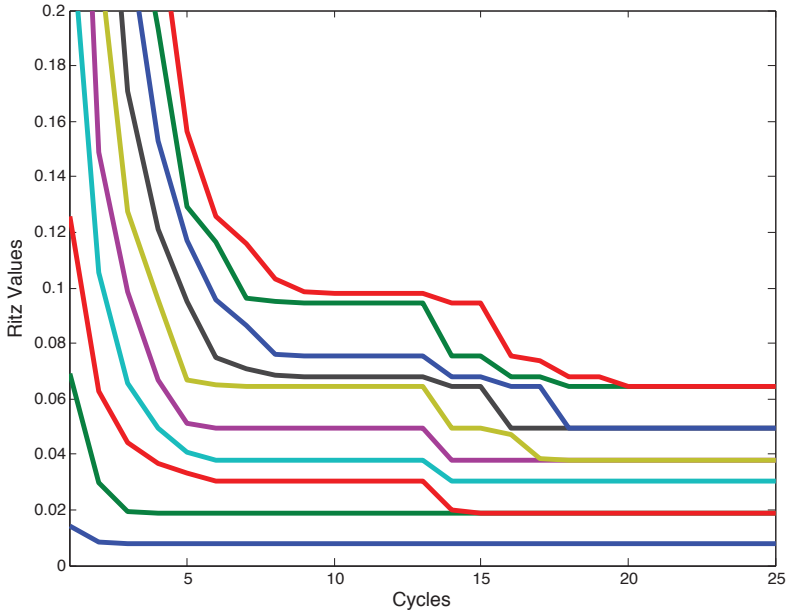

Figure 6.1: Restarted Arnoldi method, Arnoldi(35,15), with the 10 smallest Ritz values plotted against the cycle. The matrix is the 2-D Laplacian with $n = 2500$. By cycle 20, we have approximations to all 10 of the smallest eigenvalues, including the 4 double eigenvalues.

Next, Figures 6.2 and 6.3 give similar graphs for the residuals of the eigenvectors corresponding to the first set and sixth set of double eigenvalues respectively. Notice how the behavior of the second eigenvector corresponding to the first double eigenvalues is like that of the non-restarted Arnoldi method; the appearance of the second copy in the Krylov subspace roughly matches the convergence of the first copy. Figure 6.3 shows the residuals of the eigenvectors corresponding to the sixth set of double eigenvalues. Since these eigenvalues are more interior, we have to build a larger dimension Krylov subspace and we choose to restart with additional Ritz

vectors. We used Arnoldi(40,20) instead of the previous Arnoldi(35,15).

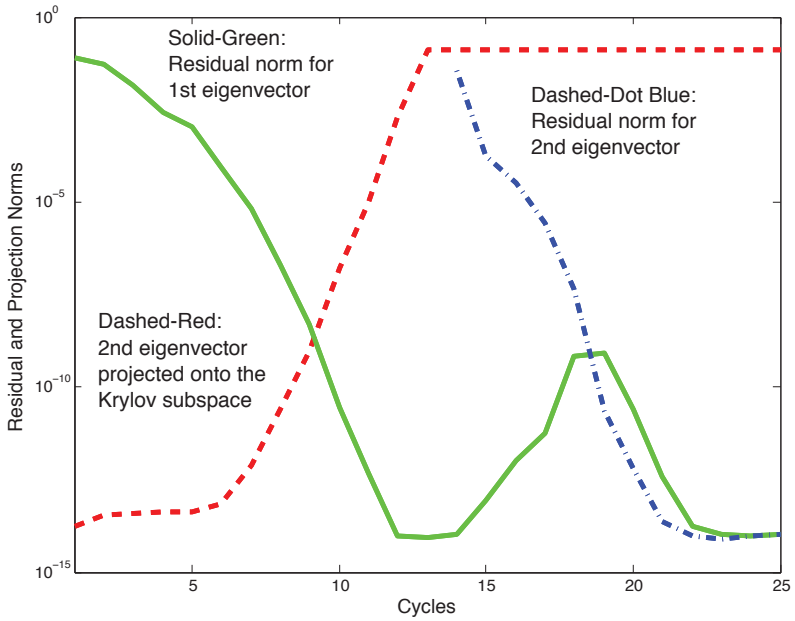

Figure 6.2: Restarted Arnoldi method, Arnoldi(35,15), with convergence shown for both eigenvectors corresponding with the multiple eigenvalue at 0.0190, plus the norm of the projection of the second eigenvector onto the Krylov subspace. The matrix is the 2-D Laplacian with $n = 2500$. This shows how the 2nd and 3rd eigenvalues converge.

Example 6. We use the the 3-D Laplacian matrix from the previous examples. Again, we see similar results to that of the non-restarted Arnoldi. Figure 6.4 plots the Ritz values against the cycles. We see convergence of all three of the first triple eigenvalues at cycle 11, or 235 iterations.

To give a more complete comparison, we present tables as in Examples 5 and 6. In Table 6.1 and we present various sizes of the 2-D Laplacian to give more perspective in comparing the methods. In Table 6.2, we do the same for the 3-D Laplacian. In general, restarted Arnoldi is more expensive than the standard method. The restricted size of the Krylov subspace will require that the restarted method will need to run longer than without restarting.

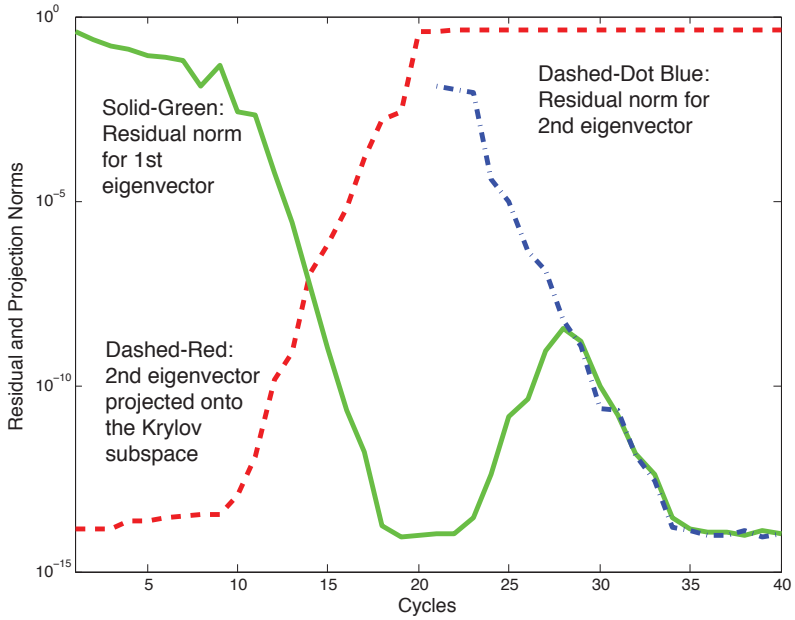

Figure 6.3: Restarted Arnoldi method, Arnoldi(40,20), with convergence shown for both eigenvectors corresponding with the 6th set of multiple eigenvalue plus the norm of the projection of the second eigenvector onto the subspace. The matrix is the 2-D Laplacian with $n = 2500$.

Table 6.1: Comparisons for 2-D Laplacian of various size n. Displays the direct comparisons between Arnoldi(35,15) with non-restarted Arnoldi. The results displayed indicate at what iteration the method will find the first double eigenvalues.

	Iterations		
Size (n)	c	$\mathcal{I}(c)$	$\mathcal{A}_n$
900	7	175	160
1600	10	235	210
2500	13	295	260
3600	16	355	310
4900	19	415	370
6400	22	475	420
8100	24	515	475
10000	29	615	525
12100	31	655	580
14400	38	795	625

In conclusion, restarted Arnoldi generally has slower convergence in comparison with non-restarted Arnoldi, since we are restricting the size of the Krylov sub-

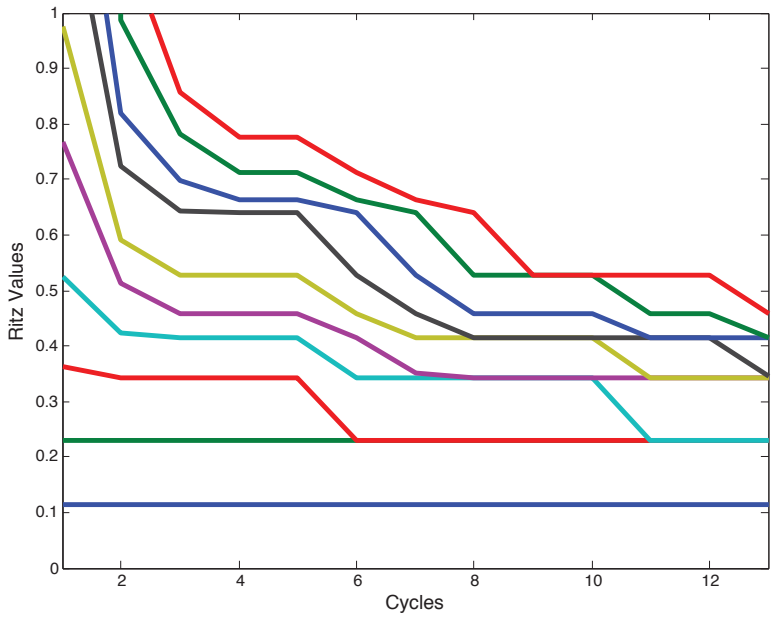

Figure 6.4: Restarted Arnoldi method, Arnoldi(35,15), with the 10 smallest Ritz values plotted against the cycle. The matrix is the 3-D Laplacian with $n = 3375$. By cycle 11, we have approximations to the first set of triple eigenvalues.

Table 6.2: Comparisons for 3-D Laplacian of various size n. Displays the direct comparisons between Arnoldi(35,15) with non-restarted Arnoldi. The iterations displayed indicates at what iteration the method will find all the first triple eigenvalues.

	Iterations		
Size (n)	c	$\mathcal{I}(c)$	$\mathcal{A}_n$
1000	6	155	140
2197	9	215	180
3375	10	235	205
4913	12	275	230
6859	13	295	260
8000	14	315	280

space. However, restarting is necessary for large problems due to practical storage considerations. Even though some of the information from the previous cycle is discarded because of restarting, the most important information is retained for the next cycle. This makes restarted methods very effective and adaptable [11, 13, 14, 17].

CHAPTER SEVEN

Perturbed Arnoldi Methods

In this thesis, we are concerned about the effective computation of multiple eigenvalues of a large matrix. In the previous chapters we outlined and presented numerical results of the Arnoldi method and a couple variations of it in the form of a block method, and a restarted method. In practice, the restarted Arnoldi is the standard procedure for industrial work and Sorensen's implicitly restarted Arnoldi is among the most well-known eigenvalue algorithms to date. The Arnoldi algorithm's convergence, as such with all Krylov subspace methods, depends on the eigenvalue spectrum of the matrix. The case when a matrix yields eigenvalues of higher multiplicity can present a challenging problem. We have seem how round-off error can bring about the desired copies of the eigenvalues in all three previous methods. In this chapter, we examine how matrix perturbations can help separate multiple eigenvalues, and thus improve convergence of the Arnoldi method.

7.1 Perturbing Multiple Eigenvalues

Matrix perturbation theory is a rich field and numerous amounts of literature can be found discussing the behavior of perturbing a general non-Hermitian matrix. The specific problem of examining the eigenvalue spectrum of various matrices gave rise to the investigation of explaining the behavior of multiple eigenvalues of a general matrix when the matrix is perturbed slightly. More specifically, we are interested in under what circumstances might an eigenvalue of multiplicity p spawn p simple eigenvalues after perturbation. Furthermore, what explanation is required to claim that the new eigenvalue will be found at a varying distance from the original eigenvalue. According to Sun in [19] and Stewart in [20], the distance at which eigenvalues are separated is a characteristic of the original matrix rather than the

perturbation itself. Thus it is reasonable to assume that a multiple eigenvalue can have several sensitivity factors in how each multiple eigenvalue is separated; that is, multiple eigenvalues do not separate uniformly and can greatly differ. This is demonstrated by example in [20]. The claim here is that almost any randomly chosen perturbation matrix would cause separation. This implies that the case where separation of multiple eigenvalues does not occur is a singular case. It is unfortunate that we cannot say with complete mathematical rigor that we are guaranteed the separation of multiple eigenvalues of a general non-Hermitian matrix (the Hermitian case is discussed in [21]). We leave this investigation for further work. We turn our attention now to some numerical examples that help verify that multiple eigenvalues will separate and improve the convergence of the Arnoldi method.

7.2 Diagonal Matrix Perturbations

In this section, we look at computing multiple eigenvalues of a general matrix A as before, but this time, we will perturb the matrix. Specifically, we will use a diagonal matrix. Given our matrix A, we apply our method of choice to the perturbed matrix:

$$\hat{A} = A + \sigma P,$$

where we define σ to be the perturbation factor, and P is a diagonal perturbation matrix whose diagonal elements are generated via Normal$(0, 1)$, and $\| P \| = 1$, the spectral norm of P.

Example 7. We use the 2-D finite difference Laplacian as in the previous examples and apply the non-restarted Arnoldi method to the perturbed matrix $\hat{A} = A + \sigma P$, where $\sigma = 10^{-6}$. Figure 7.1 has a plot similar to Figure 4.1, however, the multiple eigenvalues appear sooner. The second copy of the smallest double eigenvalue appears at iteration 200 compared to the earlier result at iteration 260 for the unperturbed Arnoldi. We get approximations to all 10 of the smallest eigen-

values, including the 4 doubles, by iteration 280 versus 350 for the unperturbed example.

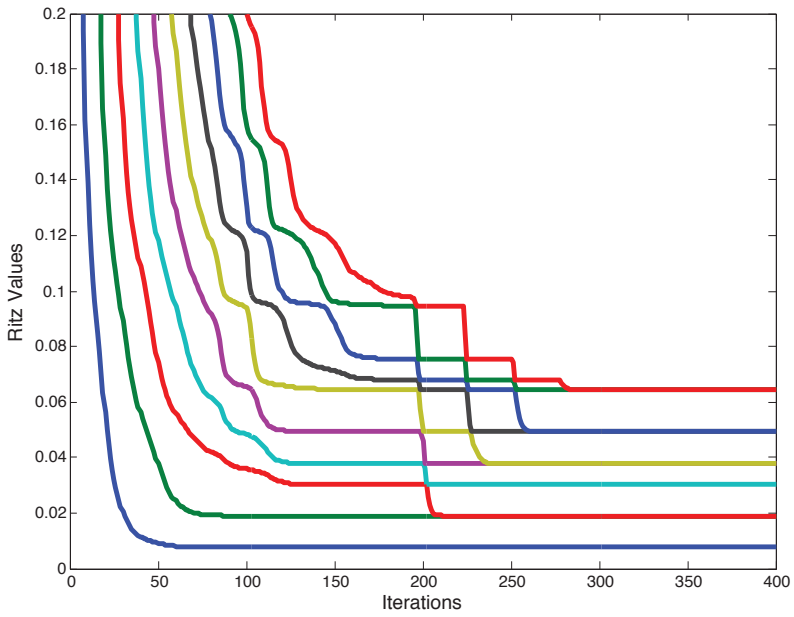

Figure 7.1: Non-restarted perturbed Arnoldi method with $\sigma = 10^{-6}$ and the 10 smallest Ritz values plotted at each iteration. The matrix is the 2-D Laplacian with $n = 2500$. By iteration 280, there are approximations to all 10 of the smallest eigenvalues, including the 4 double eigenvalues.

We now switch to the restarted method, Arnoldi(35,15), in order to have a more realistic method. The results are fairly similar in both how extra copies appear and in how much the perturbation quickens the appearance of the double eigenvalues. Applying Arnoldi(35,15) to the same matrix, we restart the subspace when it reaches dimension 35, then compute 15 Ritz vectors for the next cycle. Figure 7.2 shows a similar plot to Figure 7.1 with $\sigma = 10^{-6}$. The second copy of the smallest double eigenvalue appears at cycle 10, or iteration 215, compared to 295 without perturbation. We get approximations to all 10 Ritz values by iteration 315, or cycle 15. This is an improvement from 415 iterations without perturbation.

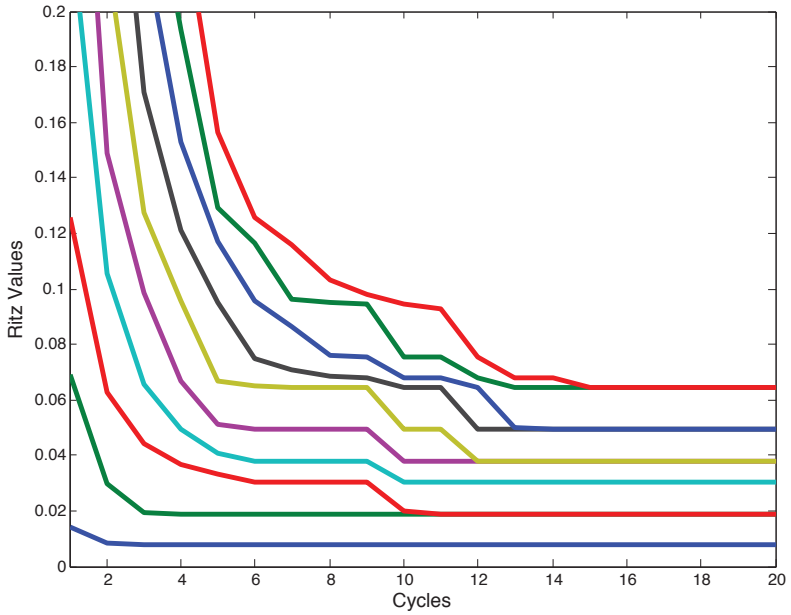

Figure 7.2: Restarted Arnoldi method, Arnoldi(35,15), with $\sigma = 10^{-6}$ and the 10 smallest Ritz values plotted against the cycle. The matrix is the 2-D Laplacian with $n = 2500$. By cycle 15, we have approximations to all 10 of the smallest eigenvalues, including the 4 double eigenvalues.

To help examine how the perturbation further affects the restarted Arnoldi method, we now monitor the residuals of the target eigenvectors with varying σ. Figure 7.3 shows the residual norm for the second copy of the first double eigenvalue (the second and third overall). It is clear that as σ increases, the copies appear much faster, however we are sacrificing accuracy. Therefore, depending on the application in mind, the trade off is left to the user to decide what is tolerable. Figure 7.4 shows the same plot, but for the second copy of the fourth double eigenvalue corresponding with the ninth and tenth eigenvalues.

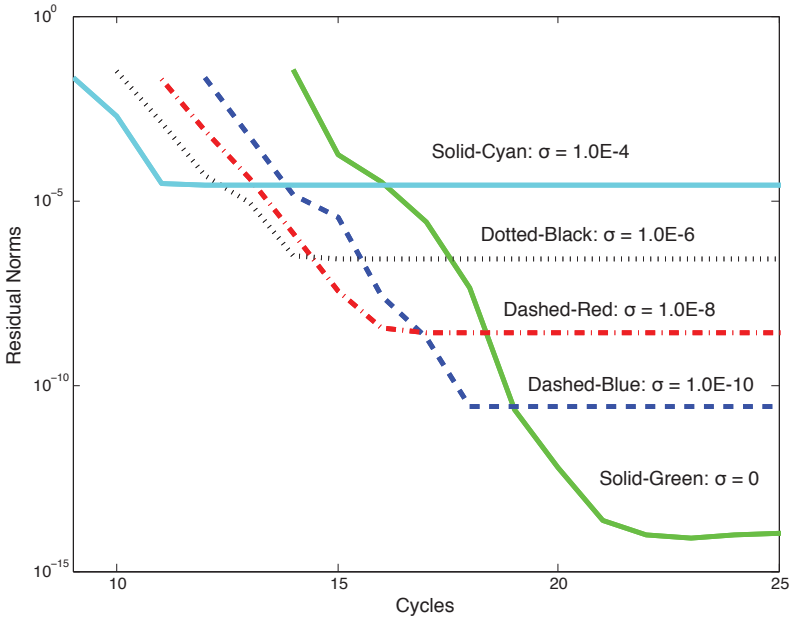

Figure 7.3: Restarted Arnoldi method, Arnoldi(35,15), with varying diagonal perturbations σ. The residual norms for the first double eigenvector is shown. The matrix is the 2-D finite difference Laplacian with $n = 2500$.

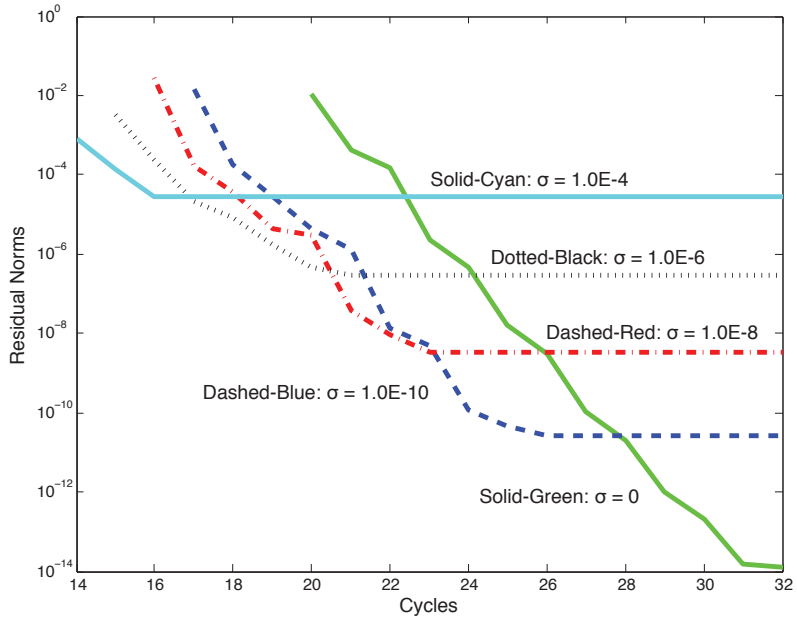

Figure 7.4: Restarted Arnoldi method, Arnoldi(35,15), with varying diagonal perturbations σ. The residual norms for the fourth double eigenvector is shown. The matrix is the 2-D finite difference Laplacian with $n = 2500$.

Example 8. We repeat the same structure as in Example 8, but this time we use the 3-D finite difference Laplacian of size $n = 3375$. Figure 7.5 plots the 10 smallest Ritz values against the iteration count for the non-restarted Arnoldi with perturbation factor $\sigma = 10^{-6}$. As with before, we see that the triple eigenvalues appear sooner. At iteration 143, the first triple eigenvalues have been approximated, and by iteration 175 we get the second set of triple eigenvalues. See Figure 4.4 for a direct comparison.

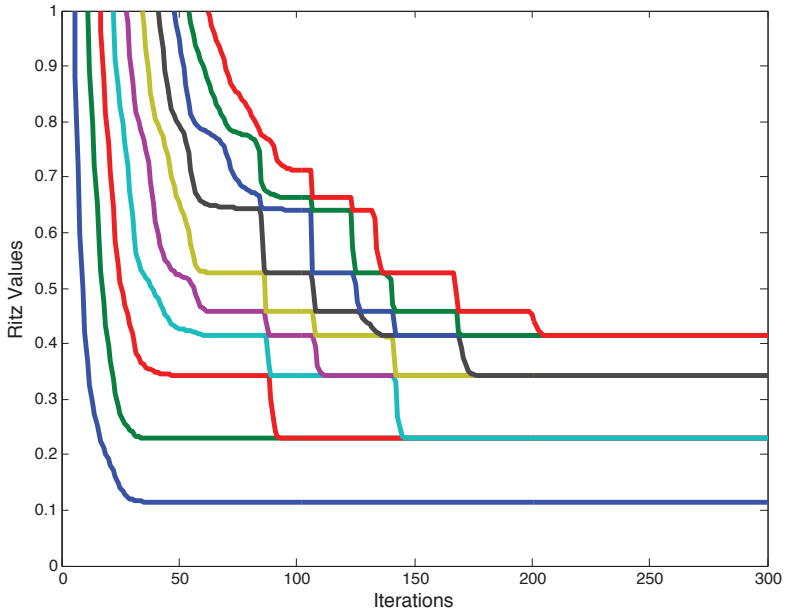

Figure 7.5: Non-restarted perturbed Arnoldi method with $\sigma = 10^{-6}$ and the 10 smallest Ritz values plotted at each iteration. The matrix is the 3-D Laplacian with $n = 3375$. By iteration 143, we have approximated the first set of triple eigenvalues. By iteration 175 we get the second set.

Switching to restarted Arnoldi, we have similar results. Figure 7.6 shows how Arnoldi(35,15) with perturbation factor $\sigma = 10^{-6}$ brings about the triple eigenvalues. The second copy corresponding with the first triple eigenvalues appears at cycle 4, or at 95 iterations, and the third copy drops down at cycle 7, or at iteration 155. At

cycle 11, we have the first three sets of triple eigenvalues approximated.

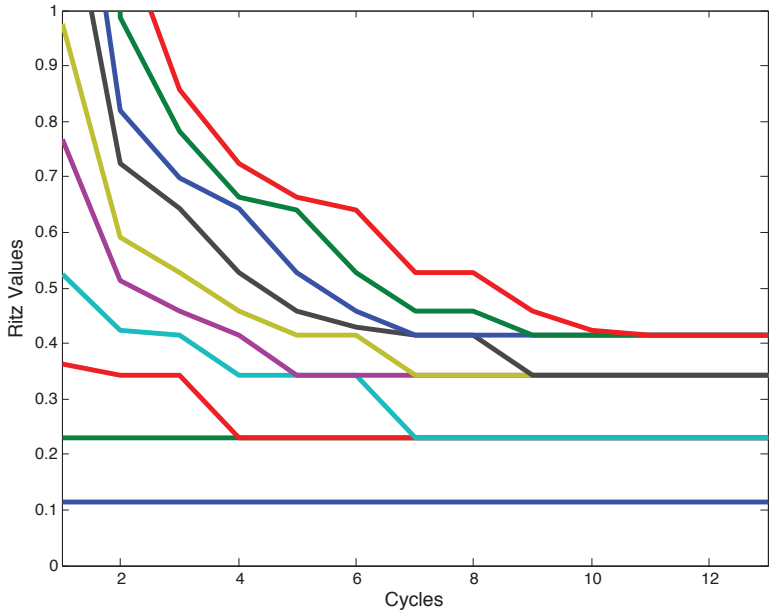

Figure 7.6: Restarted Arnoldi method, Arnoldi(35,15), with $\sigma = 10^{-6}$ and the 10 smallest Ritz values plotted against the cycle. The matrix is the 3-D Laplacian with $n = 3375$. By cycle 7, we have approximations to the first set of triple eigenvalues.

Figure 7.7 gives the residual of the 4th eigenvector, the third copy of the first triple eigenvalue, plotted against the cycle with varying perturbation factor σ. Figure 7.8 is the same plot, but for the interior eigenvector corresponding to the third copy of the third set of triple eigenvalues (10th eigenvector).

The method of perturbing via a random diagonal matrix is a simple heuristic approach that presents an alternative to waiting for round-off error, though it only one of many different possible strategies. An obvious concern is that too large of a perturbation factor will create results with too large of an error for certain applications. For that reason it is necessary to monitor the residual norm carefully as you perturb the matrix. This approach illustrates the effectiveness of separating multiple eigenvalues with a small matrix perturbation, as claimed in [19, 20, 21].

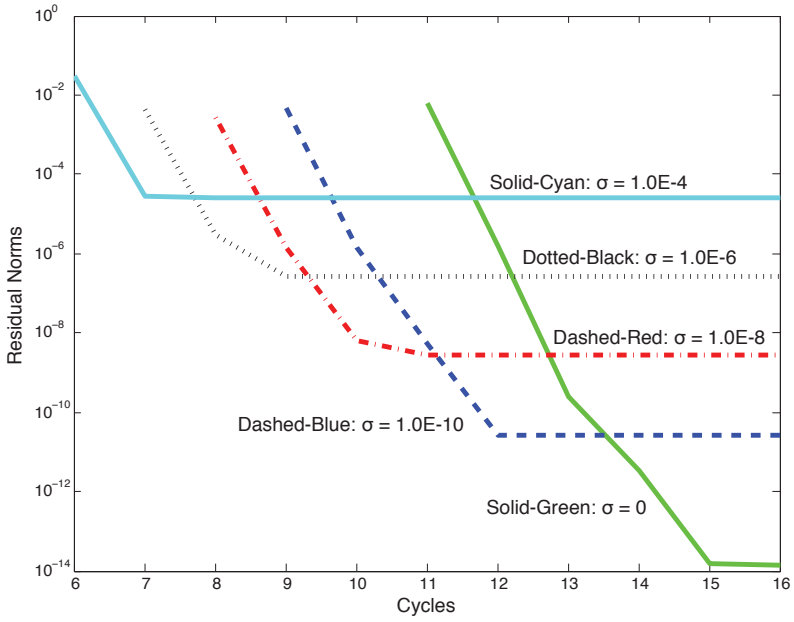

Figure 7.7: Restarted Arnoldi method, Arnoldi(35,15), with varying diagonal perturbations σ. The residual norms for the fourth eigenvector is shown. The matrix is the 3-D finite difference Laplacian with $n = 3375$.

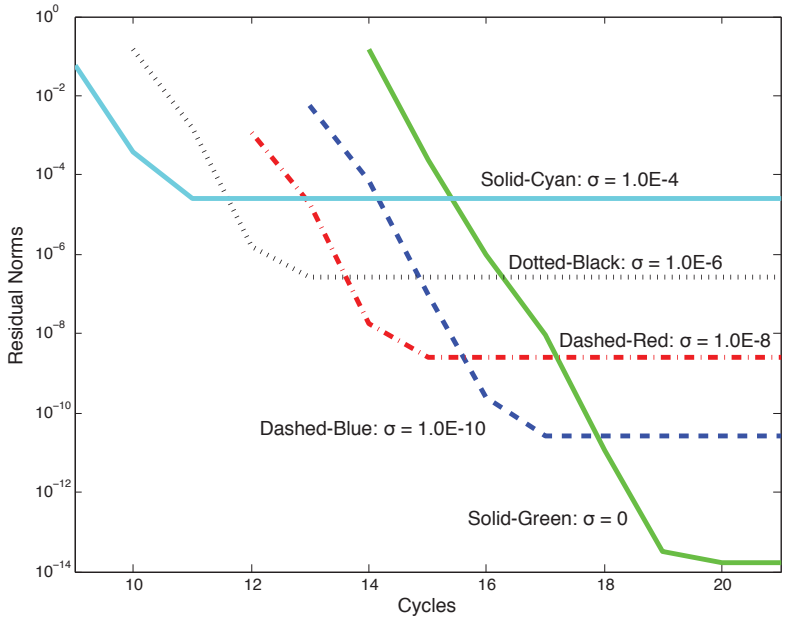

Figure 7.8: Restarted Arnoldi method, Arnoldi(35,15), with varying diagonal perturbations σ. The residual norms for the tenth eigenvector is shown. The matrix is the 3-D finite difference Laplacian with $n = 3375$.

CHAPTER EIGHT

Conclusion

We have presented, starting from establishing the Arnoldi method as presented in [9], a few variations of well established and known methods for computing eigenvalues of a large sparse matrix. More specifically, we examined the problem of computing multiple eigenvalues and finding their corresponding eigenvectors. During this investigation, we found that the role of round-off error can play a significant role in revealing multiple eigenvalues.

This topic has many possible directions for further study. From a theoretical point of view, perturbation theory provides us with useful insight and tools to help strengthen our mathematical claims. In particular, showing that we are able to guarantee separation of multiple eigenvalues by a small matrix perturbation. If the not, then showing that the case where multiple eigenvalues do not separate is a singular case would be tremendously beneficial as well.

Another point of further exploration is extrapolating on the idea of matrix perturbations; instead of a simple diagonal matrix perturbations, we may also experiment with rank-one or low-rank matrix perturbations. Currently, we are in the process of studying how these types of perturbations incorporated into an Arnoldi method perform for computing multiple eigenvalues.

BIBLIOGRAPHY

[1] Arioli, Gianni, and Gazzola, Filippo, *A new mathematical explanation of the Tacoma Narrows Bridge collapse*, Modellistica e Calcolo Scientifico, 2008.

[2] Peters, Tom F., *Transitions in Engineering: Guillaume Henri Dufour and the Early 19th Century Cable Suspension Bridges*, Birkhauser, 1987.

[3] Noble, Benjamin, and Daniel, James W., *Applied Linear Algebra*, Pearson Hall; 3rd Edition, 1987. Print.

[4] Meyer, Carl D., *Matrix Analysis and Applied Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 2001. Print.

[5] Golub N. Gene, and Van Loan, Charles E., *Matrix Computations*. Johns Hopkins University Press; 3rd Edition, 1996.

[6] Krylov, A. N., *On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined*. Izvestiya Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk 7:4. 1931.

[7] Cullen, Charles C., *An Introduction to Numerical Linear Algebra*. The Prindle, Weber & Schmidt Series in Calculus and Upper-Division Mathematics, 1993.

[8] Trefethen, Lloyd, N., and Bau III, David, *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 1997.

[9] Arnoldi, W. E., *The principle of minimized iterations in the solution of the matrix eigenvalue problem*. Quart. Appl. Math., 9:171729, 1951.

[10] Saad, Yousef, *Numerical Methods for Large Eigenvalue Problems*, Halsted Press, New York, NY, 1992. MR 93h:65052.

[11] Saad, Yousef, *Variations on Arnoldi method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl. 34:26917295, 1980.

[12] Sadkane, Miloud, *Block-Arnoldi and Davidson methods for unsymmetric large eigenvalue problems*, Numerische Mathematik, 1993.

[13] Morgan, Ronald B., *On restarting the Arnoldi method for large nonsymmetric eigenvalue problems*, Math. Comp., 65:1213-1230, 1996.

[14] Morgan, Ronald B., and Zeng, M. *A harmonic restarted Arnoldi for calculating eigenvalues and determining multiplicity*. Linear Algebra Appl., 415:9617113, 2006.

[15] Morgan, Ronald B., Restarted block-GMRES with deflation of eigenvalues. Appl. Numer. Math., 54:22217236, 2005.

[16] Ruhe, A., *Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices*. Mathematics of Computation, 33:68017687, 1979.

[17] Sorensen, D. C., *Implicit application of polynomial filters in a k-step Arnoldi method*. SIAM J. Matrix Anal. Appl., 13:357-385, 1992.

[18] Daniel, J. W., and Gragg, W. B., *Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization*. Mathematics of Computation, 30:77217795, 1976.

[19] Sun, J. G., *A Note on Local Behavior of Multiple Eigenvalues*. SIAM Journal on Matrix Analysis and Applications, 10:53317541, 1989.

[20] Stewart, G. W., and Zhang, G., *Eigenvalues of Graded Matrices and the Condition Numbers of a Multiple Eigenvalue*. 1991.

[21] Li, Chi-Kwong, and Li, Ren-Cang, *A note on eigenvalues of perturbed Hermitian matrices*. Linear Algebra and its Applications, 2005.