Spring 2012

Rapid Approximation of Bilinear Forms Involving Matrix Functions Through Asymptotic Analysis of Gaussian Node Placement

Elisabeth Marie Palchak
University of Southern Mississippi

The University of Southern Mississippi

RAPID APPROXIMATION OF BILINEAR FORMS INVOLVING MATRIX FUNCTIONS THROUGH ASYMPTOTIC ANALYSIS OF GAUSSIAN NODE PLACEMENT

by

Elisabeth Marie Palchak

A Thesis
Submitted to the Graduate School
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Master of Science

Approved:

[redacted]

Dean of the Graduate School

May 2012

ABSTRACT

RAPID APPROXIMATION OF BILINEAR FORMS INVOLVING MATRIX FUNCTIONS THROUGH ASYMPTOTIC ANALYSIS OF GAUSSIAN NODE PLACEMENT

by Elisabeth Marie Palchak

May 2012

Technological advancements have allowed computing power to generate high resolution models. As a result, greater stiffness has been introduced into systems of ordinary differential equations (ODEs) that arise from spatial discretization of partial differential equations (PDEs). The components of the solutions to these systems are coupled and changing at widely varying rates, which present problems for time-stepping methods. Krylov Subspace Spectral methods, developed by Dr. James Lambers, bridge the gap between explicit and implicit methods for stiff problems by computing each Fouier coefficient from an individualized approximation of the solution operator. KSS methods demonstrate a high order of accuracy, but their efficiency needs to be improved. We will carry out an asymptotic study to determine how these approximations behave at high frequencies to develop a formula to reduce the computation of each node while still achieving a high level of accuracy. Our numerical results will reveal that our method does prove to increase the efficiency as well as the accuracy of KSS methods.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. James Lambers, for introducing me to the field of Numerical Analysis. I am appreciative and grateful for your patience, guidance, and encouragement. I would also like to thank Dr. Joseph Kolibal and Dr. Haiyan Tian for taking the time to serve on my committee and for offering their comments and suggestions. Last, but not least, I would like to thank my family and friends for being there every step of the way.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure

LIST OF TABLES

Table

NOTATION AND GLOSSARY

General Usage and Terminology

The notation used in this text represents fairly standard mathematical and computational usage. In many cases these fields tend to use different preferred notation to indicate the same concept, and these have been reconciled to the extent possible, given the interdisciplinary nature of the material. In particular, the notation for partial derivatives varies extensively, and the notation used is chosen for stylistic convenience based on the application. While it would be convenient to utilize a standard nomenclature for this important symbol, the many alternatives currently in the published literature will continue to be utilized.

The blackboard fonts are used to denote standard sets of numbers: $\mathbb{R}$ for the field of real numbers, $\mathbb{C}$ for the complex field. The capital letters, $A, B, \cdots$ are used to denote matrices. Functions which are denoted in boldface type typically represent vector valued functions, and real valued functions usually are set in lower case roman or greek letters. Lower case letters such as i, j, k, l, m, n and sometimes p and d are used to denote indices.

Vectors are typset in square brackets, e.g., $[\cdot]$, and matrices are typeset in parenthesese, e.g., $(\cdot)$. In general the norms are typeset using double pairs of lines, e.g., $||\cdot||$, and the abolute value of numbers is denoted using a single pairs of lines, e.g., $|\cdot|$. Single pairs of lines around matrices indicates the determinant of the matrix. Denote $\langle\cdot,\cdot\rangle$ as the standard inner product on $[0, 2\pi]$. Define $\left[\begin{array}{cc} f(x) & g(x) \end{array}\right]$ as the limit of a block on n rows and 2 columns of vectors of values of $f(x)$ and $g(x)$ as $n \to \infty$.

Chapter 1

INTRODUCTION

In recent years, technology has advanced dramatically. These advancements in technology have allowed computing power to generate high resolution models, but with new advancements comes new challenges. The components of the solution of the ordinary differential equations (ODEs) that arise from spatial discretization of partial differential equations (PDEs) are coupled and changing at widely varying rates. Therein lies the challenge. This coupling has increased the stiffness [2] of the ODEs which presents problems for time-stepping methods.

We first discuss the Hockbruck-Lubich method [16, 17] which is a representative approach to using a polynomial approximation to compute $e^{-At}\mathbf{v}$. Since solutions to stiff systems have components that are changing at widely varying rates, this method is not efficient because the time step needs to be chosen very small to compute each component to achieve an acceptable order of accuracy. To handle these components differently, we introduce the next approach which applies Krylov Subspace Spectral (KSS) methods, developed by Dr. James Lambers [20]. KSS methods compute each Fourier coefficient from an individualized approximation of the solution operator. As a result, these KSS methods demonstrate a high order of accuracy, but are lacking in efficiency.

To improve these KSS methods, we will carry out an asymptotic study to determine how these approximations behave at high frequencies. In our anaylsis, we want to recognize any patterns to develop a formula to approximate the largest node to assign as a sharp lower bound to alleviate the expense of computing each node at a high level of accuracy.

In Chapter 2 we discuss the background material. In Section 1 of Chapter 2 we discuss the Hockbruck-Lubich method for solving the problem. In Section 2, we describe the background material in detail as it relates to matrices, quadrature, and partial differential equations. In Section 3, we discuss the block approach, and in Section 4 we introduce block Krylov Subspace Spectral methods. In Chapter 3, we discuss the solution to the problem through the asymptotic block Lanczos iteration. Section 1 of Chapter 3 consider the case when we have a constant leading coefficient, and Section 2 considers the case when we have a variable leading coefficient. Chapter 4 presents our numerical results. In Section 1 of Chapter 4, we examine a one-dimensional parabolic partial differential equation. In Section

2, we examine a two-dimensional example of a parabolic partial differential equation. In Chapter 5, we state our conclusions based on these results.

Chapter 2
BACKGROUND

2.1 Polynomial Approximation of the Solution Operator

For simplicity, we consider the one-dimensional case which can be generalized to higher dimensions [13, 15]. Consider the parabolic PDE

$$u_t + Lu = 0, \quad t > 0, \tag{2.1}$$

with $\exp[-Lt]$ as the solution operator on $(0, 2\pi)$, where L is a second order, self-adjoint, positive definite differential operator. The first approach to approximate the solution, developed by Hochbruck and Lubich [16, 17], follows. Consider

$$\mathbf{u}'(t) + A\mathbf{u} = 0, \quad \mathbf{u}(t_0) = \mathbf{u}_0, \tag{2.2}$$

where A is an $N \times N$ matrix, $\mathbf{u}(t)$ and $\mathbf{u}_0$ are N-vectors. This system is a result of the spatial discretization of the previous PDE (2.1). We find the exact solution of the ODE to be $\mathbf{u}(t) = e^{-At}\mathbf{u}_0$, which results in an approximate solution of the PDE.

We use a polynomial of A produced by the Lanczos iteration as an approximation of exponential. For example, consider $\mathbf{w} = e^{-At}\mathbf{v}$ for a given symmetric matrix A and vector $\mathbf{v}$. If we apply the Lanczos iteration to the matrix A with initial vector $\mathbf{v}$, j times, where $j \ll N$, at the end of the jth iteration, the Lanczos iteration produces an $N \times j$ orthogonal matrix X_j and a $j \times j$ tridiagonal matrix T_j such that $X_j^T A X_j = T_j$. Then we can compute the approximation

$$\mathbf{w}_j = \|\mathbf{v}\|_2 X_j e^{-T_j t}\mathbf{e}_1, \tag{2.3}$$

where each column $\mathbf{x}_k$, where $k = 1, \ldots, j$, of X_j is $\mathbf{x}_k = p_{k-1}(A)\mathbf{v}$. The polynomial $p_n(A)$ is of degree n in A, the polynomials p_j are orthonormal, and $\mathbf{w}_j$ is a product of a polynomial in A of degree $j-1$ and $\mathbf{v}$ [21, 17].

It is not possible to approximate exponential functions on a large interval with a polynomial without using a large number of terms. Similarly, it is not possible to approximate a matrix exponential with widely varying eigenvalues without many Lanczos iterations, which is the case if A arises from a stiff system. So if the eigenvalues are not clustered, this approach is not practical. A new approach is needed that approximates the high frequency components and low frequency components separately. Specifically, we need to apply a different polynomial approximation of the exponential for each component.

2.2 Matrices

A componentwise technique is used in the next approach. We execute this componentwise approach by using an approximation of the solution operator that is ideal for each Fourier coefficient of the solution. Given the solution $\tilde{u}(x,t_n)$ at time t_n, the solution at time t_{n+1} is

$$\hat{u}(\omega,t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, \exp[-L\Delta t]\tilde{u}(x,t_n) \right\rangle. \tag{2.4}$$

The result from discretizing the (2.4) is

$$\mathbf{u}^T f(A)\mathbf{v}, \tag{2.5}$$

where $\mathbf{u} = \frac{1}{\sqrt{2\pi}} e^{i\omega x}$ and $\mathbf{v} = \tilde{u}(x,t_n)$ are N-vectors, $A = L_N$ is an $N \times N$ symmetric positive definite matrix that comes from discretizing the operator L, and $f(\lambda) = \exp(-\lambda\Delta t)$. Methods for computing the bilinear form in (2.4) are discussed by Golub and Meurant in [10]. As given by properties of symmetric positive definite matrices, the matrix A has real eigenvalues

$$b = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N = a > 0, \tag{2.6}$$

and corresponding orthonormal eigenvectors $\mathbf{q}_j$, where $j = 1,\ldots,N$. Thus we can rewrite $\mathbf{u}^T f(A)\mathbf{v}$ in terms of its spectral decomposition,

$$\mathbf{u}^T f(A)\mathbf{v} = \sum_{j=1}^{N} f(\lambda_j)\mathbf{u}^T \mathbf{q}_j \mathbf{q}_j^T \mathbf{v}. \tag{2.7}$$

The matrix function (2.7) can also be regarded as the Riemann-Stieltjes integral

$$\mathbf{u}^T f(A)\mathbf{v} = I[f] = \int_a^b f(\lambda)\, d\alpha(\lambda) \tag{2.8}$$

where

$$\alpha(\lambda) = \begin{cases} 0, & \text{if } \lambda < a \\ \sum_{j=i}^{N} \alpha_j \beta_j, & \text{if } \lambda_i \leq \lambda < \lambda_{i-1} \\ \sum_{j=1}^{N} \alpha_j \beta_j, & \text{if } b \leq \lambda \end{cases}, \quad \alpha_j = \mathbf{u}^T \mathbf{q}_j, \quad \beta_j = \mathbf{q}_j^T \mathbf{v}. \tag{2.9}$$

We can approximate the integral $I[f]$ by Gaussian, Gauss-Radau, or Gauss-Lobatto Quadrature rules [3, 8, 9, 10], but the scope of my research only considers Gaussian. Approximating $I[f]$ yields

$$I[f] = \sum_{j=1}^{K} w_j f(\lambda_j) + R[f] \tag{2.10}$$

where we use the Lanczos algorithm to acquire the nodes λ_j, $j = 1,\ldots,K$ and weights w_j, $j = 1,\ldots,K$ [4, 6, 7, 12]. In the case where $\mathbf{u} = \mathbf{v}$, we apply the Lanczos algorithm [32]:

```
$\beta_0 = 0, \mathbf{p} = 0, \mathbf{u} = given, \mathbf{q} = \mathbf{u}/\|\mathbf{u}\|_2$

for $n = 1, 2, \ldots, K$
    $\mathbf{v} = A\mathbf{q}$
    $\alpha_n = \mathbf{q}^T\mathbf{v}$
    if $n < K$
        $\mathbf{v} = \mathbf{v} - \beta_{n-1}\mathbf{p} - \alpha_n\mathbf{q}$
        $\beta_n = \|\mathbf{v}\|_2$
        $\mathbf{p} = \mathbf{q}$
        $\mathbf{q} = \mathbf{v}/\beta_n$
    end
end
```

The α_js and β_js produced by the Lanczos algorithm form a tridiagonal matrix $\mathcal{T}_K$, where the diagonal elements are the αs and the super and sub diagonal elements are the βs. The resulting tridiagonal matrix produces the nodes and the weights [12] in our approximation. The nodes λ_j are the eigenvalues of

$$\mathcal{T}_K = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{K-2} & \alpha_{K-1} & \beta_{K-1} \\ & & & \beta_{K-1} & \alpha_K \end{bmatrix}, \tag{2.11}$$

and the weights are obtained by squaring the first components of the eigenvectors.

2.3 Block Gaussian Quadrature

Now we consider the case when $\mathbf{u} \neq \mathbf{v}$. As a consequence, the weights could be negative and destabilize the quadrature rule [1, 21]. Thus, we consider the block approach

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}^T f(A) \begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}. \tag{2.12}$$

The block matrix function (2.12) can be regarded as the Riemann-Stieltjes integral

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \begin{bmatrix} \mathbf{u}^T f(A)\mathbf{u} & \mathbf{u}^T f(A)\mathbf{v} \\ \mathbf{v}^T f(A)\mathbf{u} & \mathbf{v}^T f(A)\mathbf{v} \end{bmatrix} \tag{2.13}$$

where $\mu(\lambda)$ is a 2×2 matrix, each entry of which is a measure of the form $\alpha(\lambda)$. Using the most general quadrature formula with K matrix nodes [10, 21] we get

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \sum_{j=1}^{K} W_j f(T_j) W_j + error, \tag{2.14}$$

where T_j and W_j are 2×2 matrices. If we diagonalize each T_j, the resulting quadrature formula is given by

$$\int_a^b f(\lambda)\,d\mu(\lambda) = \sum_{j=1}^{2K} f(\lambda_j)\mathbf{v}_j\mathbf{v}_j^T + error, \tag{2.15}$$

where λ_j is a scalar and $\mathbf{v}_j$ is a 2-vector. The nodes and weights for our quadrature formula are obtained by applying the block Lanczos algorithm:

```
$X_0 = 0, R_0 = \left[\ \mathbf{u} \quad \mathbf{v}\ \right], R_0 = X_1 B_0$

for $n = 1, 2, \ldots,$K
    $V = AX_n$
    $M_n = X_n^T V$
    if$n < K$
        $R_n = V - X_{n-1}B_{n-1}^T - X_n M_n$
        $R_n = X_{n+1}B_n$
    end
end
```

The block Lanczos algorithm produces 2×2 matrices M_j and B_j, as stated in [11], that form a block tridiagonal matrix $\mathcal{T}_K$, where each B_j is upper triangular. The resulting block tridiagonal matrix produces the nodes and the weights. The nodes λ_j are the eigenvalues of

$$\mathcal{T}_K = \begin{bmatrix} M_1 & B_1^T & & & \\ B_1 & M_2 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-2} & M_{K-1} & B_{K-1}^T \\ & & & B_{K-1} & M_K \end{bmatrix}, \tag{2.16}$$

and the weights are the 2×2 matrices obtained by taking the outer product of the first two components of each eigenvector with itself.

2.4 Block KSS Methods

The general method for Block KSS [19, 24] begins by defining

$$R_0 = \left[\ \hat{\mathbf{e}}_\omega \quad \mathbf{u}(t_n)\ \right], \tag{2.17}$$

where $\hat{\mathbf{e}}_\omega$ is a discretization of $\frac{1}{\sqrt{2\pi}}e^{i\omega x}$ and

$$\mathbf{u}_\omega^n = \mathbf{u}^n - \hat{\mathbf{e}}_\omega \hat{\mathbf{e}}_\omega^H \mathbf{u}^n. \tag{2.18}$$

Next we compute the QR factorization of R_{j-1} for $j = 1$, such that

$$R_0 = X_1(\omega)B_0(\omega) \tag{2.19}$$

which outputs

$$X_1(\omega) = \left[\begin{array}{cc} \hat{\mathbf{e}}_\omega & \frac{\mathbf{u}_\omega^n}{\|\mathbf{u}_\omega^n\|_2} \end{array}\right] \tag{2.20}$$

and

$$B_0(\omega) = \left[\begin{array}{cc} 1 & \hat{\mathbf{e}}_\omega^H \mathbf{u}^n \\ 0 & \|\mathbf{u}_\omega^n\|_2 \end{array}\right]. \tag{2.21}$$

Then we apply the block Lanczos algorithm to the operator L_N, which comes from the discretization of L, with initial block $X_1(\omega)$. The result produces a block tridiagonal matrix $\mathcal{T}_K$ like (2.16), where every entry of $\mathcal{T}_K$ is a function of ω. Then, at time t_{n+1}, each Fourier coefficient of the solution is

$$[\hat{\mathbf{u}}^{n+1}]_\omega = [B_0^H E_{12}^H exp[\mathcal{T}_K(\omega)\Delta t]E_{12}B_0]_{12}, \quad E_{12} = \left[\begin{array}{cc} \mathbf{e}_1 & \mathbf{e}_2 \end{array}\right]. \tag{2.22}$$

For the parabolic problem we will examine in Chapter 4, we will show that block KSS methods have a high-order of accuracy $O(\Delta t^{2K-1})$ [20] for K block Gaussian nodes, whereas the Hochbruck and Lubich method only has $O(\Delta t^{K-1})$ [17].

Compared to the Hochbruck and Lubich method, it would seem that KSS methods have a high amount of computation expense. However, this is not actually the case. We compute an approximation of the solution operator for each Fourier coefficient, and all of the subspaces generated are closely related by the wave number ω [21]. Although KSS methods take less time computationally in comparison to the Hochbruck and Lubich method, we still do not achieve a desired level of efficiency. In Chapter 3, we will discuss how we address this issue.

Chapter 3

ASYMPTOTIC BLOCK LANCZOS ITERATION

3.1 Constant Leading Coefficient

Previously in our discussion of block KSS methods, we stated a Jacobi matrix is computed for each Fourier coefficient by using an approximation that is ideal for that component [16, 17, 18, 31, 33]. Then each component is approximated using a different polynomial of A. Thus, the computed solution including all the components is defined by

$$\mathbf{u}^{n+1} = \tilde{f}(L_N;\Delta t)\mathbf{u}^n = \sum_{j=0}^{2K} D_j(\Delta t)A^j\mathbf{u}^n, \tag{3.1}$$

where $\mathbf{u}^n$ is the solution from the previous time step and $D_j(\Delta t)$ is a diagonal matrix in the Fourier basis.

Block KSS methods are more flexible than time-stepping methods based on polynomial or rational approximations of $\exp[-At]$ and can be used to improve accuracy and stability for stiff problems. However, compared to other time-stepping methods with the same number of grid points, more computation time is required per time step to compute the nodes and weights. Given the high order of accuracy and stability of KSS methods [14, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30], we wish to improve their efficiency without sacrificing any accuracy. We now examine how we can take advantage of this flexibility.

First we will examine the case where we have a constant leading coefficient. We use (2.20) to form the initial block recursion coefficient

$$M_1(\omega) = \begin{bmatrix} \hat{\mathbf{e}}_\omega^H L_N \hat{\mathbf{e}}_\omega & \hat{\mathbf{e}}_\omega^H L_N \mathbf{u}_\omega^n \\ [\mathbf{u}_\omega^n]^H L_N \hat{\mathbf{e}}_\omega & [\mathbf{u}_\omega^n]^H L_N \mathbf{u}_\omega^n \end{bmatrix}. \tag{3.2}$$

Suppose L is a second-order differential operator where

$$Lu = -pu_{xx} + q(x)u \tag{3.3}$$

and p is constant. Then

$$M_1(\omega) = \begin{bmatrix} p\omega^2 + \bar{q} & \hat{\mathbf{e}}_\omega^H \tilde{\mathbf{q}} \mathbf{u}_\omega^n \\ [\mathbf{u}_\omega^n]^H \tilde{\mathbf{q}} \hat{\mathbf{e}}_\omega & [\mathbf{u}_\omega^n]^H L_N \mathbf{u}_\omega^n \end{bmatrix}, \tag{3.4}$$

where $\overline{q}$ is the mean of a function $q(x)$ defined on $[0,2\pi]$ and $\tilde{q}(x) = q(x) - \overline{q}$. Denote $\tilde{\mathbf{q}}$ by the vector with components $[\tilde{\mathbf{q}}]_j = \tilde{q}(x_j)$.

As $|\omega|$ increases, the Fourier coefficients of a function go to zero and $\mathbf{u}^n_\omega \to \mathbf{u}^n$. So the off-diagonal entries are virtually the Fourier coefficient of $\tilde{q}u(\cdot,t_n)$, corresponding to ω, which decay to 0 as $|\omega| \to \infty$. So,

$$M_1(\omega) \sim \begin{bmatrix} p\omega^2 + \bar{q} & 0 \\ 0 & \frac{[\mathbf{u}^n]^H L_N \mathbf{u}^n}{[\mathbf{u}^n]^H \mathbf{u}^n} \end{bmatrix}, \tag{3.5}$$

and the non-diagonal entries become negligible. Thus as ω increases, $B_1(\omega)$ becomes small so the overall matrix $\mathcal{T}_K(\omega)$ approximately decouples. Thus, for higher frequencies, the diagonal entries of $M_1(\omega)$ are approximately two eigenvalues of $\mathcal{T}_K(\omega)$.

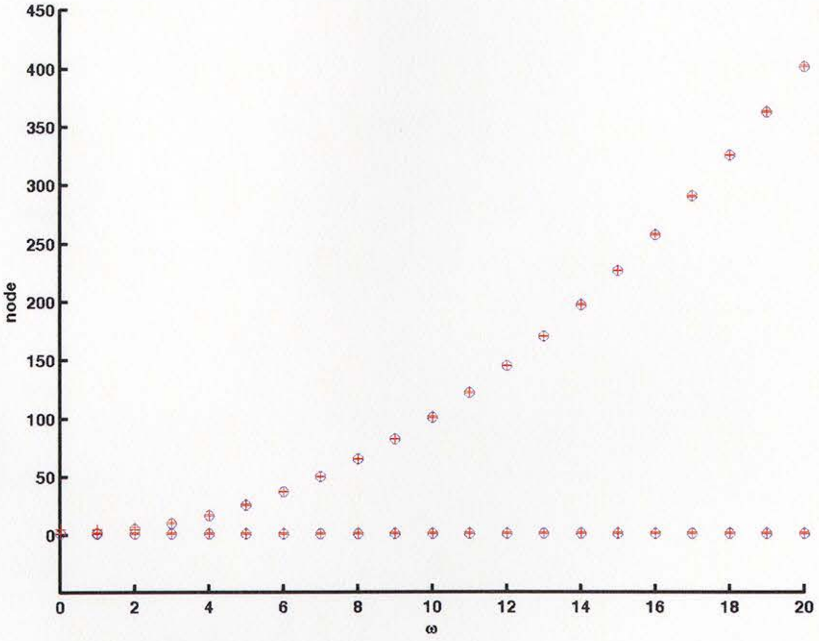

Figure 3.1: Quadrature Nodes and Block Gaussian Nodes

Figure 3.1 above shows the prescribed quadrature nodes by a 4-node KSS method with the blue circles. The red crosses show a 2-node block KSS method with 4 block Gaussian nodes. The previous approximation is a good approximation of the Gaussian nodes, but the work that goes into computing the actual Gaussian nodes is excessive. It has been observed in numerical experiments for the 2-node Block KSS method, that half of the four scalar nodes λ_j are clustered around the values of the diagonal entries of $M_1(\omega)$. So we prescribe the smallest and largest nodes

$$\lambda_1(\omega) = \frac{[\mathbf{u}^n]^H L_N \mathbf{u}^n}{[\mathbf{u}^n]^H \mathbf{u}^n}, \quad \lambda_{2K}(\omega) = \frac{\hat{\mathbf{e}}_\omega^H L_N \hat{\mathbf{e}}_\omega}{\hat{\mathbf{e}}_\omega^H \hat{\mathbf{e}}_\omega} \tag{3.6}$$

as seen in Figure 3.1. The rest of the nodes are equally spaced between $\lambda_1(\omega)$ and $\lambda_{2K}(\omega)$ [21].

3.2 Variable Leading Coefficient

The previous approach does not work when the operator contains a variable leading coefficient as demonstrated Figure 3.2. Different from the case in (3.3) where the operator contains a constant leading coefficient, we define the operator

$$Lu = -(p(x)u_x)_x + q(x)u. \tag{3.7}$$

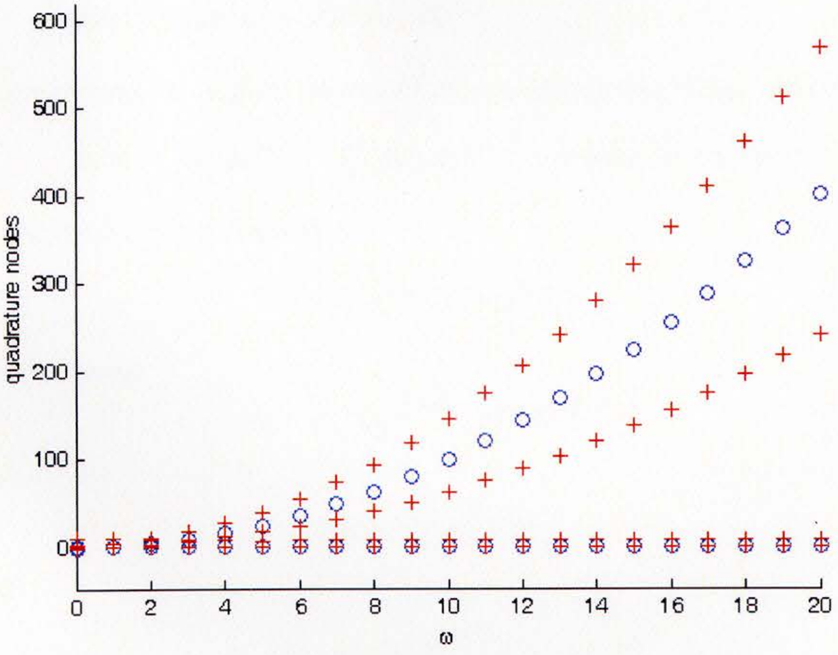

Figure 3.2: Variable Leading Coefficient Fails

We now propose to assign the largest node for each component to serve as a sharper lower bound, as opposed to the lower bound we could obtain from the $(1,1)$ entry of (3.5). To achieve our approximation of the largest node, we examine the columns of the R_j blocks generated by block Lanczos, only considering the highest powers of ω, to develop a formula that maximizes the Rayleigh quotient. From block Lanczos, we compute $M_2(\omega)$ in the following manner. We calculate

$$R_1 = LX_1 - X_1M_1 \tag{3.8}$$

Thus,

$$\begin{aligned} R_1 &= \Big[\ L(\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}) - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}(M_1)_{11} + u_\omega(x)(M_1)_{21}], \\ &\quad L(u_\omega(x)) - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}(M_1)_{12} + u_\omega(x)(M_1)_{22}]\ \Big] \\ &= \Big[\ L(\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}) - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}(\overline{p}\omega^2 + \overline{q}) + u_\omega(x)[\tfrac{\omega^2}{\sqrt{2\pi}}\hat{pu} - \tfrac{i\omega}{\sqrt{2\pi}}\hat{p'}u + \tfrac{1}{\sqrt{2\pi}}\hat{qu}]], \\ &\quad L(u_\omega(x)) - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}\overline{[\tfrac{\omega^2}{\sqrt{2\pi}}\hat{pu} - \tfrac{i\omega}{\sqrt{2\pi}}\hat{p'}u + \tfrac{1}{\sqrt{2\pi}}\hat{qu}]} + \\ &\quad u_\omega(x)(-puu'' - p'uu' + qu^2]\ \Big] \\ &= \Big[\ \tfrac{\omega^2}{\sqrt{2\pi}}pe^{i\omega x} - \tfrac{i\omega}{\sqrt{2\pi}}p'e^{i\omega x} + \tfrac{1}{\sqrt{2\pi}}qe^{i\omega x} - \tfrac{\omega^2}{\sqrt{2\pi}}\overline{p}e^{i\omega x} - \tfrac{1}{\sqrt{2\pi}}\overline{q}e^{i\omega x} - \tfrac{\omega^2}{\sqrt{2\pi}}\hat{pu}u + \\ &\quad \tfrac{\omega}{\sqrt{2\pi}}\hat{p'}uu - \tfrac{1}{\sqrt{2\pi}}\hat{qu}u, -pu'' - p'u' + qu - \tfrac{\omega^2}{\sqrt{2\pi}}\hat{pu}e^{i\omega x} + \tfrac{i\omega}{\sqrt{2\pi}}\hat{p'}ue^{i\omega x} - \\ &\quad \tfrac{1}{\sqrt{2\pi}}\hat{qu}e^{i\omega x} + pu^2u'' + p'u^2u' + qu^3\ \Big]. \end{aligned} \tag{3.9}$$

After only considering the highest powers of ω in each column, we determine

$$R_1 = \begin{bmatrix} \frac{\omega^2}{\sqrt{2\pi}}\tilde{p}e^{i\omega x} & \frac{-\omega^2}{2\pi}\hat{pu}(x)e^{i\omega x} \end{bmatrix} + \text{lower order terms.} \tag{3.10}$$

Neglecting the constants we conclude

$$R_1 = \begin{bmatrix} e^{i\omega x} & \tilde{p}(x)e^{i\omega x} \end{bmatrix}. \tag{3.11}$$

From R_1 we determine

$$X_2 = \begin{bmatrix} \frac{1}{\sqrt{2\pi}}e^{i\omega x} & \tilde{p}(x)e^{i\omega x} \end{bmatrix}, \tag{3.12}$$

which leads to

$$M_2(\omega) = X_2^H L X_2. \tag{3.13}$$

We calculate

$$\begin{aligned} M_2(\omega) &= \begin{bmatrix} \frac{1}{\sqrt{2\pi}}e^{i\omega x} & \tilde{p}(x)e^{i\omega x} \end{bmatrix}^H L \begin{bmatrix} \frac{1}{\sqrt{2\pi}}e^{i\omega x} & \tilde{p}(x)e^{i\omega x} \end{bmatrix} \\ &= \begin{bmatrix} \left\langle \frac{1}{\sqrt{2\pi}}e^{i\omega x}, L\frac{1}{\sqrt{2\pi}}e^{i\omega x} \right\rangle & \left\langle \frac{1}{\sqrt{2\pi}}e^{i\omega x}, L\tilde{p}(x)e^{i\omega x} \right\rangle \\ \left\langle \tilde{p}(x)e^{i\omega x}, L\frac{1}{\sqrt{2\pi}}e^{i\omega x} \right\rangle & \left\langle \tilde{p}(x)e^{i\omega x}, L\tilde{p}(x)e^{i\omega x} \right\rangle \end{bmatrix}. \end{aligned} \tag{3.14}$$

After computing the inner products of the entries we get

$$M_2(\omega) = \left[\begin{array}{cc} \dfrac{\overline{p}\omega^2 + \overline{q}}{\frac{\omega^2}{\sqrt{2\pi}}\|\tilde{p}\|_2 + \frac{1}{\sqrt{2\pi}}\frac{\langle \tilde{q}, \tilde{p}\rangle}{\|\tilde{p}\|_2}} & \begin{array}{c} \frac{\omega^2}{\sqrt{2\pi}}\|\tilde{p}\|_2 + \frac{1}{\sqrt{2\pi}}\frac{\langle \tilde{q}, \tilde{p}\rangle}{\|\tilde{p}\|_2} \\ \omega^2\frac{\|\tilde{p}\|_p^2}{\|\tilde{p}\|^2} + \frac{\|\tilde{p'}\|_p^2}{\|\tilde{p}\|^2} + \frac{\|\tilde{p}\|_q^2}{\|\tilde{p}\|^2} \end{array} \end{array} \right]. \tag{3.15}$$

Using $M_2(\omega)$, we compute

$$R_2 = LX_2 - X_2M_2 - X_1{B_1}^T. \tag{3.16}$$

Since we determined ${B_1}^T \sim 0$ is negligible, we only need to compute

$$R_2 = LX_2 - X_2M_2 \tag{3.17}$$

where

$$X_2 = \left[\begin{array}{cc} \frac{1}{\sqrt{2\pi}}e^{i\omega x} & \tilde{p}(x)e^{i\omega x} \end{array}\right]. \tag{3.18}$$

Thus,

$$\begin{aligned}
R_2 \;=\; & \Big[\; L(\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}) - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}(M_2)_{11} + \tilde{p}(x)e^{i\omega x}(M_2)_{21}], \\
& L(\tilde{p}(x)e^{i\omega x}) - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}(M_2)_{12} + \tilde{p}(x)e^{i\omega x}(M_2)_{22}] \;\Big] \\
=\; & \Big[\; L(\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}) - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}(\overline{p}\omega^2 + \overline{q}) + \tilde{p}(x)e^{i\omega x}\overline{(\tfrac{\omega^2}{\sqrt{2\pi}}\|\tilde{p}\|_2 + \tfrac{1}{\sqrt{2\pi}}\tfrac{\langle\tilde{q},\tilde{p}\rangle}{\|\tilde{p}\|_2})}], \\
& L(\tilde{p}(x)e^{i\omega x}) - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}(\tfrac{\omega^2}{\sqrt{2\pi}}\|\tilde{p}\|_2 + \tfrac{1}{\sqrt{2\pi}}\tfrac{\langle\tilde{q},\tilde{p}\rangle}{\|\tilde{p}\|_2}) + \\
& \tilde{p}(x)e^{i\omega x}(\omega^2\tfrac{\|\tilde{p}\|_p^2}{\|\tilde{p}\|^2} + \tfrac{\|\tilde{p'}\|_p^2}{\|\tilde{p}\|^2} + \tfrac{\|\tilde{p}\|_q^2}{\|\tilde{p}\|^2})] \;\Big] \\
=\; & \Big[\; \tfrac{\omega^2}{\sqrt{2\pi}}pe^{i\omega x} - \tfrac{i\omega}{\sqrt{2\pi}}p'e^{i\omega x} + \tfrac{1}{\sqrt{2\pi}}qe^{i\omega x} - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}(\overline{p}\omega^2 + \overline{q}) \\
& +\tilde{p}(x)e^{i\omega x}\overline{(\tfrac{\omega^2}{\sqrt{2\pi}}\|\tilde{p}\|_2 + \tfrac{1}{\sqrt{2\pi}}\tfrac{\langle\tilde{q},\tilde{p}\rangle}{\|\tilde{p}\|_2})}], p_0\omega^2\tilde{p}e^{i\omega x} - p_0 i\omega\tilde{p}'e^{i\omega x} + \\
& \omega^2\tilde{p}^2e^{i\omega x} - 2i\omega\tilde{p}\tilde{p}'e^{i\omega x} - p_0 i\omega\tilde{p}'e^{i\omega x} - p_0\tilde{p}''e^{i\omega x} - \tilde{p}'^2e^{i\omega x} - \tilde{p}\tilde{p}''e^{i\omega x} \\
& -i\omega\tilde{p}\tilde{p}'e^{i\omega x} + q\tilde{p}e^{i\omega x} - [\tfrac{1}{\sqrt{2\pi}}e^{i\omega x}(\tfrac{\omega^2}{\sqrt{2\pi}}\|\tilde{p}\|_2 + \tfrac{1}{\sqrt{2\pi}}\tfrac{\langle\tilde{q},\tilde{p}\rangle}{\|\tilde{p}\|_2}) + \\
& \tilde{p}(x)e^{i\omega x}(\omega^2\tfrac{\|\tilde{p}\|_p^2}{\|\tilde{p}\|^2} + \tfrac{\|\tilde{p'}\|_p^2}{\|\tilde{p}\|^2} + \tfrac{\|\tilde{p}\|_q^2}{\|\tilde{p}\|^2})] \;\Big].
\end{aligned} \tag{3.19}$$

After only considering the highest powers of ω in each column, we determine

$$R_2 = \left[\begin{array}{cc} \tilde{p}\omega^2\frac{1}{\sqrt{2\pi}}e^{i\omega x} + \tilde{p}(x)e^{i\omega x}\frac{\omega^2}{\sqrt{2\pi}}\|\tilde{p}\|_2 & \tilde{p}^2\omega^2e^{i\omega x} - \frac{\omega^2}{2\pi}e^{i\omega x}\|\tilde{p}\|_2 \end{array}\right]. \tag{3.20}$$

Neglecting the constants, we conclude

$$R_2 = \left[\begin{array}{cc} \tilde{p}(x)e^{i\omega x} & \tilde{p}(x)^2e^{i\omega x} - e^{i\omega x} \end{array}\right]. \tag{3.21}$$

Using R_2, we compute X_3

$$X_3 = \left[\begin{array}{cc} \tilde{p}(x)e^{i\omega x} & \tilde{p}(x)^2e^{i\omega x} - e^{i\omega x} \end{array}\right] \tag{3.22}$$

which yields

$$M_3(\omega) = X_3^H LX_3. \tag{3.23}$$

So

$$M_3(\omega) = \left[\begin{array}{cc} \tilde{p}(x)e^{i\omega x} & \tilde{p}(x)^2 e^{i\omega x} - e^{i\omega x} \end{array}\right]^H L \left[\begin{array}{cc} \tilde{p}(x)e^{i\omega x} & \tilde{p}(x)^2 e^{i\omega x} - e^{i\omega x} \end{array}\right]. \tag{3.24}$$

We note that the columns of the X_j blocks are linear combinations of

$$e^{i\omega x}, \tilde{p}e^{i\omega x}, \tilde{p}^2 e^{i\omega x}, \ldots, \tilde{p}^{j-1}e^{i\omega x} \tag{3.25}$$

$+$ lower order terms of ω. We recall our goal is to estimate the largest eigenvalue of T_k such that

$$\lambda_1(T_{k,\omega}) = \lambda_1(Q_{k,\omega}^H L Q_{k,\omega}) \tag{3.26}$$

where

$$Q_{k,\omega} = \left[\begin{array}{c} X_1 X_2 \ldots X_k \end{array}\right]. \tag{3.27}$$

The largest eigenvalue of a symmetric matrix maximizes the Raleigh quotient, where

$$\max_{\mathbf{v} \in \text{range}(Q_{k.\omega})} \frac{\mathbf{v}^H L \mathbf{v}}{\mathbf{v}^H \mathbf{v}}. \tag{3.28}$$

By computing R_1 and R_2, we observed a nice pattern that will allow us to consider a much smaller space. Since we only consider the high frequency components, the useful part of the range$(Q_{k,\omega})$ is well approximated by the span$\{e^{i\omega x}, \tilde{p}e^{i\omega x}, \tilde{p}^2 e^{i\omega x}, \ldots, \tilde{p}^{k-1}e^{i\omega x}\}$. As $|\omega| \to \infty$, the largest eigenvalue is well approximated by

$$\max_{\mathbf{v} \in \text{range}(P_{k.\omega})} \frac{\mathbf{v}^H L \mathbf{v}}{\mathbf{v}^H \mathbf{v}} \tag{3.29}$$

where

$$P_{k,\omega} = \left[\begin{array}{cccc} e^{i\omega x} & \tilde{p}e^{i\omega x} & \cdots & \tilde{p}^{k-1}e^{i\omega x} \end{array}\right] \tag{3.30}$$

All the vectors $\mathbf{v}$ are a linear combination of

$$\mathbf{v} = c_1 \mathbf{p}_1 + c_2 \mathbf{p}_2 + \cdots + c_k \mathbf{p}_k \tag{3.31}$$

so

$$\mathbf{v} = P_{k,\omega}\mathbf{c}. \tag{3.32}$$

Since $\mathbf{v}$ is a linear combination of the columns of $P_{k,\omega}$, we determine

$$\max_{\mathbf{v} \in \text{range}(Q_{k.\omega})} \frac{\mathbf{v}^H L \mathbf{v}}{\mathbf{v}^H \mathbf{v}} \approx \max_{\mathbf{c} \in \mathbb{C}} \frac{\mathbf{c}^H (P_{k,\omega}^H L P_{k,\omega})\mathbf{c}}{\mathbf{c}^H (P_{k,\omega}^H P_{k,\omega})\mathbf{c}} = \max_{\mathbf{c} \in \mathbb{C}} \frac{\mathbf{c}^H B \mathbf{c}}{\mathbf{c}^H F \mathbf{c}} = \lambda_1(B,F) \tag{3.33}$$

where λ_1 is the largest generalized eigenvalue [5] of the matrix pair (B,F). For

$$\mathbf{u}_k = \tilde{p}^{k-1}e^{i\omega x} \tag{3.34}$$

and

$$\mathbf{u}_j = \tilde{p}^{j-1} e^{i\omega x} \tag{3.35}$$

we determine formulas for B and F. We compute F in the following manner.

$$\begin{aligned}
F_{kj} &= \langle \mathbf{u}_k, \mathbf{u}_j \rangle \\
&= \langle \tilde{p}^{k-1} e^{i\omega x}, \tilde{p}^{j-1} e^{i\omega x} \rangle \\
&= \int_0^{2\pi} (\overline{\tilde{p}^{k-1} e^{i\omega x}})(\tilde{p}^{j-1} e^{i\omega x}) dx \\
&= \int_0^{2\pi} (\tilde{p}^{k-1})(\tilde{p}^{j-1}) dx \\
&= \int_0^{2\pi} (\tilde{p}^{k+j-2}) dx
\end{aligned} \tag{3.36}$$

We compute our formula for B similarly

$$\begin{aligned}
B_{kj} &= \langle \mathbf{u}_k, L\mathbf{u}_j \rangle \\
&= \langle \tilde{p}^{k-1} e^{i\omega x}, L(\tilde{p}^{j-1} e^{i\omega x}) \rangle \\
&= \int_0^{2\pi} (\overline{\tilde{p}^{k-1} e^{i\omega x}})(L(\tilde{p}^{j-1} e^{i\omega x})) dx \\
&= \int_0^{2\pi} (\overline{\tilde{p}^{k-1} e^{i\omega x}})(p_0 \tilde{p}^{j-1} \omega^2 e^{i\omega x} + \tilde{p}^j \omega^2 e^{i\omega x} - 2(j-1) p_0 \tilde{p}^{j-2} (\tilde{p})' i\omega e^{i\omega x} \\
&\quad -2(j-1)\tilde{p}^{j-1}(\tilde{p})' i\omega e^{i\omega x} - (j-1)(j-2) p_0 \tilde{p}^{j-3}((\tilde{p})')^2 e^{i\omega x} - \\
&\quad (j-1)(j-2)\tilde{p}^{j-2}((\tilde{p})')^2 e^{i\omega x} - (j-1) p_0 \tilde{p}^{j-2} (\tilde{p})'' e^{i\omega x} - \\
&\quad (j-1)\tilde{p}^{j-1}(\tilde{p})'' e^{i\omega x} - p' \tilde{p}^{j-1} i\omega e^{i\omega x} - (j-1) p' \tilde{p}^{j-2} (\tilde{p})' e^{i\omega x} + \\
&\quad q(\tilde{p}^{j-1} e^{i\omega x})) dx \\
&= \int_0^{2\pi} (p_0 \omega^2 \tilde{p}^{k+j-2} + \omega^2 \tilde{p}^{k+j-1} - (j-1)(j-2) p_0 \tilde{p}^{k+j-4} \tilde{p}'^2 - \\
&\quad (j-1)(j-2)\tilde{p}^{k+j-3} \tilde{p}'^2 - (j-1) p_0 \tilde{p}^{k+j-3} \tilde{p}'' - (j-1)\tilde{p}^{k+j-2} \tilde{p}'' \\
&\quad -(j-1)\tilde{p}^{k+j-3} \tilde{p}'^2 + \tilde{p}^{k+j-2} q) dx
\end{aligned} \tag{3.37}$$

As shown in Figure 3.3, this method is very accurate for the 4-node case in approximating the external nodes when p is variable. To summarize, in the case where p is variable, we prescribe the largest node accoring to the aforementioned method. When p is constant, we follow the given procedure in (3.5) and (3.6) to use the quadrature rule. Our numerical results are shown in Chapter 4.

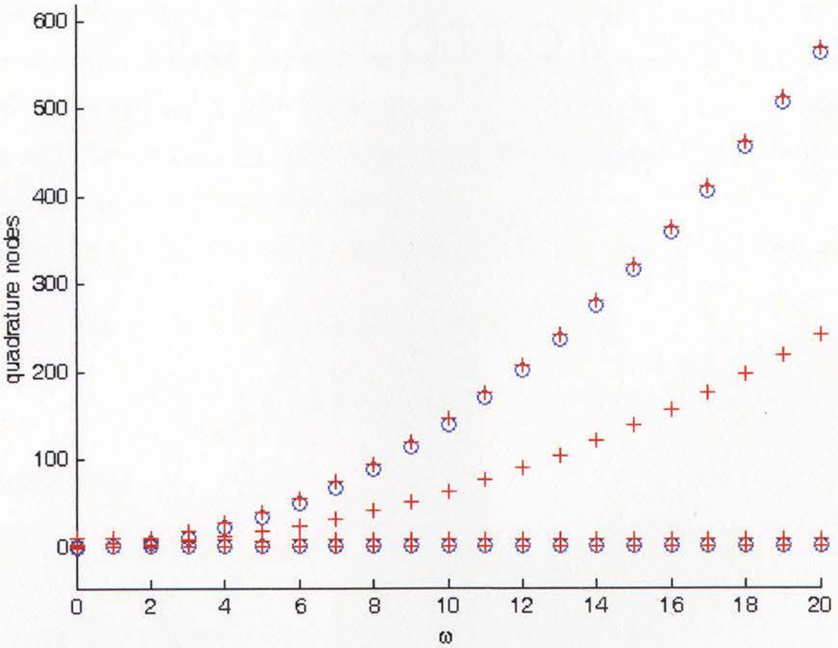

Figure 3.3: Well-Approximated Nodes for Variable Coefficient Case

Chapter 4

NUMERICAL RESULTS

4.1 One-Dimensional Parabolic PDE

In this chapter we solve examples of the one-dimensional and two-dimensional heat equation using the Hochbruck-Lubich method of approximating e^{-LT}, an accelerated KSS method with nodes prescribed in the same way as when p is constant, Dr. Lambers' original KSS method, and our new method. All experiments are conducted on 128 and 256 point grids with 4- or 6- dimensional Krylov subspaces. The exact solutions are unknown, so we compute the error by comparing each solution to the solution at the smallest time step and taking the 2-norm of the relative difference.

To compare our results to previous methods, we solve (2.1), (3.7) using the coefficients

$$\begin{aligned} p(x) &= 1+\frac{1}{2}\cos x-\frac{1}{4}\sin 2x+\frac{1}{8}\cos 3x \\ q(x) &= 1+\frac{1}{4}\sin x-\frac{1}{4}\cos 2x+\frac{1}{8}\sin 3x-\frac{1}{8}\cos 4x \end{aligned} \tag{4.1}$$

with initial conditions

$$u(x,0)=1+\frac{3}{10}\cos x-\frac{3}{20}\sin 2x+\frac{3}{40}\cos 3x, \quad 0<x<2\pi. \tag{4.2}$$

First we compare our methods using 4-node KSS methods. The results are given in Figure 4.1, Figure 4.2, and Table 4.1. Figure 4.1 gives the relative error for the solution of (2.1) using four different methods. The black curve demonstrates our improved KSS method. The blue curve demonstrates Dr. Lambers' original KSS method. The red curve is an accelerated KSS method with nodes prescribed in the manner as when p is constant. The green curve demonstrates the Hockbruck-Lubich method of approximating e^{-Lt}. All methods should be 3rd-order accurate, but only the our new KSS method achieves this level of accuracy. Similarly, Figure 4.2 shows our 4-node comparison on a 256 point grid. Table 4.1 gives estimates of the relative error of the four methods we discussed.

Then we compare our methods using 6-node KSS methods. The results are given in Figure 4.3, Figure 4.4, and Table 4.2. Figure 4.3 gives the relative error for the solution of (2.1) using the four different methods we discussed. The black curve demonstrates our improved KSS method. The blue curve demonstrates Dr. Lambers' original KSS method.

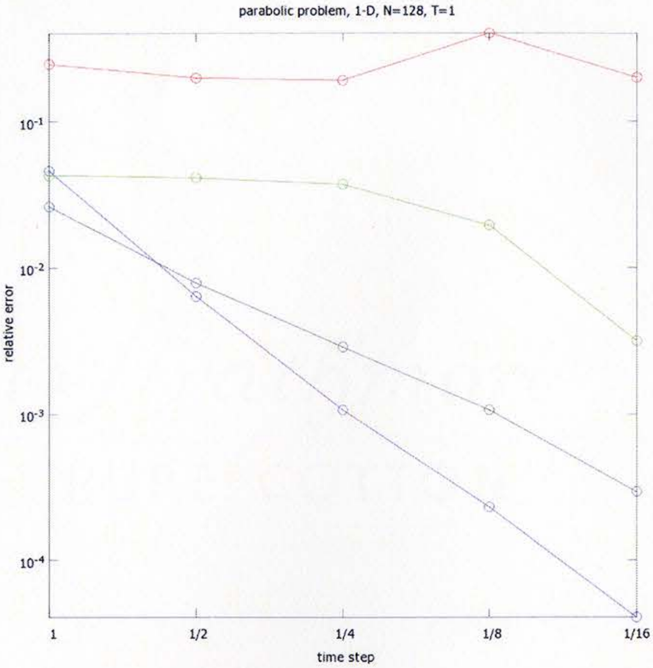

Figure 4.1: 4-Node Comparison on 128 Point Grid

N	Δt	Improved KSS	KSS-p constant	Original KSS	Hockbruck-Lubich
128	1	4.560e-002	2.440e-001	2.605e-002	4.272e-002
	1/2	6.375e-003	1.964e-001	7.848e-003	4.077e-002
	1/4	1.073e-003	1.891e-001	2.877e-003	3.694e-002
	1/8	2.321e-004	3.967e-001	1.069e-003	1.947e-002
	1/16	4.083e-005	1.983e-001	2.935e-004	3.157e-003
256	1	4.560e-002	2.440e-001	2.604e-002	4.272e-002
	1/2	6.375e-003	1.964e-001	7.839e-003	4.077e-002
	1/4	1.073e-003	1.891e-001	2.868e-003	3.694e-002
	1/8	2.321e-004	3.967e-001	1.134e-003	1.947e-002
	1/16	4.083e-005	1.983e-001	3.089e-004	3.157e-003

Table 4.1: Relative Error Estimates of 4-node Comparisons

The red curve is an accelerated KSS method with nodes prescribed in the manner as when p is constant. The green curve demonstrates the Hockbruck-Lubich method of approximating e^{-Lt}. All methods should be 5th-order accurate, but only the our new KSS method achieves this level of accuracy. Similarly, Figure 4.4 shows our 6-node comparison on a 256 point grid. Table 4.2 gives estimates of the relative error of the four methods we discussed. Again we find that all of these methods should be 5th-order accurate, but only our new method is.

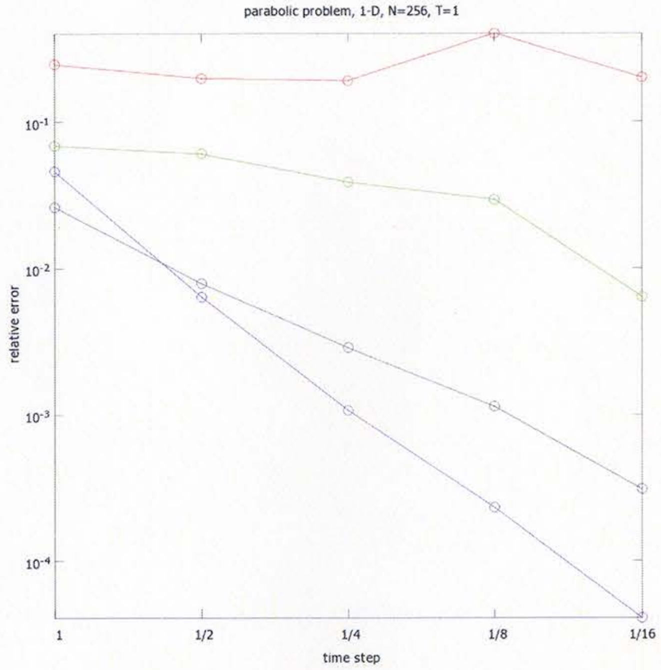

Figure 4.2: 4-Node Comparison on 256 Point Grid

N	Δt	Improved KSS	KSS-p constant	Original KSS	Hockbruck-Lubich
128	1	1.090e-001	6.085e+002	1.534e-002	4.665e-002
	1/2	2.323e-002	1.002e+008	2.070e-003	1.755e-002
	1/4	9.684e-004	1.351e+005	1.150e-003	6.993e-003
	1/8	4.323e-005	1.280e+003	4.171e-004	1.441e-002
	1/16	4.416e-006	3.848e-002	8.049e-005	2.407e-002
256	1	1.090e-001	1.407e+002	1.533e-002	4.665e-002
	1/2	2.323e-002	4.898e+007	2.065e-003	1.755e-002
	1/4	9.684e-004	6.936e+002	1.506e-003	6.993e-003
	1/8	4.323e-005	7.051e+004	4.763e-004	1.441e-002
	1/16	4.416e-006	1.244e-001	8.375e-005	2.407e-002

Table 4.2: Relative Error Estimates of 6-node Comparisons

4.2 Two-Dimensional Parabolic PDE

We now consider high dimensions of the heat equation. To test our methods, we use the two-dimensional coefficients

$$\begin{aligned} p(x,y) &= 1+\frac{1}{2}\cos(x+y)-\frac{1}{4}\sin(2(x-y))+\frac{1}{8}\cos(3(x+y)) \\ q(x,y) &= 1+\frac{1}{4}\sin x-\frac{1}{4}\cos 2y+\frac{1}{8}\sin 3x-\frac{1}{8}\cos 4y \end{aligned} \tag{4.3}$$

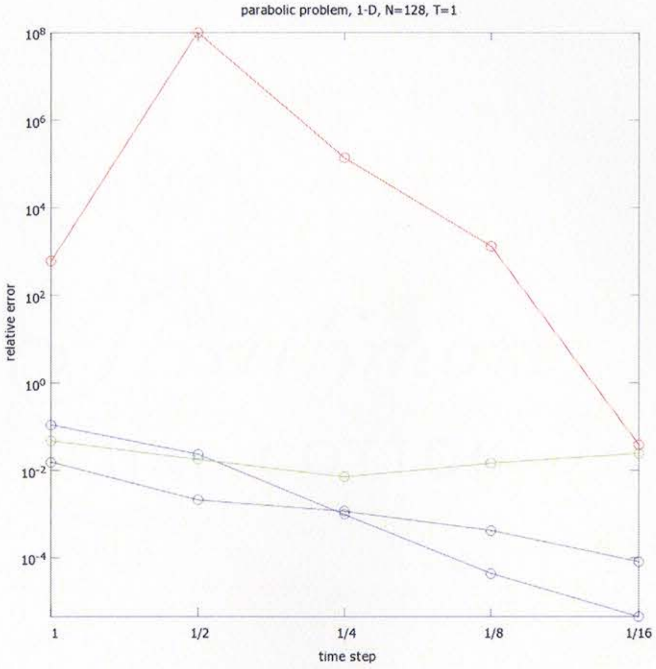

Figure 4.3: 6-Node Comparison on 128 Point Grid

with initial conditions

$$u(x,y,0) = 1 + \frac{3}{10}\cos x - \frac{3}{20}\sin(2(x+y)) + \frac{3}{40}\cos 3x, \quad 0 < x,y < 2\pi. \tag{4.4}$$

The results are given in Figure 4.5, Figure 4.6, and Table 4.3. Figure 4.5 gives the relative error for the solution of (2.1) using the four different methods we discussed. The black curve demonstrates our improved KSS method. The blue curve demonstrates Dr. Lambers' original KSS method. The red curve is an accelerated KSS method with nodes prescribed in the manner as when p is constant. The green curve demonstrates the Hockbruck-Lubich method of approximating e^{-Lt}. All methods should be 3rd-order accurate, but only the our new KSS method achieves this level of accuracy. Similarly, Figure 4.6 shows our 4-node comparison on 32 points per dimension. Table 4.3 gives estimates of the relative error of the four methods we discussed, with 4 nodes, on a grid with 16 points per dimension.

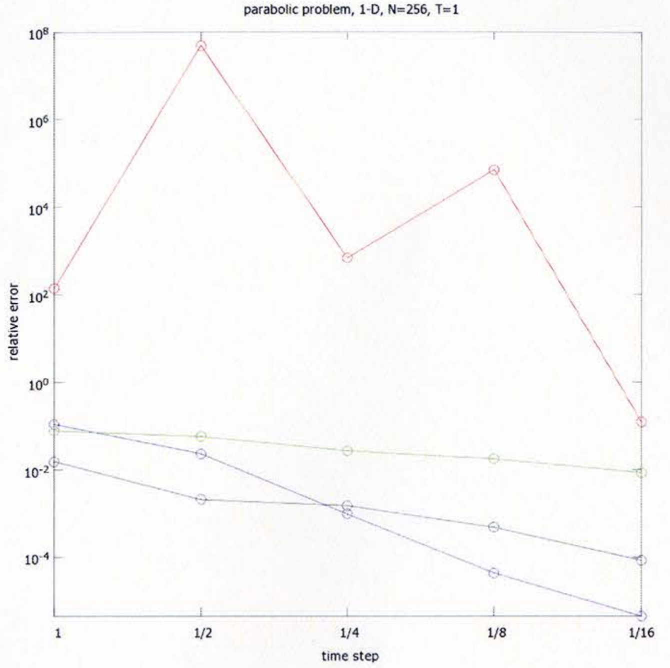

Figure 4.4: 6-Node Comparison on 256 Point Grid

N	Δt	Improved KSS	KSS-p constant	Original KSS	Hockbruck-Lubich
16	1	6.723e-002	5.510e-001	2.644e-002	1.351e-001
	1/2	1.443e-002	3.301e-001	8.910e-003	9.480e-002
	1/4	3.191e-003	6.699e-001	2.956e-003	4.589e-002
	1/8	4.519e-004	4.469e-001	7.558e-004	6.060e-003
	1/16	1.101e-004	1.133e-002	1.360e-004	1.270e-003
32	1	6.724e-002	6.013e-001	2.646e-002	1.351e-001
	1/2	1.320e-002	4.232e-001	8.593e-003	9.480e-002
	1/4	1.511e-003	7.755e-001	3.273e-003	4.589e-002
	1/8	3.186e-004	5.897e-001	1.090e-003	6.060e-003
	1/16	1.189e-004	2.141e-001	2.662e-004	1.270e-003

Table 4.3: Relative Error Estimates of 2D Comparisons

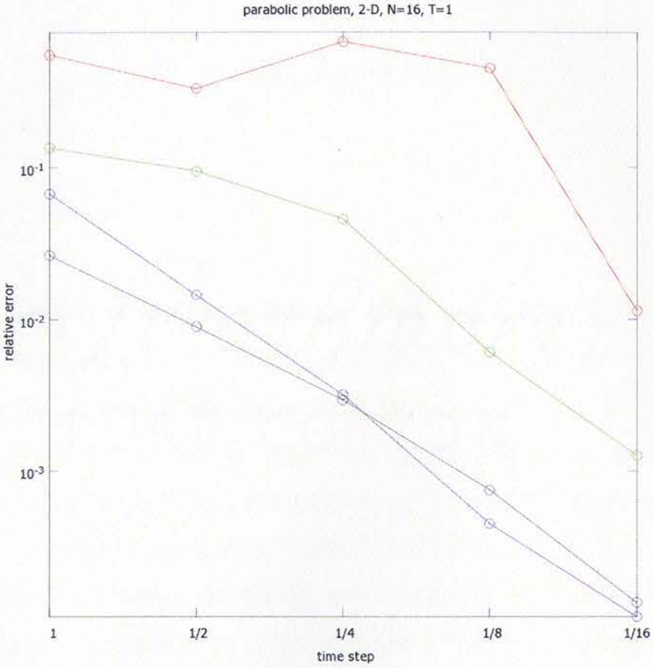

Figure 4.5: 2D, 4-Node Comparison on 16 Points per Dimension

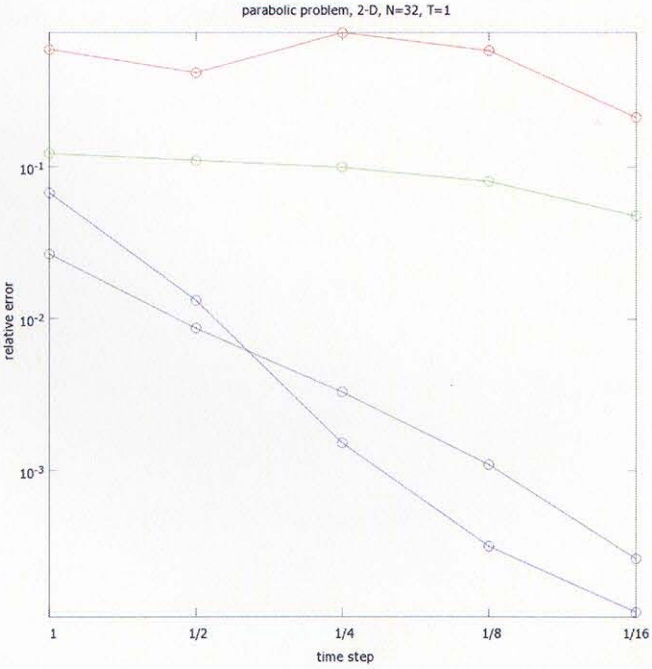

Figure 4.6: 2D, 4-Node Comparison on 32 Points per Dimension

Chapter 5

CONCLUSION

We examined the problem of stiffness that has been introduced into systems of ordinary differential equations that arise from the spatial discretization of partial differential equations. We discussed the Hockbruck-Lubich method of approximating e^{-Lt}, and determined that it takes too many Lanczos iterations to achieve a suitable level of accuracy. We discussed Krylov Subspace Spectral methods to address this stiffness issue. Although KSS methods are more scalable than the Hockbruck-Lubich method, they are too computationally expensive per time step. From our asymptotic study, we were led to develop a formula that uses the largest generalized eigenvalue to approximate the largest quadrature node for each component. This method has allowed us improve the efficiency of the existing KSS methods without sacrificing any accuracy. In this thesis, we solved a one-dimensional and two-dimensional paraboic PDE to analyze our results. We found that in addition to not sacrificing any accuracy, we actually improved the accuracy of the existing KSS methods. Now we have set the foundation to develop an efficient method that is more practical for stiff problems.

BIBLIOGRAPHY

[1] Atkinson, K.: *An Introduction to Numerical Analysis, 2nd Ed.* Wiley (1989)

[2] Burden, R., Faires, J.: *Numerical Analysis, 8th Ed.* Thompson Brooks/Cole (2005)

[3] Dahlquist, G., Eisenstat, S. C., Golub, G. H.: Bounds for the Error of Linear Systems of Equations Using the Theory of Moments. *J. Math. Anal. Appl.* **37** (1972) 151-166.

[4] Davis, P., Rabinowitz, P.: *Methods of Numerical Integration*, 2nd Ed. Academic Press (1984)

[5] Demmel, J.:*Applied Numerical Linear Algebra* Siam (1997)

[6] Gautschi, W.: Construction of Gauss-Christoffel Quadrature Formulas. *Math. Comp.* **22** (1986) 251-270.

[7] Gautschi, W.: Orthogonal Polynomials–Constructive Theory and Applications. *J. of Comp. and Appl. Math.* **12/13** (1985) 61-76.

[8] Golub, G. H.: Some Modified Matrix Eigenvalue Problems. *SIAM Review* **15** (1973) 318-334.

[9] Golub, G. H.: Bounds for Matrix Moments. *Rocky Mnt. J. of Math.* **4** (1974) 207-211.

[10] Golub, G. H., Meurant, G.: Matrices, Moments and Quadrature. *Proceedings of the 15th Dundee Conference*, June-July 1993, Griffiths, D. F., Watson, G. A. (eds.), Longman Scientific & Technical (1994)

[11] Golub, G. H., Underwood, R.: The Block Lanczos Method for Computing Eigenvalues. *Mathematical Software III*, J. Rice Ed., (1977) 361-377.

[12] Golub, G. H., Welsch, J.: Calculation of Gauss Quadrature Rules. *Math. Comp.* **23** (1969) 221-230.

[13] Guidotti, P., Lambers, J. V., Wong, E.: A Nonlinear Nonlocal Diffusion Model for Color Image Denoising. In preparation.

[14] Guidotti, P., Lambers, J. V., Sølna, K.: Analysis of 1-D Wave Propagation in Inhomogeneous Media. *Numer. Funct. Anal. Opt.* **27** (2006) 25-55.

[15] Guidotti, P., Longo, K.: Two Enhanced Fourth Order Diffusion Models for Image Denoising. *Journal of Mathematical Imaging and Vision* **40** (2011) 188-198.

[16] Hochbruck, M., Lubich, C.: A Gautschi-type Method for Oscillatory Second-order Differential Equations. *Numer. Math.* **83** (1999) 403-426.

[17] Hochbruck, M., Lubich, C.: On Krylov Subspace Approximations to the Matrix Exponential Operator. *SIAM J. Numer. Anal.* **34** (1996) 1911-1925.

[18] Hochbruck, M., Lubich, C., Selhofer, H.: Exponential Integrators for Large Systems of Differential Equations. *SIAM J. Sci. Comput.* **19** (1998) 1552-1574.

[19] Lambers, J. V.: Derivation of High-Order Spectral Methods for Time-dependent PDE using Modified Moments. *Electron. T. Numer. Ana.* **28** (2008) 114-135.

[20] Lambers, J. V.: Enhancement of Krylov Subspace Spectral Methods by Block Lanczos Iteration. *Electron. T. Numer. Ana.* **31** (2008) 86-109.

[21] Lambers, J. V.: Explicit High-Order Time-Stepping Based on Componentwise Application of Asymptotic Block Lanczos Iteration. *Numerical Linear Algebra with Applications* (2012) to appear.

[22] Lambers, J. V.: An Explicit, Stable, High-Order Spectral Method for the Wave Equation Based on Block Gaussian Quadrature. *IAENG Journal of Applied Mathematics* **38** (2008) 333-348. Available at `http://www.iaeng.org/IJAM/issues_v38/issue_4/IJAM_38_4_10.pdf`.

[23] Lambers, J. V.: Implicitly Defined High-Order Operator Splittings for Parabolic and Hyperbolic Variable-Coefficient PDE Using Modified Moments. *International Journal of Computational Science* **2** (2008) 376-401.

[24] Lambers, J. V.: Krylov Subspace Methods for Variable-Coefficient Initial-Boundary Value Problems. Ph.D. Thesis, Stanford University (2003).
Available at `http://www.math.usm.edu/lambers/pub/thesis.pdf`.

[25] Lambers, J. V.: Krylov Subspace Spectral Methods for the Time-Dependent Schrödinger Equation with Non-Smooth Potentials. *Numer. Algorithms* **51** (2009) 239-280.

[26] Lambers, J. V.: Krylov Subspace Spectral Methods for Variable-Coefficient Initial-Boundary Value Problems. *Electron. T. Numer. Ana.* **20** (2005) 212-234.

[27] Lambers, J. V.: A Multigrid Block Krylov Subspace Spectral Method for Variable-Coefficient Elliptic PDE. *IAENG Journal of Applied Mathematics* **39** (2009) 236-246.
Available at `http://www.iaeng.org/IJAM/issues_v39/issue_4/IJAM_39_4_07.pdf`.

[28] Lambers, J. V.: Practical Implementation of Krylov Subspace Spectral Methods. *J. Sci. Comput.* **32** (2007) 449-476.

[29] Lambers, J. V.: Spectral Methods for Time-dependent Variable-coefficient PDE Based on Block Gaussian Quadrature. *Proceedings of the 2009 International Conference on Spectral and High-Order Methods* (2010) in press.
Available at `http://www.math.usm.edu/lambers/papers/icosahom_lambers_final.pdf`.

[30] Lambers, J. V.: A Spectral Time-Domain Method for Computational Electrodynamics. *Adv. Appl. Math. Mech.* **1**(6) (2009) 781-798.
Available at `http://www.global-sci.org/aamm/volumes/v1n6/pdf/16-781.pdf`.

[31] Moret, I., Novati, P.: RD-rational Approximation of the Matrix Exponential Operator. *BIT* **44** (2004) 595-615.

[32] Trefethen, L., Bau III, D.:*Numerical Analysis* Siam (1997)

[33] van den Eshof, J., Hochbruck, M.: Preconditioning Lanczos Approximations to the Matrix Exponential. *SIAM J. of Sci. Comput.* **27** (2006) 1438-1457.