An Explicit, Stable, High-Order Spectral Method for the Wave Equation Based on Block Gaussian Quadrature

James V. Lambers *

Abstract—**This paper presents a modification of Krylov Subspace Spectral (KSS) Methods, which build on the work of Golub, Meurant and others pertaining to moments and Gaussian quadrature to produce high-order accurate approximate solutions to the variable-coefficient second-order wave equation. Whereas KSS methods currently use Lanczos iteration to compute the needed quadrature rules, the modification uses block Lanczos iteration in order to avoid the need to compute two quadrature rules for each component of the solution, or use perturbations of quadrature rules that tend to be sensitive in problems with oscillatory coefficients or data. It will be shown that under reasonable assumptions on the coefficients of the problem, a 1-node KSS method is second-order accurate and unconditionally stable, and methods with more than one node are shown to possess favorable stability properties as well, in addition to very high-order temporal accuracy. Numerical results demonstrate that block KSS methods are significantly more accurate than their non-block counterparts, especially for problems that feature oscillatory coefficients.**

Keywords: ***spectral methods, Gaussian quadrature, variable-coefficient, block Lanczos method, stability, wave equation***

1 Introduction

Consider the following initial-boundary value problem in one space dimension,

$$u_{tt} + Lu = 0 \quad \text{on } (0, 2\pi) \times (0, \infty), \tag{1}$$

$$u(x, 0) = f(x), \quad u_t(x, 0) = g(x), \quad 0 < x < 2\pi, \tag{2}$$

with periodic boundary conditions

$$u(0, t) = u(2\pi, t), \quad t > 0. \tag{3}$$

The operator L is a second-order differential operator of the form

$$Lu = -(p(x)u_x)_x + q(x)u, \tag{4}$$

where $p(x)$ is a positive function and $q(x)$ is a nonnegative (but nonzero) smooth function. It follows that L is self-adjoint and positive definite.

In [18, 20] a class of methods, called Krylov subspace spectral (KSS) methods, was introduced for the purpose of solving variable-coefficient parabolic problems. These methods are based on the application of techniques developed by Golub and Meurant in [7], originally for the purpose of computing elements of the inverse of a matrix, to elements of the matrix exponential of an operator. It has been shown in these references that KSS methods, by employing different approximations of the solution operator for each Fourier component of the solution, achieve higher-order accuracy in time than other Krylov subspace methods (see, for example, [14]) for stiff systems of ODE, and, as shown in [15], they are also quite stable, considering that they are explicit methods.

In [16], we considered whether these methods can be enhanced, in terms of accuracy, stability or any other measure, by using a single block Gaussian quadrature rule to compute each Fourier component of the solution, instead of two standard Gaussian rules. KSS methods take the solution from the previous time step into account only through a perturbation of initial vectors used in Lanczos iteration. While this enables KSS methods to handle stiff systems very effectively by giving individual attention to each Fourier component, and also yields high-order operator splittings (see [17]), it is worthwhile to consider whether it is best to use quadrature rules whose nodes are determined primarily by each basis function used to represent the solution, instead of the solution itself. Intuitively, a block quadrature rule that uses a basis function and the solution should strike a better balance between the competing goals of computing each component with an approximation that is, in some sense, optimal for that component in order to deal with stiffness, and giving the solution a prominent role in computing the quadrature rules that are used to evolve it forward in time.

In this paper, we apply this block approach to the second-order wave equation (1), (2), (3). KSS methods have previously been applied to this problem in [12, 15, 17], and have exhibited even higher-order accuracy than for

*Submitted October 12, 2008. Stanford University, Department Energy Resources Engineering, Stanford CA USA 94305-2220 Tel/Fax: 650-725-2729/2099 Email: lambers@stanford.edu

parabolic problems. It is our hope that the superiority of block KSS methods for parabolic problems extends to hyperbolic problems. Section 2 reviews the main properties of KSS methods, including algorithmic details and results concerning local accuracy. They use perturbations of quadratic forms to compute Fourier components of the solution, where the perturbation is in the direction of the solution from the previous time step. Section 3 discusses their application to the wave equation, and reviews previous convergence analysis. In Section 4, we present the modified KSS method that uses block Lanczos iteration to approximate each Fourier component of the solution by a single Gaussian quadrature rule. In Section 5, we study the convergence behavior of the block method. Numerical results are presented in Section 6. In Section 7, various extensions and future directions are discussed.

2 Krylov Subspace Spectral Methods

We begin with a review of the main aspects of KSS methods, which are easier to describe for parabolic problems. Let $S(t) = \exp[-Lt]$ represent the exact solution operator of the problem

$$u_t + Lu = 0, \quad 0 < x < 2\pi, \quad t > 0, \tag{5}$$

$$u(x,0) = f(x), \quad 0 < x2, \pi, \tag{6}$$

$$u(0,t) = u(2\pi,t), \quad t > 0, \tag{7}$$

and let $\langle \cdot, \cdot \rangle$ denote the standard inner product of functions defined on $[0, 2\pi]$,

$$\langle f(x), g(x) \rangle = \int_0^{2\pi} \overline{f(x)} g(x)\, dx. \tag{8}$$

Krylov subspace spectral methods, introduced in [18, 20], use Gaussian quadrature on the spectral domain to compute the Fourier components of the solution. These methods are time-stepping algorithms that compute the solution at time $t_1, t_2, \ldots$, where $t_n = n\Delta t$ for some choice of Δt. Given the computed solution $\tilde{u}(x, t_n)$ at time t_n, the solution at time t_{n+1} is computed by approximating the Fourier components that would be obtained by applying the exact solution operator to $\tilde{u}(x, t_n)$,

$$\hat{u}(\omega, t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, S(\Delta t)\tilde{u}(x, t_n) \right\rangle. \tag{9}$$

Krylov subspace spectral methods approximate these components with higher-order temporal accuracy than traditional spectral methods and time-stepping schemes. We briefly review how these methods work.

We discretize functions defined on $[0, 2\pi]$ on an N-point uniform grid with spacing $\Delta x = 2\pi/N$. With this discretization, the operator L and the solution operator $S(\Delta t)$ can be approximated by $N \times N$ matrices that represent linear operators on the space of grid functions, and the quantity (9) can be approximated by a bilinear form

$$\hat{u}(\omega, t_{n+1}) \approx \sqrt{\Delta x} \hat{\mathbf{e}}_\omega^H S_N(\Delta t)\mathbf{u}^n, \tag{10}$$

where

$$[\hat{\mathbf{e}}_\omega]_j = \frac{1}{\sqrt{2\pi}} e^{i\omega j \Delta x}, \quad [\mathbf{u}^n]_j = u(j\Delta x, t_n), \tag{11}$$

and

$$S_N(t) = \exp[-L_N t], \tag{12}$$

$$[L_N]_{jk} = -p(j\Delta x)[D_N^2]_{jk} + p'(j\Delta x)[D_N]_{jk} + q(j\Delta x), \tag{13}$$

where D_N is a discretization of the differentiation operator that is defined on the space of grid functions. Our goal is to approximate (10) by computing an approximation to

$$[\hat{\mathbf{u}}^{n+1}]_\omega = \hat{\mathbf{e}}_\omega^H \mathbf{u}^{n+1} = \hat{\mathbf{e}}_\omega^H S_N(\Delta t)\mathbf{u}^n. \tag{14}$$

In [7] Golub and Meurant describe a method for computing quantities of the form

$$\mathbf{u}^T f(A)\mathbf{v}, \tag{15}$$

where $\mathbf{u}$ and $\mathbf{v}$ are N-vectors, A is an $N \times N$ symmetric positive definite matrix, and f is a smooth function. Our goal is to apply this method with $A = L_N$ where L_N was defined in (12), $f(\lambda) = \exp(-\lambda t)$ for some t, and the vectors $\mathbf{u}$ and $\mathbf{v}$ are derived from $\hat{\mathbf{e}}_\omega$ and $\mathbf{u}^n$.

The basic idea is as follows: since the matrix A is symmetric positive definite, it has real eigenvalues

$$b = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N = a > 0, \tag{16}$$

and corresponding orthogonal eigenvectors $\mathbf{q}_j$, $j = 1, \ldots, N$. Therefore, the quantity (15) can be rewritten as

$$\mathbf{u}^T f(A)\mathbf{v} = \sum_{j=1}^N f(\lambda_j)\mathbf{u}^T \mathbf{q}_j \mathbf{q}_j^T \mathbf{v}. \tag{17}$$

We let $a = \lambda_N$ be the smallest eigenvalue, $b = \lambda_1$ be the largest eigenvalue, and define the measure $\alpha(\lambda)$ by

$$\alpha(\lambda) = \begin{cases} 0, & \text{if } \lambda < a \\ \sum_{j=i}^N \alpha_j \beta_j, & \text{if } \lambda_i \leq \lambda < \lambda_{i-1} \\ \sum_{j=1}^N \alpha_j \beta_j, & \text{if } b \leq \lambda \end{cases}, \tag{18}$$

where $\alpha_j = \mathbf{u}^T \mathbf{q}_j$ and $\beta_j = \mathbf{q}_j^T \mathbf{v}$. If this measure is positive and increasing, then the quantity (15) can be viewed as a Riemann-Stieltjes integral

$$\mathbf{u}^T f(A)\mathbf{v} = I[f] = \int_a^b f(\lambda)\, d\alpha(\lambda). \tag{19}$$

As discussed in [4, 5, 6, 7], the integral $I[f]$ can be bounded using either Gauss, Gauss-Radau, or Gauss-Lobatto quadrature rules, all of which yield an approximation of the form

$$I[f] = \sum_{j=1}^K w_j f(t_j) + R[f], \tag{20}$$

where the nodes t_j, $j = 1, \ldots, K$, as well as the weights w_j, $j = 1, \ldots, K$, can be obtained using the symmetric Lanczos algorithm if $\mathbf{u} = \mathbf{v}$, and the unsymmetric Lanczos algorithm if $\mathbf{u} \neq \mathbf{v}$ (see [11]).

In the case $\mathbf{u} \neq \mathbf{v}$, there is the possibility that the weights may not be positive, which destabilizes the quadrature rule (see [1] for details). Therefore, it is best to handle this case by rewriting (15) using decompositions such as

$$\mathbf{u}^T f(A)\mathbf{v} = \frac{1}{\delta}[\mathbf{u}^T f(A)(\mathbf{u} + \delta\mathbf{v}) - \mathbf{u}^T f(A)\mathbf{u}], \qquad (21)$$

where δ is a small constant. Guidelines for choosing an appropriate value for δ can be found in [20, Section 2.2].

Employing these quadrature rules yields the following basic process (for details see [18, 20]) for computing the Fourier coefficients of $\mathbf{u}^{n+1}$ from $\mathbf{u}^n$. It is assumed that when the Lanczos algorithm (symmetric or unsymmetric) is employed, K iterations are performed to obtain the K quadrature nodes and weights.

for $\omega = -N/2 + 1, \ldots, N/2$
  Choose a scaling constant δ_ω
  Compute $u_1 \approx \hat{\mathbf{e}}_\omega^H S_N(\Delta t)\hat{\mathbf{e}}_\omega$
    using the symmetric Lanczos algorithm
  Compute $u_2 \approx \hat{\mathbf{e}}_\omega^H S_N(\Delta t)(\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n)$
    using the unsymmetric Lanczos algorithm
  $[\hat{\mathbf{u}}^{n+1}]_\omega = (u_2 - u_1)/\delta_\omega$
end

It should be noted that the constant δ_ω plays the role of δ in the decomposition (21), and the subscript ω is used to indicate that a different value may be used for each wave number $\omega = -N/2+1, \ldots, N/2$. Also, in the presentation of this algorithm in [20], a polar decomposition is used instead of (21), and is applied to sines and cosines instead of complex exponential functions.

This algorithm has high-order temporal accuracy, as indicated by the following theorem. Let $BL_N([0, 2\pi]) = \text{span}\{\ e^{-i\omega x}\ \}_{\omega=-N/2+1}^{N/2}$ denote a space of bandlimited functions with at most N nonzero Fourier components.

Theorem 1 *Let L be a self-adjoint m-th order positive definite differential operator on $C_p([0, 2\pi])$ with coefficients in $BL_N([0, 2\pi])$, and let $f \in BL_N([0, 2\pi])$. Then the preceding algorithm, applied to the problem (1), (2), (3), is consistent; i.e.*

$$[\hat{\mathbf{u}}^1]_\omega - \hat{u}(\omega, \Delta t) = O(\Delta t^{2K}),$$

for $\omega = -N/2 + 1, \ldots, N/2$.

Proof. See [20, Lemma 2.1, Theorem 2.4]. □

The preceding result can be compared to the accuracy achieved by an algorithm described by Hochbruck and Lubich in [14] for computing $e^{-A\Delta t}\mathbf{v}$ for a given symmetric positive definite matrix A and vector $\mathbf{v}$ using the symmetric Lanczos algorithm. As discussed in [14], this algorithm can be used to compute the solution of some ODEs without time-stepping, but this becomes less practical for ODEs arising from a semi-discretization of problems such as (5), (6), (7), due to their stiffness. In this situation, it is necessary to either use a high-dimensional Krylov subspace, in which case reorthogonalization is required, or one can resort to time-stepping, in which case the local temporal error is only $O(\Delta t^K)$, assuming a K-dimensional Krylov subspace. Regardless of which remedy is used, the computational effort needed to compute the solution at a fixed time T increases substantially.

The difference between Krylov subspace spectral methods and the approach described in [14] is that in the former, a different K-dimensional Krylov subspace is used for each Fourier component, instead of the same subspace for all components as in the latter. As can be seen from numerical results comparing the two approaches in [20], using the same subspace for all components causes a loss of accuracy as the number of grid points increases, whereas Krylov subspace spectral methods do not suffer from this phenomenon.

Using a perturbation of the form (21) is only one approach for computing bilinear forms such as (15) in the case where $\mathbf{u} \neq \mathbf{v}$. In [15], this approach was numerically stabilized by the use of formulas for the derivatives of the nodes and weights with respect to the parameter δ. However, two quadrature rules are needed to compute each component, as well as the unsymmetric Lanczos algorithm, which is much less well-behaved than its symmetric counterpart. A polar decomposition may be used, but that also requires two quadrature rules, although the symmetric Lanczos algorithm can be used for both. Even so, as shown in [19], the performance of this approach can be sensitive to the size of the perturbation, especially when the solution is oscillatory.

An approach that requires only one quadrature rule per component, and gives the solution a greater role in the computation of these rules than merely a perturbation, involves block Lanczos iteration. The result is a block-tridiagonal, Hermitian matrix from which the nodes and weights for the quadrature rule can be obtained. It is worthwhile to examine whether a block approach might be more effective than the original algorithm for hyperbolic problems as well as the parabolic problems for which this method was successfully applied in [16].

3 Application to the Wave Equation

In this section we review the application of Krylov subspace spectral methods to the problem (1), (2), (3). A spectral representation of the operator L allows us the obtain a representation of the solution operator (the *propa-*

gator) in terms of the sine and cosine families generated by L by a simple functional calculus. Introduce

$$R_1(t) = L^{-1/2}\sin(t\sqrt{L}) = \sum_{n=1}^{\infty} \frac{\sin(t\sqrt{\lambda_n})}{\sqrt{\lambda_n}} \langle \varphi_n^*, \cdot \rangle \varphi_n, \quad (22)$$

$$R_0(t) = \cos(t\sqrt{L}) = \sum_{n=1}^{\infty} \cos(t\sqrt{\lambda_n}) \langle \varphi_n^*, \cdot \rangle \varphi_n\,, \quad (23)$$

where $\lambda_1, \lambda_2, \ldots$ are the (positive) eigenvalues of L, and $\varphi_1, \varphi_2, \ldots$ are the corresponding eigenfunctions. Then the propagator of (1) can be written as

$$P(t) = \begin{bmatrix} R_0(t) & R_1(t) \\ -L\,R_1(t) & R_0(t) \end{bmatrix}. \quad (24)$$

The entries of this matrix, as functions of L, indicate which functions are the integrands in the Riemann-Stieltjes integrals used to compute the Fourier components of the solution.

3.1 Solution Using KSS Methods

We briefly review the use of Krylov subspace spectral methods for solving (1), first outlined in [12].

Since the exact solution $u(x,t)$ is given by

$$u(x,t) = R_0(t)f(x) + R_1(t)g(x), \quad (25)$$

where $R_0(t)$ and $R_1(t)$ are defined in (22), (23), we can obtain $[\mathbf{u}^{n+1}]_\omega$ by approximating each of the quadratic forms

$$\begin{aligned} c_\omega^+(t) &= \langle \hat{\mathbf{e}}_\omega, R_0(\Delta t)[\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n] \rangle & (26)\\ c_\omega^-(t) &= \langle \hat{\mathbf{e}}_\omega, R_0(\Delta t)\hat{\mathbf{e}}_\omega \rangle & (27)\\ s_\omega^+(t) &= \langle \hat{\mathbf{e}}_\omega, R_1(\Delta t)[\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}_t^n] \rangle & (28)\\ s_\omega^-(t) &= \langle \hat{\mathbf{e}}_\omega, R_1(\Delta t)\hat{\mathbf{e}}_\omega \rangle\,, & (29) \end{aligned}$$

where δ_ω is a nonzero constant. It follows that

$$[\hat{\mathbf{u}}^{n+1}]_\omega = \frac{c_\omega^+(t) - c_\omega^-(t)}{\delta_\omega} + \frac{s_\omega^+(t) - s_\omega^-(t)}{\delta_\omega}. \quad (30)$$

Similarly, we can obtain the coefficients $\tilde{v}_\omega$ of an approximation of $u_t(x,t)$ by approximating the quadratic forms

$$\begin{aligned} c_\omega^+(t)' &= -\langle \hat{\mathbf{e}}_\omega, LR_1(\Delta t)[\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n] \rangle & (31)\\ c_\omega^-(t)' &= -\langle \hat{\mathbf{e}}_\omega, LR_1(\Delta t)\hat{\mathbf{e}}_\omega \rangle & (32)\\ s_\omega^+(t)' &= \langle \hat{\mathbf{e}}_\omega, R_0(\Delta t)[\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}_t^n] \rangle & (33)\\ s_\omega^-(t)' &= \langle \hat{\mathbf{e}}_\omega, R_0(\Delta t)\hat{\mathbf{e}}_\omega \rangle\,. & (34) \end{aligned}$$

As noted in [18], this approximation to $u_t(x,t)$ does not introduce *any* error due to differentiation of our approximation of $u(x,t)$ with respect to t–the latter approximation can be differentiated *analytically*.

It follows from the preceding discussion that we can compute an approximate solution $\tilde{u}(x,t)$ at a given time T using the following algorithm.

Algorithm 1 *Given functions $c(x)$, $f(x)$, and $g(x)$ defined on the interval $(0, 2\pi)$, and a final time T, the following algorithm from [12] computes a function $\tilde{u}(x,t)$ that approximately solves the problem (1), (2) from $t = 0$ to $t = T$.*

$t = 0$
while $t < T$ **do**
  Select a time step Δt
  $f(x) = \tilde{u}(x,t)$
  $g(x) = \tilde{u}_t(x,t)$
  for $\omega = -N/2 + 1$ **to** $N/2$ **do**
    Choose a nonzero constant δ_ω
    Compute the quantities $c_\omega^+(\Delta t)$, $c_\omega^-(\Delta t)$, $s_\omega^+(\Delta t)$, $s_\omega^-(\Delta t)$, $c_\omega^+(\Delta t)'$, $c_\omega^-(\Delta t)'$, $s_\omega^+(\Delta t)'$, *and* $s_\omega^-(\Delta t)'$
    $\tilde{u}_\omega(\Delta t) = \frac{1}{\delta_\omega}(c_\omega^+(\Delta t) - c_\omega^-(\Delta t)) + \frac{1}{\delta_\omega}(s_\omega^+(\Delta t) - s_\omega^-(\Delta t))$
    $\tilde{v}_\omega(\Delta t) = \frac{1}{\delta_\omega}(c_\omega^+(\Delta t)' - c_\omega^-(\Delta t)') + \frac{1}{\delta_\omega}(s_\omega^+(\Delta t)' - s_\omega^-(\Delta t)')$
  end
  $\tilde{u}(x, t+\Delta t) = \sum_{\omega=1}^{N} \hat{\mathbf{e}}_\omega(x)\tilde{u}_\omega(\Delta t)$
  $\tilde{u}_t(x, t+\Delta t) = \sum_{\omega=1}^{N} \hat{\mathbf{e}}_\omega(x)\tilde{v}_\omega(\Delta t)$
  $t = t + \Delta t$
end

In this algorithm, each of the quantities inside the **for** loop are computed using K quadrature nodes. The nodes and weights are obtained in exactly the same way as for the parabolic problem (5), (6), (7). It should be noted that although 8 bilinear forms are required for each wave number ω, only three sets of nodes and weights need to be computed, and then they are used with different integrands.

3.2 Convergence Analysis

We now study the convergence behavior of the preceding algorithm, which we denote by KSS-W(K), where K is the number of Gaussian quadrature nodes used to approximate each Riemann-Stieltjes integral. Following the reformulation of Krylov subspace spectral methods presented in Section 4, we let $\delta_\omega \to 0$ to obtain

$$\begin{bmatrix} \hat{\mathbf{u}}^{n+1} \\ \hat{\mathbf{u}}_t^{n+1} \end{bmatrix}_\omega = \left(\sum_{k=1}^{K} w_k A_k\right) \begin{bmatrix} \hat{\mathbf{u}}^n \\ \hat{\mathbf{u}}_t^n \end{bmatrix} + \sum_{k=1}^{K} A_k \begin{bmatrix} w_k' \\ \tilde{w}_k' \end{bmatrix} - \sum_{k=1}^{K} w_k \frac{t}{2\sqrt{\lambda_k}} B_k \begin{bmatrix} \lambda_k' \\ \tilde{\lambda}_k' \end{bmatrix} - w_k C_k \begin{bmatrix} \lambda_k' \\ \tilde{\lambda}_k' \end{bmatrix},$$

where λ_k' and w_k' are the derivatives of the nodes and weights, respectively, in the direction of $\mathbf{u}^n$, and $\tilde{\lambda}_k'$ and $\tilde{w}_k'$ are the derivatives in the direction of $\mathbf{u}_t^n$. The matrices A_k, B_k and C_k are defined by

$$A_k = \begin{bmatrix} c_k & \frac{1}{\sqrt{\lambda_k}} s_k \\ -\sqrt{\lambda_k} s_k & c_k \end{bmatrix}$$

$$B_k = \begin{bmatrix} s_k & -\frac{1}{\sqrt{\lambda_k}}c_k \\ \sqrt{\lambda_k}c_k & s_k \end{bmatrix}$$
$$C_k = \begin{bmatrix} 0 & \frac{1}{2(\lambda_k)^{3/2}}s_k \\ \frac{1}{2\sqrt{\lambda_k}}s_k & 0 \end{bmatrix}$$

where $c_k = \cos(\sqrt{\lambda_k}t)$, $s_k = \sin(\sqrt{\lambda_k}t)$.

We first recall a result concerning the accuracy of each component of the approximate solution.

Theorem 2 *Assume that $f(x)$ and $g(x)$ satisfy (3), and let $u(x, \Delta t)$ be the exact solution of (1), (2) at $(x, \Delta t)$, and let $\tilde{u}(x, \Delta t)$ be the approximate solution computed by Algorithm 1. Then*

$$|\langle \hat{\mathbf{e}}_\omega, u(\cdot, \Delta t) - \tilde{u}(\cdot, \Delta t)\rangle| = O(\Delta t^{4K}), \tag{35}$$

$$|\langle \hat{\mathbf{e}}_\omega, u_t(\cdot, \Delta t) - \tilde{u}_t(\cdot, \Delta t)\rangle| = O(\Delta t^{4K-1}) \tag{36}$$

where K is the number of quadrature nodes used in Algorithm 1.

Proof. See [12]. □

To prove stability, we use the following norm,

$$\|(u, v)\|_L = \left(\langle u, Lu\rangle + \langle v, v\rangle\right)^{1/2}, \tag{37}$$

where L is a differential operator, used in [12] to show conservation for the wave equation. Specifically, if $u(x, t)$ is the solution to (1), (2), (3), then

$$\|(u(\cdot, t), u_t(\cdot, t)\|_L = \|(f(\cdot), g(\cdot))\|, \quad t > 0. \tag{38}$$

We also use the corresponding discrete norm,

$$\|(\mathbf{u}, \mathbf{v})\|_{L_N} = \left(\mathbf{u}^H L_N \mathbf{u} + \mathbf{v}^H \mathbf{v}\right)^{1/2}. \tag{39}$$

Let L be a constant-coefficient, self-adjoint, positive definite second-order differential operator with corresponding spectral discretization L_N, and let $\mathbf{u}^n$ be the discretization of the solution of (1) at time t_n. Then it is easily shown, in a manner analogous to [12, Lemma 2.8], that

$$\|(\mathbf{u}^n, \mathbf{u}_t^n)\|_{L_N} = \|(\mathbf{f}, \mathbf{g})\|_{L_N}, \tag{40}$$

where $\mathbf{f}$ and $\mathbf{g}$ are the discretizations of the initial data $f(x)$ and $g(x)$ from (2).

Theorem 3 *Let $p(x)$ be a positive constant function and $q(x)$ in (4) belong to $BL_M([0, 2\pi])$ for some integer M. Then, for the problem (1), (2), (3), KSS-W(1) is unconditionally stable.*

Proof. We write $L = C + V$, where C is the constant-coefficient operator obtained by averaging the coefficients of L. That is,

$$Cu = pu_{xx} + \overline{q}u, \quad V = \tilde{q}u,$$

where we use the notation

$$\overline{f} = \text{Avg}\, f = \frac{1}{2\pi}\int_0^{2\pi} f(x)\, dx, \quad \tilde{f} = f - \overline{f}.$$

As shown in [15, Theorem 6.3], the computed solution has the form

$$\begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{u}_t^{n+1} \end{bmatrix} = \tilde{S}_N(\Delta t) \begin{bmatrix} \mathbf{u}^n \\ \mathbf{u}_t^n \end{bmatrix}.$$

The approximate solution operator $\tilde{S}_N(\Delta t)$ has the form

$$\tilde{S}_N(\Delta t) = \left\{ P_N(\Delta t)\left[I_N + \frac{\Delta t}{2}B_N\right] + \frac{1}{2}G_N(\Delta t)B_N \right\}, \tag{41}$$

where $P_N(\Delta t)$ is the discrete solution operator on an N-point uniform grid for the problem $u_{tt} + Cu = 0$, and the operators $B_N(\Delta t)$ and $G_N(\Delta t)$ are defined by

$$B_N \begin{bmatrix} \mathbf{u}^n \\ \mathbf{u}_t^n \end{bmatrix} = \begin{bmatrix} C_N^{-1} V_N \mathbf{u}_t^n \\ V_N \mathbf{u}^n \end{bmatrix},$$

$$G_N(\Delta t) = C_N^{-1/2} \sin(C_N^{1/2}\Delta t).$$

Using the induced operator C_N-norm, we obtain

$$\|B_N\|_{C_N} \leq Q\overline{q}^{-1} \quad Q = \max_{0 \leq x \leq 2\pi} |\tilde{q}(x)|,$$

$$\|G_N(\Delta t)\|_{C_N} \leq \Delta t,$$

which, in combination with (40), yields

$$\|\tilde{S}_N(\Delta t)\|_{C_N} \leq 1 + \Delta t Q \overline{q}^{-1},$$

from which the result follows. □

Theorem 4 *Let $p(x)$ be a positive constant function and $q(x)$ in (4) belong to $BL_M([0, 2\pi])$ for some integer M. Then, for the problem (1), (2), (3), KSS-W(1) is convergent of order $(3, p)$, where the exact solution $u(x, t)$ belongs to $C^p([0, 2\pi])$ for each t in $[0, T]$.*

Proof. See [15, Theorem 6.4]. □

4 Block Formulation

In this section, we describe how we can compute elements of functions of matrices using block Gaussian quadrature. We then present a modification of KSS methods that employs this block approach.

4.1 Block Gaussian Quadrature

If we compute (15) using the formula (21) or the *polar decomposition*

$$\frac{1}{4}[(\mathbf{u} + \mathbf{v})^T f(A)(\mathbf{u} + \mathbf{v}) - (\mathbf{v} - \mathbf{u})^T f(A)(\mathbf{v} - \mathbf{u})], \tag{42}$$

then we would have to run the process for approximating an expression of the form (15) with two starting vectors. Instead we consider

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

which results in the 2×2 matrix

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \begin{bmatrix} \mathbf{u}^T f(A)\mathbf{u} & \mathbf{u}^T f(A)\mathbf{v} \\ \mathbf{v}^T f(A)\mathbf{u} & \mathbf{v}^T f(A)\mathbf{v} \end{bmatrix}, \qquad (43)$$

where $\mu(\lambda)$ is a 2×2 matrix function of λ, each entry of which is a measure of the form $\alpha(\lambda)$ from (18).

In [7] Golub and Meurant show how a block method can be used to generate quadrature formulas. We will describe this process here in more detail. The integral $\int_a^b f(\lambda)\, d\mu(\lambda)$ is now a 2×2 symmetric matrix and the most general K-node quadrature formula is of the form

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \sum_{j=1}^{K} W_j f(T_j) W_j + error \qquad (44)$$

with T_j and W_j being symmetric 2×2 matrices. Equation (44) can be simplified using

$$T_j = Q_j \Lambda_j Q_j^T$$

where Q_j is the eigenvector matrix and Λ_j the 2×2 diagonal matrix containing the eigenvalues. Hence,

$$\sum_{j=1}^{K} W_j f(T_j) W_j = \sum_{j=1}^{K} W_j Q_j f(\Lambda_j) Q_j^T W_j$$

and if we write $W_j Q_j f(\Lambda_j) Q_j^T W_j$ as

$$f(\lambda_1)\mathbf{z}_1\mathbf{z}_1^T + f(\lambda_2)\mathbf{z}_2\mathbf{z}_2^T,$$

where $\mathbf{z}_k = W_j Q_j \mathbf{e}_k$ for $k = 1, 2$, we get for the quadrature rule

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

where t_j is a scalar and $\mathbf{v}_j$ is a vector with two components.

We now describe how to obtain the scalar nodes t_j and the associated vectors $\mathbf{v}_j$. In [7] it is shown that there exist orthogonal matrix polynomials such that

$$\lambda p_{j-1}(\lambda) = p_j(\lambda)B_j + p_{j-1}(\lambda)M_j + p_{j-2}(\lambda)B_{j-1}^T$$

with $p_0(\lambda) = I_2$ and $p_{-1}(\lambda) = 0$. We can write the last equation as

$$\lambda \begin{bmatrix} p_0(\lambda) \\ p_1(\lambda) \\ \vdots \\ p_{K-1}(\lambda) \end{bmatrix} = \mathcal{T}_K \begin{bmatrix} p_0(\lambda) \\ p_1(\lambda) \\ \vdots \\ p_{K-1}(\lambda) \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ p_K(\lambda)B_K^T \end{bmatrix}$$

with

$$\mathcal{T}_K = \begin{bmatrix} M_1 & B_1^T & & & \\ B_1 & M_2 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-2} & M_{K-1} & B_{K-1}^T \\ & & & B_{K-1} & M_K \end{bmatrix} \qquad (45)$$

which is a block-triangular matrix. Therefore, we can define the quadrature rule as

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \sum_{j=1}^{2K} f(\lambda_j)\mathbf{v}_j\mathbf{v}_j^T + error \qquad (46)$$

where $2K$ is the order of the matrix $\mathcal{T}_K$, λ_j an eigenvalue of $\mathcal{T}_K$ and $\mathbf{u}_j$ is the vector consisting of the first two elements of the corresponding normalized eigenvector.

To compute the matrices M_j and B_j, we use the block Lanczos algorithm, which was proposed by Golub and Underwood in [10]. Let X_0 be an $N \times 2$ given matrix, such that $X_1^T X_1 = I_2$. Let $X_0 = 0$ be an $N \times 2$ matrix. Then, for $j = 1, \ldots$, we compute

$$\begin{aligned} M_j &= X_j^T A X_j, \\ R_j &= A X_j - X_j M_j - X_{j-1} B_{j-1}^T, \qquad (47) \\ X_{j+1} B_j &= R_j. \end{aligned}$$

The last step of the algorithm is the QR decomposition of R_j (see [9]) such that X_j is $n \times 2$ with $X_j^T X_j = I_2$. The matrix B_j is 2×2 upper triangular. The other coefficient matrix M_j is 2×2 and symmetric. The matrix R_j can eventually be rank deficient and in that case B_j is singular. The solution of this problem is given in [10].

4.2 Block KSS Methods

We are now ready to describe block KSS methods. For each wave number $\omega = -N/2 + 1, \ldots, N/2$, we define

$$R_0 = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \mathbf{u}^n \end{bmatrix}, \quad \tilde{R}_0 = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \mathbf{u}_t^n \end{bmatrix},$$

and then compute the QR factorizations

$$R_0 = X_1 B_0, \quad \tilde{R}_0 = \tilde{X}_1 \tilde{B}_0,$$

which yields

$$X_1 = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \mathbf{u}_\omega^n / \|\mathbf{u}_\omega^n\|_2 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 1 & \hat{\mathbf{e}}_\omega^H \mathbf{u}^n \\ 0 & \|\mathbf{u}_\omega^n\|_2 \end{bmatrix},$$

$$\tilde{X}_1 = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \mathbf{u}_{t,\omega}^n / \|\mathbf{u}_{t,\omega}^n\|_2 \end{bmatrix}, \quad \tilde{B}_0 = \begin{bmatrix} 1 & \hat{\mathbf{e}}_\omega^H \mathbf{u}_t^n \\ 0 & \|\mathbf{u}_{t,\omega}^n\|_2 \end{bmatrix},$$

where

$$\begin{aligned} \mathbf{u}_\omega^n &= \mathbf{u}^n - \hat{\mathbf{e}}_\omega \hat{\mathbf{e}}_\omega^H \mathbf{u}^n, \\ \mathbf{u}_{t,\omega}^n &= \mathbf{u}_t^n - \hat{\mathbf{e}}_\omega \hat{\mathbf{e}}_\omega^H \mathbf{u}_t^n. \end{aligned}$$

We then carry out the block Lanczos iteration described in (47) to obtain block tridiagonal matrices $\mathcal{T}_K$ and $\tilde{\mathcal{T}}_K$, where

$$\mathcal{T}_K = \begin{bmatrix} M_1 & B_1^H & & & \\ B_1 & M_2 & B_2^H & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-2} & M_{K-1} & B_{K-1}^H \\ & & & B_{K-1} & M_K \end{bmatrix} \tag{48}$$

and $\tilde{\mathcal{T}}_K$ is defined similarly. Then, we can express each Fourier component of the approximate solution at the next time step as

$$\begin{aligned}[\hat{\mathbf{u}}^{n+1}]_\omega &= \left[B_0^H E_{12}^H \cos[\mathcal{T}_K^{1/2}\Delta t] E_{12} B_0\right]_{12} + \\ &\quad \left[\tilde{B}_0^H E_{12}^H \mathcal{T}_K^{-1/2} \sin[\tilde{\mathcal{T}}_K^{1/2}\Delta t] E_{12}\tilde{B}_0\right]_{12},\end{aligned}$$

and each Fourier component of its time derivative is approximated by

$$\begin{aligned}[\hat{\mathbf{u}}_t^{n+1}]_\omega &= -\left[B_0^H E_{12}^H \mathcal{T}_K^{1/2} \sin[\mathcal{T}_K^{1/2}\Delta t] E_{12} B_0\right]_{12} + \\ &\quad \left[\tilde{B}_0^H E_{12}^H \cos[\tilde{\mathcal{T}}_K^{1/2}\Delta t] E_{12}\tilde{B}_0\right]_{12},\end{aligned}$$

where

$$E_{12} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}.$$

We denote this method by KSS-WB(K).

The computation of $E_{12}^H \cos[\mathcal{T}_K^{1/2}\Delta t] E_{12}$, and similar expressions above, consists of computing the eigenvalues and eigenvectors of $\mathcal{T}_K$ in order to obtain the nodes and weights for Gaussian quadrature, as described earlier in this section.

4.3 Implementation

In [21], it was demonstrated that recursion coefficients for all wave numbers $\omega = -N/2+1, \ldots, N/2$ can be computed simultaneously, by regarding them as functions of ω and using symbolic calculus to apply differential operators analytically, as much as possible. As a result, KSS methods require $O(N \log N)$ floating-point operations per time step, which is comparable to other time-stepping methods. The same approach can be applied to block KSS methods. For both types of methods, it can be shown that for a K-node Gaussian rule or block Gaussian rule, K applications of the operator L_N to the previous solution $\mathbf{u}^n$ are needed.

5 Convergence Analysis

We now examine the convergence of block KSS methods by first investigating their consistency and stability. As shown in [15, 20], the original KSS methods are high-order accurate in time, but are also explicit methods that possess stability properties characteristic of implicit methods, so it is desired that block KSS methods share both of these traits with their predecessors.

5.1 Consistency

The error in a K-node block Gaussian quadrature rule of the form (46) is

$$R(f) = \frac{f^{(2K)}(\eta)}{(2K)!} \int_a^b \prod_{j=1}^{2K} (\lambda - \lambda_j)\, d\mu(\lambda). \tag{49}$$

It follows that the rule is exact for polynomials of degree up to $2K-1$, which is proven in [2]. The above form of the remainder can be obtained using results from [24]. We now use this remainder to prove the consistency of block KSS methods for the wave equation.

Theorem 5 *Let L be a self-adjoint 2nd-order positive definite differential operator on $C_p([0,2\pi])$ with coefficients in $BL_M([0,2\pi])$ for a fixed integer M, and let $f, g \in C^n([0,2\pi])$ for $n \geq 4K$ for a positive integer K. Let $N \geq M$, and that for each $\omega = -N/2+1, \ldots, N/2$, the recursion coefficients in (45) are computed on a $2^K N$-point uniform grid. Then a block KSS method that uses a K-node block Gaussian rule to compute each Fourier component $[\hat{\mathbf{u}}^1]_\omega$, for $\omega = -N/2+1, \ldots, N/2$, of the solution to (1), (2), (3), and each Fourier component $[\hat{\mathbf{u}}_t^1]_\omega$ of its time derivative, satisfies*

$$\left|[\hat{\mathbf{u}}^1]_\omega - \hat{u}(\omega, \Delta t)\right| = O(\Delta t^{4K}),$$

$$\left|[\hat{\mathbf{u}}_t^1]_\omega - \hat{u}_t(\omega, \Delta t)\right| = O(\Delta t^{4K-1}),$$

where $\hat{u}(\omega, \Delta t)$ is the corresponding Fourier component of the exact solution at time Δt, and $\hat{u}_t(\omega, \Delta t)$ is the corresponding Fourier component of its time derivative at time Δt.

Proof. The result follows from the substitution of $\cos(\sqrt{\lambda}\Delta t)$, $\lambda^{-1/2}\sin(\sqrt{\lambda}\Delta t)$, and $-\lambda^{1/2}\sin(\sqrt{\lambda}\Delta t)$ for the integrand $f(\lambda)$ in the quadrature error (49), and the elimination of spatial error from the computation of the recursion coefficients by refining the grid to the extent necessary to resolve all Fourier components of pointwise products of functions.

It is important to note that the error term for each Fourier component of $\mathbf{u}^1$ and $\mathbf{u}_t^1$ is actually a Fourier component of the application of a fixed pseudodifferential operator to $f_N(x)$, the N-point Fourier interpolant of $f(x)$. This pseudodifferential operator is of order at most $4K$.

Specifically, we have the following error terms for each Fourier component of $\mathbf{u}^1$ and $\mathbf{u}_t^1$:

$$E_{\omega,N}(\Delta t) = \Delta x \frac{\Delta t^{4K}}{(2K)!} \hat{\mathbf{e}}_\omega^H \prod_{j=1}^{2K} (L_N - \lambda_j I)\mathbf{f} +$$

$$\Delta x\frac{\Delta t^{4K+1}}{(2K)!}\hat{\mathbf{e}}_\omega^H\prod_{j=1}^{2K}(L_N-\tilde{\lambda}_jI)\mathbf{g},$$

$$\tilde{E}_{\omega,N}(\Delta t) = \Delta x\frac{\Delta t^{4K-1}}{(2K)!}\hat{\mathbf{e}}_\omega^H\prod_{j=1}^{2K}(L_N-\lambda_jI)\mathbf{f}+$$
$$\Delta x\frac{\Delta t^{4K}}{(2K)!}\hat{\mathbf{e}}_\omega^H\prod_{j=1}^{2K}(L_N-\tilde{\lambda}_jI)\mathbf{g}.$$

The factor of Δx is needed to normalize $\hat{\mathbf{e}}_\omega$ so that each Fourier component of the approximate solution is an approximation of the corresponding Fourier component of the exact solution.

The nodes, being the eigenvalues of $\mathcal{T}_K$ and $\tilde{\mathcal{T}}_K$, grow like $O(\omega^2)$, as they can be expressed as quadratic forms $\mathbf{w}^HL_N\mathbf{w}$ where $\mathbf{w}$ is a unit vector, and L_N is a discretization of a second-order differential operator. Therefore, for $j=1,\ldots,K$, each node λ_j or $\tilde{\lambda}_j$, as a function of $\omega\in\mathbb{Z}$, defines the spectrum of a second-order pseudodifferential operator, with corresponding eigenfunctions $e^{i\omega x}$.

It follows that the constant in the error term for each coefficient is bounded independently of N, and due to the smoothness of the coefficients and initial data, the overall local errors, $\|u(\cdot,\Delta t)-\tilde{u}(\cdot,\Delta t)\|_{L^2}$ and $\|u_t(\cdot,\Delta t)-\tilde{u}_t(\cdot,\Delta t)\|_{L^2}$, are also bounded independently of N. □

5.2 Stability for the One-Node Case

When $K=1$, we simply have $\mathcal{T}_1=M_1$, where

$$M_1=\begin{bmatrix}\hat{\mathbf{e}}_\omega^HL_N\hat{\mathbf{e}}_\omega & \hat{\mathbf{e}}_\omega^HL_N\mathbf{u}_\omega^n\\ [\mathbf{u}_\omega^n]^HL_N\hat{\mathbf{e}}_\omega & [\mathbf{u}_\omega^n]^HL_N\mathbf{u}_\omega^n\end{bmatrix}. \tag{50}$$

We now examine the stability of the 1-node method in the case where $p(x)\equiv p=$ constant. We then have

$$\mathcal{T}_1=\begin{bmatrix}p\omega^2+\bar{q} & \hat{\mathbf{e}}_\omega^H\tilde{\mathbf{q}}\mathbf{u}_\omega^n\\ [\mathbf{u}_\omega^n]^H\tilde{\mathbf{q}}\hat{\mathbf{e}}_\omega & [\mathbf{u}_\omega^n]^HL_N\mathbf{u}_\omega^n\end{bmatrix}. \tag{51}$$

We use the notation $\overline{f}$ to denote the mean of a function $f(x)$ defined on $[0,2\pi]$, and define $\tilde{q}(x)=q(x)-\overline{q}$. We denote by $\tilde{\mathbf{q}}$ the vector with components $[\tilde{\mathbf{q}}]_j=\tilde{q}(x_j)$. For convenience, multiplication of vectors, as in the off-diagonal elements of $\mathcal{T}_1$, denotes component-wise multiplication.

Because M_1 is Hermitian, we can write

$$M_1=U_1\Lambda_1U_1^H.$$

The Fourier component $[\hat{\mathbf{u}}^{n+1}]_\omega$ is then obtained as follows:

$$\begin{aligned}[\hat{\mathbf{u}}^{n+1}]_\omega &= \left[B_0^H\cos[\mathcal{T}_1^{1/2}\Delta t]B_0\right]_{12}+\\ &\quad\left[\tilde{B}_0^H\mathcal{T}_1^{-1/2}\sin[\tilde{\mathcal{T}}_1^{1/2}\Delta t]\tilde{B}_0\right]_{12}\\ &= \left[B_0^HU_1\cos[-\Lambda_1^{1/2}\Delta t]U_1^HB_0\right]_{12}+\\ &\quad\left[\tilde{B}_0^H\tilde{U}_1\tilde{\Lambda}_1^{-1/2}\sin[\tilde{\Lambda}_1^{1/2}\Delta t]\tilde{U}_1^H\tilde{B}_0\right]_{12}\\ &= \begin{bmatrix}u_{11}c_1 & u_{12}c_2\end{bmatrix}\times\\ &\quad\begin{bmatrix}\overline{u_{11}} & \overline{u_{21}}\\ \overline{u_{12}} & \overline{u_{22}}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{e}}_\omega^H\mathbf{u}^n\\ \|\mathbf{u}_\omega^n\|_2\end{bmatrix}+\\ &\quad\begin{bmatrix}\tilde{u}_{11}\tilde{\lambda}_1^{-1/2}\tilde{s}_1 & \tilde{u}_{12}\tilde{\lambda}_2^{-1/2}\tilde{s}_2\end{bmatrix}\times\\ &\quad\begin{bmatrix}\overline{\tilde{u}_{11}} & \overline{\tilde{u}_{21}}\\ \overline{\tilde{u}_{12}} & \overline{\tilde{u}_{22}}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{e}}_\omega^H\mathbf{u}_t^n\\ \|\mathbf{u}_{t,\omega}^n\|_2\end{bmatrix}\\ &= [|u_{11}|^2c_1+|u_{12}|^2c_2]\hat{\mathbf{e}}_\omega^H\mathbf{u}^n+\\ &\quad[u_{11}\overline{u_{21}}c_1+u_{12}\overline{u_{22}}c_2]\|\mathbf{u}_\omega^n\|_2+\\ &\quad[|\tilde{u}_{11}|^2\tilde{\lambda}_1^{-1/2}\tilde{s}_1+|\tilde{u}_{12}|^2\tilde{\lambda}_2^{-1/2}\tilde{s}_2]\hat{\mathbf{e}}_\omega^H\mathbf{u}_t^n+\\ &\quad[\tilde{u}_{11}\overline{\tilde{u}_{21}}\tilde{\lambda}_1^{-1/2}\tilde{s}_1+\tilde{u}_{12}\overline{\tilde{u}_{22}}\tilde{\lambda}_2^{-1/2}\tilde{s}_2]\|\mathbf{u}_{t,\omega}^n\|_2.\end{aligned}$$

where, for $i=1,2$,

$$\begin{aligned}c_i &= \cos(\sqrt{\lambda_i}\Delta t),\\ s_i &= \sin(\sqrt{\lambda_i}\Delta t),\\ \tilde{c}_i &= \cos(\sqrt{\tilde{\lambda}_i}\Delta t),\\ \tilde{s}_i &= \sin(\sqrt{\tilde{\lambda}_i}\Delta t).\end{aligned}$$

Similarly,

$$\begin{aligned}[\hat{\mathbf{u}}_t^{n+1}]_\omega &= [|u_{11}|^2(-\lambda_1^{1/2})s_1+|u_{12}|^2(-\lambda_2^{1/2})s_2]\hat{\mathbf{u}}_\omega^n+\\ &\quad[u_{11}\overline{u_{21}}(-\lambda_1^{1/2})s_1+u_{12}\overline{u_{22}}(-\lambda_2^{1/2})]\|\mathbf{u}_\omega^n\|_2+\\ &\quad[|\tilde{u}_{11}|^2\tilde{c}_1+|\tilde{u}_{12}|^2\tilde{c}_2]\hat{\mathbf{u}}_{t,\omega}^n+\\ &\quad[\tilde{u}_{11}\overline{\tilde{u}_{21}}\tilde{c}_1+\tilde{u}_{12}\overline{\tilde{u}_{22}}\tilde{c}_2]\|\mathbf{u}_{t,\omega}^n\|_2.\end{aligned}$$

This simple form of the approximate solution operator yields the following result. As before, we denote by $\tilde{S}_N(\Delta t)$ the matrix such that

$$\begin{bmatrix}\mathbf{u}^{n+1}\\ \mathbf{u}_t^{n+1}\end{bmatrix}=\tilde{S}_N(\Delta t)\begin{bmatrix}\mathbf{u}^n\\ \mathbf{u}_t^n\end{bmatrix},$$

for given N and Δt. For simplicity, we use the notation $\tilde{S}_N(\Delta t)^n$ in place of $\left[\tilde{S}_N(\Delta t)\right]^n$.

Theorem 6 *Let $p(x)$ be a positive constant function and let $q(x)$ in (4) belong to $BL_M([0,2\pi])$ for a fixed integer M. Then, for the problem (1), (2), (3), the block KSS method with $K=1$, KSS-WB(1), is unconditionally stable. That is, given $T>0$, there exists a constant C_T, independent of N and Δt, such that*

$$\|\tilde{S}_N(\Delta t)^n\|_C\le C_T, \tag{52}$$

for $0\le n\Delta t\le T$, where C is the differential operator defined by $Cu=-pu_{xx}+\overline{q}u$.

Proof. As in the proof of Theorem 3, we write $L=C+V$. We also write

$$\mathbf{u}^{n+1}=\mathbf{v}^{n+1}+\mathbf{w}^{n+1},\quad \mathbf{u}_t^{n+1}=\mathbf{v}_t^{n+1}+\mathbf{w}_t^{n+1},$$

where $\mathbf{v}^{n+1}$ is the approximate solution at time Δt to the constant-coefficient problem

$$v_{tt} + Cv = 0,$$

with initial conditions

$$v(x,0) = \tilde{u}(x,t_n), \quad v_t(x,0) = \tilde{u}_t(x,t_n),$$

and periodic boundary conditions, and $\mathbf{w}^{n+1}$ is the approximate solution at time Δt to the problem

$$w_{tt} + Vw = 0,$$

with initial conditions

$$w(x,0) = \tilde{u}(x,t_n), \quad w_t(x,0) = \tilde{u}(x,t_n).$$

Then, using notation from the proof of Theorem 3,

$$\left[\begin{array}{c} \mathbf{v}^{n+1} \\ \mathbf{v}_t^{n+1} \end{array} \right] = P_N(\Delta t) \left[\begin{array}{c} \mathbf{u}^n \\ \mathbf{u}_t^n \end{array} \right],$$

and therefore

$$\|(\mathbf{v}^{n+1}, \mathbf{v}_t^{n+1})\|_{C_N} = \|(\mathbf{u}^n, \mathbf{u}_t^n)\|_{C_N}.$$

We now need to bound the C_N-norm of the operator $Q_N(\Delta t)$ such that

$$\left[\begin{array}{c} \mathbf{w}^{n+1} \\ \mathbf{w}_t^{n+1} \end{array} \right] = Q_N(\Delta t) \left[\begin{array}{c} \mathbf{u}^n \\ \mathbf{u}_t^n \end{array} \right],$$

by Δt times a constant that is independent of N. From the previously described expressions for the Fourier components of $\mathbf{u}^{n+1}$ and $\mathbf{u}_t^{n+1}$, and the fact that for each ω, the matrix U_1 is orthogonal, we obtain

$$\begin{aligned} [\hat{\mathbf{w}}^{n+1}]_\omega &= (c_1 - c_2)[u_{11}\overline{u_{21}}\|\mathbf{u}_\omega^n\|_2 - |u_{12}|^2 \hat{\mathbf{e}}_\omega^H \mathbf{u}^n] + \\ &\quad (\tilde{\lambda}_1^{-1/2}\tilde{s}_1 - \tilde{\lambda}_2^{-1/2}\tilde{s}_2) \times \\ &\quad [\tilde{u}_{11}\overline{\tilde{u}_{21}}\|\mathbf{u}_{t,\omega}^n\|_2 - |\tilde{u}_{12}|^2 \hat{\mathbf{e}}_\omega^H \mathbf{u}_t^n], \end{aligned}$$

$$\begin{aligned} [\hat{\mathbf{w}}_t^{n+1}]_\omega &= [\lambda_1^{1/2}s_1 - \lambda_2^{1/2}s_2][|u_{12}|^2 \hat{\mathbf{u}}_\omega^n - u_{11}\overline{u_{21}}\|\mathbf{u}_\omega^n\|_2] + \\ &\quad (\tilde{c}_1 - \tilde{c}_2)[\tilde{u}_{11}\overline{\tilde{u}_{21}}\|\mathbf{u}_{t,\omega}^n\|_2 - |\tilde{u}_{12}|^2 \hat{\mathbf{u}}_{t,\omega}^n]. \end{aligned}$$

If the nodes λ_1 and λ_2 are equal, then the contributions to these Fourier components due to $\mathbf{u}^n$ are zero, and if $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ are equal, then the contributions due to $\mathbf{u}_t^n$ are zero. From this point on, we assume these nodes are distinct. Let $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$. Let

$$\alpha_1 = \hat{\mathbf{e}}_\omega^H L_N \hat{\mathbf{e}}_\omega = p\omega^2 + \overline{q}, \quad \alpha_2 = [\mathbf{u}_\omega^n]^H L_N \mathbf{u}_\omega^n,$$

$$\tilde{\alpha}_1 = \hat{\mathbf{e}}_\omega^H L_N \hat{\mathbf{e}}_\omega = \alpha_1, \quad \tilde{\alpha}_2 = [\mathbf{u}_{t,\omega}^n]^H L_N \mathbf{u}_{t,\omega}^n.$$

By direct computation of the elements of U_1, whose columns are the eigenvectors of $\mathcal{T}_1$, we obtain

$$|u_{12}|^2 = \frac{\epsilon}{\lambda_1 - \lambda_2}, \quad u_{11}\overline{u_{21}} = \frac{\hat{\mathbf{e}}_\omega^H \tilde{\mathbf{q}} \mathbf{u}_\omega^n}{\|\mathbf{u}_\omega^n\|_2(\lambda_1 - \lambda_2)},$$

where

$$\epsilon = \lambda_1 - \alpha_1 = \frac{|\hat{\mathbf{e}}_\omega^H \tilde{\mathbf{q}} \mathbf{u}_\omega^n|^2}{\|\mathbf{u}_\omega^n\|_2^2(\lambda_1 + \alpha_2)}.$$

Similarly,

$$|\tilde{u}_{12}|^2 = \frac{\tilde{\epsilon}}{\tilde{\lambda}_1 - \tilde{\lambda}_2}, \quad \tilde{u}_{11}\overline{\tilde{u}_{21}} = \frac{\hat{\mathbf{e}}_\omega^H \tilde{\mathbf{q}} \mathbf{u}_{t,\omega}^n}{\|\mathbf{u}_{t,\omega}^n\|_2(\tilde{\lambda}_1 - \tilde{\lambda}_2)},$$

$$\tilde{\epsilon} = \tilde{\lambda}_1 - \tilde{\alpha}_1 = \frac{|\hat{\mathbf{e}}_\omega^H \tilde{\mathbf{q}} \mathbf{u}_{t,\omega}^n|^2}{\|\mathbf{u}_{t,\omega}^n\|_2^2(\tilde{\lambda}_1 + \tilde{\alpha}_2)}.$$

Since α_1 is a quadratic function of ω and

$$\lim_{N\to\infty} \alpha_2 = \frac{\langle \tilde{u}(\cdot,t_n), L\tilde{u}(\cdot,t_n)\rangle}{\langle \tilde{u}(\cdot,t_n), \tilde{u}(\cdot,t_n)\rangle},$$

it follows that ϵ and $\tilde{\epsilon}$ are eigenvalues of a constant-coefficient pseudodifferential operator of order -2.

We now have

$$\begin{aligned} [\hat{\mathbf{w}}^{n+1}]_\omega &= \frac{c_1 - c_2}{\lambda_1 - \lambda_2}[\hat{\mathbf{e}}_\omega^H \tilde{\mathbf{q}} \mathbf{u}_\omega^n - \epsilon \hat{\mathbf{e}}_\omega^H \mathbf{u}^n] + \\ &\quad \frac{\tilde{\lambda}_1^{-1/2}\tilde{s}_1 - \tilde{\lambda}_2^{-1/2}\tilde{s}_2}{\tilde{\lambda}_1 - \tilde{\lambda}_2}[\hat{\mathbf{e}}_\omega^H \tilde{\mathbf{q}} \mathbf{u}_{t,\omega}^n - \tilde{\epsilon} \hat{\mathbf{e}}_\omega^H \mathbf{u}_t^n], \end{aligned}$$

$$\begin{aligned} [\hat{\mathbf{w}}_t^{n+1}]_\omega &= \frac{\lambda_1^{1/2}s_1 - \lambda_2^{1/2}s_2}{\lambda_1 - \lambda_2}[\epsilon \hat{\mathbf{u}}_\omega^n - \hat{\mathbf{e}}_\omega^H \tilde{\mathbf{q}} \mathbf{u}_\omega^n] + \\ &\quad \frac{\tilde{c}_1 - \tilde{c}_2}{\tilde{\lambda}_1 - \tilde{\lambda}_2}[\hat{\mathbf{e}}_\omega^H \tilde{\mathbf{q}} \mathbf{u}_{t,\omega}^n - \tilde{\epsilon} \hat{\mathbf{u}}_{t,\omega}^n]. \end{aligned}$$

From Taylor expansions of sin and cos, we obtain

$$|c_1 - c_2| \leq \frac{\Delta t^2}{2} \max\{\lambda_1, \lambda_2\}, \tag{53}$$

$$|\tilde{c}_1 - \tilde{c}_2| \leq \frac{\Delta t^2}{2} \max\{\tilde{\lambda}_1, \tilde{\lambda}_2\}, \tag{54}$$

$$|\tilde{\lambda}_1^{-1/2}\tilde{s}_1 - \tilde{\lambda}_2^{-1/2}\tilde{s}_2| \leq 2\Delta t, \tag{55}$$

$$|\lambda_1^{1/2}s_1 - \lambda_2^{1/2}s_2| \leq \Delta t(\lambda_1 + \lambda_2). \tag{56}$$

It follows that

$$\begin{aligned} \left[\begin{array}{c} \mathbf{w}^{n+1} \\ \mathbf{w}_t^{n+1} \end{array} \right] &= \left[\begin{array}{cc} Q_{11,N}(\Delta t) & Q_{12,N}(\Delta t) \\ Q_{21,N}(\Delta t) & Q_{22,N}(\Delta t) \end{array} \right] \times \\ &\quad \left[\begin{array}{cc} V_N - \mathcal{E}_N & 0 \\ 0 & V_N - \tilde{\mathcal{E}}_N \end{array} \right] \left[\begin{array}{c} \mathbf{u}^n \\ \mathbf{u}_t^n \end{array} \right], \end{aligned}$$

where V_N is the discretization of the operator V defined by $Vu = \tilde{q}u$, and $\mathcal{E}_N$ is defined by

$$\mathcal{E}_N = \mathcal{F}_N^{-1} \hat{\mathcal{E}}_N \mathcal{F}_N,$$

where $\mathcal{F}_N$ denotes the discrete Fourier transform and $\hat{\mathcal{E}}_N$ is the diagonal matrix of eigenvalues, which, for each ω, are equal to ϵ. The matrix $\tilde{\mathcal{E}}_N$ is defined similarly, with eigenvalues given by $\tilde{\epsilon}$. It follows from earlier discussion

that $\mathcal{E}_N$ and $\tilde{\mathcal{E}}_N$ are discretizations of pseudodifferential operators whose symbols are of order -2, which implies that

$$\left\| \begin{bmatrix} V_N - \mathcal{E}_N & 0 \\ 0 & V_N - \tilde{\mathcal{E}}_N \end{bmatrix} \right\|_{C_N} \leq Q,$$

where Q is a constant that is independent of N. Note that the smoothness of the coefficient $q(x)$ allows us to bound V_N independently of N.

The pseudodifferential operators $Q_{ii}(\Delta t)$, for $i = 1, 2$, with discretizations $Q_{ii,N}(\Delta t)$, are of order Δt^2, and bounded independently of N, as their eigenvalues,

$$\lambda_\omega(Q_{11}(\Delta t)) = \frac{c_1 - c_2}{\lambda_1 - \lambda_2},$$

$$\lambda_\omega(Q_{22}(\Delta t)) = \frac{\tilde{c}_1 - \tilde{c}_2}{\tilde{\lambda}_1 - \tilde{\lambda}_2},$$

converge to $\Delta t^2/2$ as $|\omega| \to \infty$, as can be seen from (53) and (54), and the fact that λ_2 and $\tilde{\lambda}_2$, being equal to $\alpha_2 - \epsilon$ and $\tilde{\alpha}_2 - \tilde{\epsilon}$, respectively, are bounded independently of ω. This yields

$$\left\| \begin{bmatrix} Q_{11,N}(\Delta t) & 0 \\ 0 & Q_{22,N}(\Delta t) \end{bmatrix} \right\|_{C_N} \leq \Delta t^2 D,$$

where D is a constant independent of N and Δt.

The operator $Q_{12}(\Delta t)$ is equal to Δt times an operator whose symbol is of order -2, while $Q_{21}(\Delta t)$ is equal to Δt times a bounded operator. We then have

$$\|(\mathbf{w}^{n+1}, \mathbf{w}_t^{n+1})\|_{C_N} \leq \Delta t^2 QD \|(\mathbf{u}^n, \mathbf{u}_t^n)\|_{C_N} + \mathcal{R}^{1/2},$$

where

$$\mathcal{R} = [Q_{12}(\Delta t)(\tilde{V}_N - \tilde{\mathcal{E}})\mathbf{u}_t^n]^H C_N [Q_{12}(\Delta t)(\tilde{V}_N - \tilde{\mathcal{E}})\mathbf{u}_t^n] + \|Q_{21}(\Delta t)(\tilde{V}_N - \tilde{\mathcal{E}})\mathbf{u}^n\|_2^2.$$

The first term in $\mathcal{R}$ can be written as

$$[(\tilde{V}_N - \tilde{\mathcal{E}})\mathbf{u}_t^n]^H R_N(\Delta t)[(\tilde{V}_N - \tilde{\mathcal{E}})\mathbf{u}_t^n],$$

where $R_N(\Delta t) = \mathcal{F}^{-1}\hat{R}_N(\Delta t)\mathcal{F}$ and has eigenvalues that, for each ω, satisfy

$$|\lambda_\omega(R_N(\Delta t))| \leq \frac{4\Delta t^2 \lambda_1}{|\lambda_1 - \lambda_2|^2}.$$

It follows that

$$\mathcal{R}^{1/2} \leq 2\Delta t \tilde{Q}^{-1} Q \|(\mathbf{u}^n, \mathbf{u}_t^n)\|_{C_N},$$

where

$$\tilde{Q} = \min\left\{\overline{q}, \min_{\omega \in Z} \frac{(\lambda_1 - \lambda_2)^2}{\lambda_1}\right\}.$$

We conclude that

$$\|Q_N(\Delta t)\|_{C_N} \leq 2\Delta t \tilde{Q}^{-1} Q + \Delta t^2 QD,$$

and therefore

$$\|\tilde{S}_N(\Delta t)\|_{C_N} \leq 1 + 2\Delta t \tilde{Q}^{-1} Q + \Delta t^2 QD,$$

from which the result follows. □

Theorems 5 and 6 immediately imply the following result.

Theorem 7 *Let the exact solution $u(x,t)$ of the problem (1), (2), (3) belong to $C^p([0, 2\pi])$ for each t in $[0, T]$. Let $q(x)$ in (4) belong to $BL_M([0, 2\pi])$ for some integer M. Then, the 1-node block KSS method, applied to this problem, is convergent of order $(2, p)$. That is, there exist constants C_t and C_x, independent of the time step Δt and grid spacing $\Delta x = 2\pi/N$, such that*

$$\|u(\cdot, t) - \mathbf{u}(\cdot, t)\|_{L^2} \leq C_t \Delta t^2 + C_x \Delta x^p, \quad 0 \leq t \leq T. \quad (57)$$

Proof. The proof proceeds in a manner analogous to [15, Theorem 6.4]. □

It is important to note that although stability and convergence were only shown for the case where the leading coefficient $p(x)$ is constant, it has been demonstrated that KSS methods exhibit similar stability on more general problems, such as in [15] where it was applied to a second-order wave equation with time steps that greatly exceeded the CFL limit, even though the leading coefficient was not constant. Furthermore, [15] also introduced homogenizing similarity transformations that can be used to extend the applicability of theoretical results concerning stability that were presented in that paper, as well as the one given here.

6 Numerical Results

In this section, we will present numerical results for comparisons between the original KSS method (as described in [15]) and the new block KSS method, applied to the second-order wave equation. The comparisons will focus on the accuracy of the temporal approximations employed by each method. A thorough analysis of the spatial discretization error, along with modifications needed to achieve high accuracy in the case of oscillatory or discontinuous initial data, will be deferred for future work.

In [17] it was shown that the original KSS method KSS-W(K) for the wave equation compared favorably to the standard ODE solvers provided in MATLAB (described in [23]), as was demonstrated for parabolic problems in [21]. We now compare that method to our new block approach. In [17], we computed solutions at $T = 1$ and compared all methods at time steps $\Delta t = 2^{-j}$, where j is a small nonnegative integer. Here, we use $T = 10$, with all time steps increased by a factor of 10 as well. For time steps this large, we found that MATLAB's ODE solvers performed so poorly that they did not exhibit any sign of convergence, so we do not include comparisons to those time-stepping methods here.

6.1 Construction of Test Cases

We introduce some differential operators and functions that will be used in the experiments described in this section. As most of these functions and operators are randomly generated, we will denote by $R_1, R_2, \ldots$ the sequence of random numbers obtained using MATLAB's random number generator `rand` after setting the generator to its initial state. These numbers are uniformly distributed on the interval $(0, 1)$.

- We will make frequent use of a two-parameter family of functions defined on the interval $[0, 2\pi]$. First, we define

$$f^0_{j,k}(x) = \mathrm{Re}\left\{ \sum_{|\omega|<N/2} \hat{f}_j (1+|\omega|)^{-(k+1)} e^{i\omega x} \right\}, \tag{58}$$

for $j, k = 0, 1, \ldots,$ where

$$\hat{f}_j(\omega) = R_{jN+2(\omega+N/2)-1} + iR_{jN+2(\omega+N/2)}. \tag{59}$$

The parameter j indicates how many functions have been generated in this fashion since setting MATLAB's random number generator to its initial state, and the parameter k indicates how smooth the function is. Figure 1 shows selected functions from this collection.

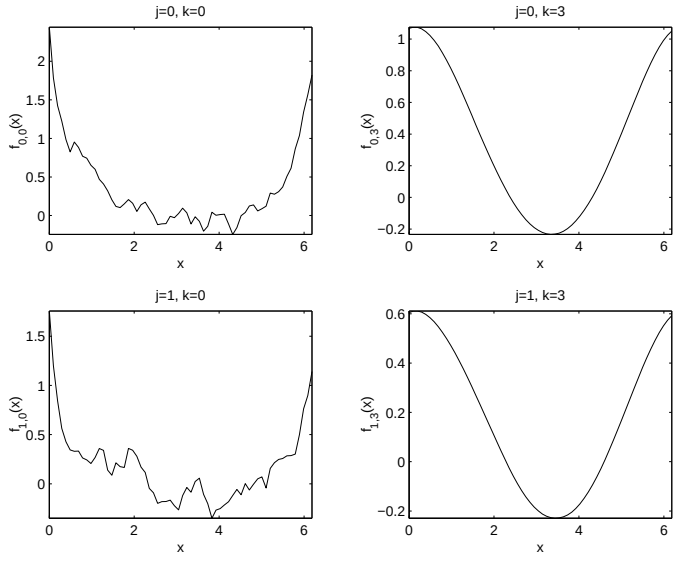

Figure 1: Functions from the collection $f_{j,k}(x)$, for selected values of j and k.

In many cases, it is necessary to ensure that a function is positive or negative, so we define the translation operators E^+ and E^- by

$$E^+ f(x) = f(x) - \min_{x\in[0,2\pi]} f(x) + 1, \tag{60}$$

$$E^- f(x) = f(x) - \max_{x\in[0,2\pi]} f(x) - 1. \tag{61}$$

- We define a similar two-parameter family of functions defined on the rectangle $[0, 2\pi] \times [0, 2\pi]$:

$$g_{j,k}(x, y) = \mathrm{Re}\left\{ \sum_{|\omega|,|\xi|<N/2} \hat{g}_j(\omega, \xi) e^{i(\omega x+\xi y)} \right\}, \tag{62}$$

where j and k are nonnegative integers, and

$$\begin{aligned} \hat{g}_j(\omega,\xi) \;=\; & (1+|\omega|)^{-(k+1)}(1+|\xi|)^{-(k+1)} \times \\ & \left\{ R_{jN^2+2[N(\omega+N/2-1)+(\xi+N/2)]-1} + \right. \\ & \left. iR_{jN^2+2[N(\omega+N/2-1)+(\xi+N/2)]} \right\}. \end{aligned} \tag{63}$$

Figure 2 shows selected functions from this collection.

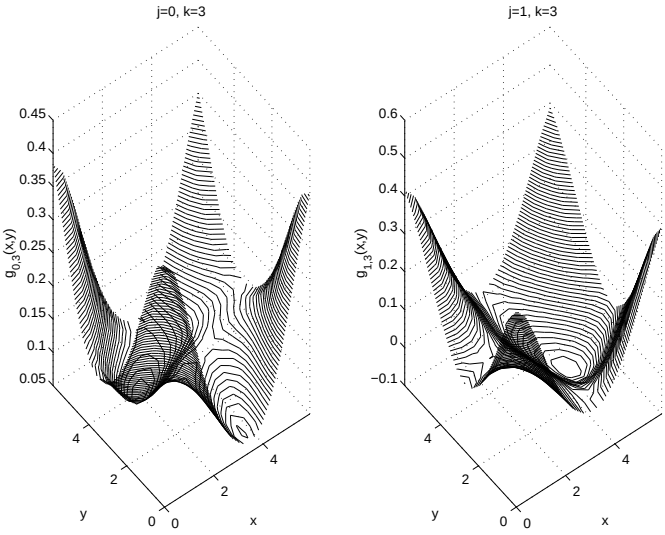

Figure 2: Functions from the collection $g_{j,k}(x, y)$, for selected values of j and k.

In all experiments, unless otherwise noted, solutions $u^{(j)}(x, t)$ are computed using time steps $\Delta t = 10 \cdot 2^{-j}$, for $j = 0, \ldots, 6$. The error estimates are obtained by computing $\|\tilde{u}^{(j)}(\cdot, 10) - \tilde{u}^{(6)}(\cdot, 10)\|_{L^2} / \|\tilde{u}^{(6)}(\cdot, 1)\|_{L^2}$ for $j = 0, \ldots, 5$. This method of estimating error assumes that $u^{(6)}(x, t)$ is a sufficiently accurate approximation to the exact solution, but this has proven in practice to be a valid assumption by comparing $u^{(6)}$ against approximate solutions computed using established methods, and by comparing $u^{(6)}$ against solutions obtained using various methods with smaller time steps.

In [21], and in Section 6.3 of this paper, errors measured by comparison to exact solutions of problems with source terms further validate the convergence behavior. It should be noted that we are not seeking a sharp estimate of the error, but rather an indication of the rate of convergence, and for this goal, using $u^{(6)}$ as an approximation to the exact solution is sufficient.

6.2 Smooth Coefficients

We now show the accuracy of our approach in one and two space dimensions. We solve the wave equation in which the spatial operator has a constant leading coefficient but a variable zero-order coefficient, constructed from randomly generated Fourier coefficients as discussed in Section 6.1. Specifically, we solve the following problems:

- $$\frac{\partial^2 u}{\partial t^2}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) - E^- f_{0,3}(x)u(x,t) = 0, \quad (64)$$
$$u(x,0) = E^+ f_{1,3}(x), \quad u_t(x,0) = E^+ f_{2,3}(x), \quad (65)$$
$$u(x,t) = u(x+2\pi,t). \quad (66)$$

- $$\frac{\partial u^2}{\partial t^2}(x,y,t) - \Delta u(x,y,t) - E^- g_{0,3}(x,y)u(x,t) = 0, \quad (67)$$
$$u(x,y,0) = E^+ g_{1,3}(x,y), \quad u_t(x,y,0) = E^+ g_{2,3}, \quad (68)$$
$$u(x,y,t) = u(x+2\pi,t) = u(x,y+2\pi,t). \quad (69)$$

  In [21], it is shown that the methods for efficiently computing recursion coefficients generalizes in a straightforward manner to higher spatial dimensions.

The results are shown in Figures 3 and 4, and Tables 1 and 2, and compared to those obtained using the original KSS method. In the 2-D case, the variable coefficient of the PDE is smoothed to a greater extent than in the 1-D case, because the prescribed decay rate of the Fourier coefficients is imposed in both the x- and y-directions. This results in greater accuracy in the 2-D case, which is consistent with the result proved in [20] that the local truncation error varies linearly with the variation in the coefficients. We see that significantly greater accuracy is obtained with block KSS methods, especially in two space dimensions. Both methods exhibit sixth-order accuracy in time, consistent with the theoretical results proved earlier concerning local error.

In an attempt to understand why the block KSS method is significantly more accurate, we examine the approximate solution operator for the simple case of $K = 1$. As shown in (41), the original 1-node KSS method is equivalent to the simple splitting involving the averaged-coefficient operator. On the other hand, the 1-node block KSS method is not equivalent to such a splitting, because every node and weight of the quadrature rule used to compute each Fourier component is influenced by the solution from the previous time step.

Furthermore, an examination of the nodes for both methods reveals that for the original KSS method, all of the

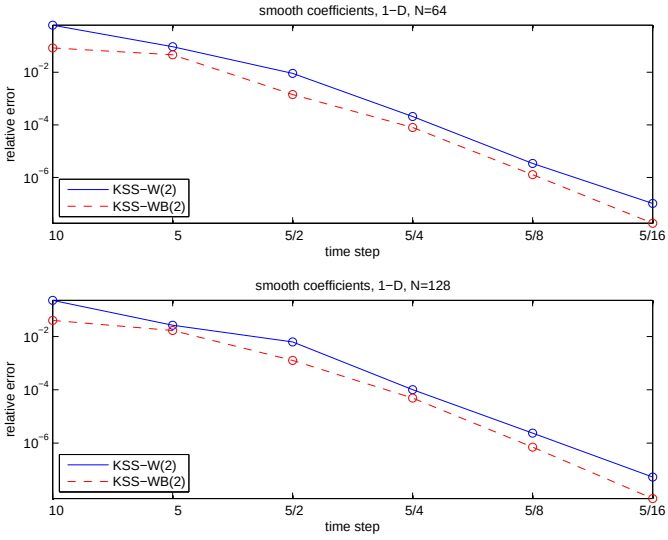

Figure 3: (a) Top plot: Estimates of relative error in the approximate solution of (64), (65), (66) at $T = 10$. Solutions are computed using the original 2-node KSS method (solid curve), and a 2-node block KSS method (dashed curve), both with $N = 128$ grid points. (b) Bottom plot: Estimates of relative error in the approximate solution of the same problem, using the same methods, with $N = 256$ grid points. In both cases, both methods use time steps $\Delta t = 2^{-j}$, $j = 0, \ldots, 5$.

nodes used to compute $[\hat{\mathbf{u}}^{n+1}]_\omega$ tend to be clustered around $\hat{\mathbf{e}}_\omega^H L_N \hat{\mathbf{e}}_\omega$, whereas with the block KSS method, half of the nodes are clustered near this value, and the other half are clustered near $[\mathbf{u}_\omega^n]^H L_N \mathbf{u}_\omega^n$, so the previous solution plays a much greater role in the construction of the quadrature rules.

A similar effect was achieved with the original KSS method by using a Gauss-Radau rule in which the prescribed node was an approximation of the smallest eigenvalue of L_N, and while this significantly improved accuracy for parabolic problems, as shown in [20], the solution-dependent approach used by the block method makes more sense, especially if the initial data happens to be oscillatory.

6.3 Oscillatory Coefficients

We now apply a 2-node block Krylov subspace spectral method to the problem

$$\frac{\partial^2 u}{\partial t^2}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) - E^- f_{0,0}(x)u(x,t) = 0, \quad (70)$$
$$u(x,0) = E^+ f_{1,3}(x), \quad u_t(x,0) = E^+ f_{2,3}(x), \quad (71)$$
$$u(x,t) = u(x+2\pi,t). \quad (72)$$

In this problem, the zero-order coefficient exhibits high-frequency oscillations, as shown in the top left plot of Figure 1.

Table 1: Estimates of relative error and temporal order of convergence in the approximate solution of (64), (65), (66) at $T = 10$, using 2-node original and block Krylov subspace spectral methods. Error is the relative difference, in the 2-norm sense, between approximate solutions and a solution computed using a smaller time step, since no exact solution is available. N denotes the number of grid points and Δt denotes the time step used.

N	Δt	KSS-W(2)	KSS-WB(2)
64	10	0.623	0.0844
	5	0.0942	0.0466
	2.5	0.00912	0.00142
	1.25	0.000208	7.96e-005
	0.625	3.41e-006	1.27e-006
	0.3125	1.02e-007	1.81e-008
128	10	0.23	0.0404
	5	0.0267	0.0169
	2.5	0.0063	0.00127
	1.25	0.000102	4.9e-005
	0.625	2.37e-006	7.04e-007
	0.3125	5.29e-008	8.29e-009

Table 3 lists the relative errors for various time steps and grid sizes. The errors are obtained by comparing the approximate solution to the known exact solution in the 2-norm sense. While the performance of both methods are at least comparable in the case of smooth coefficients, KSS-W(2) exhibits severe instability for larger time steps, eventually recovering to achieve fourth-order convergence for smaller time steps. KSS-WB(2), on the other hand, while not as accurate as with smooth coefficients, still demonstrates fifth-order accuracy in time, and is stable even for $\Delta t = 10$. Unfortunately, some accuracy is lost as the number of grid points increases. The error estimates are also plotted in Figure 5.

6.4 Variable Leading Coefficient and Source Term

We will not prove stability for the 2-node case in this paper. Instead, we will provide numerical evidence of stability and a contrast with another high-order explicit method. In particular, we use the method KSS-W(2) to solve a second-order wave equation featuring a variable leading coefficient and a source term. First, we note that if $p(x,t)$ and $u(x,t)$ are solutions of the system of first-order wave equations

$$\begin{bmatrix} p \\ u \end{bmatrix}_t = \begin{bmatrix} 0 & a(x) \\ b(x) & 0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x + \begin{bmatrix} F \\ G \end{bmatrix}, \quad t \geq 0 \quad (73)$$

with source terms $F(x,t)$ and $G(x,t)$, then $u(x,t)$ also satisfies the second-order wave equation

$$\frac{\partial^2 u}{\partial t^2} = a(x)b(x)\frac{\partial^2 u}{\partial x^2} + a'(x)b(x)\frac{\partial u}{\partial x} + bF_x + G \quad (74)$$

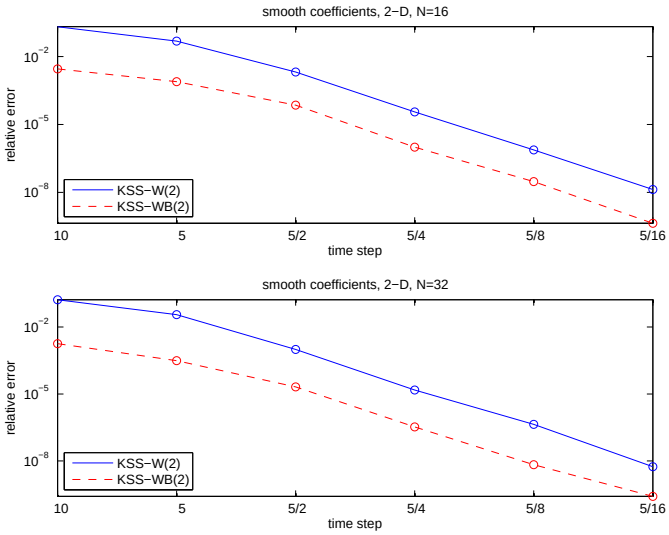

Figure 4: (a) Top plot: Estimates of relative error in the approximate solution of (67), (68), (69) at $T = 10$. Solutions are computed using the original 2-node KSS method (solid curve), and a 2-node block KSS method (dashed curve), both with $N = 16$ grid points per dimension. (b) Bottom plot: Estimates of relative error in the approximate solution of the same problem, using the same methods, with $N = 32$ grid points per dimension. In both cases, both methods use time steps $\Delta t = 2^{-j}$, $j = 0, \ldots, 5$.

with the source term $b(x)F_x(x,t) + G(x,t)$. In [13], a time-compact fourth-order finite-difference scheme is applied to a problem of the form (73), with

$$\begin{aligned} F(x,t) &= (a(x) - \alpha^2)\sin(x - \alpha t), \\ G(x,t) &= \alpha(1 - b(x))\sin(x - \alpha t), \\ a(x) &= 1 + 0.1\sin x, \\ b(x) &= 1, \end{aligned}$$

which has the exact solutions

$$\begin{aligned} p(x,t) &= -\alpha\cos(x - \alpha t), \\ u(x,t) &= \cos(x - \alpha t). \end{aligned}$$

We convert this problem to the form (74) and solve it with initial data

$$u(x,0) = \cos x, \quad (75)$$

$$u_t(x,0) = \sin x. \quad (76)$$

The results of applying both methods to this problem are shown in Figure 6, for the case $\alpha = 1$. Due to the smoothness of the coefficients, the spatial discretization error in the KSS methods is dominated by the temporal error, resulting in sixth-order accuracy in time for KSS-W(2), and seventh-order accuracy for KSS-WB(2).

Table 2: Estimates of relative error and temporal order of convergence in the approximate solution of (67), (68), (69) at $T = 10$, using 2-node original and block Krylov subspace spectral methods. Error is the relative difference, in the 2-norm sense, between approximate solutions and a solution computed using a smaller time step. N denotes the number of grid points per dimension, and Δt denotes the time step used.

N	Δt	KSS-W(2)	KSS-WB(2)
16	10	0.207	0.00283
	5	0.0479	0.000778
	2.5	0.00207	7.12e-005
	1.25	3.6e-005	1e-006
	0.625	7.54e-007	2.99e-008
	0.3125	1.35e-008	4.37e-010
32	10	0.168	0.0018
	5	0.0357	0.000311
	2.5	0.000993	2.08e-005
	1.25	1.51e-005	3.34e-007
	0.625	4.32e-007	6.76e-009
	0.3125	5.49e-009	2.58e-010

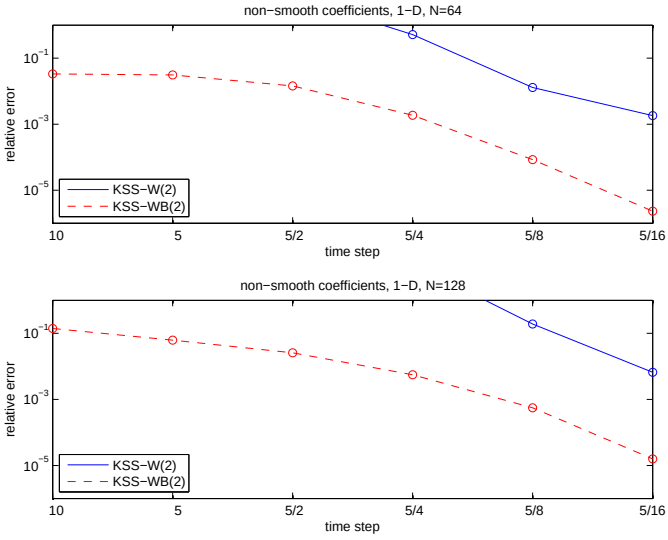

Figure 5: (a) Top plot: Estimates of relative error in the approximate solution of (70), (71), (72) at $T = 10$. Solutions are computed using the original 2-node KSS method (solid curve), and a 2-node block KSS method (dashed curve). Both methods use $N = 128$ grid points. (b) Bottom plot: Estimates of relative error in the approximate solution of the same problem at $T = 10$. Solutions are computed using the same methods, with $N = 256$ grid points. In both cases, both methods use time steps $\Delta t = 2^{-j}$, $j = 0, \ldots, 5$.

The source term is handled by applying Duhamel's principle, with a 4-node Gaussian rule over each time step, as first described in [21]. This has the effect of reducing the average time step to $\Delta t/5$ (in general, $\Delta 5/(M+1)$ for an M-node Gaussian rule over each time step). For a more informative comparison, we therefore use this reduced average time step in reporting the results for KSS methods.

Table 4 illustrates the differences in stability between the two methods. For the fourth-order finite-difference scheme from [13], the greatest accuracy is achieved for $c_{\max}\Delta t/\Delta x$ close to the CFL limit of 1, where $c_{\max} = \max_x \sqrt{a(x)b(x)}$. However, for KSS-W(2) and KSS-WB(2), this limit can be greatly exceeded and reasonable accuracy can still be achieved. In fact, while the results reported here were obtained using $N = 64$ grid points, nearly identical results are also obtained from substantial increase in the number of grid points, such as $N = 256$, *with the same time steps.*

7 Discussion

In this concluding section, we consider various generalizations of the problems and methods considered in this paper.

7.1 Higher Space Dimension

In [21], it is demonstrated how to compute the recursion coefficients α_j and β_j for operators of the form $Lu = -p\Delta u + q(x, y)u$, and the expressions are straightforward generalizations of the expressions for the one-dimensional case. It is therefore reasonable to suggest that for operators of this form, the consistency and stability results given here for the one-dimensional case generalize to higher dimensions. This will be investigated in the near future.

7.2 Discontinuous Coefficients and Data

As shown in [21] and again in the previous section of this paper, rough or discontinuous coefficients reduce the accuracy of KSS methods, because they introduce significant spatial discretization error into the computation of recursion coefficients.

Furthermore, for the stability result reported in this paper, the assumption that the coefficients are bandlimited is crucial. It can be weakened to some extent and replaced by an appropriate assumption about the regularity of the coefficients, but for simplicity that was not pursued here.

Regardless, this result does not apply to problems in which the coefficients are discontinuous, because Gibbs' phenomenon prevents their discrete Fourier transforms from being uniformly bounded for all N. Similar difficulties arise for hyperbolic problems when the initial data is not smooth, whereas this was not a concern in the parabolic case.

Table 3: Estimates of relative error and temporal order of convergence in the approximate solution of (70), (71), (72) using 2-node original and block Krylov subspace spectral methods. Error is the relative difference, in the 2-norm sense, between the exact solution $u(x,t) = \cos(x-t)$ and the computed solution at $T = 10$. N denotes the number of grid points and Δt denotes the time step used.

N	Δt	KSS-W(2)	KSS-WB(2)
64	10	4.64	0.0334
	5	490	0.031
	2.5	7.39	0.0144
	1.25	0.517	0.00187
	0.625	0.0129	8.44e-005
	0.3125	0.00181	2.32e-006
128	10	17.7	0.14
	5	4.04e+058	0.0622
	2.5	4.56e+009	0.0259
	1.25	12.7	0.00557
	0.625	0.192	0.000558
	0.3125	0.00662	1.58e-005

Table 4: Relative error in the solution of (74) with the time-compact fourth-order finite difference scheme from [13], for various values of N, and the KSS methods KSS-W(2) and KSS-WB(2).

T	Δt	FD	KSS-W(2)	KSS-WB(2)
8π	0.62832	0.0024	0.00377	0.00318
	0.31416	0.00014	6.26e-005	2.67e-005
	0.15708	8.8e-006	2.09e-007	6.52e-010
56π	0.62832	0.014	0.0195	0.0211
	0.31416	0.0009	0.000423	0.00018
	0.15708	5.6e-005	1.46e-006	4.5e-009

Ongoing work, described in [19], involves the use of the polar decomposition (42), to alleviate difficulties caused by such coefficients and initial data; future work will explore possible combinations of this approach with block KSS methods in order to generalize the superior accuracy of the block approach to these more difficult problems.

7.3 Other Boundary Conditions and Maxwell's Equations

While we only considered periodic boundary conditions in this paper, KSS methods for the wave equation can be used with other boundary conditions. Dirichlet boundary conditions were used in [12]. Inhomogeneous Dirichlet or Neumann boundary conditions can be handled by the standard technique of subtracting from the solution a function that satisfies the boundary conditions, and solving a modified problem with an appropriate source term. Future work will explore the adaptation of KSS methods for Maxwell's equations, including the use of boundary conditions such as perfectly matched layers (PML), introduced by Berenger in [3], which can be implemented by modifying the symbol of L during the computation of the recursion coefficients, although they must be implemented carefully in view of the recent analysis of PML for variable-coefficient problems in [22].

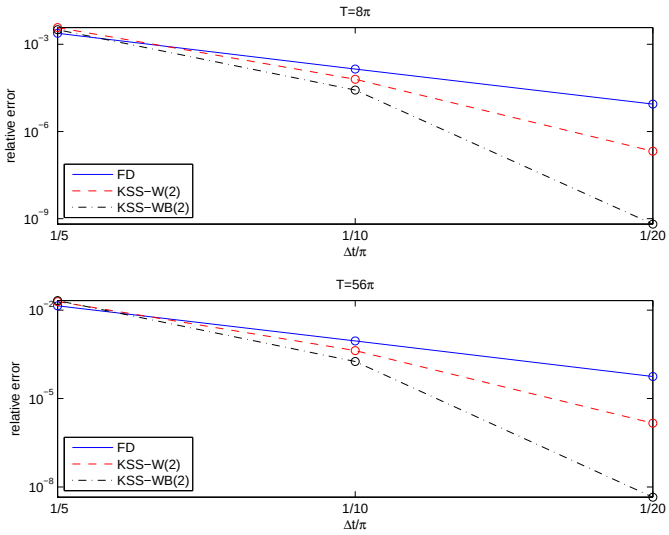

Figure 6: Estimates of relative error in the approximate solution of problem (74), (75), (76) with periodic boundary conditions, at $t = 8\pi$ (top plot) and $t = 56\pi$ (bottom plot), computed with the time-compact fourth-order finite-difference scheme from [13] (solid curve), a non-block KSS method (dashed curve), and a block KSS method (dotted-dashed curve). In the finite-difference scheme, $\lambda = \Delta t/\Delta x = 0.99$, and in both KSS methods, 2-point Gaussian quadrature rules are used, and $N = 64$ grid points.

7.4 Summary

We have demonstrated that for hyperbolic variable-coefficient PDE, block KSS methods are capable of computing Fourier components of the solution with greater accuracy than the original KSS methods, and they possess similar stability properties in the case of smooth coefficients, but are much more stable for problems with oscillatory coefficients. By pairing the solution from the previous time step with each trial function in a block and applying the Lanczos algorithm to them together, we obtain a block Gaussian quadrature rule that is better suited to approximating a bilinear form involving both functions than the approach of perturbing Krylov subspaces in the direction of the solution.

References

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

[2] Basu, S., Bose, N. K.: Matrix Stieltjes series and network models. *SIAM J. Math. Anal.* **14**(2) (1983) 209-222.

[3] Berenger, J,: A perfectly matched layer for the absorption of electromagnetic waves. *J. Comp. Phys.* **114** (1994) 185-200.

[4] 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.

[5] Golub, G. H.: Some modified matrix eigenvalue problems. *SIAM Review* **15** (1973) 318-334.

[6] Golub, G. H.: Bounds for matrix moments. *Rocky Mnt. J. of Math.* **4** (1974) 207-211.

[7] 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)

[8] Golub, G. H., Gutknecht, M. H.: Modified Moments for Indefinite Weight Functions. *Numerische Mathematik* **57** (1989) 607-624.

[9] Golub G. H., van Loan, C. F.: *Matrix Computations, 3rd Ed.* Johns Hopkins University Press (1996)

[10] Golub, G. H., Underwood, R.: The block Lanczos method for computing eigenvalues. *Mathematical Software III*, J. Rice Ed., (1977) 361-377.

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

[12] Guidotti, P., Lambers, J. V., Sølna, K.: Analysis of 1-D Wave Propagation in Inhomogeneous Media. *Numerical Functional Analysis and Optimization* **27** (2006) 25-55.

[13] Gustafsson, B., Mossberg, E.: Time Compact High Order Difference Methods for Wave Propagation. *SIAM J. Sci. Comput.* **26** (2004) 259-271.

[14] Hochbruck, M., Lubich, C.: On Krylov Subspace Approximations to the Matrix Exponential Operator. *SIAM Journal of Numerical Analysis* **34** (1996) 1911-1925.

[15] Lambers, J. V.: Derivation of High-Order Spectral Methods for Time-dependent PDE using Modified Moments. *Electronic Transactions on Numerical Analysis* **28** (2008) 114-135.

[16] Lambers, J. V.: Enhancement of Krylov Subspace Spectral Methods by Block Lanczos Iteration. *Electronic Transactions on Numerical Analysis* **31** (2008) in press.

[17] 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.

[18] Lambers, J. V.: Krylov Subspace Methods for Variable-Coefficient Initial-Boundary Value Problems. Ph.D. Thesis, Stanford University, SCCM Program, 2003.

[19] Lambers, J. V.: Krylov Subspace Spectral Methods for the Time-Dependent Schrödinger Equation with Non-Smooth Potentials. Submitted.

[20] Lambers, J. V.: Krylov Subspace Spectral Methods for Variable-Coefficient Initial-Boundary Value Problems. *Electronic Transactions on Numerical Analysis* **20** (2005) 212-234.

[21] Lambers, J. V.: Practical Implementation of Krylov Subspace Spectral Methods. *Journal of Scientific Computing* **32** (2007) 449-476.

[22] Oskooi, A. F., Zhang, L., Avniel, Y., and Johnson, S. G.: The failure of perfectly matched layers, and towards their redemption by adiabatic absorbers. *Opt. Expr.* **16** (2008) 11376-11392.

[23] Shampine, L. F., Reichelt, M. W.: The MATLAB ODE suite. *SIAM Journal of Scientific Computing* **18** (1997) 1-22.

[24] Stoer, J., Burlisch, R.: *Introduction to numerical analysis, 2nd Ed.* Springer Verlag (1983)