Electronic Transactions on Numerical Analysis.
Volume 28, pp. 114-135, 2008.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

DERIVATION OF HIGH-ORDER SPECTRAL METHODS FOR TIME-DEPENDENT PDE USING MODIFIED MOMENTS*

JAMES V. LAMBERS†

In memory of Gene Golub

Abstract. This paper presents a reformulation of Krylov Subspace Spectral (KSS) Methods, which build on Gene Golub's many contributions pertaining to moments and Gaussian quadrature, to produce high-order accurate approximate solutions to variable-coefficient time-dependent PDE. This reformulation serves two useful purposes. First, it more clearly illustrates the distinction between KSS methods and existing Krylov subspace methods for solving stiff systems of ODE arising from discretizions of PDE. KSS methods rely on perturbations of Krylov subspaces in the direction of the data, and therefore rely on directional derivatives of nodes and weights of Gaussian quadrature rules. Second, because these directional derivatives allow KSS methods to be described in terms of operator splittings, they facilitate stability analysis. It will be shown that under reasonable assumptions on the coefficients of the problem, certain KSS methods are unconditionally stable. This paper also discusses preconditioning similarity transformations that allow more general problems to benefit from this property.

Key words. spectral methods, Gaussian quadrature, variable-coefficient, Lanczos method, stability, heat equation, wave equation

AMS subject classifications. 65M12, 65M70, 65D32

1. Introduction.

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

$$u_t + Lu = 0 \quad \text{on } (0, 2\pi) \times (0, \infty), \tag{1.1}$$

$$u(x, 0) = f(x), \quad 0 < x < 2\pi, \tag{1.2}$$

with periodic boundary conditions

$$u(0, t) = u(2\pi, t), \quad t > 0. \tag{1.3}$$

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

$$Lu = -pu_{xx} + qu, \tag{1.4}$$

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

In [18], [19], a class of methods, called Krylov subspace spectral (KSS) methods, was introduced for the purpose of solving time-dependent, variable-coefficient problems such as this one. These methods are based on the application of techniques developed by Golub and Meurant in [9], 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, [15]) for stiff systems of ODE. However, the essential question of the stability of KSS methods has yet to be addressed. This paper represents a first step in this direction.

*Received April 5, 2007. Accepted for publication October 10, 2007. Recommended by M. Gutknecht.
†Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305-2220 (lambers@stanford.edu).

ETNA
Kent State University
etna@mcs.kent.edu

Section 2 reviews the main properties of KSS methods, including algorithmic details and results concerning local accuracy. The main idea behind these methods is that each Fourier component of the solution is obtained from a *perturbation* of a *frequency-dependent* Krylov subspace in the direction of the initial data, instead of a single Krylov subspace generated from the data. It follows that KSS methods can be reformulated in terms of directional derivatives of moments. This leads to a new algorithm that represents the limit of a KSS method as the size of the perturbation approaches zero, thus avoiding the cancellation and parameter-tuning that is required by the original algorithm. This new algorithm is presented in section 3. Compared to the original algorithm, the new one lends itself more readily to stability analysis, which is carried out for the simplest KSS methods in sections 4 and 5. In section 6, this analysis is repeated for the application of KSS methods to the second-order wave equation, which was introduced in [12]. Section 7 presents homogenizing transformations that can be applied to more general variable-coefficient second-order differential operators, including a new transformation that can be used to homogenize a second-order operator with smoothly varying coefficients, up to an operator of negative order. These transformations allow problems featuring these more general operators to be solved using KSS methods with the same accuracy and stability as the simpler problem presented in the preceding sections. In section 8, various generalizations are discussed.

2. Krylov subspace spectral methods. We begin with a review of the main aspects of KSS methods. Let $S(t) = \exp[-Lt]$ represent the exact solution operator of the problem (1.1), (1.2), (1.3), and let $\langle \cdot, \cdot \rangle$ denote the standard inner product of functions defined on $[0, 2\pi]$,

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

Krylov subspace spectral methods, introduced in [18], [19], 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 $\tilde{u}(x, t_n)$,

$$\hat{u}(\omega, t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, S(\Delta t)\tilde{u}(x, t_n) \right\rangle. \tag{2.1}$$

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 (2.1) can be approximated by a bilinear form

$$\hat{u}(\omega, t_{n+1}) \approx \hat{\mathbf{e}}_\omega^H S_N(\Delta t) \mathbf{u}(t_n), \tag{2.2}$$

where

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

and

$$S_N(t) = \exp[-L_N t], \quad [L_N]_{jk} = -p[D_N^2]_{jk} + q(j\Delta x)\,, \tag{2.3}$$

where D_N is a discretization of the differentiation operator that is defined on the space of grid functions. Our goal is to approximate (2.2) by computing an approximation to

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

In [9], Golub and Meurant describe a method for computing quantities of the form

$$\mathbf{u}^T f(A)\mathbf{v}, \tag{2.4}$$

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 (2.3), $f(\lambda) = \exp(-\lambda t)$ for some t, and the vectors $\mathbf{u}$ and $\mathbf{v}$ are derived from $\hat{\mathbf{e}}_\omega$ and $\mathbf{u}(t_n)$.

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

$$b = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N = a > 0,$$

and corresponding orthogonal eigenvectors $\mathbf{q}_j$, $j = 1, \ldots, N$. Therefore, the quantity (2.4) can be rewritten as

$$\mathbf{u}^T f(A)\mathbf{v} = \sum_{\ell=1}^{N} f(\lambda_\ell)\mathbf{u}^T \mathbf{q}_j \mathbf{q}_j^T \mathbf{v}.$$

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

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

If this measure is positive and increasing, then the quantity (2.4) can be viewed as a Riemann-Stieltjes integral

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

As discussed in [3], [6], [7], [9], 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

$$I[f] = \sum_{j=1}^{K} w_j f(t_j) + \sum_{j=1}^{M} v_j f(z_j) + R[f], \tag{2.5}$$

where the nodes t_j, $j = 1, \ldots, K$, and z_j, $j = 1, \ldots, M$, as well as the weights w_j, $j = 1, \ldots, K$, and v_j, $j = 1, \ldots, M$, 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 [2] for details. Therefore, it is best to handle this case by rewriting (2.4) using decompositions such as

$$\mathbf{u}^T f(A)\mathbf{v} = \frac{1}{\delta}[\mathbf{u}^T f(A)(\mathbf{u} + \delta\mathbf{v}) - \mathbf{u}^T f(A)\mathbf{u}], \tag{2.6}$$

where δ is a small constant. Guidelines for choosing an appropriate value for δ can be found in [19, Section 2.2].

Employing these quadrature rules yields the following basic process (for details see [18], [19]) for computing the Fourier coefficients of $\mathbf{u}(t_{n+1})$ from $\mathbf{u}(t_n)$. It is assumed that when the Lanczos algorithm (symmetric or unsymmetric) is employed, $M + K$ iterations are performed to obtain the $M + K$ quadrature nodes and weights.

for $\omega = -N/2 + 1, \dots, N/2$
  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 (2.6), and the subscript ω is used to indicate that a different value may be used for each wave number $\omega = -N/2 + 1, \dots, N/2$. Also, in the presentation of this algorithm in [19], a polar decomposition is used instead of (2.6), 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 2.1. *Let L be a self-adjoint m-th order positive definite differential operator on $C_p([0, 2\pi])$ with coefficients in $BL_N([0, 2\pi])$. Let $f \in BL_N([0, 2\pi])$, and let $M = 0$. Then the preceding algorithm, applied to the problem (1.1), (1.2), (1.3), is consistent; i.e.*

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

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

Proof. See [19, Lemma 2.1, Theorem 2.4]. ∎

Using results in [9] regarding the error term $R[f]$ in (2.5), it can be shown that if M prescribed nodes are used in addition to the K free nodes, then the local truncation error is $O(\Delta t^{2K+M})$. As shown in [19], significantly greater accuracy can be achieved for some problems by using a Gauss-Radau rule with one prescribed node that approximates the smallest eigenvalue of L. Also, it should be noted that in [12], a variation of Krylov subspace spectral methods is applied to variable-coefficient second-order wave equations, achieving $O(\Delta t^{4K+2M})$ accuracy.

For convenience, we denote by KSS(K) a KSS method, applied to the problem (1.1), that uses a K-node Gaussian rule for each Fourier component. If a $(K + 1)$-node Gauss-Radau rule is used instead, with K free nodes and one prescribed node approximating the smallest eigenvalue of L, then the resulting KSS method is denoted by KSS-R(K). Finally, KSS-W(K) and KSS-WR(K) refer to KSS methods applied to the second-order wave equation, using a K-node Gaussian rule and a $(K + 1)$-node Gauss-Radau rule, respectively.

The preceding result can be compared to the accuracy achieved by an algorithm described by Hochbruck and Lubich in [15] for computing $e^{A\Delta t}\mathbf{v}$ for a given matrix A and vector $\mathbf{v}$ using the unsymmetric Lanczos algorithm. As discussed in [15], 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 (1.1), (1.2), (1.3), 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 [15] 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 [19], 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.

In [19], the benefit of using component-specific Krylov subspace approximations was illustrated. A problem of the form (1.1), (1.2), (1.3) was solved using the following methods:

- A two-stage, third-order scheme described by Hochbruck and Lubich in [15] for solving systems of the form $\mathbf{y}' = A\mathbf{y} + \mathbf{b}$, where, in this case, $\mathbf{b} = 0$ and A is an $N \times N$ matrix that discretizes the operator L_3. The scheme involves multiplication of vectors by $\varphi(\gamma h A)$, where γ is a parameter (chosen to be $\frac{1}{2}$), h is the step size, and $\varphi(z) = (e^z - 1)/z$. The computation of $\varphi(\gamma h A)\mathbf{v}$, for a given vector $\mathbf{v}$, is accomplished by applying the Lanczos iteration to A with initial vector $\mathbf{v}$ to obtain an approximation to $\varphi(\gamma h A)\mathbf{v}$ that belongs to the m-dimensional Krylov subspace $\mathcal{K}(A, \mathbf{v}, m) = \text{span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{m-1}\mathbf{v}\}$.
- KSS-R(2), with 2 nodes determined by Gaussian quadrature and one additional prescribed node. The prescribed node is obtained by estimating the smallest eigenvalue of L using the symmetric Lanczos algorithm.

We chose $m = 2$ in the first method, so that both algorithms performed the same number of matrix-vector multiplications during each time step. As N increased, there was virtually no impact on the accuracy of KSS-R(2). On the other hand, this increase, which resulted in a stiffer system, reduced the time step at which the method from [15] began to show reasonable accuracy.

A result like this suggests that KSS methods are relatively insensitive to the spatial and temporal mesh sizes, in comparison to other explicit methods. It is natural to consider whether they may be unconditionally stable, and if so, under what conditions. The following sections provide an answer to this question.

3. Reformulation. From the algorithm given in the preceding section, we see that each Fourier component $[\hat{\mathbf{u}}^{n+1}]_\omega$ approximates the derivative

$$\frac{d}{d\delta_\omega}\left[\hat{\mathbf{e}}_\omega^H(\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n)\mathbf{e}_1^T \exp[-T_\omega(\delta_\omega)\Delta t]\mathbf{e}_1\right]\bigg|_{\delta_\omega=0},$$

where $T_\omega(\delta_\omega)$ is the tridiagonal matrix output by the unsymmetric Lanczos algorithm applied to the matrix L_N with starting vectors $\hat{\mathbf{e}}_\omega$ and $(\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n)$ (which reduces to the symmetric Lanczos algorithm for $\delta_\omega = 0$). In this section, we will compute these derivatives analytically. In the following sections, we will use these derivatives to examine the question of stability of Krylov subspace spectral methods.

ETNA
Kent State University
etna@mcs.kent.edu

3.1. Derivatives of the nodes and weights. For a given δ_ω, let $\lambda_{\omega,j}$, $j = 1, \ldots, K$, be the nodes of the K-point Gaussian rule obtained by applying the unsymmetric Lanczos algorithm to L_N with starting vectors $\hat{\mathbf{e}}_\omega$ and $(\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n)$. Let $w_{\omega,j}$, $j = 1, \ldots, K$, be the corresponding weights. Then, letting $\delta_\omega \to 0$, we obtain the following, assuming all required derivatives exist:

$$\begin{aligned}
[\hat{\mathbf{u}}^{n+1}]_\omega &= \hat{\mathbf{e}}_\omega^H \mathbf{u}^{n+1} \\
&= \frac{d}{d\delta_\omega} \left[\hat{\mathbf{e}}_\omega^H (\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n) \mathbf{e}_1^T \exp[-T_\omega(\delta_\omega)\Delta t] \mathbf{e}_1 \right] \Big|_{\delta_\omega = 0} \\
&= \frac{d}{d\delta_\omega} \left[\hat{\mathbf{e}}_\omega^H (\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n) \sum_{j=1}^K w_j e^{-\lambda_j \Delta t} \right] \Bigg|_{\delta_\omega = 0} \\
&= \hat{\mathbf{e}}_\omega^H \mathbf{u}^n \sum_{j=1}^K w_j e^{-\lambda_j \Delta t} + \sum_{j=1}^K w_j' e^{-\lambda_j \Delta t} - \Delta t \sum_{j=1}^K w_j \lambda_j' e^{-\lambda_j \Delta t}, \qquad (3.1)
\end{aligned}$$

where the $'$ denotes differentiation with respect to δ_ω, and evaluation of the derivative at $\delta_\omega = 0$. Equivalently, these derivatives are equal to the length of $\mathbf{u}^n$ times the directional derivatives of the nodes and weights, as functions defined on $\mathbb{R}^N$, in the direction of $\mathbf{u}^n$, and evaluated at the origin.

It should be noted that in the above expression for $[\hat{\mathbf{u}}^{n+1}]$, the nodes and weights depend on the wave number ω, but for convenience, whenever a fixed Fourier component is being discussed, the dependence of the nodes and weights on ω is not explicitly indicated.

From the Lanczos algorithm, $T_\omega(\delta_\omega)$ has the structure

$$T_\omega(\delta_\omega) = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{K-2} & \alpha_{K-1} & \beta_{K-1} \\ & & & \beta_{K-1} & \alpha_K \end{bmatrix},$$

where all entries are functions of δ_ω. Because the nodes and weights are obtained from the eigenvalues and eigenvectors of this matrix, it is desirable to use these relationships to develop efficient algorithms for computing the derivatives of the nodes and weights in terms of those of the recursion coefficients. We will first describe such algorithms, and then we will explain how the derivatives of the recursion coefficients can be computed.

The nodes are the eigenvalues of $T_\omega(\delta_\omega)$. Because $T_\omega(0)$ is Hermitian, it follows that there exists a unitary matrix Q_ω^0 such that

$$T_\omega(0) = Q_\omega^0 \Lambda_\omega(0) [Q_\omega^0]^H.$$

The eigenvalues of $T_\omega(0)$ are distinct; see [11]. Because the eigenvalues are continuous functions of the entries of the matrix, they continue to be distinct for δ_ω sufficiently small, and therefore $T_\omega(\delta)$ remains diagonalizable. It follows that we can write

$$T_\omega(\delta_\omega) = Q_\omega(\delta_\omega) \Lambda_\omega(\delta_\omega) Q_\omega(\delta_\omega)^{-1}, \qquad (3.2)$$

where $Q_\omega(0) = Q_\omega^0$. Differentiating (3.2) with respect to δ_ω and evaluating at $\delta_\omega = 0$ yields

$$\mathrm{diag}(\Lambda_\omega'(0)) = \mathrm{diag}\left(Q_\omega(0)^H T_\omega'(0) Q_\omega(0) \right),$$

ETNA
Kent State University
etna@mcs.kent.edu

since all other terms that arise from application of the product rule vanish on the diagonal. Therefore, for each ω, the derivatives of the nodes $\lambda_1, \ldots, \lambda_K$ are easily obtained by applying a similarity transformation to the matrix of the derivatives of the recursion coefficients, $T'_\omega(0)$, where the transformation involves a matrix, $Q_\omega(0)$, that must be computed anyway to obtain the weights.

To compute the derivatives of the weights, we consider the equation

$$(T_\omega(\delta_\omega) - \lambda_j I)\mathbf{w}_j(\delta_\omega) = \mathbf{0}, \quad j = 1, \ldots, K,$$

where $\mathbf{w}_j(\delta_\omega)$ is an eigenvector of $T_\omega(\delta_\omega)$ with eigenvalue λ_j, normalized to have unit 2-norm. First, we differentiate this equation with respect to δ_ω and evaluate at $\delta_\omega = 0$. Then, we delete the last equation and eliminate the last component of $\mathbf{w}_j(0)$ and $\mathbf{w}'_j(0)$ using the fact that $\mathbf{w}_j(0)$ must have unit 2-norm. The result is a $(K-1) \times (K-1)$ system where the matrix is the sum of a tridiagonal matrix and a rank-one update. This matrix is independent of the solution $\mathbf{u}^n$, while the right-hand side is not. After solving this simple system, as well as a similar one for the left eigenvector corresponding to λ_j, we can obtain the derivative of the weight w_j from the first components of the two solutions. It should be noted that although $T_\omega(0)$ is Hermitian, $T_\omega(\delta_\omega)$ is, in general, complex symmetric, which is why the system corresponding to the left eigenvector is necessary.

3.2. Derivatives of the recursion coefficients. Let A be a symmetric positive definite $n \times n$ matrix and let $\mathbf{r}_0$ be an n-vector. Suppose that we have already carried out the symmetric Lanczos iteration to obtain orthogonal vectors $\mathbf{r}_0, \ldots, \mathbf{r}_K$ and the Jacobi matrix

$$T_K = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{K-2} & \alpha_{K-1} & \beta_{K-1} \\ & & & \beta_{K-1} & \alpha_K \end{bmatrix}. \tag{3.3}$$

Now, we wish to compute the entries of the modified matrix

$$\hat{T}_K = \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_1 & & & \\ \hat{\beta}_1 & \hat{\alpha}_2 & \hat{\beta}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \hat{\beta}_{K-2} & \hat{\alpha}_{K-1} & \hat{\beta}_{K-1} \\ & & & \hat{\beta}_{K-1} & \hat{\alpha}_K \end{bmatrix} \tag{3.4}$$

that results from applying the unsymmetric Lanczos iteration with the same matrix A and the initial vectors $\mathbf{r}_0$ and $\mathbf{r}_0 + \mathbf{f}$, where $\mathbf{f}$ is a given perturbation. The following iteration, introduced in [20] and based on algorithms from [5], [8], and [22], produces these values.

Algorithm 3.1. Given the Jacobi matrix (3.3), the first $K+1$ unnormalized Lanczos vectors $\mathbf{r}_0, \ldots, \mathbf{r}_K$, and a vector $\mathbf{f}$, the following algorithm generates the modified tridiagonal matrix (3.4) that is produced by the unsymmetric Lanczos iteration with left initial vector $\mathbf{r}_0$ and right initial vector $\mathbf{r}_0 + \mathbf{f}$.

$\beta_{-1} = 0, \quad \mathbf{q}_{-1} = 0, \quad \mathbf{q}_0 = \mathbf{f}, \quad \hat{\beta}_0^2 = \beta_0^2 + \mathbf{r}_0^H \mathbf{q}_0,$
$s_0 = \frac{\beta_0}{\hat{\beta}_0^2}, \quad t_0 = \frac{\beta_0^2}{\hat{\beta}_0^2}, \quad d_0 = 0$

for $j = 1, \ldots, K$
 $\hat{\alpha}_j = \alpha_j + s_{j-1}\mathbf{r}_j^H\mathbf{q}_{j-1} + d_{j-1}\beta_{j-2}t_{j-1}^{-1/2}$
 if $j < K$ **then**
  $d_j = (d_{j-1}\beta_{j-2} + (\alpha_j - \hat{\alpha}_j)t_{j-1}^{1/2})/\hat{\beta}_{j-1}$
  $\mathbf{q}_j = (A - \hat{\alpha}_j I)\mathbf{q}_{j-1} - \hat{\beta}_{j-1}^2\mathbf{q}_{j-2}$
  $\hat{\beta}_j^2 = t_{j-1}\beta_j^2 + s_{j-1}\mathbf{r}_j^H\mathbf{q}_j$
  $s_j = \frac{\beta_j}{\hat{\beta}_j^2}s_{j-1}$
  $t_j = \frac{\beta_j^2}{\hat{\beta}_j^2}t_{j-1}$
 end
end

The correctness of this algorithm is proved in [20], where it was used to efficiently obtain the recursion coefficients needed to approximate $\hat{\mathbf{e}}_\omega^H S_N(\Delta t)(\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n)$ from those used to approximate $\hat{\mathbf{e}}_\omega^H S_N(\Delta t)\hat{\mathbf{e}}_\omega$. In [20], it was shown that with an efficient implementation of this algorithm in MATLAB, KSS methods are a viable option for solving parabolic problems when compared to MATLAB's built-in ODE solvers, even though the former are explicit and the latter are implicit.

Here, we use this algorithm for a different purpose. From the expressions for the entries of $\hat{T}_K$, the derivatives of the recursion coefficients α_j, $j = 1, \ldots, K$, and β_j, $j = 1, \ldots, K-1$, can be obtained by setting $\mathbf{r}_0 = \hat{\mathbf{e}}_\omega$ and $\mathbf{f} = \delta_\omega \mathbf{u}^n$. By differentiating the recurrence relations in Algorithm 3.1 with respect to δ_ω and evaluating at $\delta_\omega = 0$, we obtain the following new algorithm.

Algorithm 3.2. Let $T_K(\delta)$ be the tridiagonal matrix produced by the unsymmetric Lanczos iteration with left initial vector $\mathbf{r}_0$ and right initial vector $\mathbf{r}_0 + \delta\mathbf{f}$. Let $\mathbf{r}_0, \ldots, \mathbf{r}_K$ be the $K+1$ unnormalized Lanczos vectors associated with $T_K(0)$. Given T_K, as defined in (3.3), whose entries are those of $T_K(\delta)$ evaluated at $\delta = 0$, the following algorithm generates the tridiagonal matrix T_K' whose entries are the derivatives of the entries of $T_K(\delta)$ with respect to δ, and evaluated at $\delta = 0$.

$\beta_{-1} = 0, \quad \mathbf{q}_{-1} = 0, \quad \mathbf{q}_0 = \mathbf{f}, \quad [\beta_0^2]' = \mathbf{r}_0^H\mathbf{q}_0$
$s_0 = \frac{1}{\beta_0}, \quad s_0' = -\frac{[\beta_0^2]'}{\beta_0^3}, \quad t_0' = -\frac{[\beta_0^2]'}{\beta_0^2}, \quad d_0' = 0$
for $j = 1, \ldots, K$
 $\alpha_j' = s_{j-1}\mathbf{r}_j^H\mathbf{q}_{j-1} + d_{j-1}'\beta_{j-2}$
 if $j < K$ **then**
  $d_j' = (d_{j-1}'\beta_{j-2} - \alpha_j')/\beta_{j-1}$
  $\mathbf{q}_j = (A - \alpha_j I)\mathbf{q}_{j-1} - \beta_{j-1}^2\mathbf{q}_{j-2}$
  $[\beta_j^2]' = t_{j-1}'\beta_j^2 + s_{j-1}\mathbf{r}_j^H\mathbf{q}_j$
  $s_j = s_{j-1}/\beta_j$
  $s_j' = s_{j-1}'/\beta_j - \frac{[\beta_j^2]'}{\beta_j^3}s_{j-1}$
  $t_j' = t_{j-1}' - \frac{[\beta_j^2]'}{\beta_j^2}$
 end
end

Note that this iteration requires about the same computational effort as Algorithm 3.1.

ETNA
Kent State University
etna@mcs.kent.edu

3.3. Summary. The new formulation of KSS methods consists of the following process for computing each Fourier component $[\hat{\mathbf{u}}^{n+1}]_\omega$ from $\mathbf{u}^n$.

1. Compute the recursion coefficients, nodes and weights that were used to compute $u_1 \approx \hat{\mathbf{e}}_\omega^H S_N(\Delta t)\hat{\mathbf{e}}_\omega$ in the original formulation.
2. Compute the derivatives of the recursion coefficients used to obtain u_1, using Algorithm 3.2 with $\mathbf{r}_0 = \hat{\mathbf{e}}_\omega$ and $\mathbf{f} = \mathbf{u}^n$. Note that because the vectors $\mathbf{r}_j$, $j = 0, \dots, K$, are only involved through inner products, they do not need to be stored explicitly. Instead, the required inner products can be computed simultaneously for all ω using appropriate FFTs.
3. Compute the derivatives of the nodes and weights from those of the recursion coefficients, as described in section 3.1.
4. Compute $[\hat{\mathbf{u}}^{n+1}]_\omega$ as in (3.1).

Because this algorithm only requires computing the eigenvalues and eigenvectors of one $K \times K$ matrix, instead of two as in the original algorithm, both algorithms require a comparable amount of computational effort. However, unlike the original algorithm, the new one does not include a subtraction of nearly equal values for each Fourier component, and therefore exhibits greater numerical stability for small values of δ_ω.

4. The one-node case. When $K = 1$, we simply have $T_\omega(\delta_\omega) = \alpha_1(\delta_\omega)$, where

$$\alpha_1(\delta_\omega) = \hat{\mathbf{e}}_\omega^H L_N(\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n),$$

which yields

$$\alpha_1'(0) = \hat{\mathbf{e}}_\omega^H (L_N - \alpha_1 I)\mathbf{u}^n.$$

From $\lambda_1 = \alpha_1$ and $w_1 = 1$, we obtain

$$[\hat{\mathbf{u}}^{n+1}]_\omega = e^{-\alpha_1 \Delta t}\hat{\mathbf{e}}_\omega^H [1 - \Delta t(L_N - \alpha_1 I)]\mathbf{u}^n.$$

Because $\alpha_1 = p\omega^2 + \overline{q}$ and $L_N\hat{\mathbf{e}}_\omega = p\omega^2\hat{\mathbf{e}}_\omega + \operatorname{diag}(\mathbf{q})\hat{\mathbf{e}}_\omega$, it follows that

$$\mathbf{u}^{n+1} = e^{-C_N \Delta t} P_N[I - \Delta t \operatorname{diag}(\tilde{\mathbf{q}})]\mathbf{u}^n,$$

where $\tilde{\mathbf{q}} = \mathbf{q} - \overline{q}$ and $L = C + V$ is a splitting of L such that C is the constant-coefficient operator obtained by averaging the coefficients of L, and the variation of the coefficients is captured by V. The operator P_N is the orthogonal projection onto $BL_N([0, 2\pi])$. This simple form of the approximate solution operator yields the following result. For convenience, we denote by $\tilde{S}_N(\Delta t)$ the matrix such that $\mathbf{u}^{n+1} = \tilde{S}_N(\Delta t)\mathbf{u}^n$, for given N and Δt.

Theorem 4.1. *Let $q(x)$ in (1.4) belong to $BL_M([0, 2\pi])$ for a fixed integer M. Then, for the problem (1.1), (1.2), (1.3), KSS(1) is unconditionally stable. That is, given $T > 0$, there exists a constant C_T, independent of N and Δt, such that*

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

for $0 \le n\Delta t \le T$.

Proof. The matrix C_N has the diagonalization

$$C_N = F_N^{-1}\Lambda_N F_N,$$

where F_N is the matrix of the N-point discrete Fourier transform, and

$$\Lambda = \operatorname{diag}(p\omega^2 + \overline{q}), \quad \omega = -N/2 + 1, \dots, N/2.$$

It follows that $\|e^{-C_N\Delta t}\|_2 = e^{-\overline{q}\Delta t}$.

Because $q(x)$ is bounded, it follows that

$$\|P_N[I - \Delta t\,\mathrm{diag}(\tilde{\mathbf{q}})]\|_2 \leq 1 + \Delta t Q,$$

where $Q = \max_{0\leq x\leq 2\pi} q(x)$. We conclude that

$$\|\tilde{S}_N(\Delta t)\|_2 \leq e^{(Q-\overline{q})\Delta t},$$

from which the result follows with $C_T = e^{(Q-\overline{q})T}$. ☐

Now we can prove that the method converges. For convenience, we define the 2-norm of a function $u(x,t)$ to be the vector 2-norm of the restriction of $u(x,t)$ to the spatial grid:

$$\|u(\cdot,t)\|_2 = \left(\sum_{j=0}^{N-1} |u(j\Delta x, t)|^2\right)^{1/2}.$$

We also say that a method is convergent of order (m,n) if there exist constants C_t and C_x, independent of the time step Δt and grid spacing $\Delta x = 2\pi/N$, such that

$$\|u(\cdot,t) - \mathbf{u}(\cdot,t)\|_2 \leq C_t\Delta t^m + C_x\Delta x^n, \quad 0 \leq t \leq T.$$

Theorem 4.2. *Let $q(x)$ in (1.4) belong to $BL_M([0,2\pi])$ for some integer M. Then, for the problem (1.1), (1.2), (1.3), KSS(1) is convergent of order $(1,p)$, where the exact solution $u(x,t)$ belongs to $C^p([0,2\pi])$ for each t in $[0,T]$.*

Proof. Let $S(\Delta t)$ be the solution operator for the problem (1.1), (1.2), (1.3). For any nonnegative integer n and fixed grid size N, we define

$$E_n = N^{-1/2}\|S(\Delta t)^n f - \tilde{S}_N(\Delta t)^n f\|_2.$$

Then, there exist constants C_1, C_2 and C such that

$$\begin{aligned}
E_{n+1} &= N^{-1/2}\|S(\Delta t)^{n+1} f - \tilde{S}_N(\Delta t)^{n+1} f\|_2 \\
&= N^{-1/2}\|S(\Delta t)S(\Delta t)^n f - \tilde{S}_N(\Delta t)\tilde{S}_N(\Delta t)^n f\|_2 \\
&= N^{-1/2}\|S(\Delta t)S(\Delta t)^n f - \tilde{S}_N(\Delta t)S(\Delta t)^n f + \\
&\quad \tilde{S}_N(\Delta t)S(\Delta t)^n f - \tilde{S}_N(\Delta t)\tilde{S}_N(\Delta t)^n f\|_2 \\
&\leq N^{-1/2}\|S(\Delta t)S(\Delta t)^n f - \tilde{S}_N(\Delta t)S(\Delta t)^n f\| + \\
&\quad N^{-1/2}\|\tilde{S}_N(\Delta t)S(\Delta t)^n f - \tilde{S}_N(\Delta t)\tilde{S}_N(\Delta t)^n f\|_2 \\
&\leq N^{-1/2}\|S(\Delta t)u(t_n) - \tilde{S}_N(\Delta t)u(t_n)\|_2 + \|\tilde{S}_N(\Delta t)\|_2 E_n \\
&\leq C_1\Delta t^2 + C_2\Delta t\Delta x^p + e^{C\Delta t}E_n,
\end{aligned}$$

where the spatial error arises from the truncation of the Fourier series of the exact solution. It follows that

$$E_n \leq \frac{e^{CT}-1}{e^{C\Delta t}-1}(C_1\Delta t^2 + C_2\Delta t\Delta x^p) \leq \tilde{C}_1\Delta t + \tilde{C}_2\Delta x^p\,,$$

for constants $\tilde{C}_1$ and $\tilde{C}_2$ that depend only on T. ☐

5. The two-node case. Using a 2-node Gaussian rule, the Fourier components of the approximate solution are given by

$$[\hat{\mathbf{u}}^{n+1}]_\omega = \hat{\mathbf{e}}_\omega^H \mathbf{u}^n \left(w_1 e^{-\lambda_1 \Delta t} + (1-w_1)e^{-\lambda_2\Delta t}\right) + w_1' \left(e^{-\lambda_1\Delta t} - e^{-\lambda_2 \Delta t}\right) - \Delta t \left(w_1 \lambda_1' e^{-\lambda_1 \Delta t} + (1-w_1)\lambda_2' e^{-\lambda_2\Delta t}\right). \tag{5.1}$$

To obtain the derivatives of the nodes and weights, we use the following recursion coefficients and their derivatives. For convenience, when two column vectors are multiplied, it represents component-wise multiplication:

$$\begin{aligned}
\alpha_1 &= p\omega^2 + \overline{q} \\
\alpha_1' &= [\hat{\mathbf{e}}_\omega \tilde{\mathbf{q}}]^H \mathbf{u}^n \\
\beta_1^2 &= \tilde{\mathbf{q}}^H \tilde{\mathbf{q}} \\
(\beta_1^2)' &= [-pi\omega \hat{\mathbf{e}}_\omega \tilde{\mathbf{q}}' - p\hat{\mathbf{e}}_\omega \tilde{\mathbf{q}}'' + \tilde{\mathbf{q}}\hat{\mathbf{e}}_\omega \tilde{\mathbf{q}} - \beta_1^2 \hat{\mathbf{e}}_\omega]^H \mathbf{u}^n \\
\alpha_2 &= \alpha_1 + \left\{p[\mathbf{q}']^H \mathbf{q}' + [\tilde{\mathbf{q}}^2]^H [\tilde{\mathbf{q}}]\right\} / \beta_1^2 \\
\alpha_2' &= -2\alpha_1' + [\tilde{\mathbf{q}}^3 \hat{\mathbf{e}}_\omega - 4\omega^2 p^2 \tilde{\mathbf{q}}'' \hat{\mathbf{e}}_\omega + 4i\omega p^2 \tilde{\mathbf{q}}''' \hat{\mathbf{e}}_\omega + p^2 \tilde{\mathbf{q}}'''' \hat{\mathbf{e}}_\omega + \\
&\quad 4i\omega p \tilde{\mathbf{q}}\tilde{\mathbf{q}}' \hat{\mathbf{e}}_\omega + 3p\tilde{\mathbf{q}}\tilde{\mathbf{q}}'' \hat{\mathbf{e}}_\omega + 2p\mathbf{q}'\tilde{\mathbf{q}}'\hat{\mathbf{e}}_\omega - \\
&\quad (\left\{p[\mathbf{q}']^H \mathbf{q} + [\tilde{\mathbf{q}}^2]^H [\tilde{\mathbf{q}}]\right\} / \beta_1^2)[2pi\omega \tilde{\mathbf{q}}' \hat{\mathbf{e}}_\omega + p\tilde{\mathbf{q}}'' \hat{\mathbf{e}}_\omega] - \\
&\quad (\alpha_2 - \alpha_1)\tilde{\mathbf{q}}^2 \hat{\mathbf{e}}_\omega - 2i\omega p \overline{q} \tilde{\mathbf{q}}' \hat{\mathbf{e}}_\omega]^H \mathbf{u}^n / \beta_1^2
\end{aligned}$$

It follows that

$$\begin{aligned}
\lambda_{1,2} &= \frac{\alpha_1 + \alpha_2}{2} \pm \frac{\sqrt{(\alpha_1 - \alpha_2)^2 + 4\beta_1^2}}{2}, \\
\lambda_{1,2}' &= \frac{\alpha_1' + \alpha_2'}{2} \pm \frac{(\alpha_1 - \alpha_2)(\alpha_1' - \alpha_2') + 2(\beta_1^2)'}{2\sqrt{(\alpha_1 - \alpha_2)^2 + 4\beta_1^2}}, \\
w_1 &= \frac{\beta_1^2}{(\alpha_1 - \lambda_1)^2 + \beta_1^2}, \\
w_1' &= \frac{(\beta_1^2)'}{(\alpha_1 - \lambda_1)^2 + \beta_1^2} - \frac{\beta_1^2[2(\alpha_1 - \lambda_1)(\alpha_1' - \lambda_1') + (\beta_1^2)']}{[(\alpha_1 - \lambda_1)^2 + \beta_1^2]}.
\end{aligned}$$

It should be emphasized that these formulas are not meant to be used to compute the derivatives of the recursion coefficients, nodes, and weights that are used in the new formulation of KSS methods introduced in section 3; the algorithms presented in that section are more practical for this purpose. However, the above formulas are still useful for analytical purposes. A key observation is that the derivatives of the nodes and weights, taken collectively for all ω, define second-order differential operators. This leads to the following result.

Lemma 5.1. *Let C be a constant-coefficient, self-adjoint, positive definite second-order differential operator, and let V be a second-order variable-coefficient differential operator with coefficients in $BL_M([0, 2\pi])$ for some integer M. Let C_N and V_N be their spectral discretizations on an N-point uniform grid. Then there exists a constant B, independent of N and Δt, such that*

$$\left\| \Delta t e^{-C_N \Delta t} V_N \right\|_\infty \leq B.$$

Proof. For fixed N and Δt, let $A_N(\Delta t) = \Delta t e^{-C_N \Delta t} V_N$. Then, the row of $A_N(\Delta t)$ corresponding to the wave number ω includes the elements

$$\Delta t e^{-C(\omega+\xi)\Delta t} \sum_{j=0}^{2} \hat{v}_j(\xi)(\omega + \xi)^j, \quad \xi = -M/2 + 1, \ldots, M/2,$$

ETNA
Kent State University
etna@mcs.kent.edu

where $C(\lambda)$, the symbol of C, is a second-degree polynomial with negative leading coefficient, and $\hat{v}_j$ is the Fourier transform of the jth-order coefficient of V. If we examine the function

$$f(N,t) = te^{-cN^2 t}N^j, \quad N, t > 0,$$

where c is a positive constant, we find that for the values of j of interest, $j = 0, 1, 2$, $f(x,t)$ is bounded independently of N and t. This, and the fact that the number of nonzero Fourier coefficients is bounded independently of N, yield the theorem's conclusion. □

It is important to note that the fact that $j \leq 2$ is crucial. For larger values of j, $f(N,t)$ is still bounded, but the bound is no longer independent of N and t. Now, we are ready to state and prove one of the main results of the paper.

Theorem 5.2. *Let $q(x)$ in (1.4) belong to $BL_M([0, 2\pi])$ for some integer M. Then, for the problem (1.1), (1.2), (1.3), there exists a constant C such that*

$$\|\tilde{S}_N(\Delta t)\|_\infty \leq C,$$

where $\tilde{S}_N(\Delta t)$ is the approximate solution operator $\tilde{S}_N(\Delta t)$ for KSS(2) on an N-point uniform grid with time step Δt. The constant C is independent of N and Δt.

Proof. It follows from (5.1) that

$$\begin{aligned}\tilde{S}_N(\Delta t) = w_1 e^{-C_1\Delta t} + (1 - w_1)e^{-C_2\Delta t} + [e^{-C_1\Delta t} - e^{-C_2\Delta t}]W_1' - \\ \Delta t[w_1 e^{-C_1\Delta t}V_1 + (1 - w_1)e^{-C_2\Delta t}V_2],\end{aligned}$$

where $W_1 = \mathrm{diag}(w_1(\omega))$, C_i is a constant-coefficient differential operator with symbol $C_i(\omega) = \lambda_i(\omega)$, and $\hat{\mathbf{e}}_\omega^H V_i \mathbf{f} = \lambda_i'(\omega)$, where $'$ denotes the directional derivative of $\lambda_i(\omega)$ in the direction of $\mathbf{f}$.

The symmetric Lanczos algorithm assures that C_i, $i = 1, 2$, is positive definite, so

$$\|e^{-C_i\Delta t}\|_\infty = e^{-\lambda_i(0)\Delta t}. \tag{5.2}$$

From Lemma 5.1, we can conclude that there exist constants V_T and W_T such that

$$\|\Delta t e^{-C_i\Delta t}V_i\|_\infty \leq V_T, \quad i = 1, 2, \quad \|\Delta t e^{-C_i\Delta t}W_i\|_\infty \leq W_T, \tag{5.3}$$

where $W_2 = I - W_1$. Because $\lambda_1(\omega) - \lambda_2(\omega)$ is a constant $\tilde{\lambda}_0$ independent of ω, we can conclude that

$$\Delta t \left\|[e^{-C_1\Delta t} - e^{-C_2\Delta t}]W_1'\right\|_\infty = \Delta t \left\|(I - e^{(C_1 - C_2)\Delta t})e^{-C_1\Delta t}W_1'\right\|_\infty \leq W_T|\tilde{\lambda}_0|. \tag{5.4}$$

Putting together the bounds (5.2), (5.3) and (5.4) yields

$$\|\tilde{S}_N(\Delta t)\|_\infty \leq \max_{i\in\{1,2\}} e^{-\lambda_i(0)\Delta t} + W_T|\tilde{\lambda}_0| + V_T,$$

from which the theorem follows. □

It should be emphasized that this result is not sufficient to conclude unconditional stability, as in the 1-node case, but it does demonstrate the scalability of KSS methods with respect to the grid size.

ETNA
Kent State University
etna@mcs.kent.edu

6. Application to the wave equation. In this section we apply Krylov subspace spectral methods developed in [18] to the problem

$$\left\{ \begin{array}{ll} \frac{\partial^2 u}{\partial t^2} + Lu = 0 & \text{in } (0, 2\pi) \times \mathbb{R}\,, \\ u(0,t) = u(2\pi,t) & \text{on } \mathbb{R}\,, \end{array} \right. \tag{6.1}$$

with the initial conditions

$$u(x,0) = f(x), \quad u_t(x,0) = g(x), \quad x \in (0, 2\pi), \tag{6.2}$$

where, as before, the operator L is as described in (1.4).

6.1. Structure of the solution. A spectral representation of the operator L allows us the obtain a representation of the solution operator (the *propagator*) in terms of the sine and cosine families generated by L by a simple functional calculus. Introduce

$$R_1(t) = L^{-1/2}\sin(t\sqrt{L}) := \sum_{n=1}^{\infty} \frac{\sin(t\sqrt{\lambda_n})}{\sqrt{\lambda_n}} \langle \varphi_n^*, \cdot \rangle \varphi_n\,, \tag{6.3}$$

$$R_0(t) = \cos(t\sqrt{L}) := \sum_{n=1}^{\infty} \cos(t\sqrt{\lambda_n}) \langle \varphi_n^*, \cdot \rangle \varphi_n\,, \tag{6.4}$$

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 (6.1) can be written as

$$P(t) = \left[\begin{array}{cc} R_0(t) & R_1(t) \\ -L\,R_1(t) & R_0(t) \end{array} \right].$$

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.

6.2. Solution using KSS methods. We briefly review the use of Krylov subspace spectral methods for solving (6.1), first outlined in [12].

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

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

where $R_0(t)$ and $R_1(t)$ are defined in (6.3), (6.4), we can obtain $[\mathbf{u}^{n+1}]_\omega$ by approximating each of the quadratic forms

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

where δ_ω is a nonzero constant. It follows that

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

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

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

ETNA
Kent State University
etna@mcs.kent.edu

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 6.1. (KSS-W(K)) *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 (6.1), (6.2) 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 (1.1), (1.2), (1.3). 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.

6.3. Convergence analysis. We now study the convergence behavior of the preceding algorithm. Following the reformulation of Krylov subspace spectral methods presented in section 3, we let $\delta_\omega \to 0$ to obtain

$$\begin{aligned}
\begin{bmatrix} \hat{\mathbf{u}}^{n+1} \\ \hat{\mathbf{u}}_t^{n+1} \end{bmatrix}_\omega = {} & \left(\sum_{k=1}^K w_k \begin{bmatrix} \cos(\sqrt{\lambda_k}t) & \frac{1}{\sqrt{\lambda_k}}\sin(\sqrt{\lambda_k}t) \\ -\sqrt{\lambda_k}\sin(\sqrt{\lambda_k}t) & \cos(\sqrt{\lambda_k}t) \end{bmatrix} \right) \begin{bmatrix} \hat{\mathbf{e}}_\omega^H \mathbf{u}^n \\ \hat{\mathbf{e}}_\omega^H \mathbf{u}_t^n \end{bmatrix} + \\
& \sum_{k=1}^K \begin{bmatrix} \cos(\sqrt{\lambda_k}t) & \frac{1}{\sqrt{\lambda_k}}\sin(\sqrt{\lambda_k}t) \\ -\sqrt{\lambda_k}\sin(\sqrt{\lambda_k}t) & \cos(\sqrt{\lambda_k}t) \end{bmatrix} \begin{bmatrix} w_k' \\ \tilde{w}_k' \end{bmatrix} - \\
& \sum_{k=1}^K w_k \frac{t}{2\sqrt{\lambda_k}} \begin{bmatrix} \sin(\sqrt{\lambda_k}t) & -\frac{1}{\sqrt{\lambda_k}}\cos(\sqrt{\lambda_k}t) \\ \sqrt{\lambda_k}\cos(\sqrt{\lambda_k}t) & \sin(\sqrt{\lambda_k}t) \end{bmatrix} \begin{bmatrix} \lambda_k' \\ \tilde{\lambda}_k' \end{bmatrix} - \\
& w_k \begin{bmatrix} 0 & \frac{1}{2(\lambda_k)^{3/2}}\sin(\sqrt{\lambda_k}t) \\ \frac{1}{2\sqrt{\lambda_k}}\sin(\sqrt{\lambda_k}t) & 0 \end{bmatrix} \begin{bmatrix} \lambda_k' \\ \tilde{\lambda}_k' \end{bmatrix},
\end{aligned} \tag{6.5}$$

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 $\mathbf{u}_t^n$.

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

Theorem 6.2. *Assume that $f(x)$ and $g(x)$ satisfy (1.3), and let $u(x,\Delta t)$ be the exact solution of (6.1), (6.2) at $(x,\Delta t)$, and let $\tilde{u}(x,\Delta t)$ be the approximate solution computed by Algorithm 6.1. Then*

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

where K is the number of quadrature nodes used in Algorithm 6.1.

Proof. See [12]. □

To prove stability, we use the following norm, analogous to that used in [12] to show conservation for the wave equation:

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

Let L be a constant-coefficient, self-adjoint, positive definite second-order differential operator, and let $\mathbf{u}(t)$ be the discretization of the solution of (6.1) at time t. Then it is easily shown, in a manner analogous to [12, Lemma 2.8], that

$$\|(\mathbf{u}(t),\mathbf{u}_t(t))\|_{L_N} = \|(\mathbf{f},\mathbf{g})\|_{L_N},$$

where $\mathbf{f}$ and $\mathbf{g}$ are the discretizations of the initial data $f(x)$ and $g(x)$ from (6.2).

Theorem 6.3. *Let $q(x)$ in (1.4) belong to $BL_M([0,2\pi])$ for some integer M. Then, for the problem (1.1), (1.2), (1.3), KSS-W(1) is unconditionally stable.*

Proof. Let $s_\omega = \sin(\sqrt{\alpha_1}\Delta t)$ and $c_\omega = \cos(\sqrt{\alpha_1}\Delta t)$. In the case $K=1$, (6.5) reduces to

$$\begin{aligned}\begin{bmatrix} \hat{\mathbf{u}}^{n+1} \\ \hat{\mathbf{u}}_t^{n+1} \end{bmatrix}_\omega = {} & \begin{bmatrix} c_\omega & \frac{1}{\sqrt{\alpha_1}} s_\omega \\ -\sqrt{\alpha_1} s_\omega & c_\omega \end{bmatrix} \begin{bmatrix} \hat{\mathbf{e}}_\omega^H \mathbf{u}^n \\ \hat{\mathbf{e}}_\omega^H \mathbf{u}_t^n \end{bmatrix} - \\ & \frac{\Delta t}{2\sqrt{\alpha_1}} \begin{bmatrix} s_\omega & -\frac{1}{\sqrt{\alpha_1}} c_\omega \\ \sqrt{\alpha_1} c_\omega & s_\omega \end{bmatrix} \begin{bmatrix} \hat{\mathbf{e}}_\omega^H V_N \mathbf{u}^n \\ \hat{\mathbf{e}}_\omega^H V_N \mathbf{u}_t^n \end{bmatrix} - \\ & \begin{bmatrix} 0 & \frac{1}{2(\alpha_1)^{3/2}} s_\omega \\ \frac{1}{2\sqrt{\alpha_1}} s_\omega & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{e}}_\omega^H V_N \mathbf{u}^n \\ \hat{\mathbf{e}}_\omega^H V_N \mathbf{u}_t^n \end{bmatrix},\end{aligned} \tag{6.6}$$

where we use the splitting $L = C + V$ as in section 3, with corresponding spectral discretizations L_N, C_N and V_N. The first two terms in (6.6) yield the Fourier component $[\hat{\mathbf{v}}^1]_\omega$ of the exact solution at time Δt to the constant-coefficient problem

$$\begin{aligned} &\frac{\partial^2 v}{\partial t^2} + Cv = 0, \\ &v(x,0) = u(x,t_n) + \frac{\Delta t}{2} P_N C^{-1} V u_t(x,t_n), \\ &v_t(x,0) = u_t(x,t_n) - \frac{\Delta t}{2} P_N V u(x,t_n). \end{aligned}$$

It follows that

$$\begin{aligned}\|(\mathbf{u}^{n+1},\mathbf{u}_t^{n+1})\|_C^2 \leq \|(\mathbf{u}^n,\mathbf{u}_t^n)\|_C^2 + \frac{\Delta t}{2}\|(C_N^{-1}\tilde{\mathbf{q}}\mathbf{u}_t^n, \tilde{\mathbf{q}}\mathbf{u}^n)\|_C^2 + \\ \frac{\Delta t}{2}\|(C_N^{-3/2}\tilde{\mathbf{q}}\mathbf{u}_t^n, C_N^{-1/2}\tilde{\mathbf{q}}\mathbf{u}^n)\|_C^2,\end{aligned}$$

which yields

$$\|\tilde{S}_N(\Delta t)\|_C^2 \leq 1 + \frac{\Delta t}{2} Q[\overline{q}^{-1} + \overline{q}^{-2}],$$

where Q is as defined in the proof of Theorem 4.1. Because $\overline{q} > 0$, we can conclude that there exists a constant α such that $\|\tilde{S}_N(\Delta t)\|_C \leq e^{\alpha \Delta t}$, and the theorem follows. ☐

Theorem 6.4. *Let $q(x)$ in (1.4) belong to $BL_M([0, 2\pi])$ for some integer M. Then, for the problem (1.1), (1.2), (1.3), 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. Analogous to the proof of Theorems 4.2 and 5.2, except that the C-norm is used instead of the 2-norm. ☐

6.4. The two-node case. We will not prove stability for the 2-node case here. 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 source term. First, we note that if $p(x, t)$ and $u(x, t)$ are solutions of the system of first-order wave equations

$$\begin{bmatrix} p \\ u \end{bmatrix}_t = \begin{bmatrix} 0 & a(x) \\ b(x) & 0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x + \begin{bmatrix} F \\ G \end{bmatrix}, \quad t \geq 0, \tag{6.7}$$

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

$$\frac{\partial^2 u}{\partial t^2} = a(x)b(x)\frac{\partial^2 u}{\partial x^2} + a'(x)b(x)\frac{\partial u}{\partial x} + bF_x + G, \tag{6.8}$$

with the source term $b(x)F_x(x, t) + G(x, t)$. In [14], a time-compact fourth-order finite-difference scheme is applied to a problem of the form (6.7), 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 (6.8) and solve it with initial data

$$u(x, 0) = \cos x, \tag{6.9}$$

$$u_t(x, 0) = \sin x. \tag{6.10}$$

The results of applying both methods to this problem are shown in Figure 6.4, for the case $\alpha = 1$. Due to the smoothness of the coefficients, the spatial discretization error in the Krylov subspace spectral method is dominated by the temporal error, resulting in greater than sixth-order accuracy in time.

Tables 6.1 and 6.2 illustrate the differences in stability between the two methods. For the fourth-order finite-difference scheme from [14], 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) this limit can be greatly exceeded and reasonable accuracy can still be achieved.

ETNA
Kent State University
etna@mcs.kent.edu

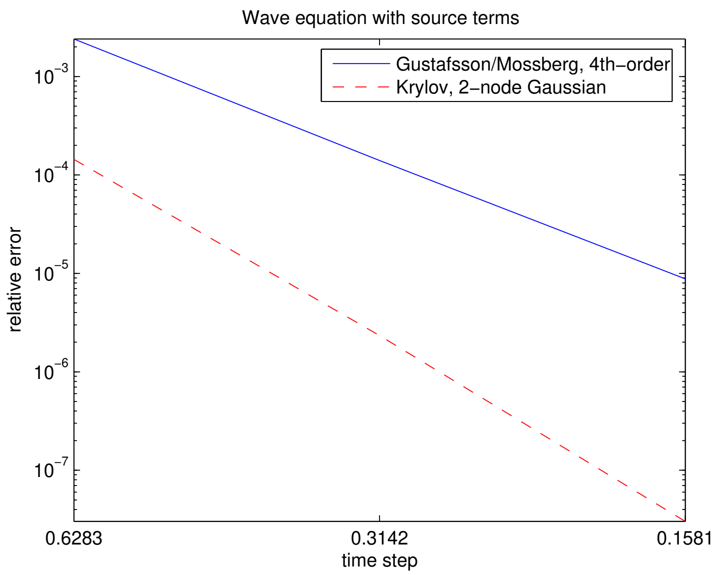

FIG. 6.1. *Estimates of relative error in the approximate solution of problem (6.8), (6.9), (6.10) with periodic boundary conditions, at $t = 8\pi$, computed with the time-compact fourth-order finite-difference scheme from [14] (solid curve) and a Krylov subspace spectral method (dashed curve). In the finite-difference scheme, $\lambda = \Delta t/\Delta x = 0.99$, and in the Krylov subspace spectral method, a 2-point Gaussian quadrature rules are used, and $N = 40$ grid points.*

TABLE 6.1
Relative error in the solution of (6.8) with the time-compact fourth-order finite difference scheme from [14], for various values of N.

$c_{\max}\Delta t/\Delta x$	N	error
	10	0.0024
0.99	20	0.00014
	40	0.0000088

7. Homogenization. So far, we have assumed that the leading coefficient of the operator L is constant, to simplify the analysis. We now consider a general second-order self-adjoint positive definite operator

$$L = -Da_2(x)D + a_0(x), \tag{7.1}$$

with symbol

$$L(x, \xi) = a_2(x)\xi^2 - a_2'(x)i\xi + a_0(x).$$

Instead of applying a KSS method directly to the problem (1.1) with this operator, we use the fact that KSS methods are most accurate when the coefficients of L are nearly constant (see [19, Theorem 2.5]) and use similarity transformations to homogenize these coefficients, effectively preconditioning the problem. In this section, we discuss these transformations. We begin with a known transformation that homogenizes the leading coefficient $a_2(x)$, and show how it can be used to generalize the stability results from the previous sections. Then, we introduce a new transformation that can homogenize $a_0(x)$ when $a_2(x)$ is constant, and demonstrate that such a transformation can improve the accuracy of KSS methods.

TABLE 6.2
Relative error in the solution of (6.8) with KSS-W(2), for various values of $\lambda = \Delta t/\Delta x$.

$\Delta t/\Delta x$	N	error
32	64	0.007524012
16	64	0.000145199
8	64	0.000008292

7.1. Homogenizing the leading coefficient. We first construct a *canonical transformation* Φ that, while defined on the region of phase space $[0, 2\pi] \times \mathbb{R}$, arises from a simple change of variable in physical space, $y = \phi(x)$, where $\phi(x)$ is a differentiable function and

$$\phi'(x) > 0, \quad \operatorname{Avg} \phi' = \frac{1}{2\pi} \int_0^{2\pi} \phi'(s)\, ds = 1.$$

The transformation Φ has the form

$$\Phi(y, \eta) \to (x, \xi), \quad x = \phi^{-1}(y), \quad \xi = \phi'(x)\eta. \tag{7.2}$$

Using this simple canonical transformation, we can homogenize the leading coefficient of L as follows: Choose $\phi(x)$ and construct a canonical transformation $\Phi(y, \eta)$ by (7.2) so that the leading term of the transformed symbol

$$\tilde{L}(y, \eta) = L(x, \xi) \circ \Phi(y, \eta) = L(\phi^{-1}(y), \phi'(\phi^{-1}(y))\eta) \tag{7.3}$$

is independent of y.

We can conclude by Egorov's theorem (see [4]) that there exists a unitary Fourier integral operator U such that if $A = U^{-1}LU$, then the symbol of A agrees with $\tilde{L}$ modulo lower-order errors. In fact, $Uf(y) = |D\phi^{-1}(y)|^{-1/2} f \circ \phi^{-1}(y)$. Therefore, using the chain rule and symbolic calculus (see below), it is a simple matter to construct this new operator $A(y, D)$.

Applying (7.3) and examining the leading coefficient of the transformed symbol yields

$$\phi'(x) = c[a_2(x)]^{-1/2},$$

where the constant c is added to ensure that $\operatorname{Avg} \phi' = 1$. This transformation is used by Guidotti, et al. in [12] to obtain approximate high-frequency eigenfunctions of a second-order operator.

In the case where $a_0(x) = 0$, 0 is an eigenvalue with corresponding eigenfunction equal to a constant function. However, because of the factor $|D\phi^{-1}(y)|^{-1/2}$ in $Uf(y)$, the constant function is not an eigenfunction of the transformed operator. It follows that in the splitting $L = C + V$, where L is the transformed operator and C, as in previous sections, is the constant-coefficient operator obtained by averaging the coefficients of L, then C is positive definite, even though L is positive semi-definite. Therefore, the stability results stated in Theorem 4.1 and Theorem 6.3 can still apply to L.

7.2. Symbolic calculus. For homogenizing lower-order coefficients, we will rely on the rules of *symbolic calculus* to work with pseudodifferential operators (see [16], [17]), or ψdO, more easily and thus perform similarity transformations of such operators with much less computational effort than would be required if we were to apply transformations that acted on matrices representing discretizations of these operators.

We will be constructing and applying unitary similarity transformations of the form

$$\tilde{L} = U^* L U,$$

where U is a Fourier integral operator, and, in some cases, a ψdO. In such cases, it is necessary to be able to compute the adjoint of a ψdO, as well as the product of ψdO.

To that end, given a differential operator A, the symbol of the adjoint A^* is given by

$$A^*(x,\xi) = \sum_\alpha \frac{1}{\alpha!}\frac{\partial^\alpha}{\partial x^\alpha}\frac{\partial^\alpha}{\partial \xi^\alpha}\overline{A(x,\xi)}, \tag{7.4}$$

while the symbol of the product of two differential operators AB, denoted by $AB(x,\xi)$, is given by

$$AB(x,\xi) = \sum_\alpha \frac{1}{\alpha!}\frac{\partial^\alpha A}{\partial \xi^\alpha}\frac{\partial^\alpha B}{\partial^\alpha x}. \tag{7.5}$$

These rules are direct consequences of the product rule for differentiation.

7.3. The pseudo-inverse of the differentiation operator. For general ψdO, the rules (7.4), (7.5) do not always apply exactly, but they do yield an approximation. However, it will be necessary for us to work with ψdO of negative order, so we must identify a class of negative-order ψdO for which these rules do apply.

Let A be an $m \times n$ matrix of rank r, and let $A = U\Sigma V^T$ be the singular value decomposition of A, where $U^TU = I_m$, $V^TV = I_n$, and $\Sigma = \operatorname{diag}(\sigma_1,\ldots,\sigma_r,0,\ldots,0)$. Then, the *pseudo-inverse* (see [10]) of A is defined as

$$A^+ = V\Sigma^+U^T,$$

where the $n \times m$ diagonal matrix Σ^+ is given by

$$\Sigma^+ = \begin{bmatrix} \sigma_1^{-1} & & & & & \\ & \ddots & & & & \\ & & \sigma_r^{-1} & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}.$$

We can generalize this concept to define the pseudo-inverse of the differentiation operator D on the space of 2π-periodic functions by

$$D^+u(x) = \frac{1}{\sqrt{2\pi}}\sum_{\omega=-\infty}^{\infty} e^{i\omega x}(i\omega)^+\hat{u}(\omega), \quad z^+ = \begin{cases} z^{-1} & z \neq 0 \\ 0 & z = 0 \end{cases}.$$

The rules (7.4) and (7.5) can be used for pseudodifferential operators defined using D^+; see [18]. This allows us to efficiently construct and apply unitary similarity transformations based on ψdO of the form

$$U = \sum_{\alpha=0}^{\infty} a_\alpha(x)[D^+]^{-\alpha}.$$

We now consider such transformations.

ETNA
Kent State University
etna@mcs.kent.edu

7.4. Lower-order coefficients. It is natural to ask whether it is possible to construct a unitary transformation U that smooths L globally, i.e. yield the decomposition

$$U^*LU = \tilde{L}(\eta).$$

In this section, we will attempt to answer this question. We seek to eliminate lower-order variable coefficients. The basic idea is to construct a transformation U_α such that

1. U_α is unitary,
2. The transformation $\tilde{L} = U_\alpha^* L U_\alpha$ yields an operator $\tilde{L} = \sum_{\alpha=-\infty}^{m} a_\alpha(x) \left(\frac{\partial}{\partial x}\right)^\alpha$ such that $a_\alpha(x)$ is constant, and
3. The coefficients $b_\beta(x)$ of L, where $\beta > \alpha$, are invariant under the similarity transformation $\tilde{L} = U_\alpha^* L U_\alpha$.

It turns out that such an operator is not difficult to construct. First, we note that if ϕ is a skew-symmetric pseudodifferential operator, then $U = \exp[\phi]$ is a unitary operator, since

$$U^*U = (\exp[\phi])^* \exp[\phi] = \exp[-\phi]\exp[\phi] = I.$$

We consider an example to illustrate how one can determine a operator ϕ so that $U = \exp[\phi]$ satisfies the second and third conditions given above. Consider a second-order self-adjoint operator of the form

$$L = a_2 D^2 + a_0(x).$$

In an effort to transform L so that the zeroth-order coefficient is constant, we apply the similarity transformation $\tilde{L} = U^*LU$, which yields an operator of the form

$$\begin{aligned}\tilde{L} = L + (L\phi - \phi L) + \frac{1}{2}[(L\phi - \phi L)\phi - \phi(L\phi - \phi L)] + \\ \frac{1}{2}[\phi(\phi L \phi) - (\phi L \phi)\phi] + \cdots .\end{aligned}$$

Since we want the first and second-order coefficients of L to remain unchanged, the perturbation E of L in $\tilde{L} = L + E$ must not have order greater than zero. If we require that ϕ has negative order $-k$, then the highest-order term in E is $L\phi - \phi L$, which has order $1-k$, so in order to affect the zeroth-order coefficient of L we must have ϕ be of order -1. By symbolic calculus, it is easy to determine that the highest-order coefficient of $L\phi - \phi L$ is $2a_2 b_{-1}'(x)$ where $b_{-1}(x)$ is the leading coefficient of ϕ. Therefore, in order to satisfy

$$a_0(x) + 2a_2 b_{-1}'(x) = \text{constant},$$

we must have $b_{-1}'(x) = -(a_0(x) - \text{Avg}\, a_0)/2a_2$. In other words, we can choose

$$b_{-1}(x) = -\frac{1}{2a_2} D^+(a_0(x)),$$

where D^+ is the pseudo-inverse of the differentiation operator D introduced in section 7.3. Therefore, for our operator ϕ, we can use

$$\phi = \frac{1}{2}[b_{-1}(x)D^+ - (b_{-1}(x)D^+)^*] = b_{-1}(x)D^+ + \text{lower-order terms}.$$

Using symbolic calculus, it can be shown that the coefficient of order -1 in $\tilde{L}$ is zero. We can use similar transformations to make lower-order coefficients constant as well. This will be explored in [21].

ETNA
Kent State University
etna@mcs.kent.edu

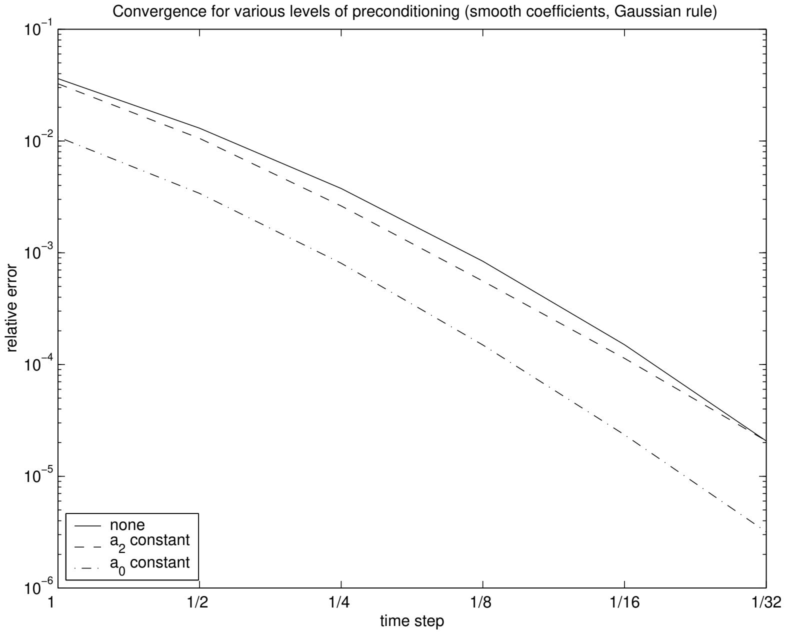

FIG. 7.1. *Estimates of relative error in the approximate solution $\tilde{u}(x,t)$ of (1.1), (1.2), (1.3) at $T = 1$, computed using no preconditioning (solid curve), a similarity transformation to make the leading coefficient of $L_1 = U^*LU$ constant (dashed curve), and a similarity transformation to make $L_2 = Q^*U^*LUQ$ constant-coefficient modulo terms of negative order. In all cases $N = 64$ grid points are used, with time steps $\Delta t = 2^{-j}$ for $j = 0, \ldots, 6$.*

We conclude this section with a demonstration of the benefit of this homogenization. Figure 7.1 depicts the temporal error for an operator L of the form (7.1), with smooth coefficients. Because the coefficients are already smooth, homogenizing $a_2(x)$ only slightly improves the accuracy, but homogenizing $a_0(x)$ as well yields a much more substantial improvement.

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

8.1. Higher space dimension. In [20], 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 given in section 5 for the one-dimensional case. It is therefore reasonable to suggest that for operators of this form, the stability results given here for the one-dimensional case generalize to higher dimensions. This will be investigated in the near future. In addition, generalization of the similarity transformations of section 7 is in progress.

8.2. Discontinuous coefficients. For the stability results reported in this paper, particularly Theorem 5.2, 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, these results do not apply to problems in which the coefficients are particularly rough or discontinuous. Future work will include the use of KSS methods with other bases of trial functions besides trigonometric polynomials, such as orthogonal wavelets or multiwavelet bases introduced in [1].

8.3. Higher-order schemes. As the number of quadrature nodes per component increases, higher-order derivatives of the coefficients are included in the expressions for the recursion coefficients, and therefore the regularity conditions that must be imposed on the coefficients are more stringent. However, even with $K = 1$ or $K = 2$, high-order accuracy in time can be achieved, so it is not a high priority to pursue this direction, except in the case of KSS-R(2), as the prescribed node significantly improves accuracy for parabolic problems, as observed in [19].

8.4. Summary. We have demonstrated that for both parabolic and hyperbolic variable-coefficient PDE, KSS methods compute Fourier components of the solution from directional derivatives of moments, where the directions are obtained from the solution from previous time steps. The resulting reformulation of these methods facilitates analysis of their stability, and in the case of sufficiently smooth coefficients, unconditional stability is achieved. Therefore, KSS methods represent a viable compromise between the computational efficiency of explicit methods and the stability of implicit methods. Although these analytical results apply to a rather narrow class of differential operators, they can be applied to problems with more general operators by means of unitary similarity transformations, which have the effect of preconditioning the problem in order to achieve greater accuracy.

REFERENCES

[1] B. ALPERT, G. BEYLKIN, D. GINES, AND L. VOZOVOI, *Adaptive solution of partial differential equations in multiwavelet bases*, J. Comput. Phys., 182 (2002), pp. 149–190.
[2] K. ATKINSON, *An Introduction to Numerical Analysis*, 2nd edition, John Wiley & Sons Inc., New York, 1989.
[3] G. DAHLQUIST, S. C. EISENSTAT, AND G. H. GOLUB, *Bounds for the error of linear systems of equations using the theory of moments*, J. Math. Anal. Appl., 37 (1972), pp. 151–166.
[4] C. FEFFERMAN, *The uncertainty principle*, Bull. Amer. Math. Soc., 9 (1983), pp. 129–206.
[5] W. GAUTSCHI, *Orthogonal polynomials: applications and computation*, Acta Numer., 5 (1996), pp. 45–119.
[6] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973) pp. 318–334.
[7] ———, *Bounds for matrix moments*, Rocky Mountain J. Math., 4 (1974), pp. 207–211.
[8] G. H. GOLUB AND M. H. GUTKNECHT, *Modified moments for indefinite weight functions*, Numer. Math., 57 (1989), pp. 607–624.
[9] G. H. GOLUB AND G. MEURANT, *Matrices, moments and quadrature*, in Numerical Analysis 1993 (Dundee, 1993), Pitman Res. Notes Math. Ser., 303 (1994), Longman Sci. Tech., Harlow, pp. 105–156.
[10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd edition, Johns Hopkins University Press, Baltimore, MD, 1996.
[11] G. H. GOLUB AND J. WELSCH, *Calculation of Gauss quadrature rules*, Math. Comp., 23 (1969), pp. 221–230.
[12] P. GUIDOTTI, J. V. LAMBERS, AND K. SØLNA, *Analysis of 1D wave propagation in inhomogeneous media*, Numer. Funct. Anal. Optim., 27 (2006), pp. 25–55.
[13] B. GUSTAFSSON, H.-O. KREISS, AND J. OLIGER, *Time Dependent Problems and Difference Methods*, John Wiley & Sons Inc., New York, 1995.
[14] B. GUSTAFSSON AND E. MOSSBERG, *Time compact high order difference methods for wave propagation*, SIAM J. Sci. Comput., 26 (2004), pp. 259–271.
[15] M. HOCHBRUCK AND C. LUBICH, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1996), pp. 1911–1925.
[16] L. HÖRMANDER, *Pseudo-differential operators*, Comm. Pure Appl. Math., 18 (1965), pp. 501–517.
[17] J. J. KOHN AND L. NIRENBERG, *An algebra of pseudo-differential operators*, Comm. Pure Appl. Math., 18 (1965), pp. 269–305.
[18] J. V. LAMBERS, *Krylov Subspace Methods for Variable-Coefficient Initial-Boundary Value Problems*, Ph.D. Thesis, Stanford University, SCCM Program, 2003.
http://sccm.stanford.edu/pub/sccm/theses/James_Lambers.pdf.
[19] ———, *Krylov subspace spectral methods for variable-coefficient initial-boundary value problems*, Electron. Trans. Numer. Anal., 20 (2005), pp. 212–234.
http://etna.math.kent.edu/vol.20.2005/pp212-234.dir/pp212-234.html.
[20] ———, *Practical implementation of Krylov subspace spectral methods*, J. Sci. Comput., 32 (2007), pp. 449–476.
[21] ———, *Unitary similarity transformations for variable-coefficient differential operators*, in preparation.
[22] R. A. SACK AND A. F. DONOVAN, *An algorithm for Gaussian quadrature given modified moments*, Numer. Math., 18 (1971/72), pp. 465–478.