Proceedings of the World Congress on Engineering 2008 Vol II
WCE 2008, July 2 - 4, 2008, London, U.K.

Implicitly Defined High-Order Operator Splittings for Parabolic and Hyperbolic Variable-Coefficient PDE Using Modified Moments

James V. Lambers *

Abstract—**This paper presents a reformulation of Krylov Subspace Spectral (KSS) Methods, which use Gaussian quadrature in the spectral domain compute high-order accurate approximate solutions to variable-coefficient parabolic and hyperbolic PDE. This reformulation serves two useful purposes. First, it improves the numerical stability of these methods by removing cancellation arising from the approximation of certain derivatives by finite differences by computing these derivatives analytically. Second, it reveals that KSS methods are actually high-order operator splittings that are defined implicitly, in terms of derivatives of the nodes and weights of Gaussian quadrature rules with respect to a parameter. Efficient algorithms for computing these derivatives are provided, as well as the first application of KSS methods to systems of coupled PDE.**

Keywords: ***spectral methods, Gaussian quadrature, Lanczos method, heat equation, wave equation***

1 Introduction

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

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

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

with periodic boundary conditions. The operator L is a second-order differential operator of the form

$$Lu = -pD^2u + q(x)u, \tag{3}$$

where $D = \frac{\partial}{\partial x}$, 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. The exact solution can be represented using a Fourier series if $q(x)$ is constant, but here we concern ourselves exclusively with the solution of variable-coefficient problems, for which numerical methods are necessary.

*Submitted March 6, 2008. Stanford University, Department Energy Resources Engineering, Stanford CA USA 94305-2220 Tel/Fax: 650-725-2729/2099 Email: lambers@stanford.edu

In [9], [10] a class of methods, called Krylov subspace spectral (KSS) methods, was introduced for the purpose of solving these problems. It has been shown in these references, as well as [7], [11], 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, [6]) for stiff systems of ODE. However, because these methods approximate derivatives of bilinear forms using finite differences, they are prone to numerical instability. In this paper, we address this issue, and also demonstrate that the resulting modified KSS methods are actually high-order operator splittings.

Section 2 reviews the main properties of KSS methods, including algorithmic details and results concerning local accuracy. It will be shown that KSS methods can be reformulated as high-order operator splittings that are implicitly defined in terms of directional derivatives of nodes and weights of Gaussian quadrature rules. This reformulation, and a description of the resulting splittings, is presented in Section 3, along with efficient algorithms for computing the derivatives that define the splittings. Generalization to the second-order wave equation is described in Section 4. Section 5 briefly discusses generalization to systems of coupled PDE. Section 6 features conclusions and future directions.

2 Krylov Subspace Spectral Methods

We begin with a review of the main aspects of KSS methods. These methods are time-stepping algorithms that compute the solution of (1), (2) 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 $S(t) = \exp[-Lt]$ to $\tilde{u}(x, t_n)$. KSS 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 each Fourier component of the solution can be approximated by a bilinear form

$$\hat{u}(\omega, t_{n+1}) \approx \hat{\mathbf{e}}_\omega^H S_N(\Delta t)\mathbf{u}(t_n), \tag{4}$$

where

$$[\hat{\mathbf{e}}_\omega]_j = \frac{1}{\sqrt{2\pi}} e^{i\omega j \Delta x}, \quad [\mathbf{u}(t_n)]_j = u(j\Delta x, t_n), \tag{5}$$

and

$$S_N(t) = \exp[-L_N t], \quad [L_N]_{jk} = \sum_{\mu=0}^{m} a_\mu(j\Delta x)[D_N^\mu]_{jk} \tag{6}$$

where D_N is a discretization of the differentiation operator D that is defined on the space of grid functions. Our goal is to approximate (4) 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 [2] Golub and Meurant describe a method for computing quantities of the form

$$\mathbf{u}^T f(A)\mathbf{v}, \tag{7}$$

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 (6), $f(\lambda) = \exp(-\lambda t)$ 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

$$b = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N = a > 0, \tag{8}$$

and corresponding orthogonal eigenvectors $\mathbf{q}_j$, $j = 1, \ldots, N$. Therefore, the quantity (7) can be rewritten as

$$\mathbf{u}^T f(A)\mathbf{v} = \sum_{\ell=1}^{N} f(\lambda_\ell)\mathbf{u}^T\mathbf{q}_j\mathbf{q}_j^T\mathbf{v}. \tag{9}$$

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

$$\alpha(\lambda) = \begin{cases} 0, & \text{if } \lambda < a \\ \sum_{j=i}^{N} \alpha_j\beta_j, & \text{if } \lambda_i \leq \lambda < \lambda_{i-1} \\ \sum_{j=1}^{N} \alpha_j\beta_j, & \text{if } b \leq \lambda \end{cases}, \tag{10}$$

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 (7) can be viewed as a Riemann-Stieltjes integral

$$\mathbf{u}^T f(A)\mathbf{v} = I[f] = \int_a^b f(\lambda)\, d\alpha(\lambda). \tag{11}$$

As discussed in [2], the integral $I[f]$ can be approximated using a Gaussian quadrature rule, where the nodes and weights 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 [4]).

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 (7) using decompositions such as

$$\mathbf{u}^T f(A)\mathbf{v} = \frac{1}{\delta}[\mathbf{u}^T f(A)(\mathbf{u} + \delta\mathbf{v}) - \mathbf{u}^T f(A)\mathbf{u}], \tag{12}$$

where δ is a small constant. Guidelines for choosing an appropriate value for δ can be found in [10, Section 2.2].

Employing these quadrature rules yields the following basic process (for details see [9], [10]) 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, 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 (12), 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 [10], a polar decomposition is used instead of (12), 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, which was proved in [10]. 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 $BL_N([0, 2\pi])$. Let $f \in BL_N([0, 2\pi])$. Then the preceding algorithm, applied to the problem (1), (2), is consistent; i.e.*

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

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

The preceding result can be compared to the accuracy achieved by an algorithm described by Hochbruck and

Lubich in [6] 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 [6], 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), (2), due to their stiffness.

In this situation, 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. The difference between KSS methods and the approach described in [6] 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 shown in [10], using the same subspace for all components causes a loss of accuracy as the number of grid points increases, whereas KSS methods do not suffer from this phenomenon.

Unfortunately, the difference quotient used to compute each Fourier component of the solution can be numerically unstable if δ_ω is chosen too small, but this parameter must also be chosen small enough to ensure stability of the quadrature rules. We will now address this issue, and realize an additional benefit in the process: operator splittings that are high-order accurate in time, and, though explicit, possess favorable stability properties.

3 Reformulation of KSS Methods as Splittings

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

$$\frac{d}{d\delta_\omega}\left[\hat{\mathbf{e}}_\omega^H(\hat{\mathbf{e}}_\omega+\delta_\omega\mathbf{u}^n)\mathbf{e}_1^T\exp[T_\omega(\delta_\omega)\Delta t]\mathbf{e}_1\right]\Big|_{\delta_\omega=0} \tag{13}$$

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. This leads to an improved algorithm, henceforth referred to as the "new formulation" of KSS methods, over the "original formulation", described in [11].

3.1 High-Order Splittings

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:

$$\begin{aligned}
[\hat{\mathbf{u}}^{n+1}]_\omega &= \hat{\mathbf{e}}_\omega^H\mathbf{u}^{n+1}\\
&= \frac{d}{d\delta_\omega}\left[\hat{\mathbf{e}}_\omega^H(\hat{\mathbf{e}}_\omega+\delta_\omega\mathbf{u}^n)\exp[-T_\omega(\delta_\omega)\Delta t]_{11}\right]\Big|_{\delta_\omega=0}\\
&= \frac{d}{d\delta_\omega}\left[\hat{\mathbf{e}}_\omega^H(\hat{\mathbf{e}}_\omega+\delta_\omega\mathbf{u}^n)\sum_{k=1}^K w_j e^{-\lambda_j\Delta t}\right]\Big|_{\delta_\omega=0}\\
&= \hat{\mathbf{e}}_\omega^H\mathbf{u}^n\sum_{k=1}^K w_j e^{-\lambda_j\Delta t}+\sum_{k=1}^K w_j' e^{-\lambda_j\Delta t}- \qquad (14)\\
&\quad \Delta t\sum_{k=1}^K w_j\lambda_j' e^{-\lambda_j\Delta t}
\end{aligned}$$

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.

It can be shown that the derivatives of the nodes and weights at $\delta_\omega=0$ are Fourier components of pseudodifferential operators applied to $\mathbf{u}^n$. It follows that by considering all Fourier components together, we find that KSS methods are actually high-order operator splittings "in disguise". These splittings have the form

$$\exp[-L\Delta t]\approx\sum_{k=1}^K W_k e^{-C_k\Delta t}[I-\Delta t V_k] \tag{15}$$

where K is the number of quadrature nodes, and the operators C_k and W_k are diagonal in the basis of trial functions (e.g., a constant-coefficient operator when using Fourier series). In fact, their eigenvalues are the kth nodes and weights, respectively. The operators V_k have the form

$$V_k=C_k'+\Delta t^{-1}W_k^{-1}W_k' \tag{16}$$

where the Fourier components of $C_k'\mathbf{u}^n$ and $W_k'\mathbf{u}^n$ are the derivatives of the nodes and weights with respect to δ_ω at $\delta_\omega=0$. Because $e^{-C_k\Delta t}\to I$ linearly as $\Delta t\to 0$, and $\sum_{j=1}^K W_k=I$, it follows that the terms in V_k of order $O(\Delta t^{-1})$ cancel and pose no difficulty.

As shown in [7], splittings such as this facilitate stability analysis of KSS methods, and such analysis demonstrates that KSS methods represent a "best-of-both-worlds" compromise between explicit and explicit time-stepping methods, as they possess the stability of implicit methods, but like explicit methods, they do not require solution of large systems of equations. However, unlike splitting such as the Strang splitting (see [13]), the stages of the splitting (15) cannot easily be described in terms of the coefficients of L. This is because they are defined

in terms of the nodes and weights of quadrature rules, which do not have a simple relation to the recursion coefficients, as they come from the eigenvalues and eigenvectors of Jacobi matrices. Nevertheless, these splittings can still be implemented efficiently, as the action of the operators V_k on a given grid function can be computed from the derivatives of the nodes and weights. We now discuss how to compute these derivatives.

3.2 Derivatives of the Nodes and Weights

We now show how to efficiently compute the derivatives of the quadrature nodes and weights used in (14), using their relationships to the matrix $T_\omega(\delta_\omega)$.

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 [4]). 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

$$T_\omega(\delta_\omega) = Q_\omega(\delta_\omega)\Lambda_\omega(\delta_\omega)Q_\omega(\delta_\omega)^{-1}, \tag{17}$$

where $Q_\omega(0) = Q_\omega^0$. Differentiating (17) with respect to δ_ω and evaluating at $\delta_\omega = 0$ yields

$$\operatorname{diag}(\Lambda_\omega'(0)) = \operatorname{diag}\left(Q_\omega(0)^H T_\omega'(0) Q_\omega(0)\right), \tag{18}$$

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, which are the products of the first components of the left and right eigenvectors of $T_\omega(\delta_\omega)$, we consider the equation

$$(T_\omega(\delta_\omega) - \lambda_j I)\mathbf{w}_j(\delta_\omega) = \mathbf{0}, \quad j = 1, \ldots, K, \tag{19}$$

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.

3.3 Derivatives of the Recursion Coefficients

From the preceding discussion, we need the derivatives of the entries of $T_\omega(\delta_\omega)$ with respect to δ_ω, at $\delta_\omega = 0$. To that end, 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. Now, suppose that we wish to compute the entries of the modified matrix $\hat{T}_K$ 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.

In [11], an iteration that produces $\hat{T}_K$ was presented, based on algorithms from [3]. 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 $\hat{\mathbf{e}}_\omega^H S_N(\Delta t)\hat{\mathbf{e}}_\omega$. 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.

In [7], this algorithm was used 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 that define $\hat{T}_K$ with respect to δ_ω and evaluating at $\delta_\omega = 0$, we obtain a Lanczos-like iteration that yields the derivatives of the recursion coefficients α_j and β_j with respect to δ_ω, evaluated at $\delta_\omega = 0$.

It is worth noting that because the initial vector for the case $\delta_\omega = 0$ is $\hat{\mathbf{e}}_\omega$, the inner products needed to obtain the derivatives of the recursion coefficients can be computed for all ω simultaneously using appropriate FFTs. As shown in [11], [7], the resulting time-stepping algorithm, in either the original or the new formulation, requires $O(N \log N)$ floating-point operations per time step.

In [8] it is shown that Theorem 1 applies to the new formulation. Then, it is demonstrated that this method yields essentially the same results as the original formulation, and much greater accuracy than the standard ODE solvers in MATLAB, which implement algorithms described in [12].

4 Application to the Wave Equation

In this section we apply KSS methods developed in [9] to the problem

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} + Lu = 0 & \text{in } (0, 2\pi) \times \mathbb{R}, \\ u(0,t) = u(2\pi, t) & \text{on } \mathbb{R}, \end{cases} \tag{20}$$

with the initial conditions

$$u(x,0) = f(x), \quad u_t(x,0) = g(x), \quad x \in (0, 2\pi), \tag{21}$$

where, as before, the operator L is as described in (3).

4.1 Solution Using KSS Methods

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. We introduce

$$R_1(t) = L^{-1/2}\sin(t\sqrt{L}), \quad R_0(t) = \cos(t\sqrt{L}). \tag{22}$$

These functions of L indicate which functions are the integrands in the Riemann-Stieltjes integrals used to compute the Fourier components of the solution. Since the exact solution $u(x,t)$ is given by

$$u(x,t) = R_0(t)f(x) + R_1(t)g(x), \tag{23}$$

we can obtain $[\mathbf{u}^{n+1}]_\omega$ by approximating each of the quadratic forms

$$\begin{aligned}
c_\omega^+(t) &= \langle \hat{\mathbf{e}}_\omega, R_0(\Delta t)[\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n]\rangle & (24)\\
c_\omega^-(t) &= \langle \hat{\mathbf{e}}_\omega, R_0(\Delta t)\hat{\mathbf{e}}_\omega\rangle & (25)\\
s_\omega^+(t) &= \langle \hat{\mathbf{e}}_\omega, R_1(\Delta t)[\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}_t^n]\rangle & (26)\\
s_\omega^-(t) &= \langle \hat{\mathbf{e}}_\omega, R_1(\Delta t)\hat{\mathbf{e}}_\omega\rangle\,, & (27)
\end{aligned}$$

where δ_ω is a nonzero constant.

We can obtain the Fourier coefficients of an approximation of $u_t(x,t)$ by approximating similar quadratic forms (see [5] for details). As noted in [9], 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*.

4.2 Reformulation

Following the reformulation of KSS methods presented in Section 3, we let $\delta_\omega \to 0$ to obtain

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

where $c_k = \cos(\sqrt{\lambda_k}t)$, $s_k = \sin(\sqrt{\lambda_k}t)$, and 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 $\mathbf{u}_t^n$.

As shown in [8], this new formulation has the same order of accuracy as the original presented in [5]. Specifically, the error in each Fourier component is $O(\Delta t^{4K})$, where K is the number of nodes in each Gaussian quadrature rule.

4.3 Numerical Results

We now compare the original formulation of KSS methods as presented in [11] with its reformulation as presented in Section 3, with the appropriate integrands used in place of $e^{-\lambda t}$. We construct the operator L and initial data $f(x)$ and $g(x)$ so that coefficients and data have three continuous derivatives. We then apply both formulations to approximate the solution to (20), (21) at $t = 1$, using 2-node Gaussian rules for each Fourier component of $u(x,t)$ and $u_t(x,t)$. As before, $N = 64$ grid points are used.

Because this KSS method is 7th-order accurate in time, very high accuracy is achieved, but as $\Delta t \to 0$, this order of convergence is not maintained by the original formulation due to catastrophic cancellation in the final step of the computation of each component. Because this subtraction is not performed in the new formulation as an implicitly-defined operator splitting, the order of convergence is maintained by this reformulation for smaller values of Δt, as we can see in Figure 1.

Figure 1 also reports the results of solving this problem using two of MATLAB's ODE solvers, `ode23s` and `ode45`, a higher-order accurate explicit solver that, while not a practical option for a parabolic problem, is much better suited to this hyperbolic one. While `ode45` exhibits slightly higher-order accuracy in this case, it is unable to achieve reasonable accuracy for larger time steps, and this threshold decreases as the number of grid points increases due to the stiffness of the problem, while KSS methods do not exhibit such sensitivity, even though they too are explicit. This is demonstrated in [8], where the same problem is solved with $N = 128$ grid points.

5 Systems of Coupled PDE

In [8], KSS methods were also applied to a system of n coupled variable-coefficient PDE of the form $\mathbf{u}_t + L\mathbf{u}$, where $\mathbf{u} : [0, 2\pi] \times [0, \infty) \to \mathbb{R}^n$ for $n > 1$. The operator L is an $n \times n$ matrix where the (i, j) entry is a differential operator L_{ij}.

Generalization of KSS methods to such a system can proceed as follows. For $i, j = 1, \ldots, n$, let $\overline{L}_{ij}$ be the constant-coefficient operator obtained by averaging the coefficients of L_{ij} over the spatial domain. Then, for each wave number ω, we define $L(\omega)$ to be the matrix with entries equal to the symbols of each $\overline{L}_{ij}$ evaluated at ω. Then, we define our trial and test functions by $\mathbf{q}_j(\omega) \otimes e^{i\omega x}$, where $\mathbf{q}_j$ is a Schur vector of $L(\omega)$.

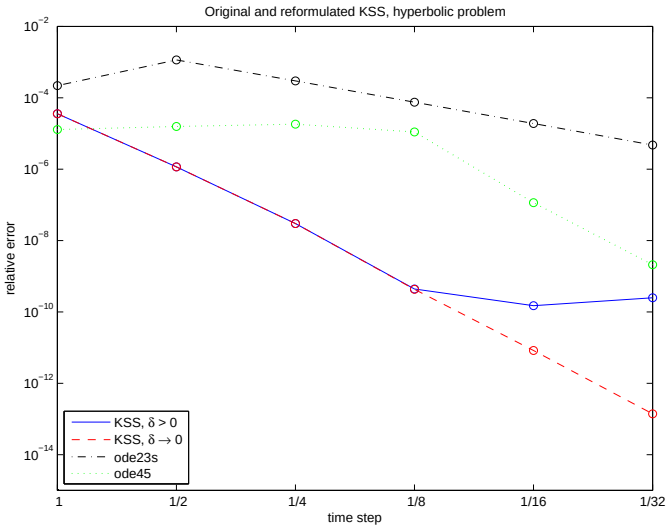

Figure 1: Estimates of relative error in the approximate solution of problem (20), (21) at $t = 1$, computed with the original KSS method from [11] with $\delta_\omega = 10^{-5}$ (solid curve), the reformulated KSS method (dashed curve), the MATLAB solver `ode23s` (dotted-dashed curve), and `ode45` (dotted curve). In both KSS methods, 2-node Gaussian quadrature rules are used, and $N = 64$ grid points are used for all methods.

As in the scalar case, this approach yields $O(\Delta t^{2K})$ local temporal error, where K is the number of nodes in each Gaussian quadrature rule, even though L is not self-adjoint. The recursion coefficients, nodes and weights can be computed in the same manner as in the scalar, self-adjoint case. The result is an implicitly-defined operator splitting of the form (15). Numerical results presented in [8] demonstrate the superior accuracy obtained by KSS methods compared to the ODE solvers of MATLAB.

6 Summary

We have demonstrated that for both parabolic and hyperbolic variable-coefficient PDE, a reformulation of KSS methods that eliminates the need to perturb quadrature rules not only improves their numerical stability, but also reveals that these methods are actually implicitly-defined high-order operator splittings. Although the stages of the splitting are not easily described, efficient algorithms for computing appropriate derivatives of the nodes and weights allow efficient implementation of these methods. We have also shown that KSS methods can also be readily generalized to systems of coupled PDE through an appropriate choice of trial functions, even if the operator $L(x, D)$ is not self-adjoint. Future work will explore the analysis, and enhancement, of KSS methods through closer examination of the splittings they define, and variation of the parameter δ_ω with respect to ω. Also, because of the improved numerical stability of these methods, using a larger number of nodes is now more viable than in the original formulation.

References

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

[2] 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)

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

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

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

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

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

[8] Lambers, J. V.: Implicitly Defined High-Order Operator Splittings for Parabolic and Hyperbolic Variable-Coefficient PDE Using Modified Moments. *International Journal of Computational Science* to appear.

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

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

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

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

[13] Strang, G.: On the construction and comparison of difference schemes. *SIAM Journal of Numerical Analysis* **5** (1968), pp. 506-517.