NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2012; **19**:970–991
Published online 19 March 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nla.1831

Explicit high-order time stepping based on componentwise application of asymptotic block Lanczos iteration

James V. Lambers*,†

University of Southern Mississippi, 118 College Dr #5045, Hattiesburg, MS, 39406-0001, USA

SUMMARY

This paper describes the development of explicit time stepping methods for linear PDEs that are specifically designed to cope with the stiffness of the system of ODEs that results from spatial discretization. As stiffness is caused by the contrasting behavior of coupled components of the solution, it is proposed to adopt a componentwise approach in which each coefficient of the solution in an appropriate basis is computed using an individualized approximation of the solution operator. This has been accomplished by Krylov subspace spectral (KSS) methods, which use techniques from 'matrices, moments and quadrature' to approximate bilinear forms involving functions of matrices via block Gaussian quadrature rules. These forms correspond to coefficients with respect to the chosen basis of the application of the solution operator of the PDE to the solution at an earlier time. In this paper, it is proposed to substantially enhance the efficiency of KSS methods through the prescription of quadrature nodes on the basis of asymptotic analysis of the recursion coefficients produced by block Lanczos iteration for each Fourier coefficient as a function of frequency. The potential of this idea is illustrated through numerical results obtained from the application of the modified KSS methods to diffusion equations and wave equations. Copyright © 2012 John Wiley & Sons, Ltd.

Received 30 April 2011; Revised 27 November 2011; Accepted 30 January 2012

KEY WORDS: Lanczos algorithm; spectral methods; Gaussian quadrature; heat equation; wave equation

1. INTRODUCTION

The rapid advancement of computing power in recent years has allowed higher resolution models. This has introduced greater stiffness into systems of ordinary differential equations (ODEs) that arise from the discretization of time-dependent partial differential equations (PDEs), which presents difficulties for time stepping methods due to the coupling of components of the solution.

Consider a system of linear ODE of the form

$$\mathbf{u}'(t) + A\mathbf{u} = \mathbf{0}, \quad \mathbf{u}(t_0) = \mathbf{u}_0, \tag{1}$$

where A is an $N \times N$ real, symmetric positive definite matrix. One-step methods such as Runge–Kutta methods yield an approximate solution of the form

$$\mathbf{u}^{n+1} = f(A; \Delta t)\mathbf{u}^n, \quad \Delta t = t_{n+1} - t_n,$$

where $\mathbf{u}^n \approx \mathbf{u}(t_n)$ and $f(A; \Delta t)$ is a polynomial of A that approximates $\exp[-A\Delta t]$ if the method is explicit or a rational function of A if the method is implicit.

Similarly, Krylov subspace methods based on exponential integrators, such as those described in [1, 2], yield approximate solutions that are equal to a polynomial of A times a vector. The polynomial is obtained by computing the exponential of a much smaller matrix that is generated by a

*Correspondence to: J. V. Lambers, University of Southern Mississippi, 118 College Dr #5045, Hattiesburg, MS, 39406-0001, USA.
†E-mail: James.Lambers@usm.edu

process such as Lanczos iteration or Arnoldi iteration. For example, consider the problem of computing $\mathbf{w} = e^{-At}\mathbf{v}$ for a given symmetric matrix A and vector $\mathbf{v}$. An approach described in [2] is to apply the Lanczos algorithm to A with initial vector $\mathbf{v}$ to obtain, at the end of the jth iteration, an orthogonal matrix X_j and a tridiagonal matrix T_j such that $X_j^{\mathrm{T}} A X_j = T_j$. Then, we can compute the approximation

$$\mathbf{w}_j = \|\mathbf{v}\|_2 X_j e^{-T_j t}\mathbf{e}_1, \tag{2}$$

where $\mathbf{e}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$. As each column $\mathbf{x}_k$, $k = 1, \ldots, j$, of X_j is of the form $\mathbf{x}_k = p_{k-1}(A)\mathbf{v}$, where $p_n(A)$ is a polynomial of degree n in A, it follows that $\mathbf{w}_j$ is the product of a polynomial in A of degree $j-1$ and $\mathbf{v}$.

The effectiveness of this approach, for general $\mathbf{v}$, depends on the eigenvalues of A. If the eigenvalues are not clustered, which is the case if A arises from a stiff system of ODEs such that obtained from spatial discretization of a time-dependent PDE, a good approximation cannot be obtained using a small number of Lanczos iterations. This can be alleviated using an outer iteration of the form

$$\mathbf{w}_j^{m+1} \approx e^{-A\Delta t}\mathbf{w}_j^m, \quad m = 0, 1, \ldots, \quad \mathbf{w}_j^0 = \mathbf{v}, \tag{3}$$

for some $\Delta t \ll t$, where the total number of outer iterations M satisfies $M\Delta t = t$. However, this approach is not practical if Δt must be chosen very small. Modifications described in [3, 4] produce rational approximations that reduce the number of Lanczos iterations, but these methods require solving systems of linear equations in which the matrix is of the form $I + hA$, where h is a parameter. Unless h is chosen very small, these systems may be ill conditioned.

In summary, time stepping methods that, when applied to a system of linear ODEs of the form (1), compute a solution of the form $\mathbf{u}(t + \Delta t) = f(A; \Delta t)\mathbf{u}(t)$, where $f(A; \Delta t)$ is a polynomial or rational approximation of $e^{-A\Delta t}$, tend to have difficulty with stiff systems, such as those that arise from the spatial discretization of time-dependent PDE. The solution of a stiff system, even one that represents a typically smooth solution of a parabolic PDE, includes rapidly changing components that are coupled with components that evolve more slowly. Because of this phenomenon, explicit methods for such systems require small time steps, whereas implicit methods typically require the solution of ill-conditioned systems of linear equations.

It is therefore proposed to employ a *componentwise* approach to time stepping in an effort to circumvent these difficulties for both parabolic and hyperbolic PDEs. This goal is to be accomplished by continuing the evolution of Krylov subspace spectral (KSS) methods [5–8]. These methods feature explicit time stepping with high-order accuracy and stability that is characteristic of implicit methods. This 'best-of-both-worlds' combination is achieved through a componentwise approach, in which each Fourier coefficient of the solution is computed using an approximation of the solution operator that is, in some sense, optimal for that coefficient.

Initial works on KSS methods [5–16] have yielded promising results, in terms of accuracy and stability. However, because they implicitly generate $O(N^d)$ low-dimensional Krylov subspaces, where N is the number of grid points per spatial dimension and d is the number of spatial dimensions, efficient implementation can be difficult except for certain special cases, even though the computational complexity per time step is $O(N \log N)$. As most of the computational expense arises from the need to compute component-dependent nodes and weights of block Gaussian quadrature rules, this paper explores the idea of using component-dependent quadrature rules in which the nodes are prescribed in such a way as to approximate the block Gaussian nodes, in an effort to achieve comparable accuracy but with far greater efficiency. The approximation strategy is based on asymptotic analysis of the recursion coefficients produced by block Lanczos iteration.

The outline of the paper is as follows. Section 2 describes how block KSS methods arise from techniques introduced by Golub and Meurant in [17] for approximating bilinear forms involving functions of matrices. Section 3 describes how asymptotic analysis of the quantities computed by block KSS methods can be used to efficiently approximate the extremal block Gaussian nodes. Numerical results are presented in Section 4. Discussion of current and future research directions is provided in Section 5, and conclusions are given in Section 6.

DOI: 10.1002/nla

2. MATRICES, MOMENTS, QUADRATURE, AND PDE

This section describes the main ideas behind block KSS methods. For simplicity, the one-dimensional case is discussed first, followed by the generalization to higher dimensions. Let $S(t) = \exp[-Lt]$ represent the solution operator of a parabolic PDE on $(0, 2\pi)$,

$$u_t + Lu = 0, \quad t > 0, \tag{4}$$

with appropriate initial conditions and periodic boundary conditions. The operator L is a linear, second-order, self-adjoint, positive definite differential operator.

Let $\langle \cdot, \cdot \rangle$ denote the standard inner product of functions defined on $[0, 2\pi]$. KSS methods, introduced in various forms in [5–8], 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 coefficients that would be obtained by applying the exact solution operator to $\tilde{u}(x, t_n)$,

$$\hat{u}(\omega, t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, S(\Delta t)\tilde{u}(x, t_n) \right\rangle, \tag{5}$$

where ω is an integer.

Clearly, such an approach requires an effective method for computing bilinear forms. KSS methods represent the first application to time-dependent PDE of techniques from the area of 'matrices, moments and quadrature': the approximation of bilinear forms involving matrix functions by treating them as Riemann–Stieltjes integrals and then applying (block) Gaussian quadrature rules that are generated by the (block) Lanczos algorithm applied to the matrix. We now provide some background on this area.

2.1. Elements of functions of matrices

In [17], Golub and Meurant described a method for computing quantities of the form

$$\mathbf{u}^{\mathrm{T}} f(A)\mathbf{v}, \tag{6}$$

where $\mathbf{u}$ and $\mathbf{v}$ are N-vectors, A is an $N \times N$ symmetric matrix, and f is a smooth function. Our goal is to apply this method with $A = L_N$, where L_N is a spectral discretization of L, $f(\lambda) = \exp(-\lambda t)$ for some t, and the vectors $\mathbf{u}$ and $\mathbf{v}$ are derived from $\hat{\mathbf{e}}_\omega$ and $\mathbf{u}^n$, where $\hat{\mathbf{e}}_\omega$ is a discretization of $(1/\sqrt{2\pi})e^{i\omega x}$ and $\mathbf{u}^n$ is a discretization of the solution at time t_n on an N-point uniform grid. In the following exposition, it is assumed that $\mathbf{u}$ and $\mathbf{v}$ are real; generalization to the complex case is straightforward.

Because the matrix A is symmetric positive definite, it has real eigenvalues $b = \lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_N = a > 0$ and corresponding orthonormal eigenvectors $\mathbf{q}_j$, $j = 1, \ldots, N$. Therefore, the quantity (6) can be rewritten as

$$\mathbf{u}^{\mathrm{T}} f(A)\mathbf{v} = \sum_{j=1}^{N} f(\lambda_j)\, \mathbf{u}^{\mathrm{T}}\mathbf{q}_j \mathbf{q}_j^{\mathrm{T}}\mathbf{v}, \tag{7}$$

which can also be viewed as a Riemann–Stieltjes integral

$$\mathbf{u}^{\mathrm{T}} f(A)\mathbf{v} = I[f] = \int_a^b f(\lambda)\, \mathrm{d}\alpha(\lambda), \tag{8}$$

where the measure $\alpha(\lambda)$ is derived from the coefficients of $\mathbf{u}$ and $\mathbf{v}$ in the basis of eigenvectors.

DOI: 10.1002/nla

As discussed in [17–20], the integral $I[f]$ can be approximated using Gaussian, Gauss–Radau or Gauss–Lobatto, quadrature rules, which yields an approximation of the form

$$I[f] = \sum_{j=1}^{K} w_j f(\lambda_j) + R[f], \tag{9}$$

where the nodes λ_j, $j = 1, \ldots, K$, and the weights w_j, $j = 1, \ldots, K$, can be obtained using the Lanczos algorithm and variations [21–24]. If $\mathbf{u} = \mathbf{v}$, then the measure $\alpha(\lambda)$ is positive and increasing, which ensures that the weights are positive.

2.2. Block Gaussian quadrature

In the case $\mathbf{u} \neq \mathbf{v}$, there is the possibility that the weights may not be positive, which destabilizes the quadrature rule [25]. Instead, one can consider the approximation of

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}^{\mathrm{T}} f(A) \begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}, \tag{10}$$

which results in the 2×2 matrix integral

$$\int_a^b f(\lambda)\, \mathrm{d}\mu(\lambda) = \begin{bmatrix} \mathbf{u}^{\mathrm{T}} f(A)\mathbf{u} & \mathbf{u}^{\mathrm{T}} f(A)\mathbf{v} \\ \mathbf{v}^{\mathrm{T}} f(A)\mathbf{u} & \mathbf{v}^{\mathrm{T}} f(A)\mathbf{v} \end{bmatrix}, \tag{11}$$

where $\mu(\lambda)$ is a 2×2 matrix, each entry of which is a measure of the form $\alpha(\lambda)$ from (8).

As discussed in [17], this matrix integral can be approximated using a quadrature rule of the form

$$\int_a^b f(\lambda)\, \mathrm{d}\mu(\lambda) = \sum_{j=1}^{2K} f(\lambda_j)\, \mathbf{v}_j \mathbf{v}_j^{\mathrm{T}} + \mathit{error}, \tag{12}$$

where, for each j, λ_j is a scalar and $\mathbf{v}_j$ is a 2-vector. Each node λ_j is an eigenvalue of the matrix

$$\mathcal{T}_K = \begin{bmatrix} M_1 & B_1^{\mathrm{T}} & & & \\ B_1 & M_2 & B_2^{\mathrm{T}} & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-2} & M_{K-1} & B_{K-1}^{\mathrm{T}} \\ & & & B_{K-1} & M_K \end{bmatrix}, \tag{13}$$

which is a block tridiagonal matrix of order $2K$. The vector $\mathbf{v}_j$ consists of the first two elements of the corresponding normalized eigenvector. The matrices M_j and B_j are computed by performing K iterations of the block Lanczos algorithm, which was proposed by Golub and Underwood in [26].

Throughout the remainder of this paper, we will refer to the scalar nodes λ_j, $j = 1, 2, \ldots, 2K$, from (12) as 'block Gaussian quadrature nodes', to distinguish them from the Gaussian quadrature nodes that are obtained through the standard (non-block) Lanczos iteration applied to a single initial vector. A key difference between the two sets of nodes is that whereas a K-node Gaussian rule is exact for polynomials of degree $2K - 1$, it takes $2K$ block Gaussian nodes to achieve the same degree of accuracy; that is, block Gaussian rules have the same degree of accuracy as general interpolatory quadrature rules. However, unlike general interpolatory quadrature rules, they are obtained from a polynomial of degree K whose coefficients are 2×2 matrices, just as K standard Gaussian quadrature nodes are the roots of a polynomial of degree K with scalar coefficients.

2.3. Block Krylov subspace spectral methods

Block KSS methods proceed as follows. We assume for convenience that N is even. For each wave number $\omega = -N/2 + 1, \ldots, N/2$, we define

$$R_0(\omega) = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \mathbf{u}^n \end{bmatrix}$$

DOI: 10.1002/nla

and then compute the QR factorization

$$R_0(\omega) = X_1(\omega)B_0(\omega),$$

which yields

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

where the 'H' superscript denotes the Hermitian transpose and

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

Then, block Lanczos iteration is applied to the discretized operator L_N with initial block $X_1(\omega)$, producing a block tridiagonal matrix $\mathcal{T}_K(\omega)$ of the form (13), where each entry is a *function* of ω. Then, each Fourier coefficient of the solution at t_{n+1} can be expressed as

$$[\hat{\mathbf{u}}^{n+1}]_\omega = \left[B_0^{\mathrm{H}} E_{12}^{\mathrm{H}} \exp[-\mathcal{T}_K(\omega)\Delta t] E_{12} B_0\right]_{12}, \quad E_{12} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix}. \tag{14}$$

In this paper, we continue to follow the convention used in [6, 10, 16, 27] and identify a block KSS method that uses K matrix nodes, or $2K$ scalar nodes, as a K-node KSS method. The term 'KSS method' is used to refer to a larger class of methods, including not only block KSS methods but also non-block variations such as those described in [5, 8] and variations introduced in the next section. What characterizes all KSS methods is that they compute each component of the solution in some basis by approximating a bilinear form involving a matrix function with an interpolatory quadrature rule that is tailored to that component.

This algorithm has local temporal accuracy $O(\Delta t^{2K-1})$ for the parabolic problem (4) [6]. By contrast, methods that apply Lanczos iteration only to the solution from the previous time step (see, for example, [2]) achieve $O(\Delta t^{K-1})$ accuracy, where K is the dimension of the Krylov subspaces used. Even higher-order accuracy, $O(\Delta t^{4K-2})$, is obtained for the second-order wave equation [10]. Furthermore, in [6, 10], it is shown that under appropriate assumptions on the coefficients of the differential operator L in (4), the one-node block KSS method is *unconditionally stable*.

2.4. Implementation

Krylov subspace spectral methods compute a Jacobi matrix corresponding to *each* Fourier coefficient, in contrast to traditional Krylov subspace methods (see, for example, [1–4, 28]) that use only a single Krylov subspace generated by the solution from the previous time step. Although it would appear that KSS methods incur a substantial amount of additional computational expense, that is not actually the case because nearly all of the Krylov subspaces that they compute are closely related by the wave number ω in the one-dimensional case, or $\vec{\omega} = (\omega_1, \omega_2, \ldots, \omega_n)$ in the n-dimensional case.

In fact, the only Krylov subspace that is explicitly computed is the one generated by the solution from the previous time step, of dimension $(K+1)$, where $2K$ is the number of block Gaussian quadrature nodes. In addition, the averages of the coefficients of L^j, for $j = 0, 1, 2, \ldots, 2K-1$, are required, where L is the spatial differential operator. When the coefficients of L are independent of time, these can be computed once, during a preprocessing step. This computation can be carried out in $O(N \log N)$ operations using symbolic calculus [12, 14].

With these considerations, the algorithm for a single time step of a one-node block KSS method for solving (4), where $Lu = -pu_{xx} + q(x)u$, with appropriate initial conditions and periodic boundary conditions, is as follows. The average of a function $f(x)$ on $[0, 2\pi]$ is denoted by $\overline{f}$.

$\hat{\mathbf{u}}^n = \mathbf{fft}(\mathbf{u}^n)$, $\mathbf{v} = L_N\mathbf{u}^n$, $\hat{\mathbf{v}} = \mathbf{fft}(\mathbf{v})$
for each ω **do**
  $\alpha_1 = -p\omega^2 + \overline{q}$ (in preprocessing step)
  $\beta_1 = \hat{\mathbf{v}}(\omega) - \alpha_1\hat{\mathbf{u}}^n(\omega)$
  $\alpha_2 = \langle \mathbf{u}^n, \mathbf{v}\rangle - 2\,\mathrm{Re}\,[\hat{\mathbf{u}}^n(\omega)\overline{\hat{\mathbf{v}}(\omega)}] + \alpha_1|\hat{\mathbf{u}}^n(\omega)|^2$
  $e_\omega = [\langle \mathbf{u}^n, \mathbf{u}^n\rangle - |\hat{\mathbf{u}}^n(\omega)|^2]^{1/2}$

DOI: 10.1002/nla

$$T_\omega = \begin{bmatrix} \alpha_1 & \beta_1/e_\omega \\ \beta_1/e_\omega & \alpha_2/e_\omega^2 \end{bmatrix}$$

$$\hat{\mathbf{u}}^{n+1}(\omega) = [e^{-T_\omega \Delta t}]_{11}\hat{\mathbf{u}}^n(\omega) + [e^{-T_\omega \Delta t}]_{12}e_\omega$$

end

$\mathbf{u}^{n+1} = \mathbf{ifft}(\hat{\mathbf{u}}^{n+1})$

As with the above-mentioned preprocessing step, the amount of work per time step is $O(N \log N)$, assuming that the FFT is used for differentiation in the computation of v. This same complexity applies to KSS methods with a higher number of quadrature nodes [7]. Furthermore, KSS methods allow substantial parallelism, as each component of the solution is obtained from its associated block Jacobi matrix, independently of other components. Coefficients of powers of L can also be computed in parallel.

Generalization of this algorithm to higher dimensions and other PDE is straightforward [29, 30]. For example, on the domain $[0, 2\pi]^n$, if $L = -p\Delta + q(\mathbf{x})$, the **for** loop would iterate over all n-tuples $\vec{\omega} = (\omega_1, \omega_2, \ldots, \omega_n)$, and for each such $\vec{\omega}$, $\alpha_1 = -p\|\vec{\omega}\|_2^2 + \overline{q}$, where $\overline{q}$ is the average of $q(\mathbf{x})$ on the domain. No additional modification to the algorithm is necessary. For the second-order wave equation, two matrices of the form T_ω are computed for each $\vec{\omega}$, corresponding to the solution from the previous time step and its time derivative. KSS methods have also been generalized to systems of coupled equations; the details are presented in [11]. In particular, it is shown in [16] that KSS methods are effective for systems of three equations in three spatial variables, through application to Maxwell's equations.

Although the one-node block KSS method described previously can easily be implemented and KSS methods can be implemented with $O(N \log N)$ complexity for any number of nodes, the asymptotic constant of the number of floating-point operations can grow quite quickly as a function of K because of the need to compute coefficients of L^j, for $j = 2, \ldots, 2K-1$. Optimization can also be made difficult by the need to maintain data structures that store representations of recursion coefficients [12, 27]. Therefore, it is necessary to consider whether alternative approaches to selecting and computing quadrature nodes for each component of the solution may be more effective.

3. SOLUTION THROUGH ASYMPTOTIC BLOCK LANCZOS ITERATION

The central idea behind KSS methods is to compute each component of the solution, in some basis, using an approximation that is specifically tailored to that component, even though all components are coupled. It follows that each component uses a different polynomial approximation of $S(L_N)$, where the function S is based on the solution operator of the PDE and L_N is the discretization of the spatial differential operator. Taking all components together reveals that the computed solution has the form

$$\mathbf{u}^{n+1} = \tilde{f}(L_N; \Delta t)\mathbf{u}^n = \sum_{j=0}^{M} D_j(\Delta t) A^j \mathbf{u}^n, \tag{15}$$

where $M = 2K$, K being the number of block Lanczos iterations, and $D_j(\Delta t)$ is a matrix that is diagonal in the chosen basis. In other words, KSS methods are explicit methods that are inherently more flexible than explicit time stepping methods that use a polynomial or rational approximation of the solution operator. We now consider how we can exploit this flexibility.

In the block KSS method, in which block Lanczos iteration is applied to the matrix L_N, the initial block recursion coefficient M_1 in (13) is given by

$$M_1(\omega) = \begin{bmatrix} \hat{\mathbf{e}}_\omega^{\mathrm{H}} L_N \hat{\mathbf{e}}_\omega & \hat{\mathbf{e}}_\omega^{\mathrm{H}} L_N \mathbf{u}_\omega^n \\ [\mathbf{u}_\omega^n]^{\mathrm{H}} L_N \hat{\mathbf{e}}_\omega & [\mathbf{u}_\omega^n]^{\mathrm{H}} L_N \mathbf{u}_\omega^n \end{bmatrix}. \tag{16}$$

Now, suppose that L is a second-order differential operator of the form

$$Lu = -pu_{xx} + q(x)u,$$

DOI: 10.1002/nla

where p is a constant. Then, we have

$$M_1(\omega) = \begin{bmatrix} p\omega^2 + \bar{q} & \hat{\mathbf{e}}_\omega^{\mathrm{H}} \tilde{\mathbf{q}} \mathbf{u}_\omega^n \\ [\mathbf{u}_\omega^n]^{\mathrm{H}} \tilde{\mathbf{q}} \hat{\mathbf{e}}_\omega & [\mathbf{u}_\omega^n]^{\mathrm{H}} L_N \mathbf{u}_\omega^n \end{bmatrix}. \tag{17}$$

As before, we use the notation $\overline{f}$ to denote the mean of a function $f(x)$ defined on $[0, 2\pi]$ and define $\tilde{q}(x) = q(x) - \overline{q}$. We denote by $\tilde{\mathbf{q}}$ the vector with components $[\tilde{\mathbf{q}}]_j = \tilde{q}(x_j)$. For convenience, multiplication of vectors, as in the off-diagonal elements of $M_1(\omega)$, denotes componentwise multiplication.

It follows that as $|\omega|$ increases, we have

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

That is, the off-diagonal entries become negligible compared with the $(1, 1)$ entry. Continuing with the block Lanczos process to compute $B_1(\omega)$, we see that for higher frequencies, the diagonal entries of $M_1(\omega)$ are approximate eigenvalues of $\mathcal{T}_K(\omega)$.

In fact, it has been observed in numerical experiments that for the two-node block KSS method, half of the four scalar nodes λ_j from (12) are clustered around each diagonal entry of $M_1(\omega)$ [6]. We therefore propose to first prescribe these values as the smallest and largest nodes:

$$a = \frac{[\mathbf{u}^n]^{\mathrm{H}} L_N \mathbf{u}^n}{[\mathbf{u}^n]^{\mathrm{H}} \mathbf{u}^n}, \quad b = \frac{\hat{\mathbf{e}}_\omega^{\mathrm{H}} L_N \hat{\mathbf{e}}_\omega}{\hat{\mathbf{e}}_\omega^{\mathrm{H}} \hat{\mathbf{e}}_\omega}, \quad \lambda_1(\omega) = \min\{a, b\}, \quad \lambda_2(\omega) = \max\{a, b\}. \tag{18}$$

Then, we prescribe the remaining nodes $\lambda_2(\omega), \ldots, \lambda_{2K-1}(\omega)$ to be equally spaced between $\lambda_1(\omega)$ and $\lambda_{2K}(\omega)$.

It is generally known that quadrature rules with equally spaced nodes are not effective when the number of nodes is large (for example, Newton–Cotes rules with 11 or more nodes are guaranteed to have at least one negative weight). However, the number of nodes, $2K$, is to be chosen on the basis of the desired temporal order of accuracy of $2K - 1$; therefore, the number of nodes can be chosen to be very small. Alternative node distributions will be discussed in Section 5.

As will be seen in the numerical experiments, the number of nodes need not be even; it is assumed to be even in this discussion because of the proposed method being described as a modification of block KSS methods that use $2K$ nodes obtained from K iterations of block Lanczos. In general, a KSS method applied to a PDE of the form $u_t + Lu = 0$ that uses M *prescribed* nodes requires a Krylov subspace of dimension M to be generated from the solution from the previous time step in order to compute an approximation of the form (15) and has temporal order of accuracy $M - 1$.

As will be demonstrated in the numerical results in the next section, this approach of prescribing the extremal nodes as in (18) is not effective for an operator with a variable leading coefficient, such as

$$Lu = -(p(x)u_x)_x + q(x)u.$$

By applying block Lanczos to a matrix L_N arising from discretization of this operator and initial block $R_0(\omega)$, we find that the block $R_1(\omega)$ produced by the block Lanczos algorithm,

$$R_1(\omega) = L_N X_1(\omega) - X_1(\omega) M_1(\omega),$$

contains vectors that are nearly parallel to $\tilde{\mathbf{p}}\hat{\mathbf{e}}_\omega$ and $\hat{\mathbf{e}}_\omega$, where $\tilde{p}(x) = p(x) - \overline{p(x)}$. Therefore, in order to approximate the largest block Gaussian node, which is the largest eigenvalue of $\mathcal{T}_K$, we seek a linear combination of these two vectors, $\mathbf{v}_\omega = \tilde{\mathbf{p}}\hat{\mathbf{e}}_\omega + c\hat{\mathbf{e}}_\omega$, that maximizes the Rayleigh quotient

$$\frac{\mathbf{v}_\omega^{\mathrm{H}} L_N \mathbf{v}_\omega}{\mathbf{v}_\omega^{\mathrm{H}} \mathbf{v}_\omega}. \tag{19}$$

DOI: 10.1002/nla

The constant c satisfies the quadratic equation

$$\|\tilde{\mathbf{p}}\|_2 c^2 + \tilde{\mathbf{p}}^{\mathrm{H}}\tilde{\mathbf{p}}^2 c - \|\tilde{\mathbf{p}}\|_2^2 = 0, \tag{20}$$

and we set $\lambda_{2K}(\omega)$ equal to the corresponding Rayleigh quotient.

We will also examine the idea of choosing the largest node for each component so that it serves as a sharp upper bound, rather than a sharp lower bound, on the largest node of a block Gaussian rule. To that end, we prescribe

$$\lambda_{2K}(\omega) = c_0\omega^2 + \overline{\mathbf{q}} \tag{21}$$

where c_0 is chosen to be an approximate upper bound on the asymptotic constant for the largest block Gaussian node as a function of ω. In all variations, the interior nodes $\lambda_2, \ldots, \lambda_{2K-1}$ are chosen to be equally spaced between λ_1 and λ_{2K}.

Once the nodes are prescribed, we use the fact that any interpolatory quadrature rule computes the exact integral of a polynomial that interpolates the integrand at the nodes, in conjunction with the fact that each integral represents a bilinear form $\hat{\mathbf{e}}_\omega f(A)\mathbf{u}^n$ for some function f, to avoid having to explicitly compute any quadrature weights. Instead, we construct the solution at time t_{n+1} by using representation (15), in which the diagonal entries of the matrices D_j are the coefficients of the polynomials that interpolate $f(\lambda)$ at the prescribed nodes.

Time stepping methods that employ a polynomial approximation of the solution operator effectively use the same quadrature nodes and weights for all components of the solution. By viewing such methods as special cases of KSS methods with component-independent nodes and weights, it can readily be seen why such methods have difficulties with stiff systems. Even though the bilinear form representing each component of the solution uses the same function of the matrix, the bilinear form is an integral with a different measure $\alpha(\lambda)$, which is influenced by both the solution and each basis function. As such, there is little hope that a single quadrature rule can yield sufficient accuracy for each member of a family of integrals with very different weight functions, unless it has a very large number of nodes.

4. NUMERICAL RESULTS

Numerical experiments comparing KSS methods with a variety of time stepping methods, including finite-difference methods, Runge–Kutta methods, and backward differentiation formulae, can be found in [8, 10, 11, 14]. Comparisons with a number of Krylov subspace methods based on exponential integrators [1,2,28], including preconditioned Lanczos iteration [3,4], are made in [15].

In this section, we will solve various PDEs by using the method described in (2) with four iterations of the Lanczos algorithm, a two-node block KSS method that uses four (scalar) block Gaussian nodes, and a four-node KSS method with equally spaced nodes chosen using asymptotic block Lanczos iteration. All methods are configured to use Krylov subspaces of the same dimension. Because the exact solutions are not known, accuracy will be gauged using the two-norm of a relative error estimate that is obtained by comparing the computed solution to a solution computed using a smaller time step, which will convey order of convergence in time.

4.1. Parabolic problems

4.1.1. One-dimensional problems. We first demonstrate the accuracy of KSS methods combined with asymptotic block Lanczos iteration on a parabolic equation in one space dimension,

$$u_t - (p(x)u_x)_x + q(x)u = 0, \quad 0 < x < 2\pi, \quad t > 0, \tag{22}$$

where the coefficients $p(x)$ and $q(x)$, given by

$$p(x) = 1, \quad q(x) = \frac{4}{3} + \frac{1}{4}\cos x, \tag{23}$$

DOI: 10.1002/nla

are chosen to be smooth functions. The initial condition is

$$u(x,0) = 1 + \frac{3}{10}\cos x - \frac{1}{20}\sin 2x, \quad 0 < x < 2\pi, \tag{24}$$

and periodic boundary conditions are imposed.

Because the leading coefficient is constant, we use (18) to prescribe the extremal nodes for the KSS method combined with asymptotic block Lanczos iteration. The results are shown in Figure 1 and Table I. We observe that both variations of KSS methods exhibit approximately third-order accuracy in time at the chosen time steps, but the Lanczos method described by (2) is far less accurate, in terms of both order of convergence and magnitude of the error; only at smaller time steps does it exhibit the expected third-order convergence. We note that the accuracy of Lanczos iteration is degraded somewhat when the number of grid points is doubled, but that does not affect the performance of either KSS method.

The effectiveness of choosing the nodes by using (18) can be explained with the assistance of Figure 2. It can be seen that the extremal nodes agree very well with the block Gaussian nodes computed by the standard block KSS method. It is interesting to note that choosing the interior nodes to achieve equal spacing is sufficient to achieve nearly as much accuracy as comput-

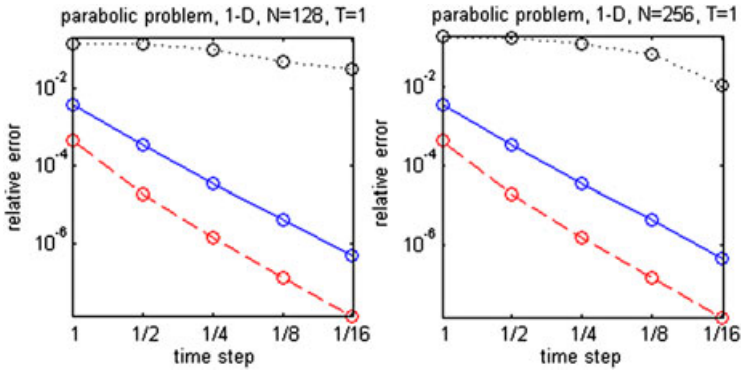

Figure 1. Estimates of relative error at $t = 1$ in the solution of (22), (23), and (24), with periodic boundary conditions, computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (solid curve), a two-node block KSS method with four block Gaussian nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are third-order accurate in time.

Table I. Estimates of relative error at $t = 1$ in the solution of (22), (23), and (24), with periodic boundary conditions.

N	Δt	KSS-equi	KSS-Gauss	Lanczos
128	1	3.591e−003	4.298e−004	1.386e−001
	1/2	3.321e−004	1.973e−005	1.384e−001
	1/4	3.625e−005	1.483e−006	9.498e−002
	1/8	4.256e−006	1.419e−007	4.903e−002
	1/16	5.120e−007	1.397e−008	3.043e−002
256	1	3.590e−003	4.298e−004	1.935e−001
	1/2	3.321e−004	1.973e−005	1.933e−001
	1/4	3.616e−005	1.483e−006	1.297e−001
	1/8	4.168e−006	1.419e−007	6.898e−002
	1/16	4.285e−007	1.397e−008	1.133e−002

Computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (KSS-equi), a two-node block KSS method with four block Gaussian nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps Δt.

DOI: 10.1002/nla

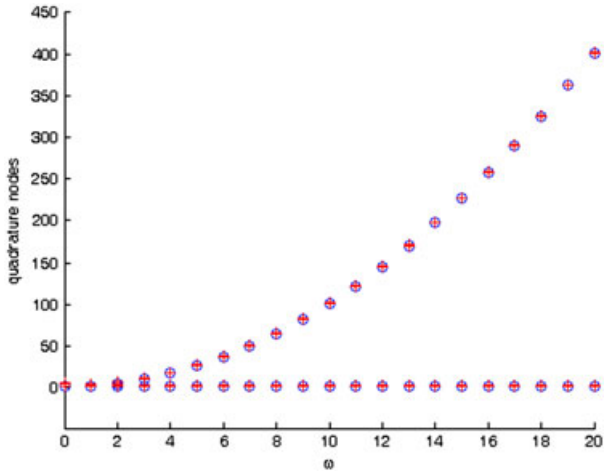

Figure 2. Quadrature nodes used by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes prescribed by (18) (blue circles; only extremal nodes shown) and a two-node block KSS method with four block Gaussian nodes (red crosses) applied to the problems (22), (23), and (24).

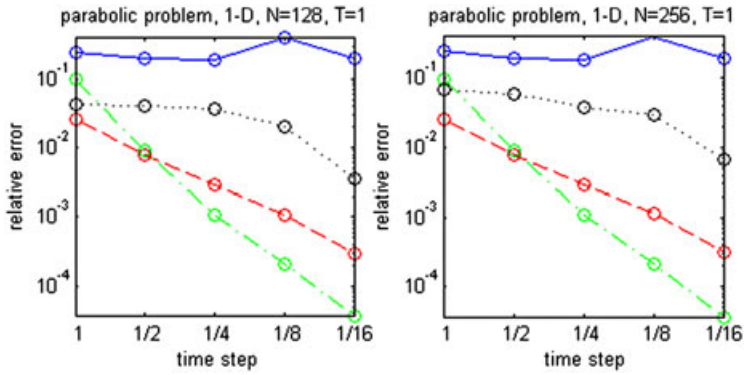

Figure 3. Estimates of relative error at $t = 1$ in the solution of (22), (25), and (26), with periodic boundary conditions, computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (solid curve) or (19) (dash-dot curve), a two-node block KSS method with four block Gaussian nodes (dashed curve), and a Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are third-order accurate in time.

ing the block Gaussian nodes for each Fourier coefficient, which requires substantially greater computational effort.

We now repeat this experiment of solving (22), except with more oscillatory coefficients

$$p(x) = 1 + \frac{1}{2}\cos x - \frac{1}{4}\sin 2x + \frac{1}{8}\cos 3x,$$
$$q(x) = 1 + \frac{1}{4}\sin x - \frac{1}{4}\cos 2x + \frac{1}{8}\sin 3x - \frac{1}{8}\cos 4x, \tag{25}$$

and initial data

$$u(x,0) = 1 + \frac{3}{10}\cos x - \frac{3}{20}\sin 2x + \frac{3}{40}\cos 3x, \quad 0 < x < 2\pi. \tag{26}$$

The results are shown in Figure 3 and Table II. We see that compared with the case of smoother coefficients and data, the KSS method combined with asymptotic block Lanczos iteration according to (18) is a failure. An explanation for this difference in performance can be found in Figure 4. The

DOI: 10.1002/nla

Table II. Estimates of relative error at $t = 1$ in the solution of (22), (25), and (26), with periodic boundary conditions.

N	Δt	KSS-equi-(18)	KSS-equi-(19)	KSS-Gauss	Lanczos
128	1	2.440e−001	9.833e−002	2.605e−002	4.274e−002
	1/2	1.964e−001	9.097e−003	7.846e−003	4.017e−002
	1/4	1.891e−001	1.068e−003	2.875e−003	3.628e−002
	1/8	3.967e−001	2.099e−004	1.074e−003	2.047e−002
	1/16	1.983e−001	3.676e−005	2.958e−004	3.492e−003
256	1	2.440e−001	9.833e−002	2.604e−002	6.822e−002
	1/2	1.964e−001	9.097e−003	7.838e−003	5.978e−002
	1/4	1.891e−001	1.068e−003	2.868e−003	3.838e−002
	1/8	3.967e−001	2.099e−004	1.134e−003	2.921e−002
	1/16	1.983e−001	3.676e−005	3.126e−004	6.651e−003

Computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (KSS-equi-(18)) or (19) (KSS-equi-(19)), a two-node block KSS method with four block Gaussian nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps Δt.-

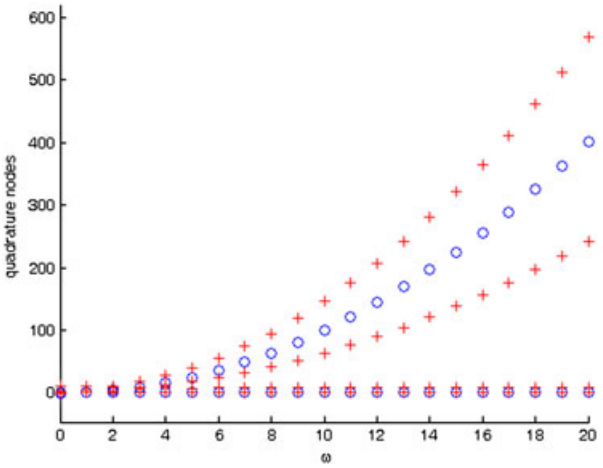

Figure 4. Quadrature nodes used by four-node Krylov subspace spectral (KSS) method with equally spaced nodes prescribed by (18) (blue circles; only extremal nodes shown) and two-node block KSS method with four block Gaussian nodes (red crosses) applied to the problems (22), (25), and (26).

extremal nodes $\lambda_4(\omega)$ are not in agreement with the largest block Gaussian nodes, except in asymptotic growth rate. It is also worth noting that the block Gaussian nodes are not as clustered around specific values as they are in the case of smoother coefficients.

On the basis of the variable leading coefficient, we instead select the extremal nodes according to (19) and solve the same problem. The results are shown in Figure 3 and Table II. We observe that not only does the KSS method with asymptotic block Lanczos iteration yield far more accuracy than when (18) was used, but it is actually *more* accurate than the standard KSS method with block Gaussian nodes. Whereas KSS with block Gaussian nodes converges only superlinearly for the chosen time steps, KSS with equally spaced nodes converges superquadratically. In Figure 5, it can be seen that the extremal nodes of the KSS method with equally spaced nodes are in much better agreement with the extremal block Gaussian nodes, which helps explain the substantial improvement in performance.

4.1.2. Two-dimensional problems. We now consider a parabolic problem in two space dimensions,

$$u_t - \nabla \cdot (p(x, y)\nabla u) + q(x, y)u = 0, \quad 0 < x, y < 2\pi, \quad t > 0, \tag{27}$$

DOI: 10.1002/nla

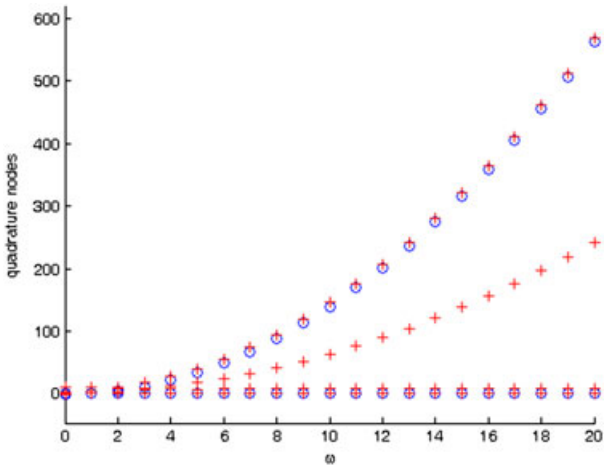

Figure 5. Quadrature nodes used by four-node Krylov subspace spectral (KSS) method with equally spaced nodes prescribed by (19) (blue circles; only extremal nodes shown) and two-node block KSS method with four block Gaussian nodes (red crosses) applied to the problems (22), (25), and (26).

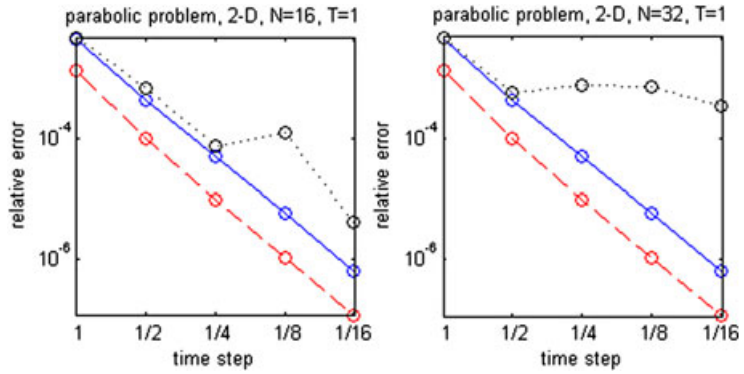

Figure 6. Estimates of relative error at $t = 1$ in the solution of (27), (28), and (29), with periodic boundary conditions, computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (solid curve), a two-node block KSS method with four block Gaussian nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid (per dimension) and various time steps. All methods are third-order accurate in time.

where the smooth coefficients are given by

$$p(x, y) = 1, \quad q(x, y) = \frac{4}{3} + \frac{1}{4}\cos x - \frac{1}{4}\sin y. \tag{28}$$

The initial condition is

$$u(x, y, 0) = 1 + \frac{3}{10}\cos x - \frac{1}{20}\sin 2y, \quad 0 < x, y < 2\pi, \tag{29}$$

and periodic boundary conditions are imposed.

The results are shown in Figure 6 and Table III. We observe essentially the same behavior of all three methods as in the one-dimensional case. The Lanczos-based exponential integrator (2) does not exhibit the theoretically expected third-order accuracy, and its performance is significantly degraded by an increase in the number of grid points. KSS methods, on the other hand, have no such difficulty with the higher resolution and actually do yield third-order accuracy in time, with block Gaussian nodes being more accurate.

DOI: 10.1002/nla

Table III. Estimates of relative error at $t = 1$ in the solution of (27), (28), and (29), with periodic boundary conditions.

N	Δt	KSS-equi	KSS-Gauss	Lanczos
16	1	4.577e−003	1.385e−003	4.822e−003
	1/2	4.455e−004	9.948e−005	6.803e−004
	1/4	4.980e−005	9.785e−006	7.680e−005
	1/8	5.819e−006	1.083e−006	1.293e−004
	1/16	6.203e−007	1.146e−007	4.175e−006
32	1	4.577e−003	1.385e−003	4.702e−003
	1/2	4.455e−004	9.948e−005	5.772e−004
	1/4	4.983e−005	9.785e−006	7.524e−004
	1/8	5.854e−006	1.083e−006	7.363e−004
	1/16	6.603e−007	1.146e−007	3.406e−004

Computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (KSS-equi), a two-node block KSS method with four block Gaussian nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid (per dimension) and various time steps Δt.

We repeat the experiment with more oscillatory coefficients

$$p(x, y) = 1 + \frac{1}{2}\cos(x + y) - \frac{1}{4}\sin(2(x - y)) + \frac{1}{8}\cos(3(x + y)),$$
$$q(x, y) = 1 + \frac{1}{4}\sin x - \frac{1}{4}\cos 2y + \frac{1}{8}\sin 3x - \frac{1}{8}\cos 4y. \tag{30}$$

The initial data

$$u(x, y, 0) = 1 + \frac{3}{10}\cos x - \frac{3}{20}\sin(2(x + y)) + \frac{3}{40}\cos 3x, \quad 0 < x, y < 2\pi, \tag{31}$$

is more oscillatory as well. The results are shown in Figure 7 and Table IV. Once again, we observe the same behavior as in the one-dimensional case: using equally spaced nodes chosen according to (18) yields extremely poor results. All three methods exhibit degradation of accuracy as the number of grid points is increased, including order of accuracy, as the KSS method with block Gaussian nodes is at best second-order accurate.

In an effort to improve accuracy, we take the variation in the leading coefficient $p(x, y)$ into account and prescribe the largest node $\lambda_4(\omega)$ for each Fourier coefficient by using (19) rather

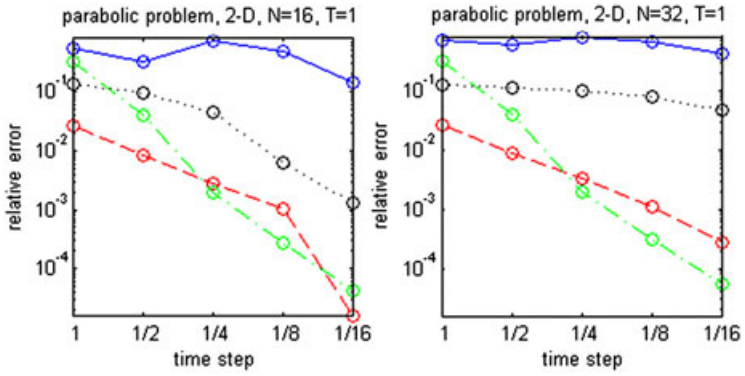

Figure 7. Estimates of relative error at $t = 1$ in the solution of (27), (30), and (31), with periodic boundary conditions, computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (solid curve) or (19) (dash-dot curve), a two-node block KSS method with four block Gaussian nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid (per dimension) and various time steps. All methods are third-order accurate in time.

DOI: 10.1002/nla

Table IV. Estimates of relative error at $t = 1$ in the solution of (27), (30), and (31), with periodic boundary conditions.

N	Δt	KSS-equi-(18)	KSS-equi-(19)	KSS-Gauss	Lanczos
16	1	5.510e−001	3.204e−001	2.653e−002	1.351e−001
	1/2	3.305e−001	4.005e−002	8.202e−003	9.480e−002
	1/4	7.355e−001	1.909e−003	2.836e−003	4.589e−002
	1/8	4.712e−001	2.761e−004	1.039e−003	6.060e−003
	1/16	1.388e−001	4.289e−005	1.581e−005	1.270e−003
32	1	6.962e−001	3.204e−001	2.645e−002	1.222e−001
	1/2	5.798e−001	3.972e−002	8.585e−003	1.109e−001
	1/4	8.134e−001	1.968e−003	3.280e−003	9.978e−002
	1/8	6.468e−001	3.171e−004	1.107e−003	8.071e−002
	1/16	4.171e−001	5.503e−005	2.788e−004	4.767e−002

Computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (KSS-equi-(18)) or (19) (KSS-equi-(19)), a two-node block KSS method with four block Gaussian nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid (per dimension) and various time steps Δt.

than (18). The result of this adjustment is shown in Figure 7 and Table IV. As in the one-dimensional case, KSS with equally spaced nodes actually outperforms KSS with block Gaussian nodes, exhibiting superquadratic convergence in time, compared with superlinear for the block Gaussian case. We also note that once again, the convergence of KSS with equally spaced nodes is negligibly affected by an increase in the number of grid points, unlike for KSS with block Gaussian nodes or the Lanczos-based exponential integrator (2).

4.2. Hyperbolic problem

We now apply all three methods to the second-order wave equation

$$u_{tt} = (p(x)u_x)_x - q(x)u, \quad 0 < x < 2\pi, \quad t > 0, \tag{32}$$

with smooth coefficients $p(x)$ and $q(x)$ defined in (23). The initial data are

$$u(x,0) = 1 + \frac{3}{10}\cos x - \frac{1}{20}\sin 2x, \quad u_t(x,0) = \frac{1}{2}\sin x + \frac{2}{25}\cos 2x, \quad 0 < x < 2\pi, \tag{33}$$

and periodic boundary conditions are imposed. KSS methods are applied to the wave equation by reducing it to a first-order system and then computing both the solution and its time derivative. Performing K block Lanczos iterations each on the solution and its time derivative, which corresponds to $2K$ scalar quadrature nodes for each, yields $O(\Delta t^{4K-2})$ accuracy. Details can be found in [9, 10].

The results are shown in Figure 8 and Table V. Because of the number of scalar nodes and the second-order derivative with respect to time, all three methods are theoretically sixth-order accurate in time for sufficiently small time steps, but for the time steps chosen in this experiment, only the two variations of KSS methods come close to exhibiting such convergence and do so independently of the spatial resolution, whereas the accuracy of the Lanczos-based exponential integrator once again deteriorates significantly from the increase in the number of grid points. We also note that although KSS with block Gaussian nodes is again more accurate than KSS with equally spaced nodes, as in the parabolic case, the gap between the error estimates for the two methods is much smaller in the hyperbolic case.

We now examine the performance over a longer time interval, with a final time of $t = 10$ rather than $t = 1$, and using time steps that are 10 times as large. Although we do not observe the same order of convergence in the two variations of KSS methods (greater than fourth-order accuracy on average), both methods still perform quite well in spite of the large time steps that are at least 25 times the CFL limit in the case of $N = 256$ grid points, whereas the Lanczos-based exponential integrator is unable to produce a solution with a reasonable accuracy. The results are shown in Figure 9 and Table VI.

DOI: 10.1002/nla

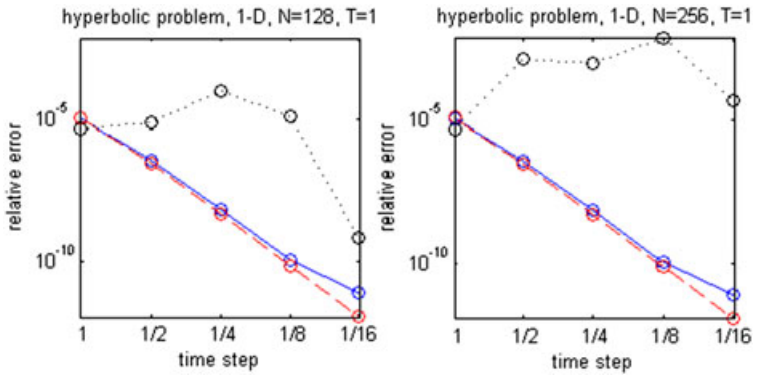

Figure 8. Estimates of relative error at $t = 1$ in the solution of (32), (23), and (33), with periodic boundary conditions, computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (solid curve), a two-node block KSS method with four block Gaussian nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are sixth-order accurate in time.

Table V. Estimates of relative error at $t = 1$ in the solution of (32), (23), and (33), with periodic boundary conditions.

N	Δt	KSS-equi	KSS-Gauss	Lanczos
128	1	1.124e−005	1.197e−005	4.831e−006
	1/2	3.976e−007	3.194e−007	7.661e−006
	1/4	7.132e−009	5.008e−009	1.015e−004
	1/8	1.144e−010	7.465e−011	1.349e−005
	1/16	8.099e−012	1.114e−012	7.373e−010
256	1	1.124e−005	1.197e−005	4.877e−006
	1/2	3.976e−007	3.194e−007	1.291e−003
	1/4	7.132e−009	5.008e−009	9.163e−004
	1/8	1.143e−010	7.465e−011	7.380e−003
	1/16	7.888e−012	1.114e−012	4.456e−005

Computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (KSS-equi), a two-node block KSS method with four block Gaussian nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps Δt.

We now solve the second-order wave equation with the more oscillatory coefficients defined in (25) and initial data

$$\begin{aligned} u(x,0) &= 1 + \frac{3}{10}\cos x - \frac{3}{20}\sin 2x + \frac{3}{40}\sin 3x, \\ u_t(x,0) &= \frac{1}{2}\sin x + \frac{1}{4}\cos 2x - \frac{1}{8}\sin 3x, \quad 0 < x < 2\pi, \end{aligned} \tag{34}$$

with periodic boundary conditions. The KSS method with equally spaced nodes employs (18) to choose the extremal nodes. The results are shown in Figure 10 and Table VII. The performance of all three method is dismal, with error actually *increasing* as the time step decreases for $N = 256$ grid points. This contrasts with the parabolic case, in which the KSS method with block Gaussian nodes and the Lanczos-based exponential integrator were both able to deliver reasonable accuracy, but in this hyperbolic problem, higher-frequency components in the solution persist over time instead of being exponentially damped, thus highlighting the importance of choosing nodes wisely for such problems.

In the interest of choosing nodes more wisely, we turn to (19) for prescribing the largest node for each Fourier coefficient, as in the parabolic case. The results of this strategy for the hyperbolic case are shown in Figure 10 and Table VII. We observe that the performance of the KSS method

DOI: 10.1002/nla

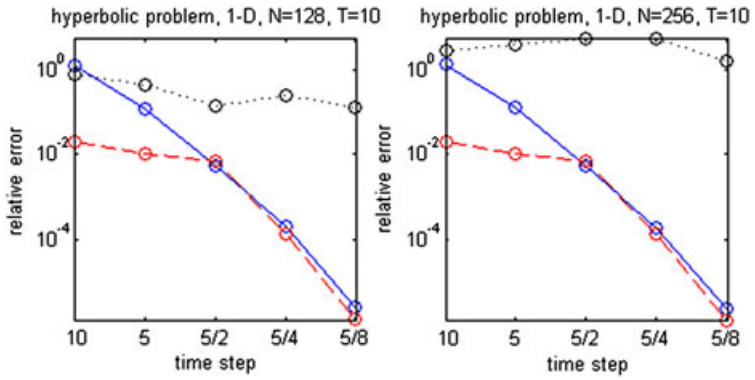

Figure 9. Estimates of relative error at $t = 10$ in the solution of (32), (23), and (33), with periodic boundary conditions, computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (solid curve), a two-node block KSS method with four block Gaussian nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are sixth-order accurate in time.

Table VI. Estimates of relative error at $t = 10$ in the solution of (32), (23), and (33), with periodic boundary conditions.

N	Δt	KSS-equi	KSS-Gauss	Lanczos
128	10	1.339e+000	2.019e−002	7.869e−001
	5	1.259e−001	1.051e−002	4.492e−001
	5/2	5.229e−003	7.018e−003	1.412e−001
	5/4	1.940e−004	1.392e−004	2.532e−001
	5/8	2.371e−006	1.190e−006	1.301e−001
256	10	1.339e+000	2.019e−002	2.874e+000
	5	1.259e−001	1.051e−002	4.020e+000
	5/2	5.229e−003	7.018e−003	5.556e+000
	5/4	1.940e−004	1.392e−004	5.545e+000
	5/8	2.371e−006	1.190e−006	1.628e+000

Computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (KSS-equi), a two-node block KSS method with four block Gaussian nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps Δt.

with equally spaced nodes is drastically improved compared with the previous experiment in which (18) was used to choose the extremal nodes. However, as the time step decreases, an increase in the relative error estimate occurs at the higher resolution, after previously exhibiting approximately fifth-order accuracy in time. Therefore, further improvement in the node selection scheme is required for robustness.

We note that in Figure 5, the largest nodes for the KSS method with equally spaced nodes are still slightly smaller than those of the KSS method with block Gaussian nodes. We therefore artificially 'force' the largest nodes for the equally spaced case to be slightly larger by using (21). The nodes of the two variations of the KSS method are shown in Figure 11, and the results of this strategy are shown in Figure 10 and Table VII. The KSS method with equally spaced nodes no longer exhibits the degradation in accuracy as the number of grid points increases and consistently exhibits greater than fifth-order accuracy in time, improving toward the theoretically expected sixth-order accuracy as Δt decreases. The insensitivity of the error to increase in the number of grid points is finally achieved in the hyperbolic case for oscillatory coefficients and data, as it had been for other 'nicer' problems.

DOI: 10.1002/nla

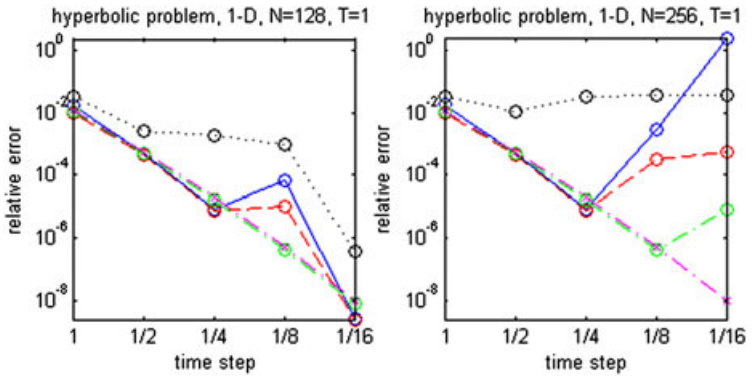

Figure 10. Estimates of relative error at $t = 1$ in the solution of (32), (25), and (34), with periodic boundary conditions, computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (solid curve) or (19) (dash-dot curve with circles) or (21) (dash-dot curve with $\times$s), a two-node block KSS method with four block Gaussian nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are sixth-order accurate in time.

Table VII. Estimates of relative error at $t = 1$ in the solution of (32), (25), and (34), with periodic boundary conditions.

N	Δt	KSS-equi-(18)	KSS-equi-(19)	KSS-equi-(21)	KSS-Gauss	Lanczos
128	1	1.750e−002	1.115e−002	1.151e−002	9.667e−003	3.133e−002
	1/2	4.911e−004	5.051e−004	5.712e−004	4.455e−004	2.586e−003
	1/4	7.951e−006	1.675e−005	2.008e−005	7.362e−006	1.830e−003
	1/8	6.793e−005	4.083e−007	4.997e−007	1.046e−005	9.908e−004
	1/16	2.524e−009	7.468e−009	9.197e−009	2.394e−009	3.646e−007
256	1	1.750e−002	1.115e−002	1.151e−002	9.667e−003	3.133e−002
	1/2	4.911e−004	5.051e−004	5.712e−004	4.455e−004	1.040e−002
	1/4	7.951e−006	1.675e−005	2.008e−005	7.395e−006	3.121e−002
	1/8	2.770e−003	4.083e−007	4.997e−007	3.067e−004	3.408e−002
	1/16	2.335e+000	8.018e−006	9.574e−009	5.368e−004	3.467e−002

Computed by a four-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (KSS-equi-(18)), (19) (KSS-equi-(19)), or (21) (KSS-equi-(21)), a two-node block KSS method with four block Gaussian nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps Δt.

4.3. Performance

We now compare the efficiency of a KSS method with equally spaced nodes against a Lanczos-based exponential integrator, as in (2), that is allowed to proceed to convergence, instead of being restricted to a Krylov subspace of the same dimension. We also compare to a preconditioned Lanczos iteration that uses a restricted denominator (RD)-rational approximation of the matrix exponential [3,4], which is designed to reduce the number of Lanczos iterations needed for convergence. They are implemented in MATLAB (The MathWorks Inc., Natick, MA, USA) and executed on a Dell Inspiron 6400/E1505 notebook with a 2.00-GHz Core 2 Duo processor (Dell Inc., Round Rock, TX, USA). Table VIII reports the execution time, in seconds, for all three methods applied to the problems (22), (23), and (24) to compute the solution at $t = 1$ on 512-point and 1024-point grids. In Figure 12, the execution time is plotted against the relative error estimates from Table VIII.

It can be seen that a KSS method with equally spaced quadrature nodes prescribed through asymptotic block Lanczos iteration yields much greater accuracy per unit of time than the Lanczos-based approximation (2). The performance of the KSS method is comparable with that of the RD-rational approximation when a larger time step is used, but for smaller time steps, the KSS method is more

DOI: 10.1002/nla

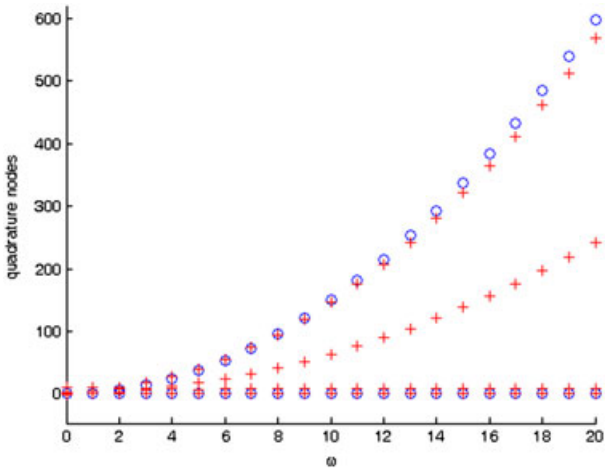

Figure 11. Quadrature nodes used by four-node Krylov subspace spectral (KSS) method with equally spaced nodes prescribed by (21) (blue circles; only extremal nodes shown) and two-node block KSS method with four block Gaussian nodes (red crosses) applied to the problems (32), (25), and (34).

efficient. It is interesting to note that as Δt decreases, the number of Lanczos iterations needed by the RD-rational approximation for convergence actually increases, whereas a decrease occurs for the standard Lanczos iteration. However, both Lanczos-based methods require higher-dimensional Krylov subspaces than the KSS method, and the RD-rational approximation incurs the additional expense of solving a system of equations during each iteration.

It is also interesting to note that both the KSS method and the RD-rational approximation exhibit favorable scalability properties in this experiment, as the execution time increases linearly with the number of grid points, as the dimensions of the Krylov subspaces on both grids are essentially the same. On the other hand, the approximation of (2) does not scale as effectively as an increase in the number of grid points leads to a corresponding increase in the number of iterations needed.

In addition to the Krylov subspace generated by the solution from the previous time step, which all three methods require and whose dimensions are indicated in Table VIII, the KSS method requires storage for $2NK$ scalars, where N is the total number of grid points and K is the number of quadrature nodes per component of the solution. This storage is needed for the nodes themselves and

Table VIII. Execution times, in seconds, and average Krylov subspace dimensions per time step, for the solution of (22), (23), and (24) at $t = 1$, with periodic boundary conditions.

		Execution time (s)			Krylov subspace dimension		
N	Δt	KSS-equi	RD-rational	Lanczos	KSS-equi	RD-rational	Lanczos
512	1	0.031	0.22	1.8	5	9.1	62.9
	0.5	0.063	0.41	1.5	5	9.2	45.6
	0.25	0.14	0.97	1.0	5	11.2	31.2
	0.125	0.25	2.3	0.94	5	12.9	21.8
	0.0625	0.47	4.4	0.97	5	12.6	15.1
1024	1	0.047	0.41	19.4	5	9.1	123.8
	0.5	0.094	0.83	14.2	5	9.5	86.7
	0.25	0.25	2.0	9.1	5	11.5	60.2
	0.125	0.42	4.7	7.5	5	13.0	41.5
	0.0625	0.91	9.0	4.4	5	12.8	28.3

Computed by a five-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (KSS-equi), a restricted denominator (RD)-rational approximation of the matrix exponential [3, 4], and Lanczos iteration as described in (2) (Lanczos) on 512-point and 1024-point grids and various time steps Δt. The KSS method is fourth-order accurate in time; the exponential integrators iterate to convergence during each time step.

DOI: 10.1002/nla

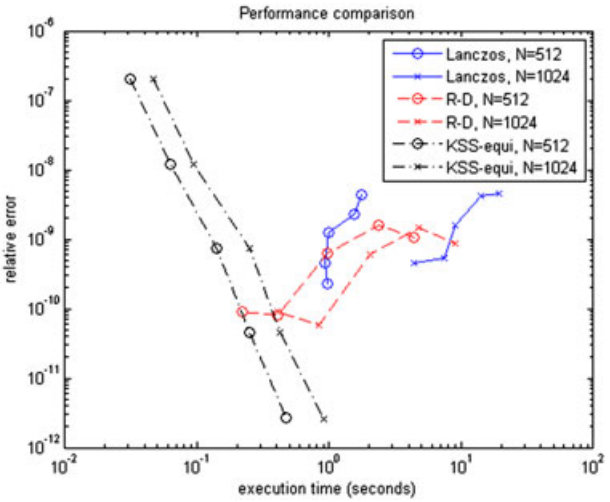

Figure 12. Plot of relative error versus execution time (in seconds) for the solution of (22), (23), and (24) at $t = 1$, with periodic boundary conditions, computed by a five-node Krylov subspace spectral (KSS) method with equally spaced nodes chosen according to (18) (KSS-equi), a restricted denominator (RD)-rational approximation of the matrix exponential [3, 4], and Lanczos iteration as described in (2) (Lanczos) on 512-point and 1024-point grids and various time steps Δt. The KSS method is fourth-order accurate in time; the exponential integrators iterate to convergence during each time step.

the coefficients of the N polynomials, each of degree $K - 1$, that interpolate $e^{-\lambda \Delta t}$ (for parabolic problems) at the K nodes associated with each component.

It can be seen from this comparison that the componentwise approximation of the solution operator e^{-Lt} allows a change in the selection of the dimension of the Krylov subspace generated by the solution from the previous time step and, by extension, the number of Arnoldi or Lanczos iterations. In standard exponential integrators, this dimension is determined according to convergence criteria and can therefore vary on the basis of the number of grid points or the time step. In a KSS method, however, the dimension is determined solely by the desired temporal order of accuracy and can therefore remain fixed throughout the entire computation.

5. DISCUSSION

We now discuss ongoing and upcoming research directions pertaining to the expansion of the applicability of KSS methods, including their enhancement through asymptotic analysis of Lanczos iteration as a function of parameters of the chosen basis functions.

5.1. Quadrature node distribution

In this paper, we have limited ourselves to using equally spaced nodes, once the extremal nodes have been selected. However, this restriction is not necessary. For example, we can apply a Chebyshev distribution to the nodes. Figure 13 compares the accuracy of equidistant and Chebyshev distributions, applied to the parabolic problems (22), (23), and (24). For $N = 512$ grid points, the results are nearly identical, but for $N = 2048$, there is a slight degradation of accuracy for the Chebyshev distribution for the smallest time steps, after performing almost identically to the equidistant distribution for larger time steps. Other experiments have suggested that the results can be sensitive to the distribution of the nodes, particularly if some of them are quite clustered. This will be investigated further in future work.

5.2. Adaptive step size control

The time step Δt can be varied adaptively by obtaining an error estimate. This can be accomplished efficiently by computing a second approximate solution that involves one additional quadrature node

DOI: 10.1002/nla

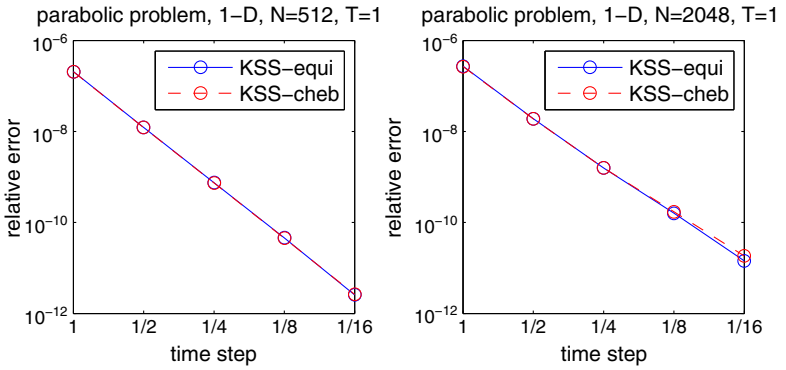

Figure 13. Estimates of relative error at $t = 0.2$ in the solution of (22), (23), and (24), with periodic boundary conditions, computed by a five-node Krylov subspace spectral method with equally spaced nodes (solid curve) and with nodes spaced according to a Chebyshev distribution (dashed curve) on an N-point grid and various time steps. Both methods are fourth-order accurate in time.

for each component of the solution. As it is not necessary to use equal spacing between nodes, this node can be chosen to lie between the existing nodes for the given coefficient. With the use of Newton interpolation to construct the approximation of each component from the chosen nodes, the additional node can be accounted for efficiently to obtain the second approximation. The time step can then be adjusted according to the size of the error estimate as in standard adaptive methods.

5.3. Other spatial discretizations

The main idea behind KSS methods, that higher-order accuracy in time can be obtained by componentwise approximation, is not limited to the enhancement of spectral methods that employ Fourier basis functions. Let A be an $N \times N$ matrix and f be an analytic function. Then, $f(A)\mathbf{v}$ can be computed efficiently by approximation of each component, with respect to a basis $\{\mathbf{u}_j\}_{j=1}^N$, by a block Gaussian quadrature rule with K block nodes (that is, $2K$ scalar nodes) if expressions of the form

$$\mathbf{u}_j^{\mathrm{H}} A^k \mathbf{u}_j, \quad \mathbf{u}_j^{\mathrm{H}} A^k \mathbf{v}, \quad \mathbf{v}^{\mathrm{H}} A^k \mathbf{v} \tag{35}$$

can be computed efficiently for $j = 1, \ldots, N$ and $k = 0, \ldots, 2K-1$ and transformation between the basis $\{\mathbf{u}_j\}_{j=1}^N$ and the standard basis can be performed efficiently.

The first expression in (35) can be computed analytically if the members of the basis $\{\mathbf{u}_j\}_{j=1}^N$ can be simply expressed in terms of j, as in Fourier spectral methods. The other two expressions in (35) are readily obtained from bases for Krylov subspaces generated by $\mathbf{v}$. If A is sparse, then each block recursion coefficient, across *all* components, can be represented as the sum of a sparse matrix and a low-rank matrix [27].

Therefore, it is worthwhile to explore the adaptation of KSS methods to other spatial discretizations for which the recursion coefficients can be computed efficiently. The temporal order of accuracy achieved in the case of Fourier spectral methods applies to such discretizations, as only the measures in the Riemann–Stieltjes integrals are changing, not the integrands. This is demonstrated in [27], in which block KSS methods are adapted to spatial discretization via finite elements. Future work will consist of asymptotic analysis of the recursion coefficients arising from these discretizations in order to develop effective algorithms for prescribing quadrature nodes.

5.4. Application to nonlinear PDE

There are several exponential integrators that can be used for solving stiff systems of nonlinear ODE, such as those proposed in [28, 31–36]. These methods involve the approximation of products of matrix functions and vectors, where the functions include the matrix exponential. A comparison of several of these methods with implicit and explicit integrators is given in [37]. It is worthwhile to explore whether methods of this type can be made more effective by replacing their Arnoldi-based

DOI: 10.1002/nla

or Lanczos-based approaches to approximation of matrix functions with a componentwise approach such as those used by KSS methods, particularly with its enhancement through appropriate prescription of quadrature nodes. The one-node block KSS method, which is first-order accurate in time for diffusion equations, has already been successfully applied to nonlinear diffusion equations from image processing [29, 30]. To achieve higher-order accuracy, an approach such as that described in [37], in which the KSS method is used to approximate the product of a function of the Jacobian with a vector, will be investigated.

6. SUMMARY

In many cases, the solution of systems of ODE can be expressed in terms of the approximate evaluation of a matrix function such as the exponential. Over the last few decades, much advancement has been made in the approximation of quantities of the form $f(A)\mathbf{v}$ and $\mathbf{u}^{\mathrm{T}} f(A)\mathbf{v}$. Techniques for computing $f(A)\mathbf{v}$, such as those described in [1, 2], have been used for several years to solve systems of ODE but can encounter the same difficulties with stiffness that other time stepping methods have.

On the other hand, techniques for computing the bilinear form $\mathbf{u}^{\mathrm{T}} f(A)\mathbf{v}$, normally used for computing selected components of the solution of a system of equations, open the door to rethinking how time stepping is carried out. Because they allow individual attention to be paid to each component of $f(A)\mathbf{v}$ in some basis, they can be used to construct new time stepping methods that are more effective for large-scale problems. KSS methods represent one approach of exploiting this flexibility, and their use of distinct block Gaussian quadrature rules for each component makes them an attractive choice for various problems in terms of their accuracy.

However, it has been established in this paper that the use of block Gaussian rules is neither necessary nor sufficient for maximizing accuracy, and alternative approaches based on prescribing quadrature nodes, whether based on asymptotic block Lanczos iteration or not, cannot only achieve comparable or greater accuracy than block Gaussian rules but can also do so with much greater efficiency. Therefore, it is worthwhile to continue the exploration of such componentwise approaches to the solution of stiff systems of ODE, both linear and nonlinear, through the approximation of matrix functions.

REFERENCES

1. Hochbruck M, Lubich C. A Gautschi-type method for oscillatory second-order differential equations. *Numerische Mathematik* 1999; **83**:403–426.
2. Hochbruck M, Lubich C. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal of Numerical Analysis* 1996; **34**:1911–1925.
3. Moret I, Novati P. RD-rational approximation of the matrix exponential operator. *BIT Numerical Mathematics* 2004; **44**:595–615.
4. van den Eshof J, Hochbruck M. Preconditioning Lanczos approximations to the matrix exponential. *SIAM Journal of Scientific Computing* 2006; **27**:1438–1457.
5. Lambers JV. Derivation of high-order spectral methods for time-dependent PDE using modified moments. *Electronic Transactions on Numerical Analysis* 2008; **28**:114–135.
6. Lambers JV. Enhancement of Krylov subspace spectral methods by block Lanczos iteration. *Electronic Transactions on Numerical Analysis* 2008; **31**:86–109.
7. Lambers JV. Krylov subspace methods for variable-coefficient initial-boundary value problems. *PhD Thesis*, Stanford University, 2003. Available from: http://www.math.usm.edu/lambers/pub/thesis.pdf.
8. Lambers JV. Krylov subspace spectral methods for variable-coefficient initial-boundary value problems. *Electronic Transactions on Numerical Analysis* 2005; **20**:212–234.
9. Guidotti P, Lambers JV, Sølna K. Analysis of 1-D wave propagation in inhomogeneous media. *Numerical Functional Analysis and Optimization* 2006; **27**:25–55.
10. Lambers JV. An explicit, stable, high-order spectral method for the wave equation based on block Gaussian quadrature. *IAENG Journal of Applied Mathematics* 2008; **38**:333–348. Available from: http://www.iaeng.org/IJAM/issues_v38/issue_4/IJAM_38_4_10.pdf.
11. Lambers JV. Implicitly defined high-order operator splittings for parabolic and hyperbolic variable-coefficient PDE using modified moments. *International Journal of Computational Science* 2008; **2**:376–401.
12. Lambers JV. Krylov subspace spectral methods for the time-dependent Schrödinger equation with non-smooth potentials. *Numerical Algorithms* 2009; **51**:239–280.

DOI: 10.1002/nla

13. Lambers JV. A multigrid block krylov subspace spectral method for variable-coefficient elliptic PDE. *IAENG Journal of Applied Mathematics* 2009; **39**:236–246. Available from: http://www.iaeng.org/IJAM/issues_v39/issue_4/IJAM_39_4_07.pdf.
14. Lambers JV. Practical implementation of Krylov subspace spectral methods. *Journal of Scientific Computing* 2007; **32**:449–476.
15. Lambers JV. Spectral methods for time-dependent variable-coefficient PDE based on block Gaussian quadrature. In *Spectral and High Order Methods for Partial Differential Equations: Selected Papers from the ICOSAHOM '09 Conference, June 22-26, Trondheim, Norway*, Hesthaven J, Rønquist E (eds). Springer: New York/Heidelberg, 2011; 429–440.
16. Lambers JV. A spectral time-domain method for computational electrodynamics. *Advances in Applied Mathematics and Mechanics* 2009; **1**(6):781–798. Available from: http://www.global-sci.org/aamm/volumes/v1n6/pdf/16-781.pdf.
17. Golub GH, Meurant G. Matrices, moments and quadrature. In *Proceedings of the 15th Dundee Conference,* June–July 1993, Griffiths DF, Watson GA (eds). Longman Scientific & Technical: Harlow, Essex, England, 1994.
18. Dahlquist G, Eisenstat SC, Golub GH. Bounds for the error of linear systems of equations using the theory of moments. *Journal of Mathematical Analysis and Applications* 1972; **37**:151–166.
19. Golub GH. Some modified matrix eigenvalue problems. *SIAM Review* 1973; **15**:318–334.
20. Golub GH. Bounds for matrix moments. *Rocky Mountain Journal of Mathematics* 1974; **4**:207–211.
21. Davis P, Rabinowitz P. *Methods of Numerical Integration*, 2nd Ed. Academic Press: New York, 1984.
22. Gautschi W. Construction of Gauss–Christoffel quadrature formulas. *Mathematics of Computation* 1986; **22**: 251–270.
23. Gautschi W. Orthogonal polynomials—constructive theory and applications. *Journal of Computational and Applied Mathematics* 1985; **12/13**:61–76.
24. Golub GH, Welsch J. Calculation of Gauss quadrature rules. *Mathematics of Computation* 1969; **23**:221–230.
25. Atkinson K. *An Introduction to Numerical Analysis*, 2nd Ed. John Wiley & Sons: Hoboken, NJ, 1989.
26. Golub GH, Underwood R. The block Lanczos method for computing eigenvalues. In *Mathematical Software III*, Rice J (ed.). Academic Press: New York, 1977; 361–377.
27. Lambers JV. Solution of time-dependent PDE through component-wise approximation of matrix functions. *IAENG Journal of Applied Mathematics* 2011; **41**:1–10. Available from: http://www.math.usm.edu/lambers/papers/ijam2010.pdf.
28. Hochbruck M, Lubich C, Selhofer H. Exponential integrators for large systems of differential equations. *SIAM Journal of Scientific Computing* 1998; **19**:1552–1574.
29. Guidotti P, Lambers JV, Wong E. A nonlinear nonlocal diffusion model for color image denoising. In preparation.
30. Guidotti P, Longo K. Two enhanced fourth order diffusion models for image denoising. *Journal of Mathematical Imaging and Vision* 2011; **40**:188–198.
31. Caliari M, Ostermann A. Implementation of exponential Rosenbrock-type integrators. *Applied Numerical Mathematics* 2009; **59**:568–582.
32. Friesner RA, Tuckerman LS, Bornblaser BC, Russo TV. A method for exponential propagation of large systems of stiff nonlinear differential equations. *Journal of Scientific Computing* 1989; **4**:5870–5876.
33. Hochbruck M, Ostermann A. Exponential Runge–Kutta methods for parabolic problems. *Applied Numerical Mathematics* 2005; **53**:323–339.
34. Ostermann A, Thalhammer M, Wright WM. A class of explicit exponential general linear methods. *BIT Numerical Mathematics* 2006; **46**:409–431.
35. Tokman M. Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods. *J. Comput. Phys.* 2006; **213**:748–776.
36. Vaissmoradi N, Malek A, Momeni-Masuleh SH. Error analysis and applications of the Fourier–Galerkin Runge–Kutta schemes for high-order Stiff PDEs. *Journal of Computational and Applied Mathematics* 2009; **231**: 124–133.
37. Loffeld J, Tokman M. Comparative Performance of Exponential, implicit and explicit integrators for stiff systems of ODEs. Submitted.

DOI: 10.1002/nla