Linear Algebra and its Applications 468 (2015) 233–259

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Solution of time-dependent PDE through rapid estimation of block Gaussian quadrature nodes

Elisabeth M. Palchak [a,*], Alexandru Cibotarica [b], James V. Lambers [b]

[a] *Pearl River Community College, 5448 Highway 49 South, Hattiesburg, MS 39401, USA*
[b] *Department of Mathematics, University of Southern Mississippi, 118 College Dr #5045, Hattiesburg, MS 39406, USA*

ARTICLE INFO

Article history:
Received 4 November 2013
Accepted 1 July 2014
Available online 22 July 2014
Submitted by C. Greif

MSC:
65M70
65F60

Keywords:
Lanczos algorithm
Spectral methods
Gaussian quadrature
Heat equation
Wave equation

ABSTRACT

The stiffness of systems of ODEs that arise from spatial discretization of PDEs causes difficulties for both explicit and implicit time-stepping methods. Krylov Subspace Spectral (KSS) methods present a balance between the efficiency of explicit methods and the stability of implicit methods by computing each Fourier coefficient from an individualized approximation of the solution operator of the PDE. While KSS methods are explicit methods that exhibit a high order of accuracy and stability similar to that of implicit methods, their efficiency needs to be improved. A previous asymptotic study of block Lanczos iteration yielded estimates of extremal block Gaussian quadrature nodes for each Fourier component and led to an improvement in efficiency. In this paper, a more detailed asymptotic study is performed in order to rapidly estimate *all* nodes, thus drastically reducing computational expense without sacrificing accuracy. Numerical results demonstrate that the new node estimation scheme does in fact accomplish these aims.

[*] Corresponding author.

http://dx.doi.org/10.1016/j.laa.2014.07.009
0024-3795/© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Advances in computing power over the last several years have allowed for the solution of mathematical models at higher spatial resolution. In such models, systems of ordinary differential equations (ODEs) that come from the spatial discretization of time-dependent partial differential equations (PDEs) have even greater stiffness. This presents challenges for both explicit and implicit time-stepping methods, as explicit methods must use smaller time steps, due to the CFL condition, while implicit methods must solve increasingly ill-conditioned systems of equations, which tends to increase the number of iterations needed by iterative methods. Either way, computational expense is increased nonlinearly with the increase in the amount of data.

For concreteness, we consider a parabolic PDE on the interval $(0, 2\pi)$,

$$u_t + Lu = 0, \quad t > 0, \tag{1}$$

where L is a second order, self-adjoint, positive definite differential operator. Spatial discretization of the previous PDE (1) results in the system of ODEs

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

where A is an $N \times N$ matrix, and $\mathbf{u}(t)$ and $\mathbf{u}_0$ are N-vectors. The solution of the PDE is approximated by finding the exact solution of the system of ODEs, $\mathbf{u}(t) = e^{-At}\mathbf{u}_0$.

Time-stepping methods approximate this product of a matrix function and vector by a polynomial, in the case of explicit methods, or a rational function, in the case of implicit methods. An approach described in [10,11] is to perform Arnoldi or Lanczos iteration to produce a polynomial of A which can then be used as an approximation of the exponential. For example, consider $\mathbf{w} = e^{-At}\mathbf{v}$ for a given symmetric matrix A and vector $\mathbf{v}$. If we apply Lanczos iteration to A with initial vector $\mathbf{v}$, j times, where $j \ll N$, then the jth iteration produces an $N \times j$ matrix X_j with orthonormal columns, and a $j \times j$ symmetric tridiagonal matrix T_j such that $X_j^H A X_j = T_j$. Then we can compute the approximation

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

where each column $\mathbf{x}_k$, $k = 1, \ldots, j$, of X_j is $\mathbf{x}_k = p_{k-1}(A)\mathbf{v}$. The polynomial $p_n(A)$ is of degree n in A, the polynomials p_j are orthonormal, and $\mathbf{w}_j$ is a product of a polynomial in A of degree $j - 1$ and $\mathbf{v}$ [11,14,19].

However, this approach is not practical due to the nature of time-stepping methods. To approximate an exponential function using a polynomial on a large interval requires using a large number of terms. Similarly, approximating a matrix exponential with greatly varying eigenvalues requires many Lanczos vectors. This is the case if A comes from a stiff system, such as systems of ODEs that result in the spatial discretization of time-dependent PDEs. Therefore, this approach is only practical if the eigenvalues are clustered. An alternative is a *component-wise* approach that will approximate the high frequency components and the low frequency components of the solution separately, using approximating polynomials that are tailored to those components.

To that end, block KSS methods [13,15] have been developed. These methods feature explicit time stepping with high-order accuracy, and stability that is characteristic of implicit methods [14]. Previous works on KSS methods [13,15] have yielded promising results, in terms of accuracy and stability. However, these results are achieved through a highly computationally expensive algorithm in which block Lanczos iteration is used with a different initial block for *each* Fourier component of the solution. Because these initial blocks are parameterized using the wave number, much redundant computation can be eliminated to obtain an algorithm that requires $\mathcal{O}(N \log N)$ operations per time step [16], additional improvements in efficiency must be realized in order to make KSS methods competitive with other time-stepping methods.

Most of the computational expenses arise from the need to compute component-dependent nodes and weights of block Gaussian quadrature rules [13,14]. In [14], a much faster variant of KSS methods was introduced, in which the quadrature nodes are prescribed based on estimates of the extremal nodes of these block Gaussian rules that are obtained through a rudimentary asymptotic analysis of the recursion coefficients produced by block Lanczos iteration [7]. The remaining nodes were prescribed to be equally spaced between the estimated extremal nodes. In this variation of block KSS methods, the nodes are computed very rapidly, because block Lanczos iteration is not actually used, for any Fourier component. The nodes are simply prescribed, instead of obtained by computing the eigenvalues of a block tridiagonal matrix produced by block Lanczos iteration. However, because only the extremal nodes for each Fourier component are accurate approximations of the nodes used in block Gaussian quadrature, this approach to improving efficiency does not necessarily preserve the accuracy of the original block KSS methods.

In this paper, a more thorough asymptotic analysis is performed that provides substantial new insight into the computation of the block Gaussian nodes. This results in a simple method for accurately estimating *all* of the nodes for each Fourier component, not just the extremal ones, and therefore a dramatically more efficient implementation of the central idea behind block KSS methods that, unlike the approach used in [14], also preserves the accuracy of the original version of these methods presented in [13,15]. It can be seen that compared to traditional Krylov subspace-based approaches to computing $f(A)\mathbf{v}$ [10–12,18], the approach used by KSS, when accelerated in this manner, exhibits not only highly favorable behavior in terms of accuracy and stability, but also scalability with respect to the number of grid points used in the spatial discretization of the underlying PDE.

The outline of this paper is as follows. Section 2 provides an overview of block KSS methods. In Section 3, asymptotic analysis of the recursion coefficients produced by block Lanczos is performed, and used to demonstrate how the eigenvalue problem for the computation of quadrature nodes approximately decouples in high frequencies. Then, it is demonstrated how this decoupling can be exploited to rapidly estimate these nodes. Section 4 consists of numerical experiments that compare the accuracy and efficiency of the original and accelerated KSS methods, as well as the use of (2). Conclusions and ideas for future work are given in Section 5.

2. Krylov subspace spectral methods

We consider the parabolic PDE (1) on the interval $(0, 2\pi)$, where L is a second-order, self-adjoint, positive definite differential operator. Appropriate initial conditions and periodic boundary conditions are imposed. As such, the solution can be represented using a Fourier series

$$u(x,t) = \frac{1}{\sqrt{2\pi}} \sum_{\omega=-\infty}^{\infty} e^{i\omega x} \hat{u}(\omega, t),$$

where ω represents the wave number of each Fourier component, and the Fourier coefficients $\hat{u}(\omega, t)$ are given by

$$\hat{u}(\omega,t) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, u(x,t) \right\rangle = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} e^{-i\omega x} u(x,t)\, dx.$$

That is, $\langle \cdot , \cdot \rangle$ denotes the standard inner product of functions on $(0, 2\pi)$.

The idea behind KSS methods is to approximate all Fourier coefficients of the solution independently of one another, using an approximation of the solution operator $e^{-L\Delta t}$ that is, in some sense, optimal for each Fourier coefficient of the solution [19]. Given the computed solution $u(x, t_n)$ at time $t_n = n\Delta t$, each Fourier coefficient of the solution at time t_{n+1} is given by

$$\hat{u}(\omega, t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, \exp[-L\Delta t] u(x, t_n) \right\rangle. \tag{3}$$

Spatial discretization of (3) yields a bilinear form

$$\mathbf{u}^H f(A) \mathbf{v},$$

where $\mathbf{u}$ and $\mathbf{v}$ are N-vectors consisting of the values of $\frac{1}{\sqrt{2\pi}} e^{i\omega x}$ and $u(x, t_n)$, respectively, on a uniform N-point grid, $A = L_N$ is an $N \times N$ symmetric positive definite matrix that comes from discretizing the operator L, and $f(\lambda) = \exp(-\lambda \Delta t)$. The matrix A has real positive eigenvalues

$$b = \mu_1 \geq \mu_2 \geq \cdots \geq \mu_N = a > 0,$$

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

$$\mathbf{u}^H f(A)\mathbf{v} = \sum_{j=1}^{N} f(\mu_j)\mathbf{u}^H \mathbf{q}_j \mathbf{q}_j^H \mathbf{v}. \tag{4}$$

As discussed in [6] by Golub and Meurant, the bilinear form (4) can also be regarded as a Riemann–Stieltjes integral

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

where

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

We can then approximate $I[f]$ by Gaussian quadrature to obtain an approximation of the form

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

where we use the Lanczos algorithm to acquire the nodes λ_j, $j = 1, \ldots, K$ and weights w_j, $j = 1, \ldots, K$ [3–5,8]. This Gaussian quadrature rule is exact for polynomials of degree up to $2K - 1$; this is proven in [6, Theorem 3.6] for the case of $\mathbf{u}$ and $\mathbf{v}$ being real vectors, which generalizes to the complex case in a straightforward manner with appropriate complex conjugation.

In the case where $\mathbf{u} \neq \mathbf{v}$, the weights are generally not positive real numbers, which can numerically destabilize the quadrature rule [1]. Alternatively, we consider the block approach

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}^H f(A) \begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}. \tag{5}$$

This matrix can be regarded as a matrix-valued Riemann–Stieltjes integral

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

where $\mu(\lambda)$ is a 2×2 matrix, each entry of which is a measure of the form $\alpha(\lambda)$. Using the most general quadrature formula with K matrix nodes [6,14] we get

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

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

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \sum_{j=1}^{2K} f(\lambda_j)\mathbf{v}_j\mathbf{v}_j^H + \text{error}, \tag{6}$$

where λ_j is a scalar and $\mathbf{v}_j$ is a 2-vector. For our quadrature formula, we obtain the nodes and weights by applying the block Lanczos algorithm [7]:

```
$X_0 = 0$, $R_0 = [\,\mathbf{u} \quad \mathbf{v}\,]$, $R_0 = X_1B_0$ ($QR$ factorization)
for $j = 1, 2, \ldots, K$
    $V = AX_j$
    $M_j = X_j^H V$
    if $j < K$
        $R_j = V - X_{j-1}B_{j-1}^H - X_jM_j$
        $R_j = X_{j+1}B_j$ ($QR$ factorization)
    end
end
```

The block Lanczos algorithm produces 2×2 matrices M_j and B_j, that form a block tridiagonal matrix

$$\mathcal{T}_K = \begin{bmatrix} M_1 & B_1^H & & & \\ B_1 & M_2 & B_2^H & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-2} & M_{K-1} & B_{K-1}^H \\ & & & B_{K-1} & M_K \end{bmatrix}, \tag{7}$$

where each B_j is upper triangular. The resulting block tridiagonal matrix produces the nodes and the weights. The nodes λ_j from (6) are the eigenvalues of $\mathcal{T}_K$, and the "weights" are the 2×2 matrices $\mathbf{v}_j\mathbf{v}_j^H$ from (6), where $\mathbf{v}_j$ is a 2-vector consisting of the first two components of each eigenvector of $\mathcal{T}_K$.

The block KSS method [13,15] for (1) begins by defining

$$R_0 = [\,\hat{\mathbf{e}}_\omega \quad \mathbf{u}^n\,], \tag{8}$$

where $\hat{\mathbf{e}}_\omega$ is a discretization of $\frac{1}{\sqrt{2\pi}}e^{i\omega x}$ and $\mathbf{u}^n$ is a discretization of the approximate solution $u(x,t)$ at time $t_n = n\Delta t$. Next we compute the QR factorization of R_0,

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

which outputs

$$X_1(\omega) = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \frac{\mathbf{u}_\omega^n}{\|\mathbf{u}_\omega^n\|_2} \end{bmatrix} \tag{9}$$

and

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

where

$$\mathbf{u}_\omega^n = \mathbf{u}^n - \hat{\mathbf{e}}_\omega \hat{\mathbf{e}}_\omega^H \mathbf{u}^n = \mathbf{u}^n - \hat{\mathbf{e}}_\omega \hat{u}(\omega, t_n). \tag{10}$$

Then we apply the block Lanczos algorithm [7] to the matrix L_N, which comes from the discretization of L, with initial block $X_1(\omega)$. This produces a block tridiagonal matrix $\mathcal{T}_K$ of the form (7), where every entry of $\mathcal{T}_K$ is a function of ω. Then, at time t_{n+1}, each Fourier coefficient of the solution is

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

An inverse FFT applied to the vector of Fourier coefficients $\hat{\mathbf{u}}^{n+1}$ yields the vector $\mathbf{u}^{n+1}$, which consists of the values of the solution $u(x, t_{n+1})$ at the grid points.

This algorithm has local temporal accuracy $\mathcal{O}(\Delta t^{2K-1})$ for the parabolic problem (1) [13]. Even higher-order accuracy, $\mathcal{O}(\Delta t^{4K-2})$, is obtained for the second-order wave equation [15]. Furthermore, under appropriate assumptions on the coefficients of the differential operator L in (1), the one-node block KSS method is unconditionally stable [13,15].

Compared to the Krylov subspace methods such as those in [10–12,18] and other sources, it would seem that KSS methods would be prohibitively expensive, due to the computation of a large number of Krylov subspaces. However, there are two significant mitigating factors. First, such Krylov subspace methods tend to require a number of Arnoldi or Lanczos iterations that increases with the number of grid points in order to preserve the same level of accuracy, as will be demonstrated later in this paper. Second, all the Krylov subspaces generated by KSS methods are closely related by the wave number ω, which allows elimination of redundant computations [14]. However, there is more that can be done to make the approach used by KSS methods, individual approximations for each Fourier component, more efficient. This is investigated in the next section.

3. Asymptotic analysis of Lanczos iteration

The central idea behind KSS methods is to compute each component of the solution, in some orthonormal basis, using an approximation that is, in some sense, optimal for that component. That is, each component uses its own polynomial approximation of $S(L_N;\Delta t)$, where the function S is based on the solution operator of the PDE (e.g. $S(L_N;\Delta t) = e^{-L_N\Delta t}$ in the case of (1)), and L_N is the discretization of the spatial differential operator. These polynomial approximations are obtained by interpolation of the function $S(\lambda;\Delta t)$ at selected nodes for each component. Then, the computed solution has the form [14]

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

where $D_j(\Delta t)$ is a matrix that is diagonal in the chosen basis. The diagonal entries are the coefficients of these interpolating polynomials in the monomial basis, with each row corresponding to a particular component. In the original block KSS method [13,15], the interpolation points are obtained by performing block Lanczos iteration and then diagonalizing a $2K \times 2K$ matrix – for each component. In this section, we develop a much faster way of obtaining interpolation points, by studying the behavior of block Lanczos in the limit as $|\omega| \to \infty$, where ω is the wave number.

3.1. The block case

As in the previous section, let $\mathbf{u}^n$ be a discretization of the approximate solution $u(x,t)$ at time $t_n = n\Delta t$ on a uniform N-point grid. Then, KSS methods use the initial block $R_0 = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \mathbf{u}^n \end{bmatrix}$, for each $\omega = -N/2+1,\dots,N/2$. We start the first iteration of the block Lanczos algorithm by finding the QR-factorization of R_0:

$$R_0 = X_1B_0,$$

where

$$X_1 = \begin{bmatrix} \hat{\mathbf{e}}_\omega & \frac{\mathbf{u}^n_\omega}{\|\mathbf{u}^n_\omega\|_2} \end{bmatrix} \quad \text{and} \quad B_0 = \begin{bmatrix} 1 & \hat{u}(\omega,t_n) \\ 0 & \|\mathbf{u}^n_\omega\|_2 \end{bmatrix} \tag{11}$$

with $\mathbf{u}^n_\omega$ is defined as in (10). We note that if the solution u is continuous, then as $|\omega| \to \infty$, $|\hat{u}^n(\omega)| \to 0$, so that in the limit B_0 is diagonal.

The next step is to compute

$$M_1 = X_1^H L_N X_1, \tag{12}$$

where the matrix L_N is a spectral discretization of the operator L defined by $Lu = pu_{xx} + q(x)u$, with p being a constant. Substituting the value of X_1 from (11) into (12) yields

$$M_1 = \begin{bmatrix} \omega^2 p + \bar{q} & \frac{\widehat{L_N \mathbf{u}_\omega^n}(\omega)}{\|\mathbf{u}_\omega^n\|_2} \\ \overline{\frac{\widehat{L_N \mathbf{u}_\omega^n}(\omega)}{\|\mathbf{u}_\omega^n\|_2}} & R(L_N, \mathbf{u}_\omega^n) \end{bmatrix},$$

where $\bar{q}$ is the mean of $q(x)$ on $(0, 2\pi)$, $\widehat{L_N \mathbf{u}_\omega^n}(\omega) = \hat{\mathbf{e}}_\omega^H L_N \mathbf{u}_\omega^n$ is the Fourier coefficient of the gridfunction $L_N \mathbf{u}^n$ corresponding to the wave number ω, and $R(L_N, \mathbf{u}_\omega^n) = \frac{\langle \mathbf{u}_\omega^n, L_N \mathbf{u}_\omega^n \rangle}{\langle \mathbf{u}_\omega^n, \mathbf{u}_\omega^n \rangle}$ is the Rayleigh quotient of L_N and $\mathbf{u}_\omega^n$. As $|\omega|$ increases, the Fourier coefficients of a continuous function go to zero; therefore, as long as the solution is sufficiently regular, the non-diagonal entries of M_1 become negligible; that is,

$$M_1 \approx \begin{bmatrix} \omega^2 p + \bar{q} & 0 \\ 0 & R(L_N, \mathbf{u}^n) \end{bmatrix}.$$

Proceeding with the iteration, and neglecting any terms that are Fourier coefficients or are of lower order in ω, we obtain

$$R_1 = L_N X_1 - X_1 M_1 \approx \begin{bmatrix} \tilde{\mathbf{q}} \hat{\mathbf{e}}_\omega & \frac{L_N \mathbf{u}_\omega^n}{\|\mathbf{u}_\omega^n\|_2} - R(L_N, \mathbf{u}_\omega^n) \frac{\mathbf{u}_\omega^n}{\|\mathbf{u}_\omega^n\|_2} \end{bmatrix},$$

where $\mathbf{q}$ is a vector consisting of the value of $q(x)$ at the grid points, $\tilde{\mathbf{q}} = \mathbf{q} - \bar{q}$, and multiplication of vectors is component-wise.

To obtain X_2, we perform the QR-factorization $R_1 = X_2 B_1$. We note that the $(1,2)$ entry of B_1, modulo lower-order terms, is the Fourier coefficient $\hat{v}_1(\omega)$, where

$$\mathbf{v}_1 = \tilde{\mathbf{q}} \left(\frac{L_N \mathbf{u}_\omega^n}{\|\mathbf{u}_\omega^n\|_2} - R(L_N, \mathbf{u}_\omega^n) \frac{\mathbf{u}_\omega^n}{\|\mathbf{u}_\omega^n\|_2} \right).$$

It follows that given sufficient regularity of the solution u, in the limit as $|\omega| \to \infty$, B_1, like B_0, approaches a diagonal matrix. Continuing this process, it can be seen that every (nonzero) off-diagonal entry of M_j or B_j, for $j = 1, 2, \ldots$, is a Fourier coefficient of some function that is a differential operator applied to u. Therefore, as long as the Fourier coefficients of u decay to zero at a sufficiently high rate as $|\omega| \to \infty$, these off-diagonal entries will also decay to zero.

It follows that in this high-frequency limit, the block tridiagonal matrix $\mathcal{T}_K$ produced by block Lanczos applied to R_0 as defined above converges to the matrix that would be obtained by applying "non-block" Lanczos iteration to the two columns of R_0 *separately*, and then alternating rows and columns of the tridiagonal matrices produced by these iterations. Therefore, by reordering the rows and columns of $\mathcal{T}_K$ in such a way that odd-numbered and even-numbered rows and columns are grouped together, we find that the eigenvalue problem for this matrix *decouples*, and the block Gaussian quadrature nodes can be obtained by computing the eigenvalues of these smaller, tridiagonal matrices. For finite ω, we can then use non-block Lanczos to at least estimate the true block Gaussian quadrature nodes.

Before proceeding, we confirm that this decoupling also takes place if the leading coefficient $p(x)$ of L is *not* constant. The blocks B_0 and M_1 are the same as in the case where p is constant, except that the $(1,1)$ entry of M_1 is $\bar{p}\omega^2 + \bar{q}$. We then have

$$R_1 = \left[\,\omega^2 \tilde{\mathbf{p}}\hat{\mathbf{e}}_\omega \quad \tfrac{L_N \mathbf{u}_\omega}{\|\mathbf{u}^n_\omega\|_2} - R(L_N, \mathbf{u}^n_\omega)\tfrac{\mathbf{u}^n_\omega}{\|\mathbf{u}^n_\omega\|_2}\,\right] + \text{lower order terms}, \tag{13}$$

where, as before, $\tilde{p}(x) = p(x) - \bar{p}$, and $\tilde{\mathbf{p}}$ is a vector that contains the values of $\tilde{p}(x)$ from $x_j = j\Delta x$, $j = 0, 1, \ldots, N-1$. It can be seen that as in the case of constant p, when we compute the QR factorization $R_1 = X_2 B_1$, the $(1,2)$ entry of B_1, modulo lower-order terms, will be a Fourier coefficient of

$$\mathbf{w}_1 = \tilde{\mathbf{p}}\left(\frac{L_N \mathbf{u}^n_\omega}{\|\mathbf{u}^n_\omega\|_2} - R\big(L_N, \mathbf{u}^n_\omega\big)\frac{\mathbf{u}^n_\omega}{\|\mathbf{u}^n_\omega\|_2}\right),$$

which will approach 0 as $|\omega| \to \infty$. Continuing this process reveals that the behavior is the same as in the case where p is constant: the eigenvalue problem for $\mathcal{T}_K(\omega)$ decouples in the limit.

3.2. The non-block case

The decoupling observed in the preceding discussion reveals that we can obtain approximations of half of the block Gaussian quadrature nodes for all Fourier components by applying “non-block” Lanczos iteration to the matrix L_N with initial vector $\mathbf{u}$, the computed solution, as is done in Krylov subspace methods such as those described in [10–12]. To estimate the other half of the nodes, we perform an asymptotic analysis of Lanczos iteration applied to L_N with initial vector $\hat{\mathbf{e}}_\omega$. The algorithm for Lanczos iteration is given as follows:

```
$\beta_0 = 0$, $\mathbf{m}_0 = \mathbf{0}$, $\mathbf{m}_1 = \mathbf{u}/\|\mathbf{u}\|_2$
for $j = 1, 2, \ldots, K$
    $\mathbf{v}_j = A\mathbf{m}_j$
    $\alpha_j = \mathbf{m}_j^H \mathbf{v}_j$
    if $j < K$
        $\mathbf{v}_j = \mathbf{v}_j - \beta_{j-1}\mathbf{m}_{j-1} - \alpha_j \mathbf{m}_j$
        $\beta_j = \|\mathbf{v}_j\|_2$
        $\mathbf{m}_{j+1} = \mathbf{v}_j/\beta_j$
    end
end
```

We first consider the case where p is constant. Carrying out 3 iterations, which corresponds to a fifth-order accurate KSS method for a parabolic PDE, we obtain the following recursion coefficients, after neglecting lower-order terms:

$$\begin{bmatrix} \alpha_1 & \overline{\beta_1} & 0 \\ \beta_1 & \alpha_2 & \overline{\beta_2} \\ 0 & \beta_2 & \alpha_3 \end{bmatrix} \approx \begin{bmatrix} p\omega^2 & \|\tilde{\mathbf{q}}\|_2 & 0 \\ \|\tilde{\mathbf{q}}\|_2 & p\omega^2 & 2p|\omega|\|\mathbf{q}_x\|_2/\|\tilde{\mathbf{q}}\|_2 \\ 0 & 2p|\omega|\|\mathbf{q}_x\|_2/\|\tilde{\mathbf{q}}\|_2 & p\omega^2 \end{bmatrix}.$$

It follows that the nodes can be easily estimated as

$$\lambda_{1,\omega} = p\omega^2, \qquad \lambda_{2,\omega}, \lambda_{3,\omega} = p\omega^2 \pm \sqrt{\beta_1^2 + \beta_2^2}. \tag{14}$$

The coefficients α_j are always equal to $p\omega^2$ modulo lower-order terms. To estimate the coefficients β_j for $j > 2$, we note that in general, the updated $\mathbf{v}$ in the above algorithm is given by

$$\mathbf{v}_j = \mathbf{v}_j - \beta_{j-1}\mathbf{m}_{j-1} - \alpha_j\mathbf{m}_j, \quad j \geq 2. \tag{15}$$

Since the last term in (15) will always cancel with the highest-order term in $\mathbf{v}$, if we drop the lower-order terms, then (15) can be written as

$$\mathbf{v}_j \approx 2ip\omega\mathbf{m}_{jx} - \beta_{j-1}\mathbf{m}_{j-1}, \tag{16}$$

where $\mathbf{m}_{jx}$ is a grid function representing the derivative with respect to x of a function represented by $\mathbf{m}_j$. From (16) it follows that for $j \geq 2$, we have

$$\beta_j \approx \|2ip\omega\mathbf{m}_{jx} - \beta_{j-1}\mathbf{m}_{j-1}\|_2$$

which becomes

$$\beta_j = \sqrt{(2p\omega)^2 + \left|\beta_{j-1} - 2ip\omega\mathbf{m}_{j-1}^H\mathbf{m}_{jx}\right|^2}. \tag{17}$$

Thus, (17) gives us the relationship between β_j and β_{j-1}. Since $\mathbf{v}_1 \approx \tilde{\mathbf{q}}\hat{\mathbf{e}}_\omega$, it follows that approximate Lanczos vectors can be obtained by repeatedly differentiating $\tilde{q}(x)$ and orthogonalizing.

We now consider the case in which $p(x)$ is not constant. Expanding until α_3, as before, we obtain the following results. First,

$$\alpha_j \approx \frac{\overline{\mathbf{p}\tilde{\mathbf{p}}^{2(j-1)}}}{\|\tilde{\mathbf{p}}^{2(j-1)}\|_2}\omega^2, \quad j = 1, 2, \ldots.$$

That is, each α_j is ω^2 times a weighted average of $p(x)$, with the weight being an increasing power of $\tilde{p}(x)$, normalized. To estimate the coefficients β_j, we use (15) to obtain

$$\mathbf{v}_j \approx \tilde{\mathbf{p}}\mathbf{m}_j\omega^2 - \beta_{j-1}\mathbf{m}_{j-1}. \tag{18}$$

From (18) it follows that for $n \geq 2$, we have

$$\beta_j \approx \left\|\tilde{\mathbf{p}}\mathbf{m}_j\omega^2 - \beta_{j-1}\mathbf{m}_{j-1}\right\|_2$$

which becomes

$$\beta_j = \sqrt{\left(\|\tilde{\mathbf{p}}\|\omega^2\right)^2 + \left|\beta_{j-1} - \omega^2 \mathbf{m}_{j-1}^H \tilde{\mathbf{p}} \mathbf{m_j}\right|^2}. \tag{19}$$

Thus, (19) gives us the relationship between β_j and β_{j-1}. Since $\mathbf{v}_1 \approx \tilde{\mathbf{p}}\omega^2 \hat{\mathbf{e}}_\omega$, it follows that $\beta_1 \approx \omega^2 \|\tilde{\mathbf{p}}\|_2$, and that, in view of (18), approximate Lanczos vectors can be obtained by repeatedly computing powers of $\tilde{p}(x)$ and orthogonalizing, as previously observed in [19]. Since the estimates of all recursion coefficients have a factor of ω^2, the quadrature nodes for all ω can quickly be estimated by computing the tridiagonal matrix of recursion coefficients, without the factor of ω^2, and computing its eigenvalues.

To summarize, the algorithm for computing $\mathbf{u}^{n+1}$ from $\mathbf{u}^n$ is as follows:

1. Perform K iterations of the symmetric Lanczos algorithm on L_N, with initial vector $\mathbf{u}^n$, and compute the K eigenvalues $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_K$ of the resulting symmetric tridiagonal matrix T_K.
2. Use the asymptotic analysis described in this section to estimate the eigenvalues of the Hermitian tridiagonal matrix $T_{K,\omega}$ obtained by applying K iterations of the symmetric Lanczos algorithm to L_N with initial vector $\hat{\mathbf{e}}_\omega$, for each wave number $\omega = -N/2+1, \ldots, N/2$. These K eigenvalue estimates $\lambda_{1,\omega}, \ldots, \lambda_{K,\omega}$ are functions of ω (see, for example, (14)).
3. Combining the eigenvalues computed in the previous two steps, we obtain approximate block Gaussian quadrature nodes $\lambda_{j,\omega}$ for $j = 1, \ldots, 2K$ and $\omega = -N/2+1, \ldots, N/2$, where for $j > K$, $\lambda_{j,\omega} = \tilde{\lambda}_{j-K}$. For each ω, the nodes $\{\lambda_{j,\omega}\}_{j=1}^{2K}$ are approximate eigenvalues of the block tridiagonal matrix $\mathcal{T}_K$ obtained by performing K iterations of block Lanczos on L_N with initial block (8).
4. From these nodes, compute the coefficients (in power form) of interpolating polynomials $p_{2K-1,\omega}(\lambda)$ of degree $2K-1$ that interpolate $e^{-\lambda\Delta t}$ at the nodes $\lambda_{1,\omega}, \ldots, \lambda_{2K,\omega}$, for $\omega = -N/2+1, \ldots, N/2$. That is, for each $\omega = -N/2+1, \ldots, N/2$, we have

$$p_{2K-1,\omega}(\lambda) = \sum_{j=0}^{2K-1} c_{j,\omega}\lambda^j, \qquad p_{2K-1,\omega}(\lambda_{j,\omega}) = e^{-\lambda_{j,\omega}\Delta_t}.$$

5. For $j = 0, 1, \ldots, 2K-1$, let $C_j = \operatorname{diag}(c_{j,-N/2+1}, \ldots, c_{j,N/2})$. Then, compute $\hat{\mathbf{u}}^{n+1}$, a vector of Fourier coefficients of $\mathbf{u}^{n+1}$, as follows:

$$\hat{\mathbf{u}}^{n+1} = \sum_{j=0}^{2K-1} C_j \mathscr{F}_N L_N^j \mathbf{u}^n,$$

   where the matrix $\mathscr{F}_N$ performs an N-point FFT.
6. Perform an inverse FFT on $\hat{\mathbf{u}}^{n+1}$ to obtain $\mathbf{u}^{n+1}$.

It should be noted that this algorithm can be easily implemented in such a way that all arithmetic is real.

This approach to node estimation can also be applied to operators in higher spatial dimension, or to operators with different boundary conditions, such as homogeneous Dirichlet or Neumann boundary conditions [2]. Furthermore, by expressing all polynomial interpolants in Newton form, with the centers corresponding to the nodes obtained by applying Lanczos iteration to the solution as in other Krylov subspace methods, a block KSS method with rapid node estimation can be viewed as a *correction* of the approximate solution computed using (2), in which frequency-dependent approximation is used. In [2] this approach is exploited to reduce the number of Fourier transforms that must be performed during each time step from $2K$ to K.

4. Numerical results

In this section, we demonstrate the effectiveness of our approach to selecting component-dependent interpolants of the solution operator for PDE. The following three approaches are compared:

- block KSS, as described in [13,15],
- block KSS with rapid node estimation as described in Section 3.2, and
- using (2).

In all comparisons of these three methods, Krylov subspaces of the same dimension are used, to highlight the benefit of using component-wise polynomial approximations of the solution operator rather than a polynomial approximation.

At the end of this section, we will also compare the performance, in terms of both accuracy and efficiency, of block KSS with rapid node estimation against (2), where in the latter method the Krylov subspace dimension is not restricted; rather, Lanczos iteration is performed until convergence is achieved, as in [10–12,17,18]. In all experiments, as the exact solution to these variable-coefficient problems is not known, error is estimated by computing the solution at various time steps and comparing all solutions against the one computed with the smallest time step.

4.1. 1-D parabolic problems

We first demonstrate the accuracy of KSS methods combined with rapid node estimation on a parabolic equation in one space dimension,

$$u_t - \big(p(x)u_x\big)_x + q(x)u = 0, \quad 0 < x < 2\pi,\ t > 0, \tag{20}$$

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

$$p(x) = 1, \qquad q(x) = \frac{4}{3} + \frac{1}{4}\cos x, \tag{21}$$

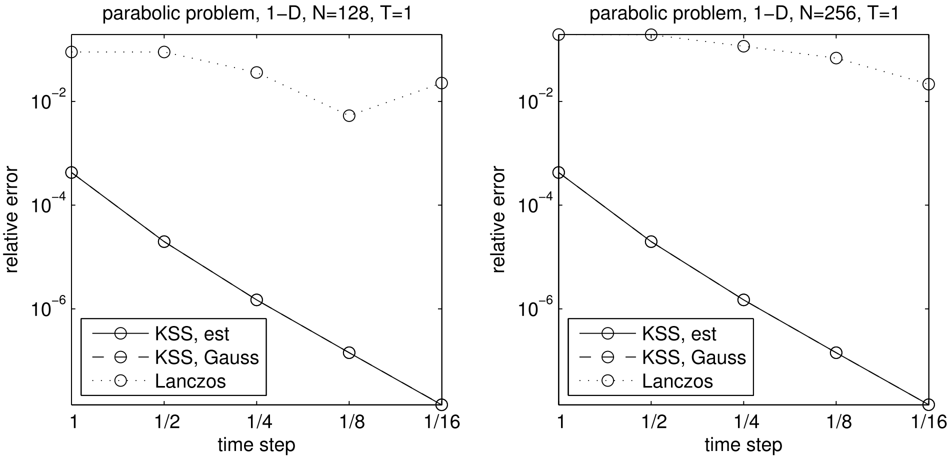

Fig. 1. Estimates of relative error at $t = 1$ in the solution of (20), (21), (22), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (solid curve), a 4-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are 3rd-order accurate in time.

Table 1
Estimates of relative error at $t = 1$ in the solution of (20), (21), (22), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (KSS-est), a 4-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps. All methods are 3rd-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
128	1	4.268e-004	4.268e-004	8.997e-002
	1/2	1.979e-005	1.979e-005	8.973e-002
	1/4	1.495e-006	1.495e-006	3.620e-002
	1/8	1.435e-007	1.435e-007	5.311e-003
	1/16	1.415e-008	1.415e-008	2.258e-002
256	1	4.268e-004	4.268e-004	1.944e-001
	1/2	1.979e-005	1.979e-005	1.937e-001
	1/4	1.495e-006	1.495e-006	1.163e-001
	1/8	1.435e-007	1.435e-007	6.864e-002
	1/16	1.415e-008	1.415e-008	2.149e-002

are chosen to be smooth functions. The initial condition is

$$u(x,0) = 1 + \frac{3}{10}\cos x - \frac{1}{20}\sin 2x, \quad 0 < x < 2\pi, \tag{22}$$

and periodic boundary conditions are imposed.

First, we use a Krylov subspace of dimension 4, so as to achieve 3rd-order accuracy in time. The results are shown in Fig. 1 and Table 1. For both KSS methods, we observe slightly greater than 3rd-order convergence in time, and the error estimates are virtually identical. However, efficiency is not: block KSS with rapid node estimation is approximately 10 times faster than standard block KSS for $N = 128$, and 20 times faster when $N = 256$. While (2) can achieve 3rd-order accuracy as $\Delta t \to 0$, this order of convergence is not exhibited at the time steps used in this experiment, and a slight

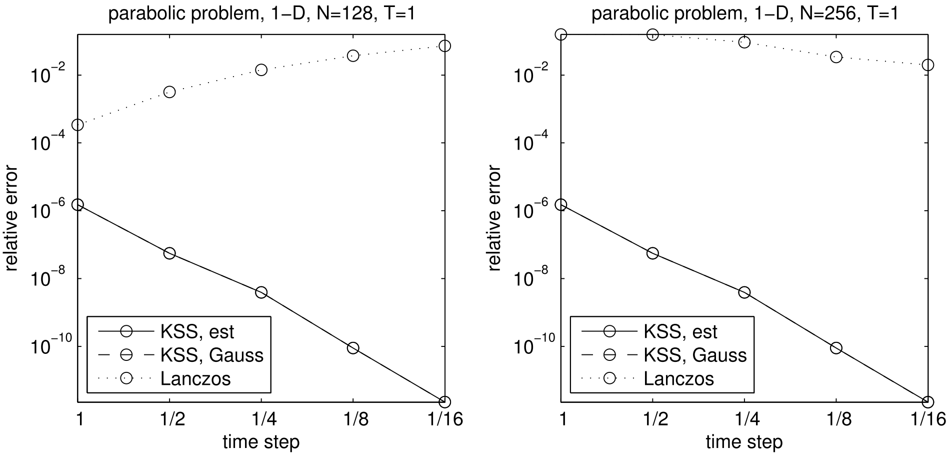

Fig. 2. Estimates of relative error at $t = 1$ in the solution of (20), (21), (22), with periodic boundary conditions, computed by a 6-node KSS method with rapidly estimated nodes (solid curve), a 6-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are 5th-order accurate in time.

Table 2
Estimates of relative error at $t = 1$ in the solution of (20), (21), (22), with periodic boundary conditions, computed by a 6-node KSS method with rapidly estimated nodes (KSS-est), a 6-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps. All methods are 5th-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
128	1	1.518e-006	1.520e-006	3.407e-004
	1/2	5.579e-008	5.579e-008	3.189e-003
	1/4	3.913e-009	3.913e-009	1.425e-002
	1/8	8.989e-011	8.989e-011	3.707e-002
	1/16	2.285e-012	2.286e-012	7.244e-002
256	1	1.518e-006	1.520e-006	1.567e-001
	1/2	5.579e-008	5.579e-008	1.565e-001
	1/4	3.913e-009	3.913e-009	9.261e-002
	1/8	8.989e-011	8.989e-011	3.391e-002
	1/16	2.288e-012	2.285e-012	1.987e-002

degradation of performance is observed as N increases. By contrast, the accuracy of the two KSS methods with the two grid sizes is virtually identical.

Next, we increase the Krylov subspace dimension to 6, so that all methods should be fifth-order accurate. The results are shown in Fig. 2 and Table 2. Again, in terms of accuracy, the performance of both KSS methods is independent of the number of grid points and the node selection scheme; however, as before, KSS with rapid node estimation is approximately 10 times faster for $N = 128$ and 20 times faster for $N = 256$. Using (2) with the same Krylov subspace dimension is not competitive in terms of accuracy; a larger dimension is needed, as will be considered at the end of this section. Unfortunately, both KSS methods achieve only about fourth-order accuracy in time; the expected fifth-order accuracy can only be observed at smaller time steps.

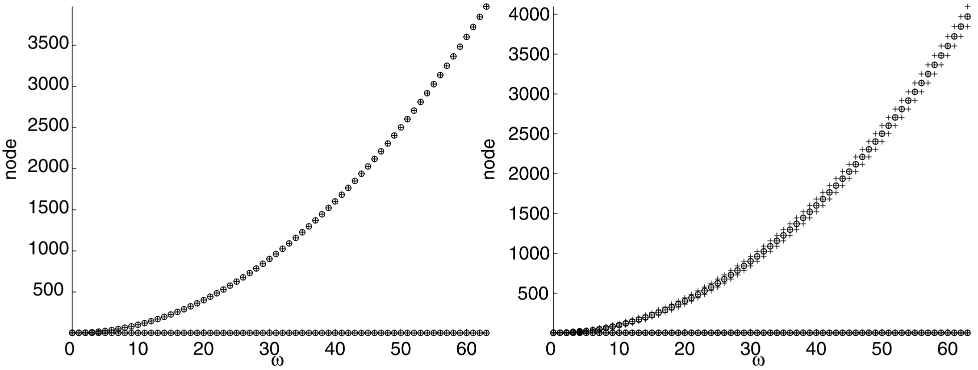

Fig. 3. Quadrature nodes used by block KSS with Gaussian nodes (crosses) and estimated Gaussian nodes (circles) with 2 (left plot) and 3 (right plot) block Lanczos iterations applied to the operator $Lu = -(pu_x)_x + qu$, with p and q defined in (21), for a total of 4 or 6 scalar nodes per frequency component (indicated by ω) in the left and right plots, respectively.

The agreement between the two KSS methods can be explained by the accuracy with which the block Gaussian nodes can be estimated using the analysis of the previous section. Fig. 3 plots the nodes used by both methods for Krylov subspace dimensions 4 and 6.

We now repeat this experiment of solving (20), except with more oscillatory coefficients

$$\begin{aligned} p(x) &= 1 + \frac{1}{4}\cos x - \frac{1}{4}\sin 2x + \frac{1}{8}\cos 3x, \\ q(x) &= 1 + \frac{1}{4}\sin x - \frac{1}{4}\cos 2x + \frac{1}{8}\sin 3x - \frac{1}{8}\cos 4x \end{aligned} \tag{23}$$

and initial data

$$u(x,0) = 1 + \frac{3}{10}\cos x - \frac{3}{20}\sin 2x + \frac{3}{40}\cos 3x, \quad 0 < x < 2\pi. \tag{24}$$

For third-order methods, the results are shown in Fig. 4 and Table 3. At the time steps used, both KSS methods perform identically in terms of accuracy independently of the grid size, but only exhibit second-order accuracy in time; third-order accuracy can be observed at much smaller time steps. As before, the approach of (2) is not competitive with KSS at this Krylov subspace dimension or choice of time step. As with the smoother coefficients, agreement between the two KSS methods can again be explained by the accuracy with which the block Gaussian nodes can be estimated. Fig. 5 plots the nodes used by both methods.

We now increase the Krylov subspace dimension to 6 for all three methods. The results are shown in Fig. 6 and Table 4. For the first time, we observe a disparity in the performance of the original block KSS method and block KSS with rapid node estimation, which is not as accurate in this case, although both KSS methods again yield results that are, for the most part, independent of the grid size. The reason for this disparity lies in the quadrature nodes corresponding to *low*-frequency components. While the estimation

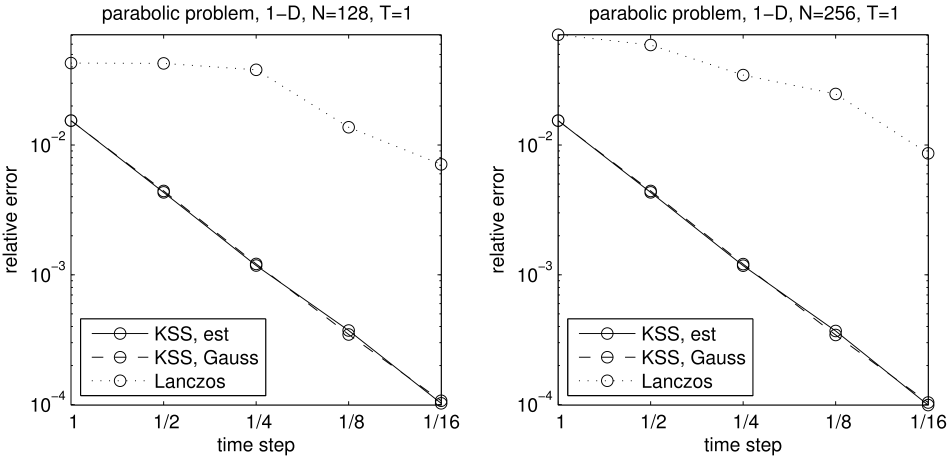

Fig. 4. Estimates of relative error at $t = 1$ in the solution of (20), (23), (24), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (solid curve), a 4-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are 3rd-order accurate in time.

Table 3
Estimates of relative error at $t = 1$ in the solution of (20), (23), (24), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (KSS-est), a 4-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps. All methods are 3rd-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
128	1	1.546e-002	1.541e-002	4.287e-002
	1/2	4.323e-003	4.434e-003	4.262e-002
	1/4	1.180e-003	1.213e-003	3.804e-002
	1/8	3.722e-004	3.466e-004	1.372e-002
	1/16	1.021e-004	1.068e-004	7.108e-003
256	1	1.545e-002	1.540e-002	7.075e-002
	1/2	4.320e-003	4.431e-003	5.914e-002
	1/4	1.177e-003	1.210e-003	3.463e-002
	1/8	3.685e-004	3.435e-004	2.480e-002
	1/16	9.918e-005	1.036e-004	8.636e-003

methods described in Section 3.2 deliver sufficiently accurate approximations of the block Gaussian nodes for nearly all frequencies in the case where the leading coefficient $p(x)$ of the operator L is constant, this is not so when $p(x)$ varies. This is illustrated in Fig. 7. While overall there is good agreement, as shown in the left plot of the figure, this is not the case for low frequencies, as shown in the right plot, compared to the 4-node case presented earlier.

4.2. 1-D hyperbolic problems

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

$$u_{tt} = \big(p(x)u_x\big)_x - q(x)u, \quad 0 < x < 2\pi, \ t > 0, \tag{25}$$

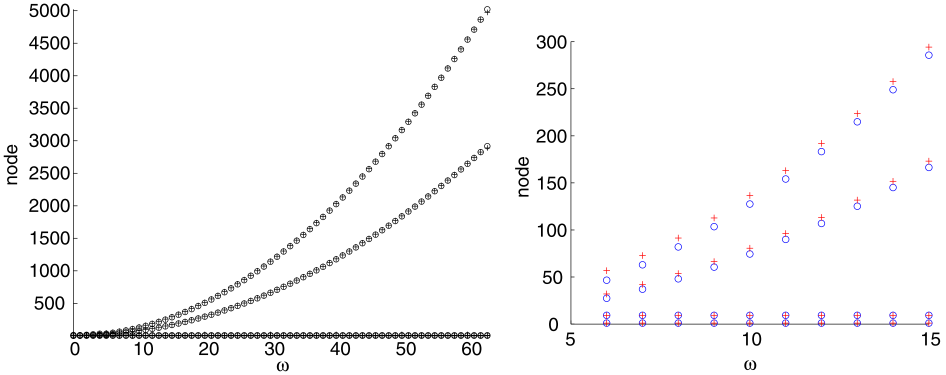

Fig. 5. Quadrature nodes used by block KSS with Gaussian nodes (crosses) and estimated Gaussian nodes (circles) with 2 block Lanczos iterations applied to the operator $Lu = -(pu_x)_x + qu$, with p and q defined in (23), for a total of 4 scalar nodes per frequency component (indicated by ω). The left plot shows frequencies $0 \leq \omega \leq 64$, while the right plot zooms in on frequencies $6 \leq \omega \leq 15$.

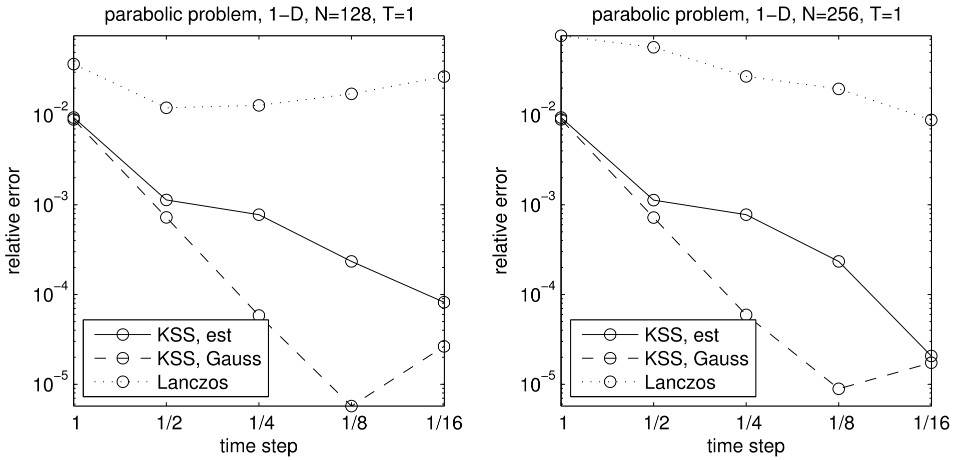

Fig. 6. Estimates of relative error at $t = 1$ in the solution of (20), (23), (24), with periodic boundary conditions, computed by a 6-node KSS method with rapidly estimated nodes (solid curve), a 6-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are 5th-order accurate in time.

with smooth coefficients $p(x)$ and $q(x)$ defined in (21). The initial data is

$$u(x,0) = 1 + \frac{3}{10}\cos x - \frac{1}{20}\sin 2x, \qquad u_t(x,0) = \frac{1}{2}\sin x + \frac{2}{25}\cos 2x, \quad 0 < x < 2\pi, \tag{26}$$

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. Details can be found in [9,15].

The results for 4-dimensional Krylov subspaces are shown in Fig. 8 and Table 5. In this case, KSS methods are generally 6th-order accurate in time [9,15], and for the

Table 4
Estimates of relative error at $t = 1$ in the solution of (20), (23), (24), with periodic boundary conditions, computed by a 6-node KSS method with rapidly estimated nodes (KSS-est), a 6-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps. All methods are 5th-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
128	1	9.370e-003	8.944e-003	3.705e-002
	1/2	1.130e-003	7.235e-004	1.207e-002
	1/4	7.759e-004	5.832e-005	1.281e-002
	1/8	2.337e-004	5.727e-006	1.722e-002
	1/16	8.212e-005	2.645e-005	2.687e-002
256	1	9.367e-003	8.941e-003	7.634e-002
	1/2	1.128e-003	7.213e-004	5.699e-002
	1/4	7.754e-004	5.931e-005	2.695e-002
	1/8	2.337e-004	8.889e-006	1.960e-002
	1/16	2.069e-005	1.737e-005	8.812e-003

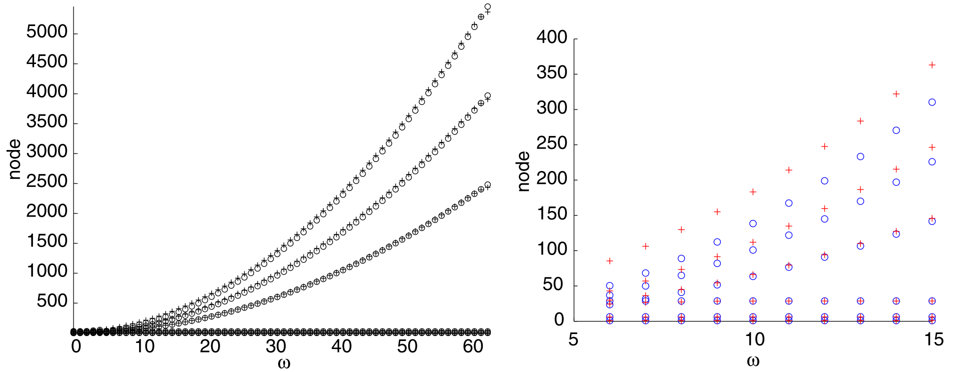

Fig. 7. Quadrature nodes used by block KSS with Gaussian nodes (crosses) and estimated Gaussian nodes (circles) with 3 block Lanczos iterations applied to the operator $Lu = -(pu_x)_x + qu$, with p and q defined in (23), for a total of 6 scalar nodes per frequency component (indicated by ω). The left plot shows frequencies $0 \leq \omega \leq 64$, while the right plot zooms in on frequencies $6 \leq \omega \leq 15$.

most part, that accuracy is achieved in this case. As before, both KSS methods deliver virtually identical results on the two different grids, whereas (2), while more competitive with KSS than in the parabolic case, exhibits a substantial degradation in accuracy as the number of grid points increases.

We now solve the second-order wave equation with the more oscillatory coefficients defined in (23), and initial data

$$\begin{aligned} u(x,0) &= 1 + \frac{3}{10}\cos x - \frac{3}{20}\sin 2x + \frac{3}{40}\sin 3x, \\ u_t(x,0) &= \frac{1}{2}\sin x + \frac{1}{4}\cos 2x - \frac{1}{8}\sin 3x, \quad 0 < x < 2\pi, \end{aligned} \tag{27}$$

with periodic boundary conditions.

The results for 4-dimensional Krylov subspaces are shown in Fig. 9 and Table 6. In this case, KSS methods are generally 6th-order accurate in time [9,15], and for the most part,

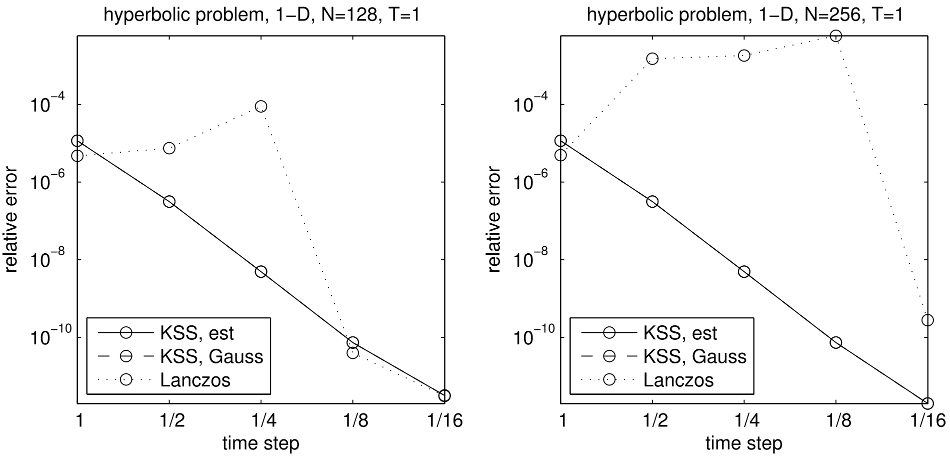

Fig. 8. Estimates of relative error at $t = 1$ in the solution of (25), (21), (26), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (solid curve), a 4-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are 6th-order accurate in time.

Table 5
Estimates of relative error at $t = 1$ in the solution of (25), (21), (26), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (KSS-est), a 4-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps. All methods are 6th-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
128	1	1.167e-005	1.167e-005	4.751e-006
	1/2	3.127e-007	3.127e-007	7.488e-006
	1/4	4.917e-009	4.917e-009	8.900e-005
	1/8	7.360e-011	7.360e-011	3.961e-011
	1/16	3.181e-012	3.181e-012	3.072e-012
256	1	1.167e-005	1.167e-005	4.915e-006
	1/2	3.127e-007	3.127e-007	1.513e-003
	1/4	4.917e-009	4.917e-009	1.814e-003
	1/8	7.346e-011	7.345e-011	5.828e-003
	1/16	1.971e-012	1.971e-012	2.768e-010

that accuracy is achieved in this case. As before, both KSS methods deliver very similar results on the two different grids, although it is notable that in this case, KSS with rapid estimation is slightly more accurate. On the other hand, (2), while more competitive with KSS than in the parabolic case, again exhibits a substantial degradation in accuracy as the number of grid points increases.

4.3. 2-D parabolic problems

We now consider a parabolic problem in two space dimensions,

$$u_t - \nabla \cdot \big(p(x,y)\nabla u\big) + q(x,y)u = 0, \quad 0 < x, y < 2\pi, \ t > 0, \tag{28}$$

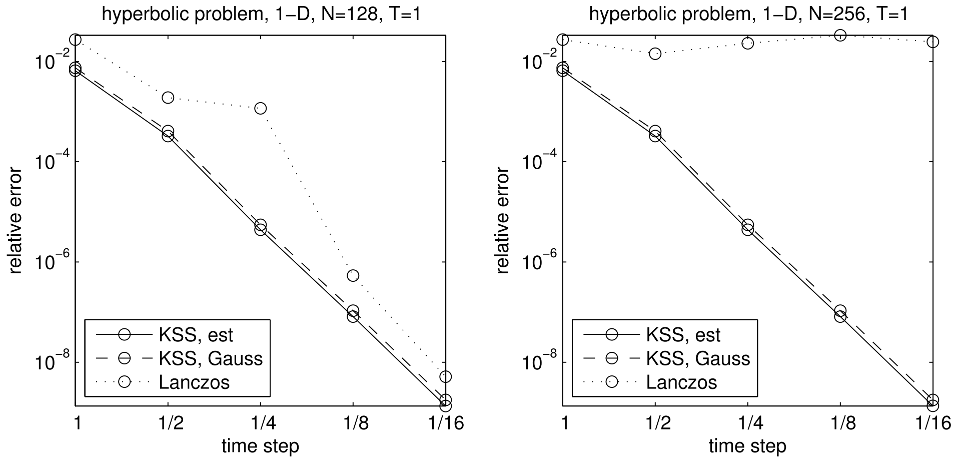

Fig. 9. Estimates of relative error at $t = 1$ in the solution of (25), (23), (27), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (solid curve), a 4-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid and various time steps. All methods are 6th-order accurate in time.

Table 6
Estimates of relative error at $t = 1$ in the solution of (25), (23), (27), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (KSS-est), a 4-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid and various time steps. All methods are 6th-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
128	1	6.543e-003	7.501e-003	2.730e-002
	1/2	3.265e-004	4.084e-004	1.909e-003
	1/4	4.390e-006	5.508e-006	1.168e-003
	1/8	8.150e-008	1.066e-007	5.349e-007
	1/16	1.344e-009	1.768e-009	5.164e-009
256	1	6.543e-003	7.501e-003	2.730e-002
	1/2	3.265e-004	4.084e-004	1.428e-002
	1/4	4.390e-006	5.508e-006	2.315e-002
	1/8	8.150e-008	1.066e-007	3.325e-002
	1/16	1.344e-009	1.768e-009	2.466e-002

where the smooth coefficients are given by

$$p(x,y) = 1, \qquad q(x,y) = \frac{4}{3} + \frac{1}{4}\cos x - \frac{1}{4}\sin y. \tag{29}$$

The initial condition is

$$u(x,y,0) = 1 + \frac{3}{10}\cos x - \frac{1}{20}\sin 2y, \quad 0 < x,y < 2\pi, \tag{30}$$

and periodic boundary conditions are imposed for both dimensions.

The results are shown in Fig. 10 and Table 7. For both KSS methods, we observe approximately 3rd-order convergence in time, and the error estimates are virtually identical. However, efficiency is not: block KSS with rapid node estimation is approximately 150 times faster than standard block KSS for $N = 16$, and 600 times faster when $N = 32$.

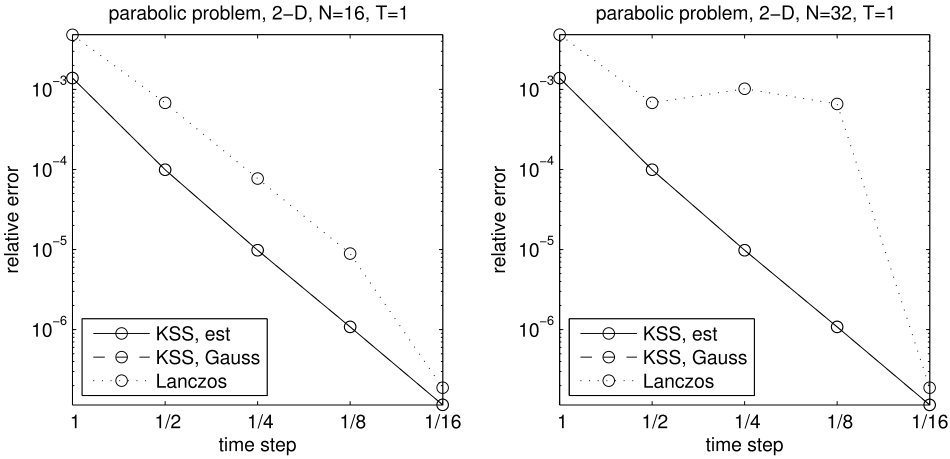

Fig. 10. Estimates of relative error at $t = 1$ in the solution of (28), (29), (30), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (solid curve), a 4-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid (per dimension) and various time steps. All methods are 3rd-order accurate in time.

Table 7
Estimates of relative error at $t = 1$ in the solution of (28), (29), (30), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (KSS-est), a 4-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid (per dimension) and various time steps. All methods are 3rd-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
16	1	1.385e-003	1.385e-003	4.823e-003
	1/2	9.948e-005	9.948e-005	6.809e-004
	1/4	9.785e-006	9.785e-006	7.708e-005
	1/8	1.083e-006	1.083e-006	8.911e-006
	1/16	1.146e-007	1.146e-007	1.884e-007
32	1	1.385e-003	1.385e-003	4.823e-003
	1/2	9.948e-005	9.948e-005	6.811e-004
	1/4	9.785e-006	9.785e-006	1.021e-003
	1/8	1.083e-006	1.083e-006	6.588e-004
	1/16	1.146e-007	1.146e-007	1.884e-007

While (2) can achieve 3rd-order accuracy as $\Delta t \to 0$, this order of convergence is not exhibited at the time steps used in this experiment, and a slight degradation of performance is observed as N increases. By contrast, the accuracy of the two KSS methods with the two grid sizes is virtually identical.

Next, we increase the Krylov subspace dimension to 6, so that all methods should be fifth-order accurate. The results are shown in Fig. 11 and Table 8. Again, in terms of accuracy, the performance of both KSS methods is independent of the number of grid points and the node selection scheme; however, as before, KSS with rapid node estimation is approximately 150 times faster for $N = 16$ and 600 times faster for $N = 32$. Using (2) with the same Krylov subspace dimension is not competitive in terms of accuracy; a larger dimension is needed, as will be considered at the end of this section. Unlike the parabolic 1-D case, both KSS methods actually do achieve the expected fifth-order

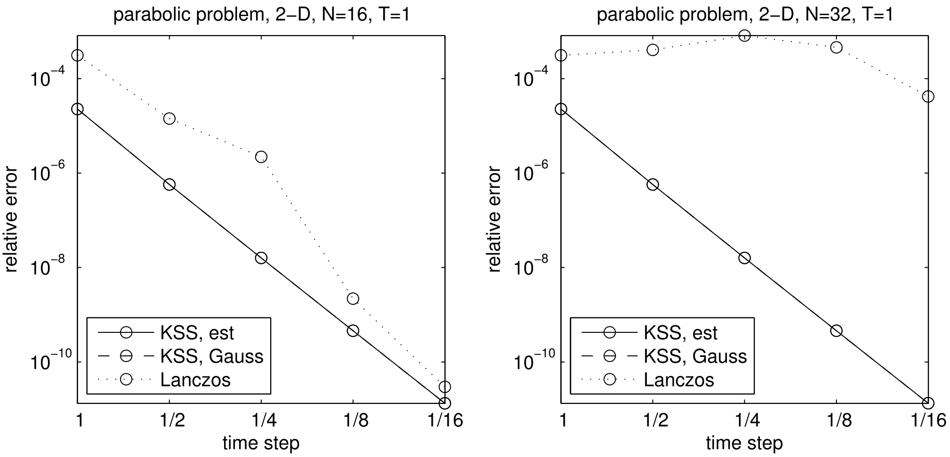

Fig. 11. Estimates of relative error at $t = 1$ in the solution of (28), (29), (30), with periodic boundary conditions, computed by a 6-node KSS method with rapidly estimated nodes (solid curve), a 6-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid (per dimension) and various time steps. All methods are 5th-order accurate in time.

Table 8
Estimates of relative error at $t = 1$ in the solution of (28), (29), (30), with periodic boundary conditions, computed by a 6-node KSS method with rapidly estimated nodes (KSS-est), a 6-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid (per dimension) and various time steps. All methods are 5th-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
16	1	2.272e-005	2.272e-005	3.114e-004
	1/2	5.715e-007	5.715e-007	1.427e-005
	1/4	1.593e-008	1.593e-008	2.204e-006
	1/8	4.591e-010	4.591e-010	2.194e-009
	1/16	1.335e-011	1.335e-011	2.984e-011
32	1	2.272e-005	2.272e-005	3.114e-004
	1/2	5.715e-007	5.715e-007	4.058e-004
	1/4	1.593e-008	1.593e-008	8.116e-004
	1/8	4.591e-010	4.591e-010	4.578e-004
	1/16	1.335e-011	1.335e-011	4.187e-005

accuracy in time. As N increases, the degradation in performance of (2) is far more substantial than in the 4-dimensional case.

We repeat the experiment with more oscillatory coefficients

$$\begin{aligned} p(x,y) &= 1 + \frac{1}{2}\cos(x+y) - \frac{1}{4}\sin\bigl(2(x-y)\bigr) + \frac{1}{8}\cos\bigl(3(x+y)\bigr), \\ q(x,y) &= 1 + \frac{1}{4}\sin x - \frac{1}{4}\cos 2y + \frac{1}{8}\sin 3x - \frac{1}{8}\cos 4y. \end{aligned} \tag{31}$$

The initial data

$$u(x,y,0) = 1 + \frac{3}{10}\cos x - \frac{3}{20}\sin\bigl(2(x+y)\bigr) + \frac{3}{40}\cos 3x, \quad 0 < x,y < 2\pi, \tag{32}$$

is more oscillatory as well.

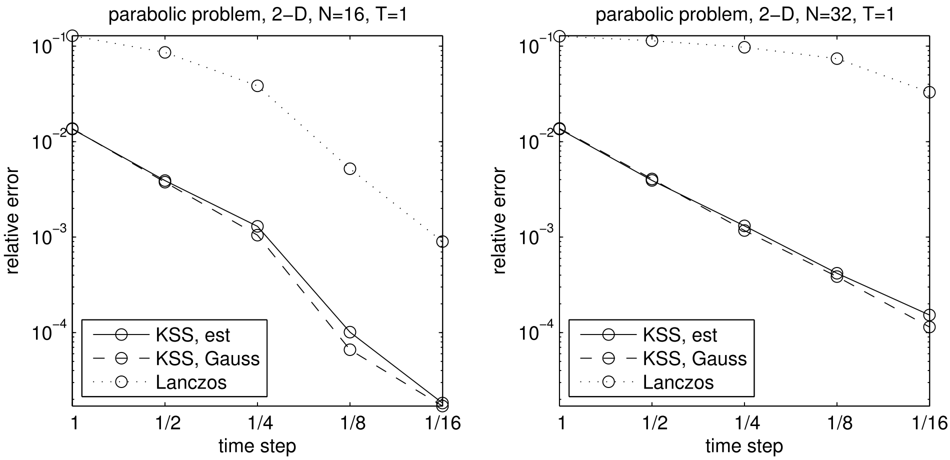

Fig. 12. Estimates of relative error at $t = 1$ in the solution of (28), (31), (32), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (solid curve), a 4-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid (per dimension) and various time steps. All methods are 3rd-order accurate in time.

Table 9
Estimates of relative error at $t = 1$ in the solution of (28), (31), (32), with periodic boundary conditions, computed by a 4-node KSS method with rapidly estimated nodes (KSS-est), a 4-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid (per dimension) and various time steps. All methods are 3rd-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
16	1	1.357e-002	1.370e-002	1.284e-001
	1/2	3.911e-003	3.766e-003	8.602e-002
	1/4	1.299e-003	1.051e-003	3.845e-002
	1/8	1.011e-004	6.626e-005	5.205e-003
	1/16	1.830e-005	1.703e-005	9.002e-004
32	1	1.357e-002	1.375e-002	1.273e-001
	1/2	3.923e-003	4.065e-003	1.142e-001
	1/4	1.310e-003	1.172e-003	9.730e-002
	1/8	4.192e-004	3.860e-004	7.405e-002
	1/16	1.529e-004	1.146e-004	3.295e-002

For third-order methods, the results are shown in Fig. 12 and Table 9. At the time steps used, both KSS methods perform identically in terms of accuracy independently of the grid size, but only exhibit slightly greater than second-order accuracy in time; third-order accuracy can be observed at much smaller time steps. As before, the approach of (2) is not competitive with KSS at this Krylov subspace dimension or choice of time step, and once again, there is significant degradation in accuracy as N increases, as this increase necessitates a corresponding increase in the Krylov subspace dimension.

We now increase the Krylov subspace dimension to 6 for all three methods. The results are shown in Fig. 13 and Table 10. As in the 1-D case, we observe a disparity in the performance of the original block KSS method and block KSS with rapid node estimation, which is not as accurate in this case. Furthermore, we also observe a disparity in the

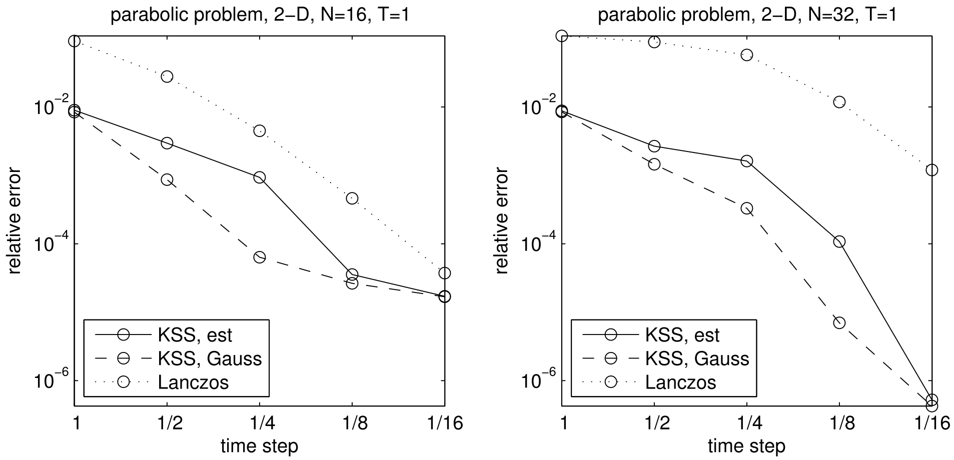

Fig. 13. Estimates of relative error at $t = 1$ in the solution of (28), (31), (32), with periodic boundary conditions, computed by a 6-node KSS method with rapidly estimated nodes (solid curve), a 6-node block KSS method with Gauss nodes (dashed curve), and Lanczos iteration as described in (2) (dotted curve) on an N-point grid (per dimension) and various time steps. All methods are 5th-order accurate in time.

Table 10
Estimates of relative error at $t = 1$ in the solution of (28), (31), (32), with periodic boundary conditions, computed by a 6-node KSS method with rapidly estimated nodes (KSS-est), a 6-node block KSS method with Gauss nodes (KSS-Gauss), and Lanczos iteration as described in (2) (Lanczos) on an N-point grid (per dimension) and various time steps. All methods are 5th-order accurate in time.

N	Δt	KSS-est	KSS-Gauss	Lanczos
16	1	8.998e-003	8.423e-003	9.219e-002
	1/2	2.971e-003	8.671e-004	2.787e-002
	1/4	9.370e-004	6.374e-005	4.485e-003
	1/8	3.549e-005	2.653e-005	4.631e-004
32	1/16	1.714e-005	1.682e-005	3.737e-005
	1	8.803e-003	8.488e-003	1.099e-001
	1/2	2.678e-003	1.457e-003	8.929e-002
	1/4	1.626e-003	3.309e-004	5.766e-002
	1/8	1.078e-004	7.003e-006	1.185e-002
	1/16	5.230e-007	4.257e-007	1.199e-003

results for the two grid sizes, except that accuracy and order of convergence *improves* as N increases, while (2), once again, shows the opposite trend. Because the time step is too large, the KSS methods only exhibit roughly second-order accuracy again, on average.

4.4. Performance

Now, we use (2) in a different way, that is consistent with its use in time-stepping methods such as those described in, among other sources, [11,17]. That is, for each product of the form $f(A)\mathbf{b}$ that needs to be computed, Lanczos iteration continues until convergence is achieved to within a specified tolerance, rather than being restricted to a Krylov subspace dimension that is determined by the desired temporal order of accuracy, as in KSS methods. We solve the 1-D parabolic problem (20), (21), (22), with smooth coefficients and initial data, with grids of dimension $N = 128, 256, 512$.

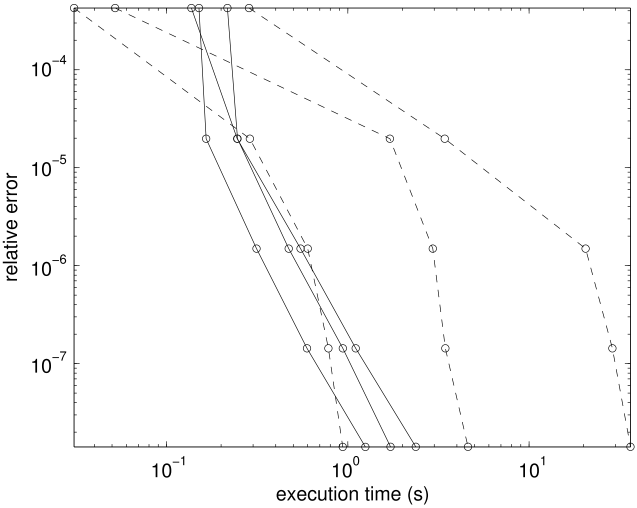

Fig. 14. Estimates of relative error at $t = 1$ in the solution of (20), (21), (22), with periodic boundary conditions, computed by a 4-node block KSS method with rapid node estimation (solid curves), and Lanczos iteration as described in (2) (dashed curves) with various time steps. For both methods, the curves, as displayed from left to right, correspond to solutions computed on N-point grids for $N = 128, 256, 512$.

The results are shown in Fig. 14. It can be seen that as N increases, the amount of time needed to achieve a given level of accuracy by (2), allowed to run until convergence is achieved to a relative error tolerance of 10^{-7}, is far greater than that required by a 4-node block KSS method with rapid node estimation. For the KSS method, the Krylov subspace dimension is always 4, whereas for (2), the maximum number of iterations needed in a time step for $N = 128, 256, 512$ was 21, 35 and 60, respectively. On the other hand, the time required by the KSS method scales approximately linearly with N.

5. Conclusion

We have demonstrated that when block Lanczos is applied to a matrix A obtained by spectral discretization of a differential operator L, for the purpose of approximating a Fourier coefficient $\hat{v}(\omega)$ of a function $v = f(L)u$, the eigenvalue problem for the block tridiagonal matrix produced by block Lanczos decouples in the limit as the frequency increases. This can be exploited to accurately estimate block Gaussian quadrature nodes for all frequencies with a fraction of the computational effort required to compute them directly. The result is a component-wise approach to time-stepping method in which the computational expense scales linearly with the resolution, as opposed to established Krylov subspace methods such as those in [10–12] that use the same polynomial approximation of $f(A)$ for every component. While additional investigation into node selection is required, particularly at low frequencies, the framework used by KSS methods opens the door to the development of similar methods with similar scalability and accuracy but even greater applicability. In particular, the techniques presented in this paper can be applied to the solution of nonlinear PDE [2] by combination with EPI methods [17].

Acknowledgements

The authors would like to thank the anonymous referee and editor for their helpful suggestions for improving the paper.

References

[1] K. Atkinson, An Introduction to Numerical Analysis, 2nd ed., Wiley, 1989.
[2] A. Cibotarica, J.V. Lambers, E.M. Palchak, Solution of time-dependent nonlinear PDE through component-wise approximation of matrix functions, SIAM J. Sci. Comput. (2014), submitted for publication.
[3] P. Davis, P. Rabinowitz, Methods of Numerical Integration, 2nd ed., Academic Press, 1984.
[4] W. Gautschi, Construction of Gauss–Christoffel quadrature formulas, Math. Comp. 22 (1986) 251–270.
[5] W. Gautschi, Orthogonal polynomials – constructive theory and applications, J. Comput. Appl. Math. 12/13 (1985) 61–76.
[6] G.H. Golub, G. Meurant, Matrices, in: D.F. Griffiths, G.A. Watson (Eds.), Moments and Quadrature. Proceedings of the 15th Dundee Conference, June–July 1993, Longman Scientific & Technical, 1994.
[7] G.H. Golub, R. Underwood, The block Lanczos method for computing eigenvalues, in: J. Rice (Ed.), Mathematical Software III, 1977, pp. 361–377.
[8] G.H. Golub, J. Welsch, Calculation of Gauss quadrature rules, Math. Comp. 23 (1969) 221–230.
[9] P. Guidotti, J.V. Lambers, K. Sølna, Analysis of 1-D wave propagation in inhomogeneous media, Numer. Funct. Anal. Optim. 27 (2006) 25–55.
[10] M. Hochbruck, C. Lubich, A Gautschi-type method for oscillatory second-order differential equations, Numer. Math. 83 (1999) 403–426.
[11] M. Hochbruck, C. Lubich, On Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal. 34 (1996) 1911–1925.
[12] M. Hochbruck, C. Lubich, H. Selhofer, Exponential integrators for large systems of differential equations, SIAM J. Sci. Comput. 19 (1998) 1552–1574.
[13] J.V. Lambers, Enhancement of Krylov subspace spectral methods by block Lanczos iteration, Electron. Trans. Numer. Anal. 31 (2008) 86–109.
[14] J.V. Lambers, Explicit high-order time-stepping based on component-wise application of asymptotic block Lanczos iteration, Numer. Linear Algebra Appl. 19 (6) (2012) 970–991.
[15] J.V. Lambers, An explicit, stable, high-order spectral method for the wave equation based on block Gaussian quadrature, IAENG Int. J. Appl. Math. 38 (2008) 333–348.
[16] J.V. Lambers, Practical implementation of Krylov subspace spectral methods, J. Sci. Comput. 32 (2007) 449–476.
[17] J. Loffeld, M. Tokman, Comparative performance of exponential, implicit and explicit integrators for stiff systems of ODEs, J. Comput. Appl. Math. 241 (2013) 45–67.
[18] I. Moret, P. Novati, RD-rational approximation of the matrix exponential operator, BIT 44 (2004) 595–615.
[19] E.M. Palchak, Rapid Approximation of Bilinear Forms Involving Matrix Functions Through Asymptotic Analysis of Gaussian Node Placement, MS thesis, University of Southern Mississippi, 2012.