

Krylov subspace spectral methods for the time-dependent Schrödinger equation with non-smooth potentials

James V. Lambers

Received: 30 June 2008 / Accepted: 19 February 2009 /
Published online: 6 March 2009
© Springer Science + Business Media, LLC 2009

Abstract This paper presents modifications of Krylov Subspace Spectral (KSS) Methods, which build on the work of Gene Golub and others pertaining to moments and Gaussian quadrature to produce high-order accurate approximate solutions to the time-dependent Schrödinger equation in the case where either the potential energy or the initial data is not a smooth function. These modifications consist of using various symmetric perturbations to compute off-diagonal elements of functions of matrices. It is demonstrated through analytical and numerical results that KSS methods, with these modifications, achieve the same high-order accuracy and possess the same stability properties as they do when applied to parabolic problems, even though the solutions to the Schrödinger equation do not possess the same smoothness.

Keywords Spectral methods · Gaussian quadrature · Schrödinger equation · Variable-coefficient · Lanczos method

Mathematics Subject Classifications (2000) 65M12 · 65M70 · 65D32 · 65F25

Dedicated to the memory of Gene H. Golub, 1932–2007.

J. V. Lambers (✉)
Department of Energy Resources Engineering, Stanford University,
Stanford, CA 94305-2220, USA
e-mail: lambers@stanford.edu

1 Introduction

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

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

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

with periodic boundary conditions

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

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

$$Lu = -pu_{xx} + V(x)u, \quad (1.4)$$

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

This equation is a simplification of the time-dependent Schrödinger equation,

$$i\hbar \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m} \nabla^2 \psi + V\psi, \quad (1.5)$$

that predicts the future behavior of a dynamic system. The constant \hbar is Planck's constant, m is mass, V is potential energy, and the solution ψ is a wavefunction that describes the quantum state of an electron [29]. By restricting ourselves to one space dimension, and using the transformation $\tau = t/\hbar$, we obtain (1.1) with $p = \hbar^2/2m$.

In [24, 25] a class of methods, called Krylov subspace spectral (KSS) methods, was introduced for the purpose of solving parabolic variable-coefficient problems such as those of the form $u_t + Lu = 0$. These methods have also been applied to the second-order wave equation (see [18, 22, 23]). These methods are based on the application of techniques developed by Golub and Meurant in [16], originally for the purpose of computing elements of the inverse of a matrix, to elements of the matrix exponential of an operator. It has been shown in these references that KSS methods, by employing different approximations of the solution operator for each Fourier component of the solution, achieve higher-order accuracy in time than other Krylov subspace methods (see, for example, [21]) for stiff systems of ODE, and, as shown in [22], they are also quite stable, considering that they are explicit methods.

In this paper, we explore the application of KSS methods to the time-dependent Schrödinger equation. Unlike the heat equation, whose solutions are infinitely differentiable, solutions of (1.1)–(1.3) retain the smoothness of the initial data. This property makes these problems more difficult for spectral methods, as they must accurately resolve a larger number of Fourier components. This difficulty is substantially increased if the potential $V(x)$, or the initial data $f(x)$, exhibit rough behavior such as oscillations or discontinuities. We will find that KSS methods, as described in the abovementioned references, are not very effective for such problems, but can easily be modified in order

to achieve the same high-order accuracy and stability as for problems with smooth coefficients and data.

Section 2 reviews the main properties of KSS methods, including algorithmic details and results concerning local accuracy. In Section 3, we discuss the application of KSS methods to the Schrödinger equation and present some results about its convergence behavior. We then show how they can be modified to most effectively handle problems with a potential or initial data that is not smooth, and indicate why such a modification is effective. In Section 4, we present a data structure and accompanying operations that can be used to efficiently implement the modified KSS methods. Numerical results are presented in Section 5. In Section 6, various extensions and future directions are discussed.

2 Krylov subspace spectral methods

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

$$\langle f(x), g(x) \rangle = \int_0^{2\pi} \overline{f(x)}g(x) dx. \tag{2.1}$$

Krylov subspace spectral methods, introduced in [24, 25], use Gaussian quadrature on the spectral domain to compute the Fourier components of the solution. These methods are time-stepping algorithms that compute the solution at time t_1, t_2, \dots , where $t_n = n\Delta t$ for some choice of Δt . Given the computed solution $\tilde{u}(x, t_n)$ at time t_n , the solution at time t_{n+1} is computed by approximating the Fourier components that would be obtained by applying the exact solution operator to $\tilde{u}(x, t_n)$,

$$\hat{u}(\omega, t_{n+1}) = \left\langle \frac{1}{\sqrt{2\pi}} e^{i\omega x}, S(\Delta t)\tilde{u}(x, t_n) \right\rangle. \tag{2.2}$$

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

We discretize functions defined on $[0, 2\pi]$ on an N -point uniform grid with spacing $\Delta x = 2\pi/N$. With this discretization, the operator L and the solution operator $S(\Delta t)$ can be approximated by $N \times N$ matrices that represent linear operators on the space of grid functions, and the quantity (2.2) can be approximated by a bilinear form

$$\hat{u}(\omega, t_{n+1}) \approx \hat{\mathbf{e}}_\omega^H S_N(\Delta t)\mathbf{u}(t_n), \tag{2.3}$$

where

$$[\hat{\mathbf{e}}_\omega]_j = \frac{1}{\sqrt{2\pi}} e^{i\omega j\Delta x}, \quad [\mathbf{u}(t_n)]_j = u(j\Delta x, t_n), \tag{2.4}$$

and

$$S_N(t) = \exp[-L_N t], \quad [L_N]_{jk} = -p[D_N^2]_{jk} + q(j\Delta x) \tag{2.5}$$

where D_N is a discretization of the differentiation operator that is defined on the space of grid functions. Our goal is to approximate (2.3) by computing an approximation to

$$[\hat{\mathbf{u}}^{n+1}]_\omega = \hat{\mathbf{e}}_\omega^H \mathbf{u}(t_{n+1}) = \hat{\mathbf{e}}_\omega^H S_N(\Delta t) \mathbf{u}(t_n). \tag{2.6}$$

In [16] Golub and Meurant describe a method for computing quantities of the form

$$\mathbf{u}^T f(A) \mathbf{v}, \tag{2.7}$$

where \mathbf{u} and \mathbf{v} are N -vectors, A is an $N \times N$ symmetric positive definite matrix, and f is a smooth function. Our goal is to apply this method with $A = L_N$ where L_N was defined in (2.5), $f(\lambda) = \exp(-\lambda t)$ for some t , and the vectors \mathbf{u} and \mathbf{v} are derived from $\hat{\mathbf{e}}_\omega$ and $\mathbf{u}(t_n)$.

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

$$b = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N = a > 0, \tag{2.8}$$

and corresponding orthogonal eigenvectors \mathbf{q}_j , $j = 1, \dots, N$. Therefore, the quantity (2.7) can be rewritten as

$$\mathbf{u}^T f(A) \mathbf{v} = \sum_{\ell=1}^N f(\lambda_\ell) \mathbf{u}^T \mathbf{q}_\ell \mathbf{q}_\ell^T \mathbf{v}. \tag{2.9}$$

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

$$\alpha(\lambda) = \begin{cases} 0, & \text{if } \lambda < a \\ \sum_{j=i}^N \alpha_j \beta_j, & \text{if } \lambda_i \leq \lambda < \lambda_{i-1}, \\ \sum_{j=1}^N \alpha_j \beta_j, & \text{if } b \leq \lambda \end{cases} \quad \alpha_j = \mathbf{u}^T \mathbf{q}_j, \quad \beta_j = \mathbf{q}_j^T \mathbf{v}, \tag{2.10}$$

If this measure is positive and increasing, then the quantity (2.7) can be viewed as a Riemann-Stieltjes integral

$$\mathbf{u}^T f(A) \mathbf{v} = I[f] = \int_a^b f(\lambda) d\alpha(\lambda). \tag{2.11}$$

As discussed in [11, 14–16], the integral $I[f]$ can be bounded using either Gauss, Gauss-Radau, or Gauss-Lobatto quadrature rules, all of which yield an approximation of the form

$$I[f] = \sum_{j=1}^K w_j f(t_j) + R[f], \tag{2.12}$$

where the nodes $t_j, j = 1, \dots, K$, as well as the weights $w_j, j = 1, \dots, K$, can be obtained using the symmetric Lanczos algorithm if $\mathbf{u} = \mathbf{v}$, and the unsymmetric Lanczos algorithm if $\mathbf{u} \neq \mathbf{v}$ (see [17]).

In the case $\mathbf{u} \neq \mathbf{v}$, there is the possibility that the weights may not be positive, which destabilizes the quadrature rule (see [6] for details). Therefore, it is best to handle this case by rewriting (2.7) using decompositions such as

$$\mathbf{u}^T f(A)\mathbf{v} = \frac{1}{\delta} [\mathbf{u}^T f(A)(\mathbf{u} + \delta\mathbf{v}) - \mathbf{u}^T f(A)\mathbf{u}], \tag{2.13}$$

where δ is a small constant. Guidelines for choosing an appropriate value for δ can be found in [25, Section 2.2].

Employing these quadrature rules yields the following basic process (for details see [24, 25]) for computing the Fourier coefficients of $\mathbf{u}(t_{n+1})$ from $\mathbf{u}(t_n)$. It is assumed that when the Lanczos algorithm (symmetric or unsymmetric) is employed, K iterations are performed to obtain the K quadrature nodes and weights.

```

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

It should be noted that the constant δ_ω plays the role of δ in the decomposition (2.13), and the subscript ω is used to indicate that a different value may be used for each wave number $\omega = -N/2 + 1, \dots, N/2$. Also, in the presentation of this algorithm in [25], a polar decomposition is used instead of (2.13), and is applied to sines and cosines instead of complex exponential functions.

If we let $\delta_\omega \rightarrow 0$ for each ω , it follows from [16] that the quadrature error for each Fourier component is

$$\frac{\Delta t^{2K}}{(2K)!} \frac{d}{d\delta_\omega} \left[\hat{\mathbf{e}}_\omega^H (\hat{\mathbf{e}}_\omega + \delta_\omega\mathbf{u}^n) \int_{\lambda_N}^{\lambda_1} e^{-\lambda\Delta t} \prod_{j=1}^K (\lambda - t_{j,\omega}(\delta_\omega))^2 d\alpha(\lambda) \right] \Bigg|_{\delta_\omega=0},$$

where, for $j = 1, \dots, K$, $t_{j,\omega}(\delta_\omega)$ is the j th Gaussian quadrature node obtained by applying the Lanczos algorithm to L_N with starting vectors $\hat{\mathbf{e}}_\omega$ and $\hat{\mathbf{e}}_\omega + \delta_\omega\mathbf{u}^n$. The leading term, in Δt , in the above derivative with respect to δ_ω is

$$\frac{\Delta t^{2K}}{(2K)!} \frac{d}{d\delta_\omega} \left[\hat{\mathbf{e}}_\omega^H \prod_{j=1}^K (L_N - t_{j,\omega}(\delta_\omega)I)^2 (\hat{\mathbf{e}}_\omega + \delta_\omega\mathbf{u}^n) \right] \Bigg|_{\delta_\omega=0}.$$

Carrying out the differentiation, we obtain

$$\frac{\Delta t^{2K}}{(2K)!} \hat{\mathbf{e}}_\omega^H \prod_{j=1}^K (L_N - t_{j,\omega}(0)I)^2 \mathbf{u}^n. \tag{2.14}$$

This is because the other terms that arise from differentiation include a factor of the form

$$\hat{\mathbf{e}}_\omega^H \left[\prod_{j=1}^K (L_N - t_{j,\omega}(0)I) \right] P(L_N) \hat{\mathbf{e}}_\omega,$$

where $P(L_N)$ is a polynomial of L_N whose degree is less than K . However, the polynomial $q_K(\lambda) = \prod_{j=1}^K (\lambda - t_{j,\omega}(0))$ is orthogonal to all polynomials of degree less than K , with respect to the measure $\alpha(\lambda)$, so these terms vanish.

The error term (2.14) suggests that KSS methods achieve high-order temporal accuracy, as the constant multiplying Δt^{2K} is a Fourier coefficient of the application of a pseudodifferential operator of order $4K$ to \mathbf{u}^n . Given sufficient regularity of the solution, it can be shown that KSS methods are consistent, and actually do achieve this high-order accuracy, as already demonstrated in numerical results in [18, 22–26]. We will discuss this further in Section 3.

This degree of accuracy can be compared to the accuracy achieved by an algorithm described by Hochbruck and Lubich in [21] for computing $e^{A\Delta t}\mathbf{v}$ for a given matrix A and vector \mathbf{v} using the unsymmetric Lanczos algorithm. As discussed in [21], this algorithm can be used to compute the solution of some ODEs without time-stepping, but this becomes less practical for ODEs arising from a semi-discretization of problems such as (1.1)–(1.3), due to their stiffness. In this situation, it is necessary to either use a high-dimensional Krylov subspace, in which case reorthogonalization is required, or one can resort to time-stepping, in which case the local temporal error is only $O(\Delta t^K)$, assuming a K -dimensional Krylov subspace. Regardless of which remedy is used, the computational effort needed to compute the solution at a fixed time T increases substantially.

The difference between Krylov subspace spectral methods and the approach described in [21] is that in the former, a different K -dimensional Krylov subspace is used for each Fourier component, instead of the same subspace for all components as in the latter. As can be seen from numerical results comparing the two approaches in [25], using the same subspace for all components causes a loss of accuracy as the number of grid points increases, whereas Krylov subspace spectral methods do not suffer from this phenomenon.

On the surface, it may appear to be a simple matter to generalize KSS methods to a problem such as the time-dependent Schrödinger equation, since the solution process differs from that used for parabolic equations only by a change of solution operator, which is reflected by a change of integrand in (2.11). While this trivial adjustment is sufficient for problems in which the variable coefficient, the potential $V(x)$, and the initial data $f(x)$ are smooth, the resulting method is much less effective when this is not the case. We will

where the ' denotes differentiation with respect to δ_ω , and evaluation of the derivative at $\delta_\omega = 0$. Algorithms for computing the derivatives of the nodes and weights are described in [22].

3.1 Relation to the time-independent Schrödinger equation

It is well known that solutions of the time-independent Schrödinger equation,

$$-\frac{\hbar^2}{2m}\nabla^2\psi_E + V\psi_E = E\psi_E, \tag{3.3}$$

where ψ_E is an energy eigenstate with energy E , are the standing wave solutions of the time-dependent equation (1.5). The integrals (2.11) that are used by KSS methods to compute the Fourier components of the solution to (1.5) are defined in terms of a basis of eigenstates and corresponding energies of the discretization on an N -point grid of (3.3) with boundary conditions (1.3).

Although the time-dependent equation can easily be solved in terms of these eigenstates, computing a basis of N eigenstates for the discretization of (3.3) on an N -point grid, to high accuracy, is computationally expensive, so we instead rely on Gaussian quadrature rules to approximate integrals of the form (2.11), which do not require computing any eigenstates or energies. However, approximate eigenstates and energies can still be obtained from the application of the Lanczos algorithm by KSS methods.

Specifically, for each wave number ω , we have

$$L_N X_{\omega,K} = X_{\omega,K} T_{\omega,K}(0) + \beta_K(0) \mathbf{x}_{K+1,\omega} \mathbf{e}_K^T,$$

where $T_{\omega,K}(\delta_\omega)$ is defined as above, after K iterations of the Lanczos algorithm, and $X_{\omega,K}$ is the matrix of Lanczos vectors. Then, from the decomposition

$$T_{\omega,K} = U_{\omega,K} \Lambda_{\omega,K} U_{\omega,K}^H,$$

where the eigenvalues and corresponding eigenvectors are ordered so that

$$|[U_{\omega,K}]_{Kj}| = \min_{1 \leq i \leq K} |[U_{\omega,K}]_{Ki}|,$$

we obtain the approximate discrete eigenstate $X_{\omega,K} U_{\omega,K} \mathbf{e}_j$, with corresponding approximate energy λ_j . The accuracy of these eigenstates depends on the number of nodes, K , and the smoothness of the potential $V(x)$.

3.2 The one-node case

When $K = 1$, we simply have $T_{\omega,1}(\delta_\omega) = \alpha_1(\delta_\omega)$, where

$$\alpha_1(\delta_\omega) = \frac{\hat{\mathbf{e}}_\omega^H L_N (\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n)}{\hat{\mathbf{e}}_\omega^H (\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n)}, \tag{3.4}$$

which yields

$$\alpha_1'(0) = \frac{\hat{\mathbf{e}}_\omega^H (L_N - \alpha_1 I) \mathbf{u}^n}{\hat{\mathbf{e}}_\omega^H \hat{\mathbf{e}}_\omega}. \tag{3.5}$$

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

$$[\hat{\mathbf{u}}^{n+1}]_\omega = e^{-i\alpha_1 \Delta t} \hat{\mathbf{e}}_\omega^H [1 - i\Delta t(L_N - \alpha_1 I)] \mathbf{u}^n. \tag{3.6}$$

For convenience, for a function f defined on $[0, 2\pi]$, we define

$$\bar{f} = \text{Avg } f = \frac{1}{2\pi} \int_0^{2\pi} f(x) dx,$$

and we denote the N -point Fourier interpolant of f by the vector \mathbf{f}_N . Because $\alpha_1 = p\omega^2 + \bar{V}$ and $L_N \hat{\mathbf{e}}_\omega = p\omega^2 \hat{\mathbf{e}}_\omega + \text{diag}(\mathbf{V}_N) \hat{\mathbf{e}}_\omega$, it follows that the approximate solution $\tilde{u}_N(x, t)$, for a given Δt and Δx , is defined by

$$\tilde{u}_N(x, t_{n+1}) = e^{-iC_N \Delta t} [I - i\Delta t R_N] \tilde{u}_N(x, t_n), \quad \tilde{u}_N(x, 0) = P_N f(x), \tag{3.7}$$

where $L = C + R$ is a splitting of L such that $C = -pD^2 + \bar{V}$ is the constant-coefficient operator obtained by averaging the coefficients of L , and the variation of the coefficients is captured by R . The discretized operator L_N is similarly split as $L_N = C_N + R_N$, with R_N defined by

$$R_N f(x) = P_N \tilde{V}_N P_N f(x),$$

where $\tilde{V} = V - \bar{V}$ and the operator P_N is the orthogonal projection onto $BL_N([0, 2\pi]) = \text{span}\{e^{-i\omega x}\}_{\omega=-N/2+1}^{N/2}$, a space of bandlimited functions with at most N nonzero Fourier components.

This simple form of the approximate solution operator yields the following results. We will denote by $\tilde{S}_N(\Delta t) f$ the result of applying a single time step of a KSS method to the function $f(x)$, using a discretization of space and time with uniform spacings $\Delta x = 2\pi/N$ and Δt , respectively. Also, for convenience, component-wise multiplication of vectors \mathbf{u} and \mathbf{v} is written as simply \mathbf{uv} .

Theorem 3.1 *Let $V(x)$ in (1.4) belong to $BL_M([0, 2\pi])$ for a fixed integer M . Let $f \in H_p^q[0, 2\pi]$, for $q \geq 5$. Then, for the problem (1.1)–(1.3), on the domain $[0, 2\pi] \times [0, T]$, the one-node KSS method (3.7) is consistent. That is,*

$$\left\| \tilde{S}_N(\Delta t) f_N - \exp[-iL\Delta t] f \right\|_{L^2} \leq C_1 \Delta t^2 + C_2 \Delta x^q,$$

where $f_N(x) = P_N f(x)$, and the constants C_1 and C_2 are independent of Δx and Δt .

Proof We split the local truncation error into two components:

$$E_1(\Delta t, \Delta x) = \exp[-i\tilde{L}_N \Delta t] f_N(x) - \tilde{S}_N(\Delta t) f_N(x)$$

$$E_2(\Delta t, \Delta x) = \exp[-iL\Delta t] f(x) - \exp[-i\tilde{L}_N \Delta t] f_N(x),$$

where we define $\tilde{L}_N = P_N L P_N$.

First, we bound $E_2(\Delta t, \Delta x)$. Because of the regularity of f , we have

$$\|f - f_N\|_{L^2} \leq C_0 \Delta x^q$$

for some constant C_0 (see [20, Theorem 2.10]). Furthermore, because

$$\sum_{\omega=-\infty}^{\infty} \omega^{2q} \left| \hat{f}(\omega) \right|^2 < \infty, \tag{3.8}$$

it follows that

$$\left| \hat{f}(\omega) \right| \leq \frac{C_f}{|\omega|^q}, \quad \omega \neq 0,$$

for some constant C_f (more precisely, $|\omega^q \hat{f}(\omega)| \rightarrow 0$ as $|\omega| \rightarrow \infty$, or the series (3.8) would diverge).

The exact solution operator $\exp[-iL\Delta t]$ is defined by

$$\exp[-iL\Delta t] f(x) = \sum_{k=1}^{\infty} e^{-i\mu_k \Delta t} v_k(x) \langle v_k, f \rangle$$

where $\{\mu_k\}_{k=1}^{\infty}$ are the real, positive eigenvalues of L , and $\{v_k(x)\}_{k=1}^{\infty}$ are the corresponding orthonormal eigenfunctions, each of which belongs to $C_p^\infty[0, 2\pi]$. Using this spectral decomposition, it can be shown using an approach similar to that used in [12, Chapter 7] for other PDE that if $f \in H_p^q[0, 2\pi]$, then $u(x, t) \in L^\infty(0, T, H_p^q[0, 2\pi])$. That is, the regularity of $f(x)$ is preserved in $u(x, \Delta t)$ for each fixed $\Delta t > 0$, and therefore there exists a constant C_2 such that

$$\| \exp[-iL\Delta t] f - \exp[-i\tilde{L}_N \Delta t] f_N \|_{L^2} \leq C_2 \Delta x^q,$$

for $0 \leq \Delta t \leq T$, due to the truncation of Fourier series.

Now, we examine $E_1(\Delta t, \Delta x)$. For each $\omega = -N/2 + 1, \dots, N/2$, we take a Taylor expansion of the corresponding Fourier coefficient of $E_1(\Delta t, \Delta x)$ centered at $\Delta t = 0$ and obtain

$$\begin{aligned} \langle \hat{e}_\omega, E_1(\Delta t, \Delta x) \rangle &= \langle \hat{e}_\omega, f_N - i\Delta t \tilde{L}_N f_N \rangle - \int_0^{\Delta t} (\Delta t - \tau) \langle \hat{e}_\omega, \tilde{L}_N^2 \exp[-i\tilde{L}_N \tau] f_N \rangle d\tau \\ &\quad - \langle \hat{e}_\omega, e^{-iC_N \Delta t} [I - i\Delta t R_N] f_N \rangle \\ &= \langle \hat{e}_\omega, f_N - i\Delta t \tilde{L}_N f_N \rangle - \int_0^{\Delta t} (\Delta t - \tau) \langle \hat{e}_\omega, \tilde{L}_N^2 \exp[-i\tilde{L}_N \tau] f_N \rangle d\tau \\ &\quad - \langle \hat{e}_\omega, f_N - i\Delta t (C_N + R_N) f_N \rangle + \int_0^{\Delta t} (\Delta t - \tau) \\ &\quad \times \langle \hat{e}_\omega, \exp[-iC_N \tau] [C_N^2 (I - i\tau R_N) + 2C_N R_N] f_N \rangle d\tau \\ &= - \int_0^{\Delta t} (\Delta t - \tau) \langle \hat{e}_\omega, \tilde{L}_N^2 \exp[-i\tilde{L}_N \tau] f_N \rangle d\tau \\ &\quad + \alpha_{1,\omega} \int_0^{\Delta t} (\Delta t - \tau) e^{-i\tau \alpha_{1,\omega}} \langle \hat{e}_\omega, [C_N (I - i\tau R_N) + 2R_N] f_N \rangle d\tau. \end{aligned}$$

In view of the regularity of $f(x)$ and $u(x, t)$, and the fact that \tilde{L}_N and C_N are discretizations of second-order differential operators with constant or

bandlimited coefficients and R_N is multiplication by a bandlimited function, there exist constants C_3 and C_4 , independent of N , such that for $\omega \neq 0$,

$$\begin{aligned} \left| \langle \hat{e}_\omega, \tilde{L}_N^2 \exp[-i\tilde{L}_N \tau] f_N \rangle \right| &\leq \frac{C_3}{|\omega|^{q-4}}, \quad \left| \langle \hat{e}_\omega, [C_N(I - i\tau R_N) + 2R_N] f_N \rangle \right| \\ &\leq \frac{C_4}{|\omega|^{q-2}}, \end{aligned}$$

for $\tau \in [0, T]$.

Because $\alpha_{1,\omega} = p\omega^2 + \bar{V}$, we have

$$|\langle \hat{e}_\omega, E_1(\Delta t, \Delta x) \rangle| \leq \frac{\Delta t^2}{2} \frac{C_3 + (p + \bar{V})C_4}{|\omega|^{q-4}}, \quad \omega \neq 0,$$

and since $q \geq 5$, it follows that

$$\begin{aligned} \|E_1(\Delta t, \Delta x)\|_{L^2} &\leq \frac{\Delta t^2}{2} \left(C_5^2 + [C_3 + (p + \bar{V})C_4]^2 \sum_{\omega \neq 0} |\omega|^{8-2q} \right)^{1/2} \\ &\leq \frac{\Delta t^2}{2} \left(C_5^2 + [C_3 + (p + \bar{V})C_4]^2 \frac{4q - 16}{2q - 9} \right)^{1/2} \\ &\leq C_1 \Delta t^2, \end{aligned}$$

where we have used $\int_1^\infty \omega^{8-2q} d\omega$ to help bound the summation over ω , and

$$\begin{aligned} C_5 &= \sup_{\Delta x \in (0, 2\pi], \Delta t \in [0, T]} |\langle \hat{e}_0, E_1(\Delta t, \Delta x) \rangle| \\ &\leq \sup_{\Delta x \in (0, 2\pi], \Delta t \in [0, T]} \left[\|\exp[-i\tilde{L}_N \Delta t] f_N\|_{L^2} + \|\tilde{S}_N(\Delta t) f_N\|_{L^2} \right] \\ &\leq \sup_{\Delta x \in (0, 2\pi], \Delta t \in [0, T]} \left[\|f_N\|_{L^2} + \|(I - i\Delta t R_N) f_N\|_{L^2} \right] \\ &\leq (2 + T \max_{0 \leq x \leq 2\pi} |V(x)|) \|f\|_{L^2}. \end{aligned}$$

We conclude that the constant C_1 is independent of N . □

Remark It is interesting to note that if we truncate the Taylor series after an additional term, we find that

$$|\langle \hat{e}_\omega, E_1(\Delta t, \Delta x) \rangle| = \frac{\Delta t^2}{2} \left| \langle \hat{e}_\omega, (\tilde{L}_N - \alpha_{1,\omega} I)^2 f_N \rangle \right| + O(\Delta t^3),$$

which is consistent with (2.14). Substituting $\alpha_{1,\omega} = p\omega^2 + \bar{V}$ yields

$$\begin{aligned} |\langle \hat{e}_\omega, E_1(\Delta t, \Delta x) \rangle| &= \frac{\Delta t^2}{2} \left| \langle \hat{e}_\omega, (\tilde{V}^2 + p\tilde{V}'' - 2i\omega p\tilde{V}') f_N \rangle \right| + O(\Delta t^3) \\ &= \frac{\Delta t^2}{2} \left| \langle \hat{e}_\omega, (\tilde{V}^2 - p\tilde{V}'') f_N - 2p\tilde{V}' f'_N \rangle \right| + O(\Delta t^3). \end{aligned}$$

We see that $q \geq 2$, rather than $q \geq 5$, is sufficient to ensure that the leading term, in Δt , of $E_1(\Delta t, \Delta x)$ can be bounded independently of N . This is due to the fact that the leading coefficient of L is constant. The proof of Theorem 3.1 does not account for any cancellation that results from this.

Now we consider the stability of the method. For convenience, we denote by $\tilde{S}_N(\Delta t)$ the matrix such that $\mathbf{u}^{n+1} = \tilde{S}_N(\Delta t)\mathbf{u}^n$, for given N and Δt .

Theorem 3.2 *Let $V(x)$ in (1.4) belong to $BL_M([0, 2\pi])$ for a fixed integer M . Then, for the problem (1.1)–(1.3), the one-node KSS method (3.7) is unconditionally stable. That is, given $T > 0$, there exists a constant C_T , independent of N and Δt , such that*

$$\|\tilde{S}_N(\Delta t)^n\|_2 \leq C_T, \tag{3.9}$$

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

Proof The matrix C_N has the diagonalization

$$C_N = F_N^{-1} \Lambda_N F_N, \tag{3.10}$$

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

$$\Lambda = \text{diag}(p\omega^2 + \bar{V}), \quad \omega = -N/2 + 1, \dots, N/2. \tag{3.11}$$

It follows that $\|e^{-iC_N\Delta t}\|_2 = 1$.

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

$$\|I - i\Delta t \text{diag}(\tilde{V}_N)\|_2 \leq 1 + \Delta t Q, \tag{3.12}$$

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

$$\|\tilde{S}_N(\Delta t)\|_2 \leq e^{Q\Delta t}, \tag{3.13}$$

from which the result follows with $C_T = e^{QT}$. □

It should be noted that unconditional stability can also be shown for the case where $V(x)$ is not bandlimited, but satisfies the condition

$$\sum_{\omega=-\infty}^{\infty} |\hat{V}(\omega)| < \infty. \tag{3.14}$$

To see this, we use the relation

$$\tilde{V}_N(\omega) = \sum_{\ell=-\infty}^{\infty} \hat{V}(\omega + \ell N), \quad \omega = -N/2 + 1, \dots, N/2, \tag{3.15}$$

where the $\tilde{V}_N(\omega)$ are the Fourier coefficients of the N -point trigonometric interpolant of $V(x)$. It follows from this relation that if (3.14) holds, then $V_N(x)$ is uniformly bounded for all integers N . However, (3.14) does *not* hold if V is not continuous on $[0, 2\pi]$.

The Schrödinger equation is unitary, which means that the solution $u(x, t)$ satisfies

$$\frac{d}{dt} \left[\int_0^{2\pi} |u(x, t)|^2 dx \right] = 0.$$

It is natural to ask whether numerical solutions computed by KSS methods also satisfy this property. For the 1-node case, we have the following result.

Theorem 3.3 *Let \mathbf{u}^n be the approximate solution of (1.1)–(1.3) by the one-node KSS method (3.7) on an N -point grid. Then*

$$\|\mathbf{u}^{n+1}\|^2 = \|\mathbf{u}^n\|^2 + O(\Delta t^2).$$

That is, the one-node KSS method is unitary in the limit as $\Delta t \rightarrow 0$.

Proof From (3.6), we have

$$\begin{aligned} \|\mathbf{u}^{n+1}\|^2 &= \sum_{\omega=-N/2+1}^{N/2} |\Delta x [\hat{\mathbf{u}}^{n+1}]_{\omega}|^2 \\ &= \sum_{\omega=-N/2+1}^{N/2} \Delta x^2 \left| e^{-i\alpha_1 \Delta t} \hat{\mathbf{e}}_{\omega}^H [1 - i\Delta t \tilde{\mathbf{V}}_N] \mathbf{u}^n \right|^2 \\ &= \Delta x^2 \sum_{\omega=-N/2+1}^{N/2} \left[|\hat{\mathbf{u}}^n_{\omega}|^2 + 2\Delta t \operatorname{Re} i \overline{\hat{\mathbf{e}}_{\omega}^H \mathbf{u}^n} \hat{\mathbf{e}}_{\omega}^H \tilde{\mathbf{V}}_N \mathbf{u}^n + \Delta t^2 \left| \hat{\mathbf{e}}_{\omega}^H \tilde{\mathbf{V}}_N \mathbf{u}^n \right|^2 \right] \\ &= \|\mathbf{u}^n\|^2 + 2\Delta t \Delta x^2 \operatorname{Re} i \sum_{\omega \neq \eta} \overline{(\hat{\mathbf{e}}_{\omega}^H \mathbf{u}^n)} \left(\hat{\mathbf{e}}_{\omega-\eta}^H \tilde{\mathbf{V}}_N \right) (\hat{\mathbf{e}}_{\eta}^H \mathbf{u}^n) + \Delta t^2 \|\tilde{\mathbf{V}}_N \mathbf{u}^n\|^2. \end{aligned}$$

By symmetry, the summation in the second term is a real number, causing the second term to vanish, from which the result follows. \square

Intuitively, one would expect this result as a natural consequence of the consistency of the scheme, but it should also be noted that the departure from unitarity is directly proportional to the heterogeneity in the potential $V(x)$.

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

$$\|u(\cdot, t)\|_2 = \left(\sum_{j=0}^{N-1} |u(j\Delta x, t)|^2 \right)^{1/2}. \tag{3.16}$$

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

$$\|u(\cdot, t) - \tilde{u}_N(\cdot, t)\|_2 \leq C_t \Delta t^m + C_x \Delta x^n, \quad 0 \leq t \leq T. \tag{3.17}$$

Theorem 3.4 *Let $V(x)$ in (1.4) belong to $BL_M([0, 2\pi])$ for some integer M . Let $f \in H_p^q[0, 2\pi]$, where $q \geq 5$. Then, for the problem (1.1)–(1.3), on the domain $[0, 2\pi] \times [0, T]$, the one-node KSS method (3.7) is convergent of order $(1, q)$.*

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

$$E_n = N^{-1/2} \|S(\Delta t)^n f - \tilde{S}_N(\Delta t)^n f\|_2. \tag{3.18}$$

Then, by Theorems 3.1 and 3.2, there exist constants C_1, C_2 and C such that

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

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

$$E_n \leq \frac{e^{CT} - 1}{e^{C\Delta t} - 1} (C_1 \Delta t^2 + C_2 \Delta t \Delta x^q) \leq \tilde{C}_1 \Delta t + \tilde{C}_2 \Delta x^q. \tag{3.19}$$

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

3.3 The multi-node case

Theorem 3.1 can be generalized to the case of more than one quadrature node per Fourier coefficient, in which case the temporal error is $O(\Delta t^{2K})$, where K is the number of quadrature nodes in each rule. This entails expanding each Fourier coefficient of the approximate solution, represented by (3.2), in a Taylor series around $\Delta t = 0$, and then establishing bounds on the nodes $\lambda_{\omega,j}$ and weights $w_{\omega,j}$, and their derivatives with respect to δ_ω , at $\delta_\omega = 0$, for $j = 1, \dots, K$ and $\omega = -N/2 + 1, \dots, N/2$.

These bounds can be proven by applying a simple inductive argument to the computation of the recursion coefficients using the Lanczos algorithm, and algorithms for computing the derivatives (see [22]). The results are

$$|\lambda_{\omega,j}| \leq C|\omega|^2, \quad |w_{\omega,j}| \leq 1,$$

for a constant C , and

$$\lambda'_{\omega,j} = \langle \hat{e}_\omega, \Lambda_j \tilde{u}(\cdot, t_n) \rangle, \quad w'_{\omega,j} = \langle \hat{e}_\omega, W_j \tilde{u}(\cdot, t_n) \rangle,$$

where Λ_j and w_j are second-order pseudodifferential operators. It follows that $\lambda'_{\omega,j}$ and $w'_{\omega,j}$ are $O(|\omega|^{2-q})$ if $\tilde{u}(x, t_n) \in H^q_p[0, 2\pi]$. Therefore, a sufficient condition for consistency of the scheme is that the initial data $f(x)$ belongs to $H^q_p[0, 2\pi]$, where $q \geq 4K + 3$. It should be noted that in [22], for the case $K = 2$, explicit formulas for the nodes, weights, and their derivatives are given, in which it can be seen that the above bounds apply.

Following steps analogous to those used in [25, Lemma 2.1, Theorem 2.4], we obtain the leading-order temporal error term for each Fourier component,

$$E_{1,\omega}(\Delta t, \Delta x) = \frac{(-i\Delta t)^{2K}}{(2K)!} \frac{d}{d\delta_\omega} \left[\prod_{j=0}^K \beta_j(\omega) \right]^2 \Big|_{\delta_\omega=0}.$$

From the definition of $\beta_K(\delta_\omega)$ and the biorthogonality of the two sets of Lanczos vectors and associated polynomials, we obtain an expression similar to (2.14),

$$E_{1,\omega}(\Delta t, \Delta x) = \frac{(-i\Delta t^{2K})}{(2K)!} \left\langle \hat{e}_\omega, \prod_{j=1}^K (\tilde{L}_N - \lambda_{\omega,j} I)^2 u_N(\cdot, t_n) \right\rangle.$$

We see that the error in each Fourier coefficient $\hat{u}(\omega, t_{n+1})$ arises from

- truncation in the Fourier series, for ω outside of the range $-N/2 + 1, \dots, N/2$
- for all other values of ω , Δt^{2K} times a discrete Fourier coefficient, with the same wave number ω , of the application of a *pseudodifferential operator* to $\tilde{u}_N(x, t_n)$.

For $K > 2$, the nodes $\lambda_{\omega,j}$ are no longer necessarily polynomials in ω , but because they are Ritz values of L_N , a second-order operator, they are at most second-order in ω .

To see this, note that at $\delta_\omega = 0$, we have

$$T_{\omega,K} = p\omega^2 I_K + E_{\omega,K}$$

where the elements of $E_{\omega,K}$ are bounded by $E|\omega|$ for some constant E that depends only on K and the coefficients of L . This follows directly from the formulas for the recursion coefficients in the Lanczos algorithm, and the choice of \hat{e}_ω as the initial vector.

By the Gerschgorin Circle Theorem (see [13]), for large ω the eigenvalues of $T_{\omega,K}$ increase like $p\omega^2$, which implies that for each $j = 1, \dots, K$, the collection of nodes $\lambda_{j,\omega}$ form the spectrum of a second-order pseudodifferential operator. In the 2-node case, explicit formulas for the nodes are given in [22, Section 5]; we see that they are zero-order corrections to $p\omega^2$.

It follows that the pseudodifferential operator from the error term, $\prod_{j=1}^K (L_N - \lambda_{\omega,j} I)^2$, is of order at most $4K$. Because the leading coefficient of L is constant, and, at least for $K \leq 2$, matches the leading coefficient of the pseudodifferential operators defined by the nodes, the highest-order terms of

this operator cancel in the 1-node and 2-node cases, resulting in an operator whose order is at most $2K$. As in the one-node case, this suggests that high-order accuracy can be achieved with less regularity in the initial data than previously assumed.

We conclude that KSS methods are capable of achieving the same high-order accuracy in time for the Schrödinger equation as for parabolic problems. However, as the number of nodes increases, the smoothness of the potential and solution needs to increase as well, in order to ensure this high order of accuracy, due to additional differentiation of both.

3.4 Non-smooth potentials

Numerical experiments, which will be reported in Section 5.2, demonstrate the effectiveness of KSS methods, in terms of estimates of global error, for problems in which the potential $V(x)$ in (1.4) and the initial data $f(x)$ are both smooth. When this is not the case, however, accuracy is significantly degraded, especially for larger time steps.

This can be resolved by using a symmetric perturbation, such as the *polar decomposition*

$$\mathbf{u}^T f(A)\mathbf{v} = \frac{1}{4\delta} [(\mathbf{u} + \delta\mathbf{v})^T f(A)(\mathbf{u} + \delta\mathbf{v}) - (\mathbf{u} - \delta\mathbf{v})^T f(A)(\mathbf{u} - \delta\mathbf{v})], \quad (3.20)$$

where \mathbf{u} and \mathbf{v} are real; a generalization to complex \mathbf{u} and \mathbf{v} is straightforward. To see why this is effective, and the original KSS method is not, we examine the effect of perturbation of the initial vectors, both symmetric and unsymmetric, on the recursion coefficients.

For convenience, we define $T(L_N, \mathbf{u}, \mathbf{v})$ to be the matrix of recursion coefficients produced by the unsymmetric Lanczos method applied to the discretized differential operator L_N , on a uniform N -point grid, with initial vectors \mathbf{u} and \mathbf{v} . We set $N = 64$ and examine the recursion coefficients for the wave number $\omega = 24$. Let \mathbf{f} be the discretized initial data, defined in Section 5.3, which we construct so as to be smooth (see Section 5.1). Furthermore, let $\delta = 10^{-4}$. We observe the following:

- When we use the smooth potential shown in Fig. 6 (left plot), we have

$$\frac{1}{\delta} [T(L_N, \hat{\mathbf{e}}_\omega, \hat{\mathbf{e}}_\omega + \delta\mathbf{f}) - T(L_N, \hat{\mathbf{e}}_\omega, \hat{\mathbf{e}}_\omega)] = \begin{bmatrix} 1.5807\text{e-}006 & 1.0388\text{e-}003 \\ 1.0388\text{e-}003 & 3.2976 \end{bmatrix}$$

- When the potential is not smooth, as in Fig. 6 (right plot), we have

$$\frac{1}{\delta} [T(L_N, \hat{\mathbf{e}}_\omega, \hat{\mathbf{e}}_\omega + \delta\mathbf{f}) - T(L_N, \hat{\mathbf{e}}_\omega, \hat{\mathbf{e}}_\omega)] = \begin{bmatrix} 7.3819\text{e-}004 & 0.5688 \\ 0.5688 & 1.7544\text{e+}003 \end{bmatrix}$$

- Using the same non-smooth potential, we obtain

$$T(L_N, \hat{\mathbf{e}}_\omega + \mathbf{f}, \hat{\mathbf{e}}_\omega + \mathbf{f}) - T(L_N, \hat{\mathbf{e}}_\omega - \mathbf{f}, \hat{\mathbf{e}}_\omega - \mathbf{f}) = \begin{bmatrix} 1.6867\text{e-}005 & 3.4202\text{e-}005 \\ 3.4202\text{e-}005 & 1.7100\text{e-}005 \end{bmatrix}$$

Figures 1 and 2 illustrate what happens across *all* Fourier components. In both figures, the top two plots show the real part of the inverse discrete Fourier transform of the Fourier components computed by approximating individual Riemann-Stieltjes integrals: $\hat{\mathbf{e}}_\omega^H \exp[-iL_N \Delta t] \hat{\mathbf{e}}_\omega$ and $\hat{\mathbf{e}}_\omega^H \exp[-iL_N \Delta t](\hat{\mathbf{e}}_\omega + \delta \mathbf{u}^n)$ in Fig. 1, and $(\hat{\mathbf{e}}_\omega \pm \mathbf{u}^n)^H \exp[-iL_N \Delta t](\hat{\mathbf{e}}_\omega + \mathbf{u}^n)$ in Fig. 2. As expected, these functions are nearly in agreement with one another.

The bottom left plot shows the real part of the difference quotient of these two functions, and the bottom right plot shows the computed solution in the complex plane. We observe in Fig. 1 that when using the asymmetric perturbation, the cancellation does not eliminate high-frequency oscillations, yielding a solution that is much less regular than the initial data, and in fact is highly inaccurate. The symmetric perturbation illustrated in Fig. 2, on the other hand, does not introduce these spurious high-frequency oscillations, yielding a much more accurate solution, as is demonstrated further in Section 5.3.

To gain some insight into why the symmetric perturbation induces much less change in the recursion coefficients, we note the following: Let $M_{\mathbf{u}_0, \mathbf{v}_0}(\mathbf{u}, \mathbf{v}) : \mathbb{C}^{K \times K} \rightarrow \mathbb{C}^{K \times K}$ denote the mapping that transforms T_1 into T_2 , where T_1 is the tridiagonal matrix produced by the unsymmetric Lanczos algorithm applied to

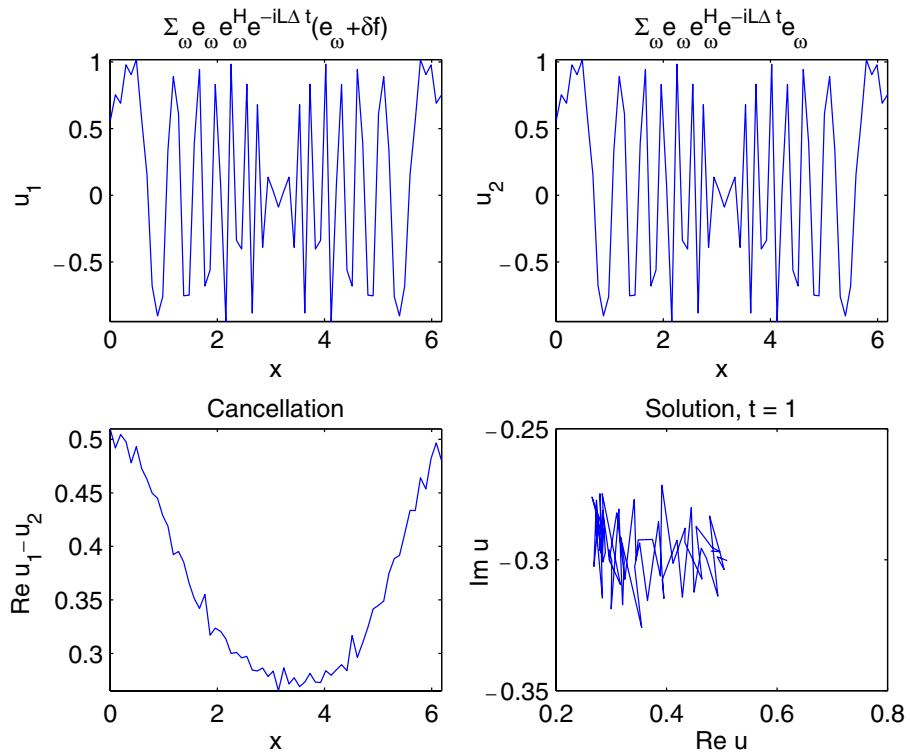


Fig. 1 Unsymmetric perturbation

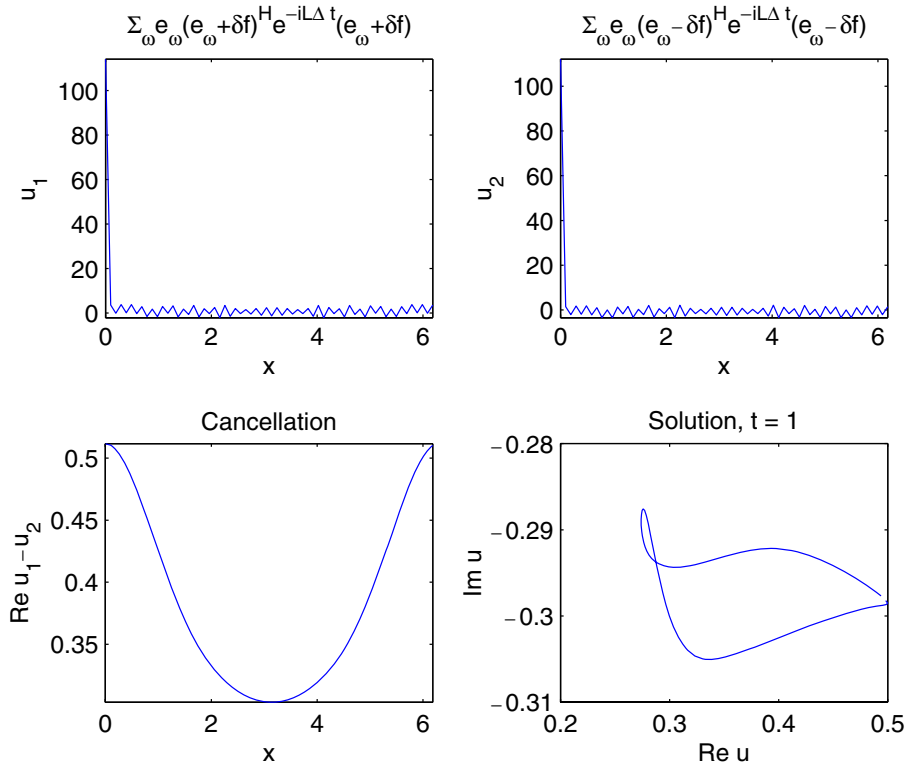


Fig. 2 Symmetric perturbation

L_N with initial vectors \mathbf{u}_0 and \mathbf{v}_0 and T_2 is the tridiagonal matrix corresponding to initial vectors $\mathbf{u}_0 + \mathbf{u}$ and $\mathbf{v}_0 + \mathbf{v}$. Then, we have

$$M_{\mathbf{u},\mathbf{u}}(\mathbf{v}, \mathbf{v}) = [M_{\mathbf{u}+\mathbf{v},\mathbf{u}+\mathbf{v}}(-\mathbf{v}, \mathbf{0})]^{-1} M_{\mathbf{u},\mathbf{u}}(\mathbf{0}, \mathbf{v}). \tag{3.21}$$

That is, the update of recursion coefficients by a symmetric perturbation of the initial vector is the composition of an update by an unsymmetric perturbation, and the *inverse* of another update by an unsymmetric perturbation. Therefore, although the two unsymmetric updates, individually, can yield large changes in the recursion coefficients, this view of the process suggests that a certain cancellation takes place, resulting in greater stability in the recursion coefficients from a symmetric perturbation of the initial vector. It should be noted that explicit formulas for the updated coefficients via an unsymmetric perturbation of the right initial vector are given in [26].

3.5 Non-smooth initial data

Now, suppose that both the potential $V(x)$ and the initial data $f(x)$ are not smooth. In this case, additional modification is necessary, as will be seen in

Section 5.4, due to loss of accuracy. In applying the Lanczos algorithm with an initial vector such as $\hat{\mathbf{e}}_\omega \pm \mathbf{u}^n$, high-frequency oscillations caused by the repeated application of L_N to \mathbf{u}^n make the recursion coefficients corresponding to higher frequencies much more sensitive to perturbations than when the solution is smooth.

To reduce this influence, we scale the perturbation by a small constant δ , just as in the original formulation of KSS methods that use an asymmetric perturbation. In the following example, we use the nonsmooth potential and data defined in Section 5.4, for a 64-point grid, and examine the recursion coefficients corresponding to the wave number $\omega = 24$. We observe the following:

- Using an unscaled symmetric perturbation, we obtain

$$T(L_N, \hat{\mathbf{e}}_\omega + \mathbf{f}, \hat{\mathbf{e}}_\omega + \mathbf{f}) - T(L_N, \hat{\mathbf{e}}_\omega - \mathbf{f}, \hat{\mathbf{e}}_\omega - \mathbf{f}) = \begin{bmatrix} 9.2421\text{e-}003 & 1.8645\text{e-}002 \\ 1.8645\text{e-}002 & 9.5283\text{e-}003 \end{bmatrix}$$

Although these values are small, they are 2–3 orders of magnitude larger than when the data is smooth, as in the previous example.

- Using the same non-smooth data, but scaling the perturbation by $\delta = 10^{-2}$, we obtain

$$\begin{aligned} & \frac{1}{\delta} [T(L_N, \hat{\mathbf{e}}_\omega + \delta\mathbf{f}, \hat{\mathbf{e}}_\omega + \delta\mathbf{f}) - T(L_N, \hat{\mathbf{e}}_\omega - \delta\mathbf{f}, \hat{\mathbf{e}}_\omega - \delta\mathbf{f})] \\ &= \begin{bmatrix} 6.3968\text{e-}004 & 4.0920\text{e-}003 \\ 4.0920\text{e-}003 & 8.3726\text{e-}004 \end{bmatrix} \end{aligned}$$

We see that the rate of change in these recursion coefficients is roughly an order of magnitude smaller than with the unscaled perturbation. It will be seen in Section 5.4 that this scaling significantly improves accuracy.

4 Practical implementation

In spite of the high-order accuracy of KSS methods, their straightforward implementation of the algorithm described in Section 2 is much too slow to be used as a time-stepping scheme, because at least $O(N^2 \log N)$ operations are required to carry out the Lanczos iteration $2N$ times, if differentiation is implemented using the FFT. Fortunately, we can take advantage of the fact that L_N represents a differential operator in order to optimize this process.

To that end, we now describe a data structure that can be used to efficiently compute and represent Lanczos vectors corresponding to all N Fourier components simultaneously, as well as the elements of the corresponding Jacobi matrices, as functions of the frequency ω .

4.1 Data structure

The data contained in this type is a representation of a function of the form

$$f(x; \omega) = \sum_{j=1}^m f_C^j(\omega) f_F^j(x) + \sum_{k=1}^n f_{\hat{C}}^k(\omega) f_{\hat{F}}^k(x) e^{i\omega x}. \tag{4.1}$$

Specifically, a function $f(x; \omega)$ is defined by four ordered collections:

$$f_F = \{f_F^1(x), \dots, f_F^m(x)\}, \tag{4.2}$$

$$f_C = \{f_C^1(\omega), \dots, f_C^m(\omega)\}, \tag{4.3}$$

$$f_{\hat{F}} = \{f_{\hat{F}}^1(x), \dots, f_{\hat{F}}^n(x)\}, \tag{4.4}$$

$$f_{\hat{C}} = \{f_{\hat{C}}^1(\omega), \dots, f_{\hat{C}}^n(\omega)\}. \tag{4.5}$$

We denote the sizes of these collections by $|f_F|$, $|f_C|$, $|f_{\hat{F}}|$, and $|f_{\hat{C}}|$, respectively. For each such collection, we use a superscript to denote a single element.

Given an N -point uniform grid of the form

$$x_j = jh, \quad j = 0, \dots, N - 1, \quad h = 2\pi/N, \tag{4.6}$$

a function $f(x; \omega)$ can be represented using matrices $F_F, F_C, F_{\hat{F}}, F_{\hat{C}}$, defined as follows:

$$F_F = [\mathbf{f}_F^1 \ \cdots \ \mathbf{f}_F^m], \quad m = |f_F|, \quad [\mathbf{f}_F^j]_k = f_F^j(kh), \tag{4.7}$$

$$F_C = [\mathbf{f}_C^1 \ \cdots \ \mathbf{f}_C^m], \quad m = |f_C|, \quad [\mathbf{f}_C^j]_k = f_C^j(k - N/2), \tag{4.8}$$

$$F_{\hat{F}} = [\mathbf{f}_{\hat{F}}^1 \ \cdots \ \mathbf{f}_{\hat{F}}^k], \quad k = |f_{\hat{F}}|, \quad [\mathbf{f}_{\hat{F}}^j]_k = f_{\hat{F}}^j(kh), \tag{4.9}$$

$$F_{\hat{C}} = [\mathbf{f}_{\hat{C}}^1 \ \cdots \ \mathbf{f}_{\hat{C}}^k], \quad k = |f_{\hat{C}}|, \quad [\mathbf{f}_{\hat{C}}^j]_k = f_{\hat{C}}^j(k - N/2). \tag{4.10}$$

We illustrate the usefulness of this representation with an example. Let \mathbf{u}^n denote the solution to (1.1)–(1.3) after n time steps, computed using Algorithm 1. Furthermore, let $L(x, D)$ be defined as in (1.4), with corresponding discretization L_N defined as in (2.5). If the unsymmetric Lanczos algorithm is applied to the matrix L_N with initial vectors $\hat{\mathbf{e}}_\omega$ and $\hat{\mathbf{e}}_\omega + \delta_\omega \mathbf{u}^n$, for $\omega = -N/2 + 1, \dots, N/2$, then the Lanczos vector \mathbf{r}_1 can be represented by the vector-valued function

$$\begin{aligned} \mathbf{r}_1(\omega) &= (L_N - \alpha_1(\omega))\mathbf{x}_1(\omega) \\ &= \frac{1}{\beta_0(\omega)} [L_N \hat{\mathbf{e}}_\omega + \delta_\omega L_N \mathbf{u}^n - \alpha_1(\omega) \hat{\mathbf{e}}_\omega - \alpha_1(\omega) \mathbf{u}^n] \\ &= \frac{1}{\beta_0(\omega)} [-p\omega^2 \hat{\mathbf{e}}_\omega + \mathbf{V}_N \hat{\mathbf{e}}_\omega + \delta_\omega p D_N^2 \mathbf{u}^n + \delta_\omega \mathbf{V}_N \mathbf{u}^n - \alpha_1(\omega) \hat{\mathbf{e}}_\omega - \delta_\omega \alpha_1(\omega) \mathbf{u}^n], \end{aligned}$$

where multiplication of vectors is component-wise, and \mathbf{V}_N is an N -point discretization of $V(x)$. It follows that $\mathbf{r}_1(\omega)$ is represented by the matrices

$$\begin{aligned}
 F_F &= [pD_N^2 \mathbf{u}^n \mathbf{V}_N \mathbf{u}^n \mathbf{u}^n], \\
 F_C &= \frac{1}{\beta_0(\omega)} [\delta_\omega \delta_\omega - \delta_\omega \alpha_1(\omega)], \\
 F_{\hat{F}} &= [p \mathbf{V}_N \mathbf{1}], \\
 F_{\hat{C}} &= \frac{1}{\beta_0(\omega)} [-\omega^2 \mathbf{1} - \alpha_1(\omega)].
 \end{aligned}$$

Note that $\hat{\epsilon}_\omega$ is absent in this representation. Its role is taken into account implicitly by using two pairs of collections instead of one, where each pair includes a collection of discretized functions and a collection of coefficients, which are functions of ω .

Now, suppose that we wish to compute $\beta_1(\omega)^2 = \mathbf{r}_1(\omega)^H \mathbf{r}_1(\omega)$. This inner product requires computing all possible inner products of vectors in F_F and $F_{\hat{F}}$, but because all dependencies on ω are relegated to the matrices F_C and $F_{\hat{C}}$, each inner products need only be computed once, rather than for each ω . We will see that the N inner products needed to compute $\mathbf{r}_1(\omega)$ for all N values of ω can be computed in $O(N \log N)$ time, rather than $O(N^2)$.

4.2 Operations

This data structure supports the following operations:

- Addition and subtraction: The sum of two functions $f(x; \omega)$ and $g(x; \omega)$ is represented by the function

$$h(x; \omega) = f(x; \omega) \oplus g(x; \omega), \tag{4.11}$$

which has values $h(x; \omega) = f(x; \omega) + g(x; \omega)$. The \oplus operation can be implemented as follows:

```

n = |f_F|
H_F = F_F
H_C = F_C
ℓ = 1
for j = 1, ..., |g_F|
    if g_F^j = f_F^k for some k
        h_C^k = h_C^k + g_C^j
    else
        h_F^{n+ℓ} = g_F^j
        h_C^{n+ℓ} = g_C^j
        ℓ = ℓ + 1
    end
end
end
    
```

```

n = |fF̂|
HF̂ = FF̂
HĈ = FĈ
ℓ = 1
for j = 1, ..., |gF̂|
  if gF̂j = fF̂k for some k
    hĈk = hĈk + gĈj
  else
    hF̂n+ℓ = gF̂j
    hĈn+ℓ = gĈj
    ℓ = ℓ + 1
  end
end

```

Similarly, the difference of two functions $f(x; \omega)$ and $g(x; \omega)$ is represented by the function

$$h(x; \omega) = f(x; \omega) \ominus g(x; \omega) \tag{4.12}$$

which has values $h(x; \omega) = f(x; \omega) - g(x; \omega)$. The implementation of \ominus simply negates the coefficients of $g(x; \omega)$ and then performs the same underlying operations as \oplus .

In the worst case, where $\{f_F\} = \{g_F\}$ and $\{f_{\hat{F}}\} = \{g_{\hat{F}}\}$, $(|f_F| + |g_F|)N$ floating-point operations are required for both \oplus and \ominus . In any case, $(|f_F| + |g_F| + |f_{\hat{F}}| + |g_{\hat{F}}|)N$ data movements and/or floating-point operations are needed.

While it is not absolutely necessary to check whether the function collections for f and g have any elements in common, it is highly recommended, since applying differential operators to these representations can cause the sizes of these collections to grow rapidly, and therefore it is wise to take steps to offset this growth wherever possible.

- **Scalar multiplication:** Given a function $f(x; \omega)$ and an expression $s(\omega)$, the operation $g(x; \omega) = s(\omega) \otimes f(x; \omega)$ scales each of the coefficients of $f(x; \omega)$ by $s(\omega)$, yielding the result $g(x; \omega)$ which is a function with values $g(x; \omega) = s(\omega) f(x; \omega)$. The \otimes operation can be implemented as follows:

```

GF = FF
for j = 1, ..., |fC|
  gCj = s fCj
end
GF̂ = FF̂
for j = 1, ..., |fĈ|
  gĈj = s fĈj
end

```

This implementation requires $(|f_C| + |f_{\hat{C}}|)N$ floating-point operations.

- Application of differential operator: The operation $g(x; \omega) = L(x, D) \wedge f(x; \omega)$ computes a representation $g(x; \omega)$ of the result of applying the m th-order differential operator $L(x, D)$ to $f(x; \omega)$ satisfying $g(x; \omega) = L(x, D)f(x; \omega)$ for each ω . The following implementation makes use of the rule

$$AB(x, \xi) = \sum_{\alpha} \frac{1}{\alpha!} \frac{\partial^{\alpha} A}{\partial \xi^{\alpha}} \frac{\partial^{\alpha} B}{\partial x^{\alpha}}. \tag{4.13}$$

for computing the symbol of the product of two differential operators $A(x, D)$ and $B(x, D)$. In describing the implementation, we do not assume a particular discretization of $L(x, D)$ or the functions in the collections f_F or $f_{\hat{F}}$; this issue will be discussed later in this section.

```

gC = fC
for j = 1, ..., |fF|
    gF^j(x) = L(x, D) fF^j(x)
end
l = 1
for j = 1, ..., |fF-hat|
    for k = 0, ..., m
        gF-hat^l = (d^k L / d xi^k)(x, D) fF-hat^j
        gC-hat^l = (i*omega)^k / k! fC-hat^j
        l = l + 1
    end
end
    
```

This implementation requires $Nm(1 + |f_{\hat{F}}|) + |f_{\hat{F}}|(Dm(m + 1)/2 + M(m + 1)(m + 2)/2) + |f_F|(Dm + M(m + 1))$ floating-point operations, where D is the number of operations required for differentiation and M is the number of operations required for pointwise multiplication of two functions. On an N -point uniform grid, $M = N$ and $D = 3N \log N + N$, provided that N is also a power of 2.

- Inner product: The operation $h(\omega) = f(x; \omega) \odot g(x; \omega)$ computes a representation of the inner product of $f(x; \omega)$ and $g(x; \omega)$, resulting in an expression $h(\omega)$ with values $\langle f(\cdot; \omega), g(\cdot; \omega) \rangle$. The following algorithm can be used to implement \odot . T represents the discrete Fourier transform operator.

```

h = 0
for j = 1, ..., |fC|
    for k = 1, ..., |gC|
        h = h + (f_C^j g_C^k) * ((f_F^j)^H g_F^k)
    end
    
```

```

for  $k = 1, \dots, |g_{\hat{C}}|$ 
     $\hat{v} = T[\mathbf{f}_{\hat{F}}^j \mathbf{g}_{\hat{F}}^k]$ 
     $\mathbf{h} = \mathbf{h} + (\mathbf{f}_{\hat{C}}^j \mathbf{g}_{\hat{C}}^k) \hat{v} \sqrt{2\pi}/h$ 
end
end
for  $j = 1, \dots, |f_{\hat{C}}|$ 
    for  $k = 1, \dots, |g_C|$ 
         $\hat{v} = T[\mathbf{f}_{\hat{F}}^j \mathbf{g}_{\hat{F}}^k]$ 
         $\mathbf{h} = \mathbf{h} + (\mathbf{f}_{\hat{C}}^j \mathbf{g}_{\hat{C}}^k) \hat{v} \sqrt{2\pi}/h$ 
    end
    for  $k = 1, \dots, |g_{\hat{C}}|$ 
         $\mathbf{h} = \mathbf{h} + (\mathbf{f}_{\hat{C}}^j \mathbf{g}_{\hat{C}}^k) * ((\mathbf{f}_{\hat{F}}^j)^H \mathbf{g}_{\hat{F}}^k)$ 
    end
end

```

The above implementation requires $3N(|f_C||g_C| + |f_{\hat{C}}||g_{\hat{C}}|) + (\frac{3}{2}N \log N + 4N + 1)(|f_C||g_{\hat{C}}| + |f_{\hat{C}}||g_C|)$ floating-point operations.

This set of operations is sufficient to carry out the Lanczos iteration symbolically given two functions as initial Lanczos vectors. The result of the iteration is the set of all Jacobi matrices for all wave numbers $\omega = -N/2 + 1, \dots, N/2$.

4.3 Computation of Jacobi matrices

For each frequency ω , the Krylov subspace spectral method, applied with $K = 2$ Gaussian quadrature nodes and $M = 0$ prescribed nodes, computes two 2×2 Jacobi matrices J_ω and $J_{\omega,n}$ in order to approximate the quantities $D_\omega = \mathbf{e}_\omega^H S_N(\Delta t) \mathbf{e}_\omega$ and $F_\omega^n = \mathbf{e}_\omega^H S_N(\Delta t) (\mathbf{e}_\omega + \delta_\omega \mathbf{u}(t_n))$, where $S_N(\Delta t) = \exp[-iL_N \Delta t]$. Clearly, D_ω need only be computed once, while F_ω^n must be recomputed at each time step. The computation of the Jacobi matrix J_ω used to obtain D_ω proceeds as follows:

$$\begin{aligned}
 [r_0]_{\hat{C}} &= 1 \\
 [r_0]_{\hat{F}} &= 1 \\
 \beta_0^2 &= r_0 \odot r_0 \\
 x_1 &= r_0 \oslash \beta_0 \\
 L_1 &= L \wedge x_1 \\
 \alpha_1 &= x_1 \odot L_1 \\
 r_1 &= L_1 \ominus (\alpha_1 \otimes x_1) \\
 \beta_1^2 &= r \odot r \\
 x_2 &= r_1 \oslash \beta_1 \\
 L_2 &= L \wedge x_2 \\
 \alpha_2 &= x_2 \odot L_2
 \end{aligned}$$

In an efficient implementation, a total of $38N + 9N \log N - 6$ floating-point operations are required. Efficiency can be improved by applying standard optimization techniques such as common subexpression elimination (see [2] for details). For example, on an operator of the form (1.4), the entries of J_ω , in the case of $K = 2$, can be computed in only $13N + \frac{3}{2}N \log N - 1$ floating-point operations using the representations of α_1, β_1 and α_2 , which were originally derived in [26].

This technique can be applied to the symmetrized KSS method described in Section 3.4, that employs the polar decomposition (3.20). In this case, it is advantageous to modify the data structure for Lanczos vectors to include an unspecified scaling factor for each term, that can assume the values $1, -1, i$, or $-i$, so that a single data structure can be used to handle all four perturbations and save storage space.

5 Numerical results

In this section, we will present numerical results for comparisons between the original KSS method (as described in Section 2) and the symmetrized KSS method described in Section 3. The comparisons will focus on the accuracy of the temporal approximations employed by each method. For convenience, we denote by $KSS(K)$ the original KSS method with K Gaussian nodes, and by $KSS-S(K)$ the symmetrized KSS method with K block Gaussian nodes, where the perturbation is unscaled, as in Section 3.4. We denote by $KSS-SD(K)$ the modified KSS method described in Section 3.5, in which symmetric perturbations are used, but they are scaled by a small constant δ .

5.1 Construction of test cases

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

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

$$f_{j,k}^0(x) = \operatorname{Re} \left\{ \sum_{|\omega| < N/2, \omega \neq 0} \hat{f}_j(\omega) (1 + |\omega|)^{-(k+1)} e^{i\omega x} \right\}, \quad j, k = 0, 1, \dots, \tag{5.1}$$

where

$$\hat{f}_j(\omega) = R_{jN+2(\omega+N/2)-1} + iR_{jN+2(\omega+N/2)}. \tag{5.2}$$

The parameter j indicates how many functions have been generated in this fashion since setting MATLAB’s random number generator to its initial state, and the parameter k indicates how smooth the function is. Figure 3 shows selected functions from this collection.

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

$$E^+ f(x) = f(x) - \min_{x \in [0, 2\pi]} f(x) + 1, \tag{5.3}$$

$$E^- f(x) = f(x) - \max_{x \in [0, 2\pi]} f(x) - 1. \tag{5.4}$$

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

$$g_{j,k}(x, y) = \text{Re} \left\{ \sum_{|\omega|, |\xi| < N/2, \omega\xi \neq 0} \hat{g}_j(\omega, \xi) (1 + |\omega|)^{-(k+1)} (1 + |\xi|)^{-(k+1)} e^{i(\omega x + \xi y)} \right\}, \tag{5.5}$$

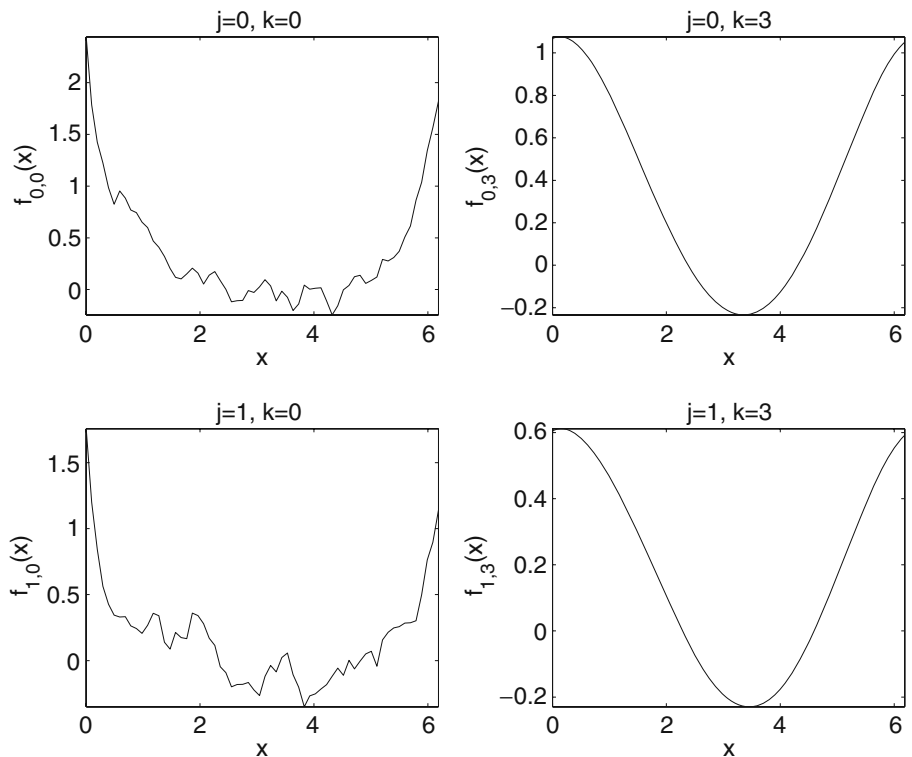


Fig. 3 Functions from the collection $f_{j,k}(x)$, for selected values of j and k

where j and k are nonnegative integers, and

$$\hat{g}_j(\omega, \xi) = R_{jN^2+2[N(\omega+N/2-1)+(\xi+N/2)]-1} + iR_{jN^2+2[N(\omega+N/2-1)+(\xi+N/2)]}. \tag{5.6}$$

Figure 4 shows selected functions from this collection.

In all experiments, solutions $u^{(j)}(x, t)$ are computed using time steps $\Delta t = 2^{-j}$, for $j = 0, \dots, 6$. The error estimates are obtained by computing $\|u^{(j)}(\cdot, 1) - u^{(6)}(\cdot, 1)\|/\|u^{(6)}(\cdot, 1)\|$. This method of estimating error assumes that $u^{(6)}(x, t)$ is a sufficiently accurate approximation to the exact solution, which is unknown for the problems under consideration, but this has proven in practice to be a valid assumption by comparing $u^{(6)}$ against approximate solutions computed using established methods, and by comparing $u^{(6)}$ against solutions obtained using various methods with smaller time steps. In [26], errors measured by comparison to exact solutions of problems with source terms further validated the convergence behavior. It should be noted that we are not seeking a sharp estimate of the error, but rather an indication of the rate of convergence, and for this goal, using $u^{(6)}$ as an approximation to the exact solution is sufficient.

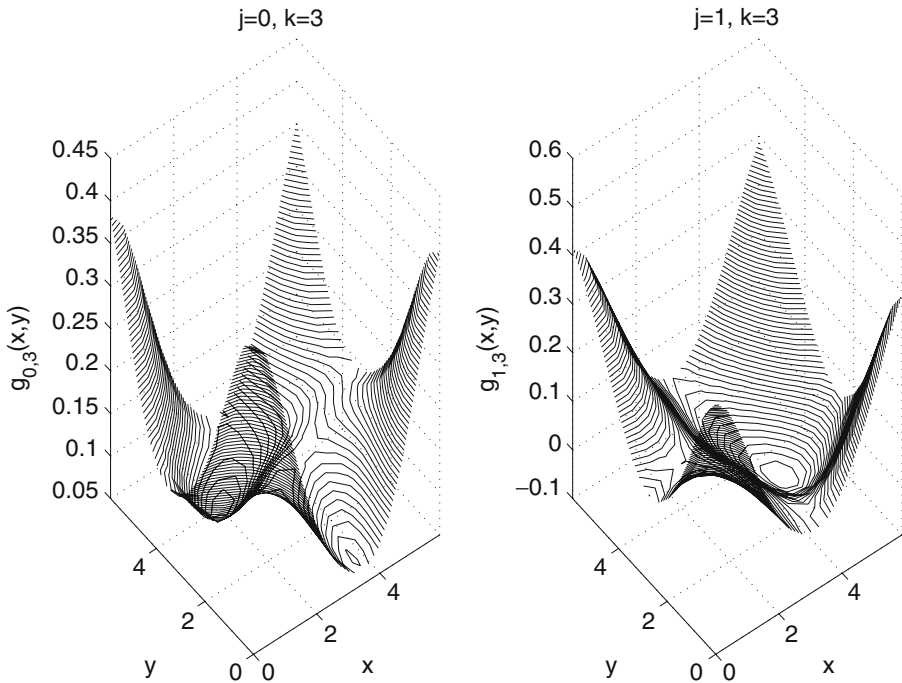


Fig. 4 Functions from the collection $g_{j,k}(x, y)$, for selected values of j and k

5.2 Smooth potential and data

We apply KSS methods to the problem

$$u_t - i \left[\overline{E^+ f_{0,3} u_{xx}} + E^- f_{1,3} u \right] = 0, \quad 0 < x < 2\pi, \quad t > 0, \quad (5.7)$$

$$u(x, 0) = E^+ f_{2,3}(x), \quad 0 < x < 2\pi, \quad (5.8)$$

with periodic boundary conditions (1.3), where we use the notation

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

For comparison, we also apply the MATLAB solver `ode23s`, which is specifically designed for stiff problems such as this. Its algorithm is described in [34]. We also applied `ode45`, due to its high order of accuracy, but we do not include

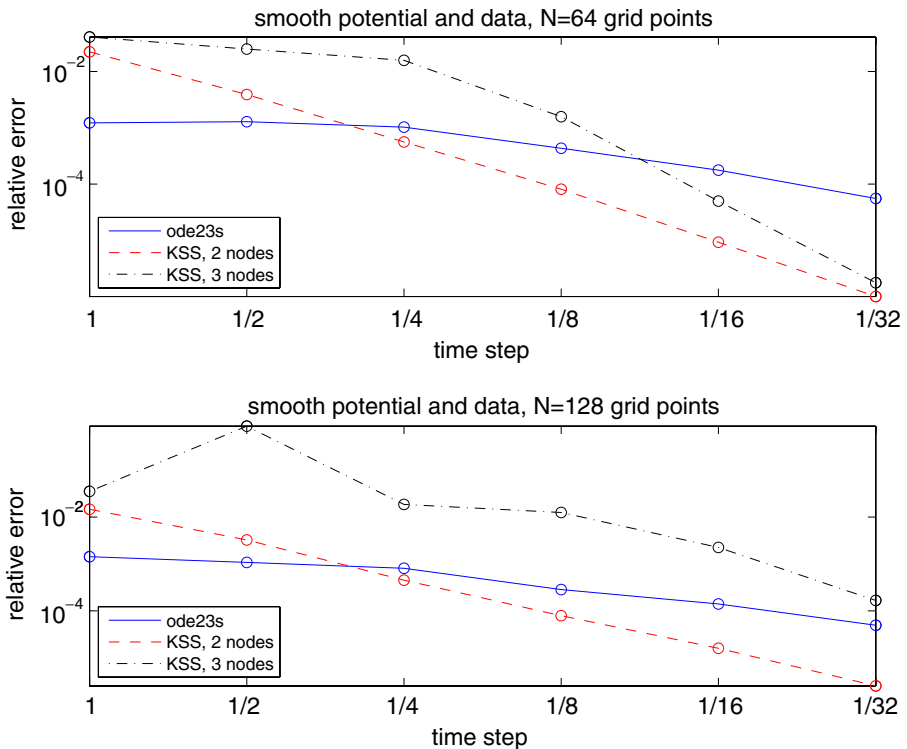


Fig. 5 Estimates of relative error in the solution of (5.7), (5.8), (1.3) by the MATLAB solver `ode23s`, the 2-node KSS method KSS(2) with unsymmetric perturbation, and the 3-node KSS method KSS(3) with unsymmetric perturbation, on uniform grids with $N = 64$ and $N = 128$ points

the results because, for the time-dependent Schrödinger equation, it is not competitive with `ode23s`.

Figure 5 and Table 1 describe the performance of `ode23s`, and 2- and 3-node KSS methods, for uniform 64- and 128-point grids. We observe that `ode23s`, a mixed 2nd- and 3rd-order method, only converges superlinearly, and in fact does not show signs of converging at all for larger time steps. KSS(2) is much more consistent with theoretical expectations of 3rd-order convergence (for all methods, the error estimates include both spatial and temporal error, but the latter is dominant due to the relatively large values of Δt). KSS(3), unfortunately, does not perform as well, only approaching its expected 5th-order convergence for smaller time steps. This issue will be examined later in this section.

Table 1 Estimates of relative error in the solution of (5.7), (5.8), (1.3) by the MATLAB solver `ode23s`, the 2-node KSS method KSS(2) with unsymmetric perturbation, and the 3-node KSS method KSS(3) with unsymmetric perturbation, on uniform grids with $N = 64$ and $N = 128$ points

Method	N	Δt	Error	Order
ode23s	64	1	0.0012	1.4
		1/2	0.0013	
		1/4	0.001	
		1/8	0.00043	
		1/16	0.00018	
		1/32	5.6e-005	
	128	1	0.0014	1.35
		1/2	0.0011	
		1/4	0.00081	
		1/8	0.00028	
		1/16	0.00014	
		1/32	4.9e-005	
KSS(2)	64	1	0.022	3.04
		1/2	0.0039	
		1/4	0.00056	
		1/8	8.1e-005	
		1/16	9.3e-006	
		1/32	1e-006	
	128	1	0.015	2.49
		1/2	0.0033	
		1/4	0.00045	
		1/8	7.9e-005	
		1/16	1.6e-005	
		1/32	2.5e-006	
KSS(3)	64	1	0.041	4.37
		1/2	0.025	
		1/4	0.016	
		1/8	0.0016	
		1/16	5e-005	
		1/32	1.8e-006	
	128	1	0.035	2.27
		1/2	0.87	
		1/4	0.019	
		1/8	0.012	
		1/16	0.0023	
		1/32	0.00017	

The last column lists estimates of the temporal order of convergence

5.3 Non-smooth potential

Next, we consider the problem

$$u_t - i \left[\overline{E^+ f_{0,1}} u_{xx} + E^- f_{1,1} u \right] = 0, \quad 0 < x < 2\pi, \quad t > 0, \quad (5.9)$$

$$u(x, 0) = E^+ f_{2,3}(x), \quad 0 < x < 2\pi, \quad (5.10)$$

with periodic boundary conditions (1.3). This differs from (5.7), (5.8) in that the potential has two fewer orders of differentiability. The two potentials are shown in Fig. 6.

Figure 7 and Table 2 describe the performance of `ode23s`, the standard 2-node KSS method KSS(2), and a symmetrized 2-node KSS method KSS-S(2), as described in Section 3.4, for 64- and 128-point uniform grids. We observe that compared to the case of a smooth potential, KSS(2), while maintaining its order of accuracy, exhibits significantly larger error, and substantial degradation of performance when the number of grid points increases. KSS-S(2), on the other hand, performs far better, actually *improving* as the grid is refined, with order of convergence very close to the expected value of 3.

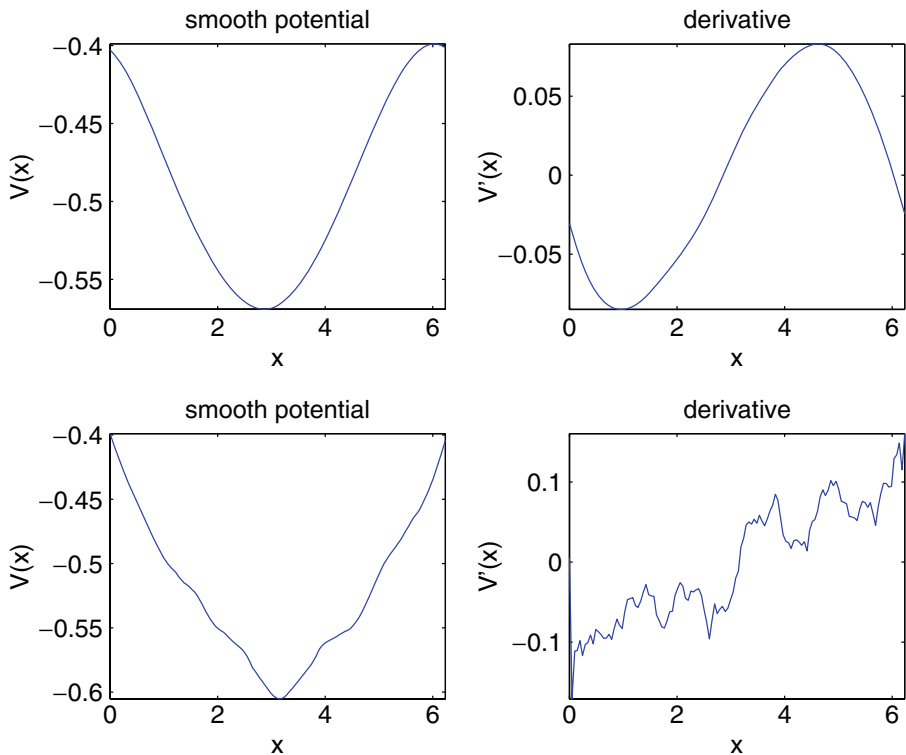


Fig. 6 Smooth and non-smooth potentials featured in (5.7) (left plot) and (5.9) (right plot) for the case of $N = 128$ grid points

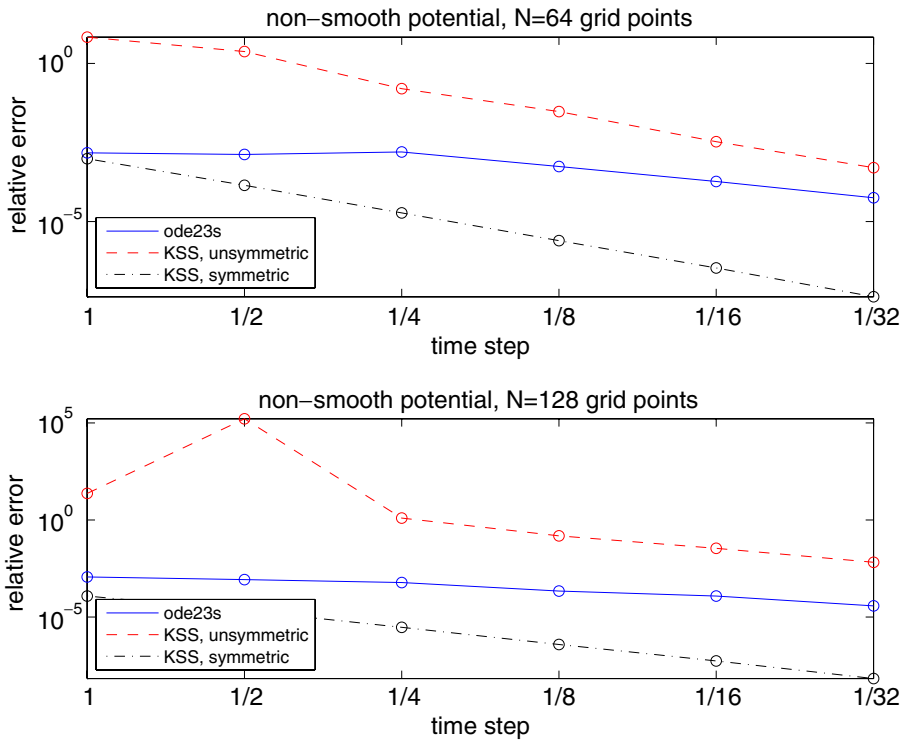


Fig. 7 Estimates of relative error in the solution of (5.9), (5.10), (1.3) by the MATLAB solver ode23s, the 2-node KSS method KSS(2) with unsymmetric perturbation, and the 2-node KSS method KSS-S(2) with symmetric perturbation, on uniform grids with $N = 64$ and $N = 128$ points

5.4 Non-smooth initial data

Next, we consider the problem

$$u_t - i \left[E^+ f_{0,1} u_{xx} + E^- f_{1,1} u \right] = 0, \quad 0 < x < 2\pi, \quad t > 0, \quad (5.11)$$

$$u(x, 0) = E^+ f_{2,1}(x), \quad 0 < x < 2\pi, \quad (5.12)$$

with periodic boundary conditions (1.3). This differs from (5.9), (5.10) in that the initial data has two fewer orders of differentiability.

Figure 8 and Table 3 describe the performance of ode23s, a symmetrized 2-node KSS method KSS-S(2), and a 2-node KSS method KSS-SD(2) with

Table 2 Estimates of relative error in the solution of (5.9), (5.10), (1.3) by the MATLAB solver ode23s, the 2-node KSS method KSS(2) with unsymmetric perturbation, and the 2-node KSS method KSS-S(2) with symmetric perturbation, on uniform grids with $N = 64$ and $N = 128$ points

Method	N	Δt	Error	Order
ode23s	64	1	0.0015	1.6
		1/2	0.0013	
		1/4	0.0016	
		1/8	0.00055	
		1/16	0.00019	
		1/32	5.7e-005	
	128	1	0.0012	1.33
		1/2	0.00085	
		1/4	0.0006	
		1/8	0.00022	
		1/16	0.00012	
		1/32	3.8e-005	
KSS(2)	64	1	6.8	2.77
		1/2	2.4	
		1/4	0.16	
		1/8	0.03	
		1/16	0.0034	
		1/32	0.00051	
	128	1	24	2.52
		1/2	1.7e+005	
		1/4	1.3	
		1/8	0.15	
		1/16	0.035	
		1/32	0.0067	
KSS-S(2)	64	1	0.00097	2.94
		1/2	0.00014	
		1/4	1.9e-005	
		1/8	2.5e-006	
		1/16	3.4e-007	
		1/32	4.2e-008	
	128	1	0.00012	2.92
		1/2	2.2e-005	
		1/4	3e-006	
		1/8	3.9e-007	
		1/16	5.5e-008	
		1/32	6.9e-009	

The last column lists estimates of the temporal order of convergence

scaled symmetric perturbation, as described in Section 3.5, for 64- and 128-point uniform grids. For KSS-S(2), we use a scaling parameter of $\delta = 10^{-2}$ on the 64-point grid, and 10^{-3} for the 128-point grid, which is necessary because the increased resolution causes greater amplification of the highest-frequency components of the solution.

We observe that the performance of KSS-S(2) is degraded significantly, compared to a problem with a non-smooth potential and smooth initial data, showing no signs of convergence for larger time steps and not even achieving quadratic convergence for smaller time steps. On the other hand, scaling the perturbation dramatically improves performance, maintaining an order of convergence close to the expected cubic, or as close as can be expected given the more significant spatial error, for both larger and smaller time steps.

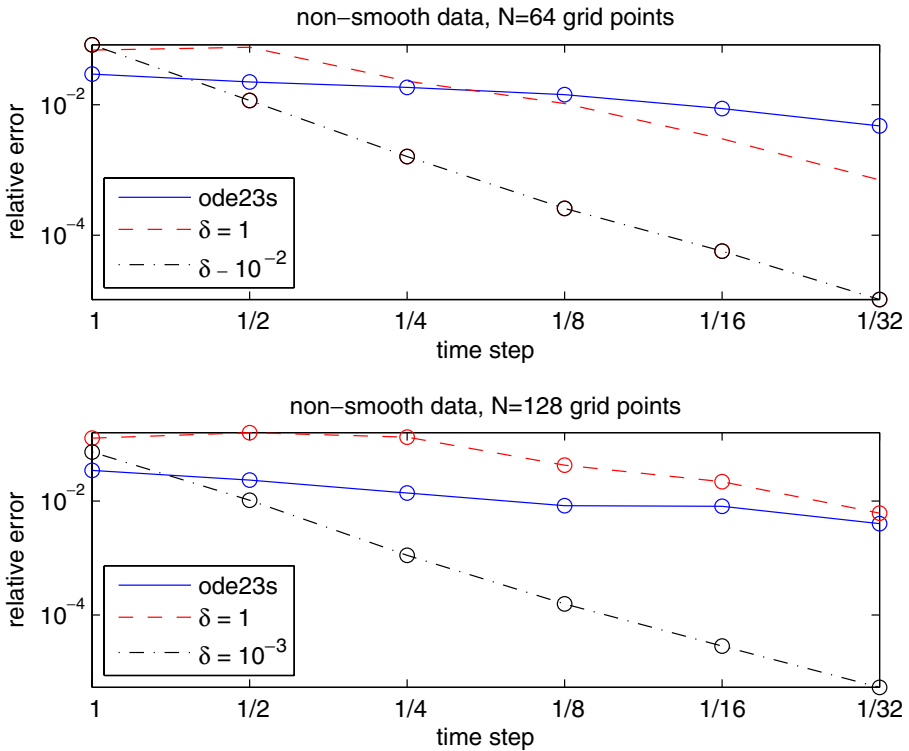


Fig. 8 Estimates of relative error in the solution of (5.11), (5.12), (1.3) by the MATLAB solver ode23s, the 2-node KSS method KSS-S(2) with symmetric perturbation, and the 2-node KSS method KSS-SD(2) with scaled symmetric perturbation, on uniform grids with $N = 64$ and $N = 128$ points. The scaling parameter δ is 10^{-2} on the 64-point grid, and 10^{-3} for the 128-point grid

5.5 Two-dimensional problems

We now attempt to solve the following two-dimensional problems:

- A problem with a smooth potential and initial data,

$$u_t - i \left[\overline{E^+ f_{0,1}} \Delta u + E^- f_{1,2} u \right] = 0, \quad 0 < x < 2\pi, \quad 0 < y < 2\pi, \quad t > 0, \tag{5.13}$$

$$u(x, y, 0) = E^+ f_{2,3}(x, y), \quad 0 < x < 2\pi, \quad 0 < y < 2\pi, \tag{5.14}$$

$$u(x + 2\pi, y, t) = u(x, y + 2\pi, t) = u(x, y, t). \tag{5.15}$$

Table 3 Estimates of relative error in the solution of (5.11), (5.12), (1.3) by the MATLAB solver ode23s, the 2-node KSS method KSS-S(2) with symmetric perturbation, and the 2-node KSS method KSS-SD(2) with scaled symmetric perturbation, on uniform grids with $N = 64$ and $N = 128$ points

Method	N	Δt	Error	Order
ode23s	64	1	0.03	0.653
		1/2	0.022	
		1/4	0.018	
		1/8	0.014	
		1/16	0.0088	
		1/32	0.0047	
	128	1	0.034	0.594
		1/2	0.023	
		1/4	0.014	
		1/8	0.0083	
		1/16	0.0081	
		1/32	0.004	
KSS-S(2)	64	1	0.069	1.68
		1/2	0.076	
		1/4	0.023	
		1/8	0.01	
		1/16	0.003	
		1/32	0.00071	
	128	1	0.13	1.48
		1/2	0.16	
		1/4	0.13	
		1/8	0.043	
		1/16	0.022	
		1/32	0.0061	
KSS-SD(2)	64	1	0.084	2.43
		1/2	0.012	
		1/4	0.0016	
		1/8	0.00026	
		1/16	5.7e-005	
		1/32	1e-005	
	128	1	0.072	2.57
		1/2	0.01	
		1/4	0.0011	
		1/8	0.00016	
		1/16	2.8e-005	
		1/32	5.4e-006	

The scaling parameter δ is 10^{-2} on the 64-point grid, and 10^{-3} for the 128-point grid. The last column lists estimates of the temporal order of convergence

- A problem with non-smooth potential,

$$u_t - i \left[\overline{E^+ f_{0,0}} \Delta u + E^- f_{1,0} u \right] = 0, \quad 0 < x < 2\pi, \quad 0 < y < 2\pi, \quad t > 0, \tag{5.16}$$

with initial condition (5.14) and periodic boundary conditions (5.15). In this problem, the potential $V(x, y) = -E^- f_{1,0}(x, y)$ has two fewer orders of differentiability per dimension than the potential of (5.13). It is shown in Fig. 9.

Figure 10 and Table 4 describe the performance of ode23s, KSS(2) and KSS-S(2). As in other experiments, ode23s converges superlinearly. KSS-S(2) converges nearly cubically, and achieves good accuracy even for larger time steps, on smooth and non-smooth potentials. KSS(2), while showing a

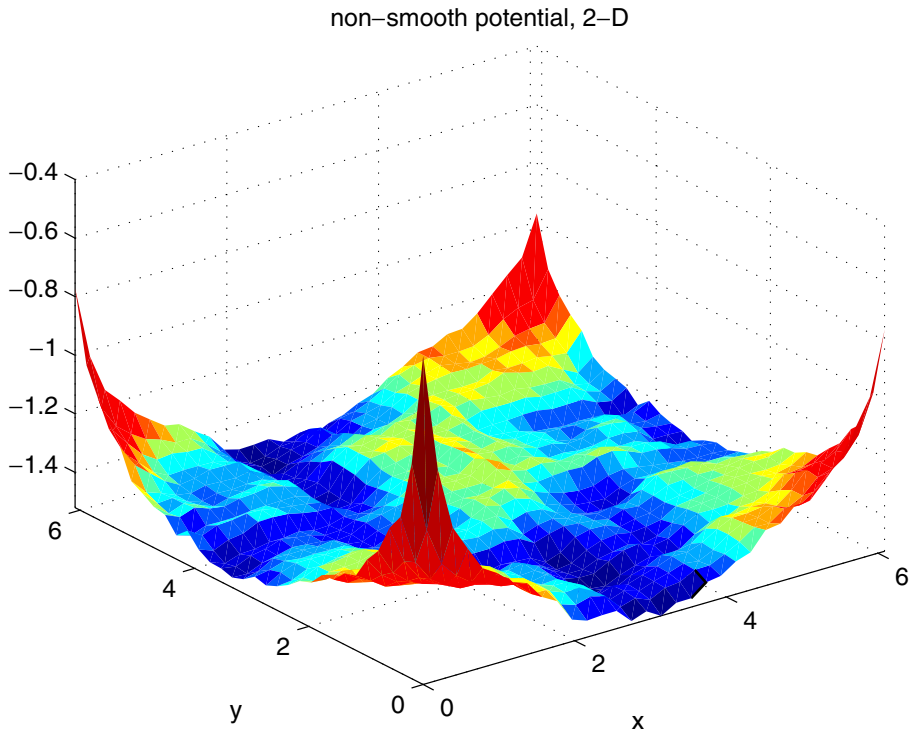


Fig. 9 Non-smooth potential $V(x, y)$ from (5.16)

higher order of convergence, only does so in recovery from relatively poor performance at larger time steps, especially for the non-smooth potential, where no accuracy is obtained at all.

5.6 Unitarity

We examine the departure from unitarity of 1-node and 2-node KSS methods, applied to the problem (5.7), (5.8), (1.3). Table 5 shows that the departure from unitarity is of $O(\Delta t^{2K-1})$ for $K = 1$ and $K = 2$. Future work will examine whether it is possible, perhaps by some alternative symmetrization, to achieve exact unitarity.

5.7 Higher-order methods

We return to the problem (5.7), (5.8), (1.3) in order to examine the performance of higher-order schemes using more than the 2 quadrature nodes used in most of the experiments in this paper. In Section 5.2 we observed that a 3-node method KSS(3) only delivered its expected 5th-order convergence when Δt was sufficiently small, and this threshold lowered as the number of

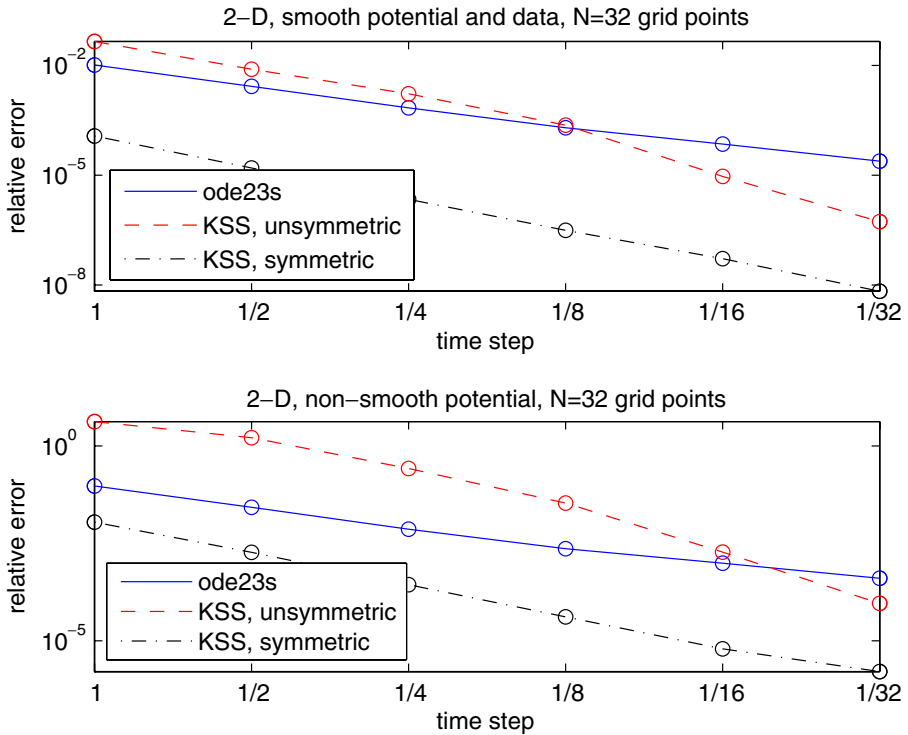


Fig. 10 *Top plot:* Estimates of relative error in the solution of (5.13)–(5.15) on uniform grids with $N = 32$ grid points per dimension. *Bottom plot:* Estimates of relative error in the solution of (5.16), (5.14), (5.15) on uniform grids with $N = 32$ grid points per dimension

grid points increased. Now, we investigate whether a symmetrized perturbation is helpful for this situation, just as it is for non-smooth coefficients or data.

As shown in Fig. 11 and Table 6, we find that with a scaled symmetric perturbation, with $\delta = 10^{-2}$, performance of the 3-node method is improved considerably. The order of convergence in time is greater than 4, and increasing with the number of grid points toward the theoretical expectation of 5th-order, since the spatial error is being reduced.

6 Discussion

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

6.1 Higher space dimension

In [26], it is demonstrated how to compute the recursion coefficients α_j and β_j for operators of the form $Lu = -p\Delta u + q(x, y)u$, and the expressions are

Table 4 Estimates of relative error in the solution of (5.7), (5.8), (1.3) by the MATLAB solver ode23s, the 2-node KSS method KSS(2) with unsymmetric perturbation, and the 2-node KSS method KSS-S(2) with symmetric perturbation, on uniform grids with $N = 32$ grid points per dimension

Method	Potential	Δt	Error	Order
ode23s	smooth	1	0.01	1.62
		1/2	0.0027	
		1/4	0.0007	
		1/8	0.0002	
		1/16	7.1e-005	
		1/32	2.4e-005	
	non-smooth	1	0.094	1.4
		1/2	0.027	
		1/4	0.0073	
		1/8	0.0023	
		1/16	0.00099	
		1/32	0.0004	
KSS(2)	smooth	1	0.045	3.24
		1/2	0.0078	
		1/4	0.0017	
		1/8	0.00023	
		1/16	9.3e-006	
		1/32	5.4e-007	
	non-smooth	1	4.2	3.24
		1/2	1.6	
		1/4	0.26	
		1/8	0.034	
		1/16	0.0019	
		1/32	9.1e-005	
KSS-S(2)	smooth	1	0.00012	2.75
		1/2	1.6e-005	
		1/4	2.2e-006	
		1/8	3.2e-007	
		1/16	5.3e-008	
		1/32	6.8e-009	
	non-smooth	1	0.011	2.75
		1/2	0.0019	
		1/4	0.00027	
		1/8	4.1e-005	
		1/16	6.2e-006	
		1/32	1.6e-006	

The last column lists estimates of the temporal order of convergence

Table 5 L^2 -norm of approximate solutions to (5.7), (5.8), (1.3) at $t = 1$ computed using KSS-S(K), for $K = 1, 2$, at various time steps, to illustrate the departure from unitarity of both methods

K	Δt	$\ \tilde{u}(t, 1)\ _{L^2}$	Order
1	1	1.363817403348700	0.99
	1/2	1.356781290497177	
	1/4	1.353184519719860	
	1/8	1.351379919687993	
2	1	1.349539476878905	2.89
	1/2	1.349570674152341	
	1/4	1.349575731121435	
	1/8	1.349576411860007	

The fourth column indicates the rate at which the departure from unitarity decreases to zero, as a function of Δt

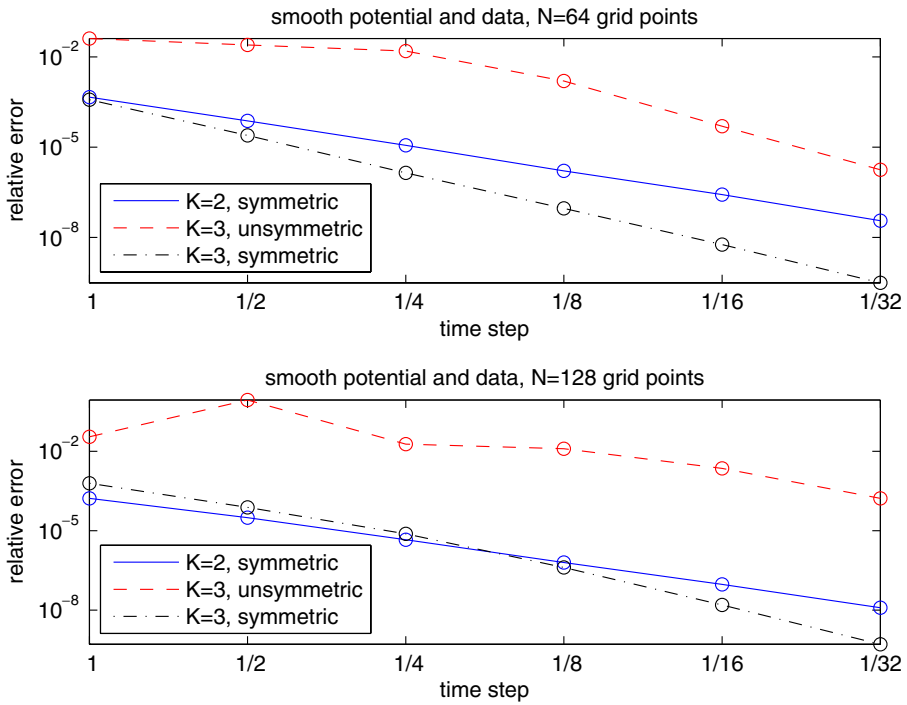


Fig. 11 Estimates of relative error in the solution of (5.7), (5.8), (1.3) by the 2-node KSS method KSS-S(2) with symmetric perturbation, the 3-node KSS method KSS-(2) with unsymmetric perturbation, and the 3-node KSS method KSS-SD(3) with scaled symmetric perturbation, on 64- and 128-point uniform grids. The scaling parameter δ is 10^{-6} for KSS(3), and 10^{-2} for KSS-SD(3)

straightforward generalizations of the expressions given in [26] for the one-dimensional case. It is therefore reasonable to suggest that for operators of this form, the consistency and stability results given here for the one-dimensional case generalize to higher dimensions, as do the data structure and operations for parametrized Lanczos vectors. This will be investigated in the near future.

6.2 Other boundary conditions

While we have used periodic boundary conditions in this paper, it is typical in practical applications of the time-dependent Schrödinger equation to use boundary conditions that are more effective at simulating an infinite domain. One such type of boundary condition is a perfectly matched layer (PML), first used by Berenger in [8] for Maxwell’s equations. A PML absorbs waves by modifying spatial differentiation operators in the PDE. For example, for absorbing waves that propagate in the x direction, $\frac{\partial}{\partial x}$ is replaced by

$$\frac{1}{1 + \frac{i\sigma(x)}{\omega}} \frac{\partial}{\partial x},$$

Table 6 Estimates of relative error in the solution of (5.7), (5.8), (1.3) by the 2-node KSS method KSS-S(2) with symmetric perturbation, the 3-node KSS method KSS-(2) with unsymmetric perturbation, and the 3-node KSS method KSS-SD(3) with scaled symmetric perturbation, on uniform grids with $N = 64$ and $N = 128$ points

Method	N	Δt	Error	Order
KSS-S(2)	64	1	0.00046	2.77
		1/2	7.5e-005	
		1/4	1.1e-005	
		1/8	1.6e-006	
		1/16	2.7e-007	
		1/32	3.6e-008	
	128	1	0.00017	2.84
		1/2	3.1e-005	
		1/4	4.6e-006	
		1/8	6.3e-007	
		1/16	9.4e-008	
		1/32	1.2e-008	
KSS(3)	64	1	0.041	4.37
		1/2	0.025	
		1/4	0.016	
		1/8	0.0016	
		1/16	5e-005	
		1/32	1.8e-006	
	128	1	0.035	2.27
		1/2	0.87	
		1/4	0.019	
		1/8	0.012	
		1/16	0.0023	
		1/32	0.00017	
KSS-SD(3)	64	1	0.00038	4.04
		1/2	2.4e-005	
		1/4	1.4e-006	
		1/8	9.2e-008	
		1/16	5.8e-009	
		1/32	3.1e-010	
	128	1	0.00062	4.62
		1/2	7.6e-005	
		1/4	7.6e-006	
		1/8	4.1e-007	
		1/16	1.6e-008	
		1/32	5.2e-010	

The scaling parameter δ is 10^{-6} for KSS(3), and 10^{-2} for KSS-SD(3). The last column lists estimates of the temporal order of convergence

where, as before, ω represents the wave number, and σ is a positive function that causes propagating waves to be attenuated.

In KSS methods, this transformation can be incorporated into the symbol of the operator L during the computation of the recursion coefficients. The dependence of the transformation on both x and ω makes the efficient application of the transformed operator more difficult, especially in higher space dimensions, but recent work on rapid application of Fourier integral operators (see [10]) can mitigate this concern. Future work will explore the use of PML, taking into account very recent analysis in [30] of the difficulties of PML with inhomogeneous media, and the remediation of these difficulties through adiabatic absorbers.

Another type of boundary condition that is useful for the Schrödinger equation is a transparent boundary condition (TBC), derived independently by several authors (see [7, 19, 31], their inhomogeneous extensions in [1, 4]),

with discrete analogues proposed in [3, 5, 7, 9, 27, 28, 32, 33] which take the form of inhomogeneous Neumann boundary conditions that are also non-local in time. KSS methods can incorporate boundary conditions of this type by a standard transformation of the problem into one with homogeneous Neumann boundary conditions with a source term, but the non-locality makes the efficient implementation of such an approach non-trivial. Future work will also explore the use of these boundary conditions.

6.3 The wave equation

In [18], KSS methods were applied to the second-order wave equation with Dirichlet boundary conditions, thus demonstrating that they are applicable to “true” IBVP, as opposed to the problems discussed in this paper that include periodic boundary conditions. It was shown that each Fourier component of the solution is computed with $O(\Delta t^{4K})$ accuracy. The symmetrization and scaling tactics presented in this paper can easily be applied to the wave equation with rough coefficients or data, as only the integrands differ, not the quadrature rules.

6.4 Scaling perturbations

In the experiments featuring non-smooth potentials or initial data, values of the scaling factor δ for KSS-SD(K) were found to significantly impact accuracy. As there are negative consequences from choices of δ that are either too large or too small, it is important to investigate whether an optimal or near-optimal value of δ can be determined, perhaps from error estimation or analysis of the recursion coefficients or quadrature rules. It is also worthwhile to consider whether different values of δ should be used for each Fourier component.

6.5 Summary

We have demonstrated that for KSS methods can be applied to the time-dependent Schrödinger equation, and achieve the same order of converge and stability as for the parabolic problems to which it has previously been applied. Although the Schrödinger equation is more difficult to solve accurately in the case of non-smooth coefficients or data, KSS methods can easily be modified to handle these difficulties, in one or more space dimensions. Furthermore, these modifications yield a method that has the same algorithmic complexity as the original version, through the use of appropriate data structures for representing parametrized families of Lanczos vectors.

Acknowledgements The author would like to thank the two anonymous referees for their helpful feedback that significantly improved the quality of this paper.

References

1. Abdallah, N.B., Pinaud, O.: A mathematical model for the transient evolution of a resonant tunneling diode. *C. R. Math. Acad. Sci. Paris* **334**, 283–288 (2002)
2. Aho, A.V., Sethi, R., Ullman, J.D.: *Compilers: Principles, Techniques and Tools*. Addison-Wesley, Reading (1988)
3. Antoine, X., Besse, C.: Unconditionally stable discretization schemes of non-reflecting boundary conditions for the one-dimensional Schrödinger equation. *J. Comput. Phys.* **188**, 157–175 (2003)
4. Arnold, A.: Mathematical concepts of open quantum boundary conditions. *Transp. Theory Stat. Phys.* **30**(4–6), 561–584 (2001)
5. Arnold, A., Ehrhardt, M., Sofronov, I.: Discrete transparent boundary conditions for the Schrödinger equation: fast calculation, approximation, and stability. *Comm. Math. Sci.* **1**(3), 501–556 (2003)
6. Atkinson, K.: *An Introduction to Numerical Analysis*, 2nd edn. Wiley, New York (1989)
7. Baskakov, V.A., Popov, A.V.: Implementation of transparent boundaries for numerical solution of the Schrödinger equation. *Wave Motion* **14**, 123–128 (1991)
8. Berenger, J.: A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.* **114**, 185–200 (1994)
9. Bruneau, C.H., Menza, L.D.: Conditions aux limites transparentes et artificielles pour l'équation de Schrödinger en dimension 1 espace. *C. R. Acad. Sci. Paris, Ser. I* **320**, 89–94 (1995)
10. Candes, E., Demanet, L., Ying, L.: Fast computation of Fourier integral operators. *SIAM J. Sci. Comput.* **29**(6), 2464–2493 (2007)
11. Dahlquist, G., Eisenstat, S.C., Golub, G.H.: Bounds for the error of linear systems of equations using the theory of moments. *J. Math. Anal. Appl.* **37**, 151–166 (1972)
12. Evans, L.C.: *Partial Differential Equations*. American Mathematical Society, Providence (1998)
13. Gerschgorin, S.: Über die Abgrenzung der Eigenwerte einer Matrix. *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk* **7**, 749–754 (1931)
14. Golub, G.H.: Some modified matrix eigenvalue problems. *SIAM Rev.* **15**, 318–334 (1973)
15. Golub, G.H.: Bounds for matrix moments. *Rocky Mt. J. Math.* **4**, 207–211 (1974)
16. Golub, G.H., Meurant, G.: *Matrices, moments and quadrature*. In: Griffiths, D.F., Watson, G.A. (eds.) *Proceedings of the 15th Dundee Conference*, June–July 1993. Longman Scientific & Technical, Harlow (1994)
17. Golub, G.H., Welsch, J.: Calculation of Gauss quadrature rules. *Math. Comput.* **23**, 221–230 (1969)
18. Guidotti, P., Lambers, J.V., Sølna, K.: Analysis of 1-D wave propagation in inhomogeneous media. *Numer. Funct. Anal. Optim.* **27**, 25–55 (2006)
19. Hellums, J.R., Frensley, W.R.: Non-Markovian open-system boundary conditions for the time-dependent Schrödinger equation. *Phys. Rev. B* **49**, 2904–2906 (1994)
20. Hesthaven, J.S., Gottlieb, S., Gottlieb, D.: *Spectral Methods for Time-Dependent Problems*. Cambridge University Press, Cambridge (2007)
21. Hochbruck, M., Lubich, C.: On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.* **34**, 1911–1925 (1996)
22. Lambers, J.V.: Derivation of high-order spectral methods for time-dependent PDE using modified moments. *Electron. Trans. Numer. Anal.* **28**, 114–135 (2007)
23. Lambers, J.V.: Implicitly defined high-order operator splittings for parabolic and hyperbolic variable-coefficient PDE using modified moments. *Int. J. Comput. Sci.* **2**(3), 376–401 (2008)
24. Lambers, J.V.: *Krylov subspace methods for variable-coefficient initial-boundary value problems*. Ph.D. thesis, Stanford University, SCCM Program (2003). Available at http://sccm.stanford.edu/pub/sccm/theses/James_Lambers.pdf
25. Lambers, J.V.: Krylov subspace spectral methods for variable-coefficient initial-boundary value problems. *Electron. Trans. Numer. Anal.* **20**, 212–234 (2005)
26. Lambers, J.V.: Practical implementation of Krylov subspace spectral methods. *J. Sci. Comput.* **32**, 449–476 (2007)
27. Mallo, I.A., Reguera, N.: Weak ill-posedness of spatial discretizations of absorbing boundary conditions for Schrödinger-type equations. *SIAM J. Numer. Anal.* **40**, 134–158 (2002)

28. Mayfield, B.: Non-local boundary conditions for the Schrödinger equation. Ph.D. thesis, University of Rhode Island, Providence (1989)
29. Moore, W.J.: Schrödinger: Life and Thought. Cambridge University Press, Cambridge (1992)
30. Oskooi, A.F., Zhang, L., Avniel, Y., Johnson, S.G.: The failure of perfectly matched layers, and towards their redemption by adiabatic absorbers. *Opt. Expr.* **16**, 11376–11392 (2008)
31. Papadakis, J.S.: Impedance formulation of the bottom boundary condition for the parabolic equation model in underwater acoustics. *NORDA Parabolic Equation Workshop, NORDA Tech. Note 143* (1982)
32. Schädle, A.: Non-reflecting boundary conditions for the two dimensional Schrödinger equation. *Wave Motion* **35**, 181–188 (2002)
33. Schmidt, F., Deuffhard, P.: Discrete transparent boundary conditions for the numerical solution of Fresnel's equation. *Comput. Math. Appl.* **29**, 53–76 (1995)
34. Shampine, L.F., Reichelt, M.W.: The *MATLAB* ODE suite. *SIAM J. Sci. Comput.* **18**, 1–22 (1997)