Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2008) 6862–6876

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Iteration methods for stability spectra of solitary waves

Jianke Yang *

Department of Mathematics and Statistics, University of Vermont, Burlington, VT 05401, United States

Received 29 November 2007; received in revised form 22 March 2008; accepted 28 March 2008
Available online 11 April 2008

Abstract

Three iteration methods are proposed for the computation of eigenvalues and eigenfunctions in the linear stability of solitary waves. These methods are based on iterating certain time evolution equations associated with the linear stability eigenvalue problem. The first method uses the fourth-order Runge–Kutta method to iterate the pre-conditioned linear stability operator, and it usually converges to the most unstable eigenvalue of the solitary wave. The second method uses the Euler method to iterate the "square" of the pre-conditioned linear stability operator. This method is shown to converge to any discrete eigenvalue in the stability spectrum. The third method is obtained by incorporating the mode elimination technique into the second method, which speeds up the convergence considerably. These methods are applied to various examples of physical interest, and are found to be efficient, easy to implement, and low in memory usage.

MSC: 65N12; 35P15; 35Q55

Keywords: Solitary waves; Linear stability; Eigenvalues; Iteration methods

1. Introduction

In the study of non-linear waves, solitary waves and their linear stability properties play important roles. In integrable systems, solitary waves (i.e. solitons) admit analytical expressions, and they are stable against small perturbations. In non-integrable systems, however, solitary waves generally do not admit analytical expressions, and they can be either stable or unstable depending on the wave equations as well as solitary wave parameters. Numerical computations of solitary waves have had a long history, and a number of numerical methods have been proposed (see [1–9] for instance). On the computation of linear stability of solitary waves, however, the situation is less satisfactory. A prevalent numerical method for the stability analysis is to simulate the evolution of a perturbed solitary wave in the non-linear wave equation, or simulate the linearized equation of the non-linear wave equation around the solitary wave, using standard evolution methods such as the pseudo-spectral method, the split-step method, or the finite-difference method. This method can determine if a solitary wave is stable or not, and can even give a rough estimate of the most unstable eigenvalue. But

* Tel.: +1 802 656 4314.
E-mail address: jyang@math.uvm.edu

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.03.039

this method is quite slow for getting a good approximation for the unstable eigenvalue, especially when the instability is weak. In addition, it is incapable of obtaining less unstable eigenvalues and stable eigenvalues in the stability spectrum. Less unstable eigenvalues may become most unstable upon parameter variation, and stable eigenmodes can induce various types of long-lasting shape oscillations to the solitary wave [10], thus such eigenvalues are often desirable as well. To obtain good approximations for eigenvalues in the stability spectrum, two methods are currently available. One is the shooting method [2]. This method is highly accurate and easy to implement, but it works only for one-dimensional (1D) problems or problems which can be reduced to 1D problems [2]. The other method is to discretize the linear stability operator (either by finite-difference, spectral differentiation [11], or spectral collocation [12] methods), and turn it into a matrix eigenvalue problem. The resulting matrix is non-Hermitian in general, and is often dense as well if spectral discretization is employed. There is a rich literature on eigenvalue computations of a general square matrix. If all eigenvalues of the matrix are desired, the most successful numerical algorithm is probably the implicitly shifted QR algorithm (see [13] and references therein for the related theory and implementations). If the matrix is very large and only a small set of eigenvalues are needed, the most successful numerical algorithm is probably the implicitly restarted Arnoldi algorithm [14,15]. Both algorithms are very efficient and have been built in softwares such as MATLAB and ARPACK [16]. This matrix eigenvalue method is attractive as it can give all (or many) eigenvalues in the linear stability spectrum at once. The main restriction of this method is often the computer memory. For instance, we have found that for the 2D and 3D linear stability problems considered in this paper (see Examples 2 and 3 in Section 4), in order to get high accuracy in eigenvalues, the resulting matrix with spectral discretization was dense and very large, which often caused MATLAB to run out of memory on our PC with 4 GB RAM. This motivates us to develop alternative numerical methods for linear stability eigenvalues which make the memory usage minimal but the computing speed maximal.

In this paper, we propose three new iteration methods for the computation of eigenvalues and eigenfunctions in the linear stability of general solitary waves. These methods are based on iterating certain pre-conditioned time evolution equations associated with the linear stability eigenvalue problem. The first method uses the fourth-order Runge–Kutta method to evolve the pre-conditioned linear stability operator, and it usually converges to the most unstable eigenvalue of the solitary wave. The second method uses the Euler method to evolve the "square" of the pre-conditioned linear stability operator. This method will be shown to converge to any discrete eigenvalue in the stability spectrum. The third method is to incorporate the mode elimination technique [8,17] into the second method, which speeds up the convergence considerably. These three methods are applied to various examples of physical interest, and are found to be robust, time-efficient, highly accurate, and simple to implement (for any spatial dimension). When compared to the matrix eigenvalue method, the memory usage of these new methods is several orders of magnitude smaller, hence they can tackle higher-dimensional stability problems on any modest PC (while the matrix eigenvalue method may often have difficulties due to memory limitations). These new methods are particularly suitable for tracing eigenvalues and studying their bifurcations as solitary wave parameters (such as the propagation constant) continuously vary. In the Appendix, a sample MATLAB code is attached to demonstrate the simple implementation of these methods.

2. The original operator iteration method

Consider a general non-linear wave equation which admits solitary wave solutions. To study the linear stability of such solitary waves, the non-linear wave equation is linearized around these waves, and a linear eigenvalue problem results:

$$\mathbf{L}\mathbf{\Psi} = \lambda\mathbf{\Psi}. \tag{2.1}$$

Here $\mathbf{L}$ is the linearization operator (which is usually non-Hermitian), λ is the eigenvalue, $\mathbf{\Psi}(\mathbf{x})$ is the corresponding eigenfunction, $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ is the spatial coordinates, and N is the number of spatial dimensions. For solitary waves (which decay to zero at large distances), the continuous spectrum of $\mathbf{L}$ is often easy to determine by examining the asymptotic operator $\mathbf{L}|_{\mathbf{x}\to\infty}$. Thus, we focus on the computation of discrete eigenvalues of $\mathbf{L}$ in this paper. Eigenvalues with positive real parts are unstable eigenvalues. The other eigenvalues are stable. Purely imaginary eigenvalues are called internal modes.

In this section, we develop an iteration method to compute the eigenmodes in Eq. (2.1). The idea is to evolve the following time evolution equation

$$\mathbf{\Psi}_t = \mathbf{M}^{-1}(\mathbf{L} - \lambda)\mathbf{\Psi}, \tag{2.2}$$

where $\mathbf{M}$ is a positive definite and Hermitian operator, and λ is updated from $\mathbf{\Psi}$ at each step of the computation. Operator $\mathbf{M}$ is introduced to speed up convergence of the computation, and will be called the acceleration operator. Its role is analogous to the pre-conditioning matrix for the iterative solution of large linear systems. Without $\mathbf{M}$, the evolution operator $\mathbf{L} - \lambda$ usually contains very large eigenvalues (which are actually infinite before discretization). This severely restricts the time stepsize one can use to compute the evolution equation (2.2). The purpose of introducing $\mathbf{M}$ is to bring all eigenvalues of the new evolution operator $\mathbf{M}^{-1}(\mathbf{L} - \lambda)$ to O(1). This way, the time stepsize for computing Eq. (2.2) can be much larger, which then significantly speeds up the convergence of iterations (see [9] for details). If the solution of the evolution equation (2.2) approaches a stationary state $\mathbf{\Psi}$, i.e. $\mathbf{M}^{-1}(\mathbf{L} - \lambda)\mathbf{\Psi} = 0$, then since $\mathbf{M}$ is positive definite, we have $\mathbf{L}\mathbf{\Psi} = \lambda\mathbf{\Psi}$, thus $(\lambda, \mathbf{\Psi})$ is an eigenmode of Eq. (2.1). The simplest way to evolve Eq. (2.2) is to use the Euler method [9]. However, we found that the Euler method diverges generically for any eigenvalue of Eq. (2.1). The reason for it will be explained later in this section. To overcome this difficulty, we propose to use the fourth-order Runge–Kutta method to evolve Eq. (2.2) instead. The resulting iteration method for the eigenvalue problem (2.1) then is

$$\mathbf{\Psi}_{n+1} = \mathbf{\Psi}_n + \frac{1}{6}(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4)\Delta t, \tag{2.3}$$

where

$$\mathbf{K}_1 = \mathcal{F}(\mathbf{\Psi}_n), \tag{2.4}$$

$$\mathbf{K}_2 = \mathcal{F}(\mathbf{\Psi}_n + \frac{1}{2}\mathbf{K}_1\Delta t), \tag{2.5}$$

$$\mathbf{K}_3 = \mathcal{F}(\mathbf{\Psi}_n + \frac{1}{2}\mathbf{K}_2\Delta t), \tag{2.6}$$

$$\mathbf{K}_4 = \mathcal{F}(\mathbf{\Psi}_n + \mathbf{K}_3\Delta t), \tag{2.7}$$

function $\mathcal{F}$ is defined as

$$\mathcal{F}(\mathbf{\Psi}) \equiv \mathbf{M}^{-1}(\mathbf{L} - \lambda)\mathbf{\Psi}, \tag{2.8}$$

λ is computed from $\mathbf{\Psi}$ by

$$\lambda = \frac{\langle \mathbf{M}^{-1}\mathbf{\Psi}, \mathbf{L}\mathbf{\Psi} \rangle}{\langle \mathbf{M}^{-1}\mathbf{\Psi}, \mathbf{\Psi} \rangle}, \tag{2.9}$$

the inner product is the standard one in the square-integrable functional space, i.e.

$$\langle \mathbf{G}_1, \mathbf{G}_2 \rangle = \int_{-\infty}^{\infty} \mathbf{G}_1^{\dagger} \cdot \mathbf{G}_2 \, \mathrm{d}\mathbf{x}, \tag{2.10}$$

$(\cdot)^{\dagger}$ denotes the Hermitian of the underlying quantity, and Δt (> 0) is an auxiliary "stepsize" parameter. Since the evolution equation (2.2) is based on the original eigenvalue Eq. (2.1), we call this method the original operator iteration method (or OOM for brevity).

To analyze the convergence properties of this OOM, we write

$$\mathbf{\Psi}_n = \mathbf{\Psi} + \widetilde{\mathbf{\Psi}}_n, \quad \widetilde{\mathbf{\Psi}}_n \ll 1, \tag{2.11}$$

where $\widetilde{\mathbf{\Psi}}_n$ is the error at the nth iteration. Substituting this formula into (2.9), we find that

$$\lambda(\mathbf{\Psi}_n) = \frac{\langle \mathbf{M}^{-1}\mathbf{\Psi}_n, \mathbf{L}\mathbf{\Psi}_n \rangle}{\langle \mathbf{M}^{-1}\mathbf{\Psi}_n, \mathbf{\Psi}_n \rangle} = \lambda + \frac{\langle \mathbf{M}^{-1}\mathbf{\Psi}, \mathbf{L}_1\widetilde{\mathbf{\Psi}}_n \rangle}{\langle \mathbf{M}^{-1}\mathbf{\Psi}, \mathbf{\Psi} \rangle} + \mathrm{O}(\widetilde{\mathbf{\Psi}}_n^2), \tag{2.12}$$

where $\mathbf{L}_1 \equiv \mathbf{L} - \lambda$. Thus, from (2.4) and (2.8), we see that

$$\mathbf{K}_1 = \mathbf{M}^{-1}[\mathbf{L} - \lambda(\mathbf{\Psi}_n)]\mathbf{\Psi}_n = \mathcal{L}_0\widetilde{\mathbf{\Psi}}_n + \mathrm{O}(\widetilde{\mathbf{\Psi}}_n^2), \tag{2.13}$$

where operator $\mathcal{L}_0$ is defined as

$$\mathcal{L}_0\mathbf{G} = \mathbf{M}^{-1}\left(\mathbf{L}_1\mathbf{G} - \frac{\langle \mathbf{M}^{-1}\mathbf{\Psi}, \mathbf{L}_1\mathbf{G}\rangle}{\langle \mathbf{M}^{-1}\mathbf{\Psi}, \mathbf{\Psi}\rangle}\mathbf{\Psi}\right). \tag{2.14}$$

Similar calculations show that

$$\mathbf{K}_2 = \mathcal{L}_0\left[\widetilde{\mathbf{\Psi}}_n + \frac{1}{2}\Delta t\mathcal{L}_0\widetilde{\mathbf{\Psi}}_n\right] + \mathrm{O}(\widetilde{\mathbf{\Psi}}_n^2), \tag{2.15}$$

and so on. When these formulas are inserted into (2.3) and quadratic as well as higher-order terms in $\widetilde{\mathbf{\Psi}}_n$ neglected, we find that the error satisfies the following linear iteration equation

$$\widetilde{\mathbf{\Psi}}_{n+1} = \mathcal{L}_{\mathrm{OOM}}\widetilde{\mathbf{\Psi}}_n, \tag{2.16}$$

where the iteration operator $\mathcal{L}_{\mathrm{OOM}}$ is

$$\mathcal{L}_{\mathrm{OOM}} = 1 + \mathcal{L}_0\Delta t + \frac{1}{2!}(\mathcal{L}_0\Delta t)^2 + \frac{1}{3!}(\mathcal{L}_0\Delta t)^3 + \frac{1}{4!}(\mathcal{L}_0\Delta t)^4. \tag{2.17}$$

Notice that $\mathbf{L}_1\mathbf{\Psi} = \mathcal{L}_0\mathbf{\Psi} = 0$, thus zero is always a discrete eigenvalue of $\mathcal{L}_0$. Assuming that the eigenvalue λ of $\mathbf{L}$ is simple (which is the generic case), then the kernel of $\mathcal{L}_0$ only contains the eigenfunction $\mathbf{\Psi}$, which does not affect the convergence of OOM iterations.[1] Thus, OOM would converge if and only if for all non-zero eigenvalues λ_0 of $\mathcal{L}_0$, $\lambda_0\Delta t$ lies inside the stability region $\mathcal{D}_{\mathrm{RK4}}$ of the fourth-order Runge–Kutta method

$$\mathcal{D}_{\mathrm{RK4}} = \left\{z : \left|1 + z + \frac{1}{2!}z^2 + \frac{1}{3!}z^3 + \frac{1}{4!}z^4\right| < 1\right\}, \tag{2.18}$$

i.e.

$$\lambda_0 \in \frac{1}{\Delta t}\mathcal{D}_{\mathrm{RK4}} \quad \text{for any non-zero } \lambda_0 \text{ of } \mathcal{L}_0. \tag{2.19}$$

Below, we demonstrate the convergence of this OOM (2.3)–(2.9) by an example.

Example 1. Consider the following 1D cubic-quintic non-linear Schrödinger (NLS) equation

$$\mathrm{i}U_t + U_{xx} - |U|^2U + |U|^4U = 0. \tag{2.20}$$

Here the cubic non-linearity is of self-defocusing type, and the quintic non-linearity is of self-focusing type. This equation admits the following solitary waves [18]

$$U(x,t) = u(x)\mathrm{e}^{\mathrm{i}\mu t}, \tag{2.21}$$

where

$$u(x) = \left(\frac{4B\mu}{\cosh 2\sqrt{\mu}x - B}\right)^{\frac{1}{2}}, \quad B = \left(1 + \frac{16}{3}\mu\right)^{-\frac{1}{2}}, \tag{2.22}$$

and $\mu > 0$ is a propagation constant. To study the linear stability of these solitary waves, we perturb them as

$$U(x,t) = \mathrm{e}^{\mathrm{i}\mu t}\{u(x) + [v(x) - w(x)]\mathrm{e}^{\lambda t} + [v(x) + w(x)]^*\,\mathrm{e}^{\lambda^* t}\}, \tag{2.23}$$

where the superscript "*" represents complex conjugation, and $v,\ w \ll 1$. Substituting this perturbation into Eq. (2.20) and linearizing, the resulting linear stability eigenvalue problem is Eq. (2.1) with

$$\mathbf{L} = -\mathrm{i}\begin{pmatrix} 0 & \partial_{xx} - \mu - u^2 + u^4 \\ \partial_{xx} - \mu - 3u^2 + 5u^4 & 0 \end{pmatrix}, \tag{2.24}$$

[1] In the non-generic case where the eigenvalue λ of $\mathbf{L}$ is not simple, the generalized eigenfunction of λ may exist in the kernels of $\mathcal{L}_0$ and $\mathcal{L}_{\mathrm{OOM}}$. In such a case, the OOM will not converge to the eigenfunction of λ. A similar phenomenon occurs for the SOM in Section 3 (see Remark 1 there).

and $\mathbf{\Psi}$ is the transpose of (v, w). Below, we take $\mu = 1$ in the solitary wave (2.22). For this wave, the slope of the power curve $P(\mu) = \int u^2\,dx$ is negative, thus $\mathbf{L}$ has a real unstable eigenvalue [18,19], which is found to be $\lambda_u = 2.1612$. Zero and $-\lambda_u$ are stable discrete eigenvalues of $\mathbf{L}$. The continuous spectrum of $\mathbf{L}$ is $i(-\infty, -\mu] \cup i[\mu, \infty)$ on the imaginary axis. This spectrum structure is shown in Fig. 1a. To compute the unstable eigenvalue λ_u by the OOM, we take the acceleration operator $\mathbf{M}$ as

$$\mathbf{M} = (c - \partial_{xx})\mathrm{diag}(1, 1), \tag{2.25}$$

with $c = 2.4$. Then the spectrum of $\mathcal{L}_0$, computed by the matrix eigenvalue method, is shown in Fig. 1b. The continuous spectrum of $\mathcal{L}_0$ consists of the two straight-line segments between points $\pm i$ and $(-\lambda_u \pm i\mu)/c$. The discrete spectrum of $\mathcal{L}_0$ contains zero and a number of eigenvalues on the left half of the complex plane. It is easy to check that for any $0 < \Delta t < 2\sqrt{2}$, all non-zero eigenvalues of $\mathcal{L}_0$ lie inside the region $\mathcal{D}_{\mathrm{RK4}}/\Delta t$. To illustrate, the region $\mathcal{D}_{\mathrm{RK4}}/\Delta t$ with $\Delta t = 2.3$ is also displayed in Fig. 1b. Clearly, the non-zero spectrum of $\mathcal{L}_0$ lies entirely inside this region, thus OOM iterations converge to the eigenvalue λ_u. To numerically compute this eigenvalue, we take the x interval as $[-10, 10]$, discretized by 128 grid points. The spatial derivatives in the OOM as well as $\mathbf{M}^{-1}$ are computed using the pseudo-spectral method (i.e. discrete Fourier transform). Starting from the initial condition

$$\mathbf{\Psi}_0 = \begin{bmatrix} v_0 \\ w_0 \end{bmatrix} = \begin{bmatrix} -0.3(1 - 2x^2)e^{-x^2} \\ i\,e^{-x^2} \end{bmatrix} \tag{2.26}$$

and with $\Delta t = 2.3$, we find that OOM iterations quickly converge to the eigenmode of λ_u. The numerically obtained eigenfunction is displayed in Fig. 1c. Here $v(x)$ is purely real, and $w(x)$ purely imaginary. The error of the eigenvalue versus the number of iterations is shown in Fig. 1d. Here the error is defined as $|\lambda_n - \lambda_{n-1}|$, which is the difference between successive eigenvalues during iterations. We see that this error drops below 10^{-10} in only 40 iterations, which is very fast.

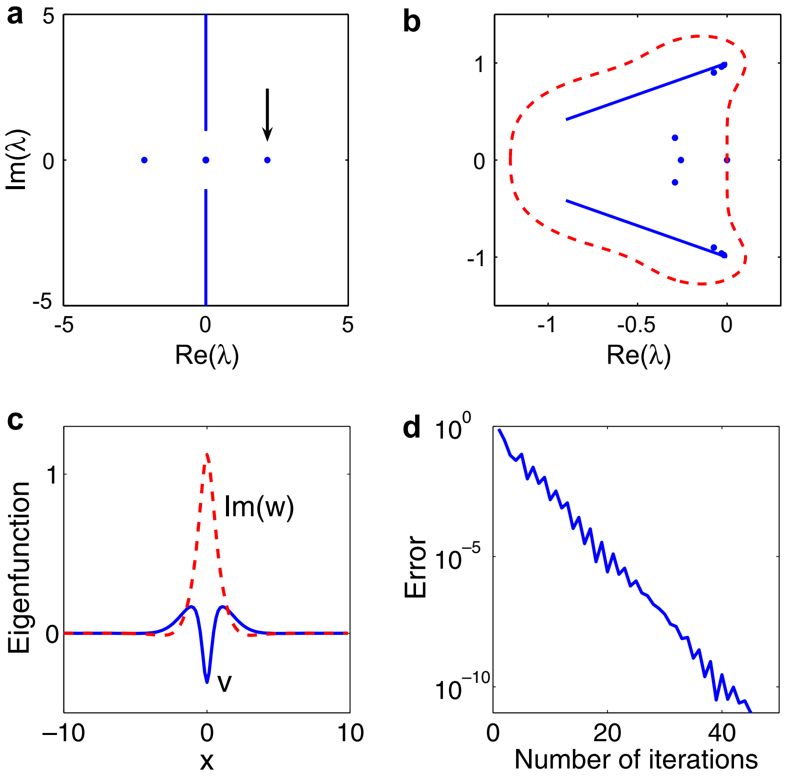

Fig. 1. (a) Spectrum of the linearization operator $\mathbf{L}$ for the solitary wave (2.22) in Eq. (2.20) with $\mu = 1$; (b) spectrum of operator $\mathcal{L}_0$ for the unstable eigenvalue λ_u marked by an arrow in (a) (with $c = 2.4$); the dashed line is the boundary of region $\mathcal{D}_{\mathrm{RK4}}/\Delta t$ with $\Delta t = 2.3$; (c) the unstable eigenfunction (v, w) of λ_u; (d) error of the eigenvalue versus the number of OOM iterations for this λ_u.

To test the sensitivity of the OOM to scheme parameters c and Δt, we have tried a wide range of these values in the above example. We find that the OOM converges to this unstable eigenvalue for any $c > 0$ when Δt is below a certain c-dependent threshold. In other words, the OOM is insensitive to the choices of c and Δt values. We also find that the c and Δt values used above (i.e. $c = 2.4,\ \Delta t = 2.3$) give near-optimal convergence rate. If other non-optimal $(c, \Delta t)$ values are used, as long as they are not far from the optimal values, then the OOM will converge at a rate comparable to the optimal one. For instance, if we take $c = 2$ and $\Delta t = 2.2$, then the error would drop below 10^{-10} in 41 iterations; and if we take $c = 2$ and $\Delta t = 2$, then the error would drop below 10^{-10} in 66 iterations. This behavior can be theoretically explained, see Fig. 1 in [8] or Fig. 3 in [9] for details. We have also tested the sensitivity of the OOM to the initial condition $\mathbf{\Psi}_0$. We find that if $\mathbf{\Psi}_0$ is reasonably close to the exact eigenfunction $\mathbf{\Psi}$, then the OOM will converge to $\mathbf{\Psi}$. In the Appendix, a sample MATLAB code for this example is included. This code shows that the OOM algorithm is quite simple to implement. The reader can run this code and change its parameters $(c, \Delta t)$ or initial conditions to verify the fast speed and robustness of this algorithm.

In addition to the robustness and simple implementation, an important advantage of the OOM is its easy extension from lower to higher spatial dimensions, and this extension does not prolong the length of the MATLAB code. For instance, to extend the MATLAB code in the Appendix from one dimension to higher dimensions, the main change is just to replace the discrete Fourier transform commands "fft" and "ifft" in the code by their higher-dimensional counterparts "fftn" and "ifftn". This contrasts the matrix eigenvalue method where one discretizes the stability operator $\mathbf{L}$ and turns Eq. (2.1) into a matrix eigenvalue problem. In this matrix method, the buildup of the matrix differs significantly from a lower dimension to a higher one (especially with spectral discretizations [11,12]). In addition, this matrix-buildup can be a time-consuming process by itself in higher dimensions (one may take the spectral collocation method as an example [12]). Thus, the easy implementation of the OOM in high spatial dimensions is a valuable property for applications. Another advantage of the OOM is its high accuracy. Since this method is compatible with the pseudo-spectral method for the computation of spatial derivatives (see Example 1), the accuracy of the OOM is spectral.

The most important advantage of the OOM over the matrix eigenvalue method is probably its low use of computer memory. The OOM only iterates the eigenfunction on the spatial grid. Thus, if N grid points are used, the memory usage of the OOM would be only $\mathrm{O}(N)$. In the matrix eigenvalue method, however, one needs to build up a $N \times N$ matrix. If spectral discretization [11,12] is used to get the same spectral accuracy of the OOM, the resulting matrix would be dense, hence the memory usage would be $\mathrm{O}(N^2)$.[2] For one-dimensional problems such as Example 1, this memory issue is often insignificant since the memory usage is generally quite small. For instance, for $N = 128$ as used in this example, the memory usage of the matrix eigenvalue method with spectral discretization would be only $\mathrm{O}(N^2) \sim \mathrm{O}(10^4)$. However, this memory issue would become critical in higher spatial dimensions. For instance, in Example 2 of Section 4 where the problem is three-dimensional, the number of grid points we take is $N = 48 \times 48 \times 48 \sim 10^5$. In this case, the matrix eigenvalue method with spectral discretization would result in a dense $10^5 \times 10^5$ matrix, which cannot be handled by most PCs currently available. For such problems, the OOM has a distinctive advantage. We will comment on this issue again in Example 3 of Section 4.

The spectrum of operator $\mathcal{L}_0$ in Fig. 1b also explains why when the Euler method is used to evolve Eq. (2.2), iterations always diverge. With the Euler method, the convergence condition is that all non-zero eigenvalues λ_0 of $\mathcal{L}_0$ lie inside the region $\mathcal{D}_{\text{Euler}}/\Delta t$, where $\mathcal{D}_{\text{Euler}} = \{z : |1 + z| < 1\}$ is the stability region of the Euler method. Obviously, for the spectrum of $\mathcal{L}_0$ in Fig. 1b, this condition is violated. Indeed, the continuous eigenvalues $\pm\mathrm{i}$ in the spectrum of $\mathcal{L}_0$ can never lie inside $\mathcal{D}_{\text{Euler}}/\Delta t$ for any Δt. Thus, Euler iterations always diverge. The advantage of the fourth-order Runge–Kutta method is that it has a larger stability region $\mathcal{D}_{\text{RK4}}$, and this region contains a good portion of the imaginary axis. This makes it possible for the convergence condition (2.19) to be met for the spectrum of $\mathcal{L}_0$ in Fig. 1b.

In Example 1, when OOM is used to compute the stable eigenvalue $-\lambda_{\mathrm{u}}$ of $\mathbf{L}$, it is found to diverge. This phenomenon is quite general. Our extensive testing of the OOM on various equations shows that the OOM

[2] If the finite-difference discretization is used, even though the resulting matrix is often sparse, but to reach the same accuracy, the number of grid points must be much larger, which often demands more memory to store the matrix, not less.

often converges to the most unstable eigenvalue of $\mathbf{L}$ (see additional examples in Section 4), but often diverges for $\mathbf{L}$'s less unstable eigenvalues and stable eigenvalues. This can be heuristically understood as follows. For the most unstable eigenvalue λ of $\mathbf{L}$, all eigenvalues of $\mathbf{L}_1$ lie on the left half of the complex plane. This tends to make the eigenvalues of $\mathcal{L}_0$ to lie on the left half of the complex plane as well when $\mathbf{M}$ is positive definite and Hermitian. Hence with Δt below a certain threshold value, condition (2.19) can be met, resulting in convergence of the OOM. For less unstable and stable eigenvalues λ, however, $\mathbf{L}_1$ has eigenvalues lying on the right half of the complex plane. This often makes $\mathcal{L}_0$ to have eigenvalues on the right half of the complex plane as well. In such a case, condition (2.19) is often violated, leading to divergence of the OOM.

Since the OOM usually diverges for less unstable and stable eigenvalues of $\mathbf{L}$, and may diverge for the most unstable eigenvalue of $\mathbf{L}$ as well in less common occasions, this motivates us to propose additional iteration methods which can converge to any discrete eigenvalue in the stability spectrum of $\mathbf{L}$. This will be done in the next section.

3. The squared-operator and modified squared-operator iteration methods

In this section, we propose two additional iteration methods for the eigenvalue problem (2.1) which can converge to any discrete eigenvalue in the stability spectrum. The ideas behind these methods are based on iterating a "squared-operator" equation rather than (2.2), similar to those developed in Ref. [8] for solitary wave computations. However, important differences exist between the proposed schemes here and those in Ref. [8] due to the linear nature of the eigenvalue problem (2.1), as we will see below.

To develop these iteration methods, we consider the new evolution equation

$$\mathbf{\Psi}_t = -\mathbf{M}^{-1}\mathbf{L}_1^{\dagger}\mathbf{M}^{-1}\mathbf{L}_1\mathbf{\Psi}, \tag{3.1}$$

where $\mathbf{L}_1$ is defined below Eq. (2.12), λ is updated from $\mathbf{\Psi}$ by formula (2.9), and $\mathbf{M}$ is an acceleration operator. The above evolution operator is the "square" of the original evolution operator in Eq. (2.2). Unlike the original operator, all eigenvalues of this squared-operator are real and non-positive (see below), hence the Euler method can be used to iterate Eq. (3.1) without causing divergence. The resulting method, called the squared-operator iteration method (SOM), for the eigenvalue problem (2.1) is

$$\mathbf{\Psi}_{n+1} = \mathbf{\Psi}_n - \mathbf{M}^{-1}\mathbf{L}_1^{\dagger}\mathbf{M}^{-1}\mathbf{L}_1\mathbf{\Psi}|_{\mathbf{\Psi}=\mathbf{\Psi}_n,\,\lambda=\lambda_n}\Delta t. \tag{3.2}$$

Here λ_n is given by formula (2.9) with $\mathbf{\Psi}$ replaced by $\mathbf{\Psi}_n$. This method resembles a similar one in Ref. [8] for solitary wave computations, except that λ here is unknown and needs to be updated by formula (2.9), while it was known and fixed in Ref. [8].

Unlike the OOM in the previous section, the above SOM can converge to any discrete eigenvalue of operator $\mathbf{L}$ under mild conditions. This result is summarized in the following theorem.

Theorem 1. *Let λ be any non-embedded discrete eigenvalue of operator $\mathbf{L}$. If λ is simple, or is multi-fold but with equal geometric and algebraic multiplicities, then the SOM (3.2) converges to this eigenvalue if $\Delta t < \Delta t_{\max}$, and the initial condition $\mathbf{\Psi}_0(\mathbf{x})$ is sufficiently close to the exact eigenfunction $\mathbf{\Psi}(\mathbf{x})$. Here $\Delta t_{\max} = -2/\Lambda_{\min}$, where $\Lambda_{\min}$ is the minimum eigenvalue of operator $\mathcal{L}_{\mathrm{SOM}}$ defined in Eq. (3.4). If $\Delta t > \Delta t_{\max}$, this SOM diverges.*

Proof. We use the linearization technique to prove this theorem. Substituting (2.11) into (3.2), utilizing (2.12), and neglecting quadratic and higher-order terms in $\widetilde{\mathbf{\Psi}}_n$, we find that the error $\widetilde{\mathbf{\Psi}}_n$ satisfies the following linear iteration equation

$$\widetilde{\mathbf{\Psi}}_{n+1} = (1 + \mathcal{L}_{\mathrm{SOM}}\Delta t)\widetilde{\mathbf{\Psi}}_n, \tag{3.3}$$

where the iteration operator $\mathcal{L}_{\mathrm{SOM}}$ is

$$\mathcal{L}_{\mathrm{SOM}}\mathbf{G} = -\mathbf{M}^{-1}\mathbf{L}_1^{\dagger}\mathbf{M}^{-1}\left(\mathbf{L}_1\mathbf{G} - \frac{\langle \mathbf{M}^{-1}\mathbf{\Psi}, \mathbf{L}_1\mathbf{G}\rangle}{\langle \mathbf{M}^{-1}\mathbf{\Psi}, \mathbf{\Psi}\rangle}\mathbf{\Psi}\right). \tag{3.4}$$

Since $\mathbf{M}$ is Hermitian and positive definite, we can rewrite $\mathcal{L}_{\mathrm{SOM}}$ as

$$\mathcal{L}_{\mathrm{SOM}} = \mathbf{M}^{-1/2}\mathcal{L}_{\mathrm{h}}\mathbf{M}^{1/2}, \tag{3.5}$$

where

$$\mathcal{L}_\mathrm{h}\mathbf{G} = -\widehat{\mathcal{L}}_\mathrm{h}^\dagger\left(\widehat{\mathcal{L}}_\mathrm{h}\mathbf{G} - \frac{\langle\widehat{\mathbf{\Psi}}, \widehat{\mathcal{L}}_\mathrm{h}\mathbf{G}\rangle}{\langle\widehat{\mathbf{\Psi}}, \widehat{\mathbf{\Psi}}\rangle}\widehat{\mathbf{\Psi}}\right), \tag{3.6}$$

$$\widehat{\mathcal{L}}_\mathrm{h} = \mathbf{M}^{-1/2}\mathbf{L}_1\mathbf{M}^{-1/2}, \quad \widehat{\mathbf{\Psi}} = \mathbf{M}^{-1/2}\mathbf{\Psi}. \tag{3.7}$$

It is easy to check that $\mathcal{L}_\mathrm{h}$ is Hermitian and semi-negative definite. Since the iteration operator $\mathcal{L}_\mathrm{SOM}$ is similar to $\mathcal{L}_\mathrm{h}$ [see (3.5)], we find that the eigenvalues Λ of $\mathcal{L}_\mathrm{SOM}$ are all real and non-positive, and its eigenfunctions form a complete set in the square-integrable functional space. Then when the error $\widetilde{\mathbf{\Psi}}_n$ is expanded into this complete set of eigenfunctions, we see from Eq. (3.3) that each Λ-eigenmode component in the error $\widetilde{\mathbf{\Psi}}_n$ is proportional to $(1+\Lambda\Delta t)^n$. If $\Delta t > \Delta t_\mathrm{max}$, where Δt_max is given in the theorem, $1+\Lambda_\mathrm{min}\Delta t < -1$, thus iterations diverge. If $\Delta t < \Delta t_\mathrm{max}$, all eigenmode components of $\mathcal{L}_\mathrm{SOM}$ with non-zero eigenvalues Λ will decay. Regarding eigenfunctions $\mathbf{G}$ of $\mathcal{L}_\mathrm{SOM}$ with zero eigenvalues, they satisfy the equation

$$\mathcal{L}_\mathrm{h}\mathbf{M}^{1/2}\mathbf{G} = 0. \tag{3.8}$$

Taking the inner product of this equation with $\mathbf{M}^{1/2}\mathbf{G}$ and using the Cauchy–Schwartz inequality, we get

$$\mathbf{L}_1\mathbf{G} = \beta\mathbf{\Psi}, \tag{3.9}$$

where β is a constant. But due to our assumption, λ is either simple, or multi-fold with equal geometric and algebraic multiplicities, thus $\mathbf{L}$ has no generalized eigenfunctions at eigenvalue λ. Consequently, Eq. (3.9) can only hold when $\beta = 0$. In other words, $\mathbf{G} = \mathbf{\Psi}$. This eigenfunction of $\mathcal{L}_\mathrm{SOM}$ does not affect the convergence of iterations. Thus Theorem 1 is proved. □

Remark 1. If eigenvalue λ in Eq. (2.1) is multi-fold with *unequal* geometric and algebraic multiplicities, the SOM (3.2) will not converge. The reason is that in this case, there exist generalized eigenfunctions $\mathbf{G}$ of operator $\mathbf{L}$ at eigenvalue λ, which satisfy Eq. (3.9) with $\beta \neq 0$. These generalized eigenfunctions are also in the kernel of the iteration operator $\mathcal{L}_\mathrm{SOM}$, thus their components in the error $\widetilde{\mathbf{\Psi}}_n$ will not decay with iterations. As a result, iterations will not converge to the eigenfunction $\mathbf{\Psi}$.

The SOM (3.2), even though always convergent, can be slow sometimes (see examples in Section 4). To improve its convergence speed, we apply a mode elimination technique to it [8,17]. The idea of this technique is to construct a modified iteration method so that the slowest-decaying eigenmode in the original method is eliminated, hence convergence is improved. Applying that technique to the SOM (3.2), we obtain the following modified squared-operator method (MSOM) for Eq. (2.1):

$$\mathbf{\Psi}_{n+1} = \mathbf{\Psi}_n - \left\{\mathbf{M}^{-1}\mathbf{L}_1^\dagger\mathbf{M}^{-1}\mathbf{L}_1\mathbf{\Psi} - \alpha_n\langle\mathbf{G}_n, \mathbf{L}_1^\dagger\mathbf{M}^{-1}\mathbf{L}_1\mathbf{\Psi}\rangle\mathbf{G}_n\right\}_{\mathbf{\Psi}=\mathbf{\Psi}_n,\,\lambda=\lambda_n}\Delta t, \tag{3.10}$$

where λ_n is given by formula (2.9) with $\mathbf{\Psi}$ replaced by $\mathbf{\Psi}_n$,

$$\mathbf{G}_n = \mathbf{\Psi}_n - \mathbf{\Psi}_{n-1} \tag{3.11}$$

is the difference between successive iterations, and

$$\alpha_n = \frac{1}{\langle\mathbf{M}\mathbf{G}_n, \mathbf{G}_n\rangle} - \frac{1}{\langle\mathbf{L}_1\mathbf{G}_n, \mathbf{M}^{-1}\mathbf{L}_1\mathbf{G}_n\rangle\Delta t}. \tag{3.12}$$

This MSOM resembles a similar method in Ref. [8] for solitary wave computations, except that λ here needs to be updated by formula (2.9) during each iteration. This MSOM delivers faster convergence than the SOM (3.2), as we will show in Section 4 below.

Regarding the implementation of this MSOM, we note that in the first iteration, since $\mathbf{G}_0$ is not yet available, we take $\alpha_0 = 0$.

The SOM and MSOM presented above share most of the advantages of the OOM. For instance, they are also highly accurate, simple to implement in any spatial dimension, and their memory usage is very low. In the next section, we will apply these three proposed methods to two examples of physical interest, and compare their performances.

4. Numerical examples

In this section, we use the above iteration methods to compute stability eigenmodes of solitary waves in two physical examples. These examples are in two or three spatial dimensions, hence their eigenmodes are difficult to compute by the old methods. However, we will show that these new methods can handle them easily and efficiently.

Example 2. Consider the 3D NLS equation with a harmonic potential,

$$\mathrm{i}U_t + \Delta U + |U|^2 U - V(x,y,z)U = 0, \tag{4.1}$$

where $\Delta = \partial_{xx} + \partial_{yy} + \partial_{zz}$, and $V(x,y,z) = x^2 + y^2 + 0.1z^2$ is a potential term. This equation governs the mean-field wave function of Bose–Einstein condensates in a magnetic potential and is called the Gross–Pitaevskii equation [20]. The harmonic potential here has a cigar-shape: it stretches longer along the z direction than along the transverse (x, y) directions. This type of potential is common in experiments on Bose–Einstein condensates. Eq. (4.1) admits a family of fundamental solitons

$$U(x,y,z,t) = u(x,y,z)\mathrm{e}^{\mathrm{i}\mu t}, \tag{4.2}$$

where $u(x,y,z) > 0$. The soliton at $\mu = -0.5$ is displayed in Fig. 2a (it was obtained by the squared-operator methods developed in [8]). Plotted here is the half-amplitude surface of the soliton, i.e. the surface of $\{(x,y,z) : u(x,y,z) = \frac{1}{2}u(0,0,0)\}$, where $u(0,0,0)$ is the amplitude of the soliton, which is 3.2712 here. This soliton has an almost circular shape inside the cigar-shape potential.

Now we consider the stability spectrum of this soliton. Perturbing this soliton in a form analogous to Eq. (2.23), we find that the eigenvalue problem for this soliton is Eq. (2.1) with

$$\mathbf{L} = -\mathrm{i}\begin{pmatrix} 0 & \Delta - \mu + u^2 - V \\ \Delta - \mu + 3u^2 - V & 0 \end{pmatrix}. \tag{4.3}$$

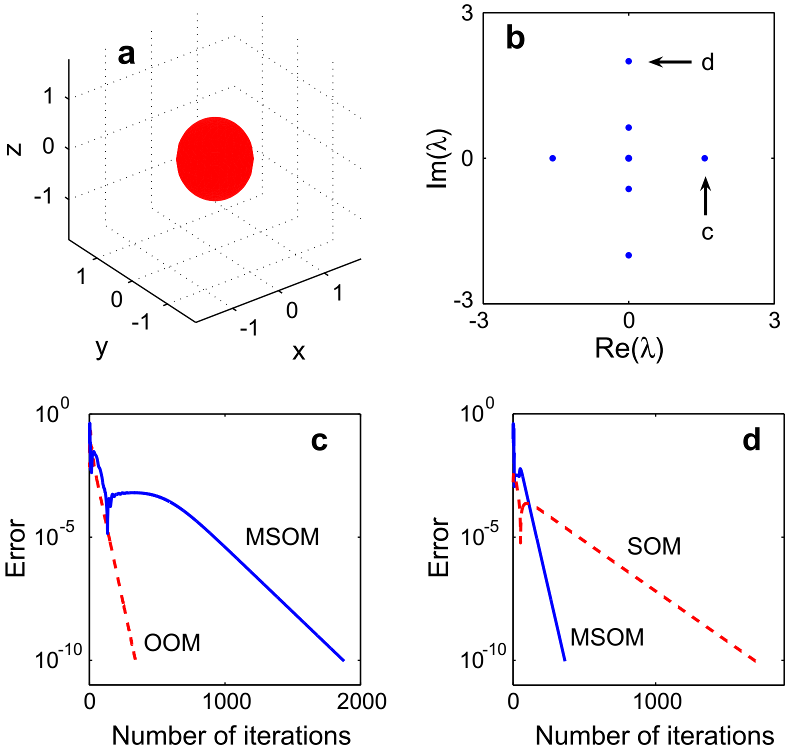

Fig. 2. (a) A 3D solitary wave in Eq. (4.1) with $\mu = -0.5$; displayed is its half-amplitude surface; (b) some eigenvalues in the stability spectrum of (a); (c) error diagrams of the OOM and MSOM for the unstable eigenvalue $\lambda_u = 1.5670$ marked in (b); (d) error diagrams of the SOM and MSOM for the internal mode $\lambda_s = 2.0000\mathrm{i}$ marked in (b).

For this soliton, $P'(\mu) < 0$, thus it has one real unstable eigenvalue [18,19], which is found to be $\lambda_u = 1.5670$. It also has a zero eigenvalue, a stable eigenvalue $-\lambda_u$, and an infinite number of internal modes such as $\lambda_s = 2.0000\mathrm{i}$. These eigenvalues are displayed in Fig. 2b.

Next we demonstrate the computation of this unstable eigenvalue λ_u and internal mode λ_s using the newly proposed methods. We take the computational domain as $-4 \leqslant x, y \leqslant 4$, and $-5 \leqslant z \leqslant 5$, discretized by 48 grid points along each direction. The acceleration operator $\mathbf{M}$ is taken as

$$\mathbf{M} = (c - \Delta)\mathrm{diag}(1, 1), \quad c > 0. \tag{4.4}$$

The spatial derivatives as well as $\mathbf{M}^{-1}$ are computed by the discrete Fourier transform. For λ_u, the initial condition is taken as

$$\mathbf{\Psi}_0 = \begin{bmatrix} -0.6(1 - 2z^2)\mathrm{e}^{-2(x^2+y^2+z^2)} \\ \mathrm{i}\,\mathrm{e}^{-2(x^2+y^2+z^2)} \end{bmatrix}. \tag{4.5}$$

For this unstable eigenvalue, all the three proposed methods can converge, and performances of the OOM and MSOM are shown below (SOM is slower than these two methods). In the OOM, we take $(c, \Delta t) = (12, 1.35)$, while in the MSOM, we take $(c, \Delta t) = (10, 0.66)$. These parameter choices are near-optimal for the underlying method. At these scheme parameters, the error diagrams of iterations are plotted in Fig. 2c. We see that the error $|\lambda_n - \lambda_{n-1}|$ drops below 10^{-10} in about 350 and 1900 iterations for the OOM and MSOM, respectively (the computing times are 14 and 50 minutes in MATLAB on our PC). Thus, the OOM converges faster than the MSOM for this computation. For the internal mode $\lambda_s = 2.0000\mathrm{i}$, the OOM does not converge, while the SOM and MSOM do. For this eigenvalue, we take the initial condition as

$$\mathbf{\Psi}_0 = \begin{bmatrix} x & \mathrm{e}^{-2(x^2+y^2+z^2)} \\ 0.5x & \mathrm{e}^{-2(x^2+y^2+z^2)} \end{bmatrix}. \tag{4.6}$$

With $(c, \Delta t) = (10, 0.5)$ for MSOM and $(12, 0.4)$ for SOM, the error diagrams of iterations for both methods are plotted in Fig. 2d. It is seen that the errors of the MSOM and SOM drop below 10^{-10} in about 360 and 1750 iterations, or 9 and 31 minutes in MATLAB on our PC. Thus, the MSOM converges faster than the SOM.

It should be pointed out that for the above example, the matrix eigenvalue method would have severe difficulties, see discussions in Section 2. Comparatively, the OOM, SOM and MSOM can compute the eigenvalues in this example quite easily and efficiently.

Example 3. The next example is the 2D NLS equation with a periodic potential

$$\mathrm{i}U_t + \Delta U + |U|^2 U - V(x, y)U = 0, \tag{4.7}$$

where $\Delta = \partial_{xx} + \partial_{yy}$, and $V(x, y) = 1.5(\cos^2 x + \cos^2 y)$. This equation arises in Bose–Einstein condensates trapped in optical lattices and light propagation in periodic non-linear media [21–23]. This equation admits vortex solitons of the form $U(x, y, t) = u(x, y)\mathrm{e}^{-\mathrm{i}\mu t}$, where $u(x, y)$ is a complex function with a unit topological charge [21]. The intensity distribution of the vortex solution at $\mu = 0$ is shown in Fig. 3a. Perturbing this soliton in a form analogous to Eq. (2.23), the stability eigenvalue problem for this soliton is found to be Eq. (2.1) with

$$\mathbf{L} = -\mathrm{i}\begin{pmatrix} \frac{1}{2}(u^{*2} - u^2) & \Delta + \mu - V + 2|u|^2 - \frac{1}{2}(u^2 + u^{*2}) \\ \Delta + \mu - V + 2|u|^2 + \frac{1}{2}(u^2 + u^{*2}) & \frac{1}{2}(u^2 - u^{*2}) \end{pmatrix}. \tag{4.8}$$

The spectrum of this operator is displayed in Fig. 3b. This spectrum was obtained by the matrix eigenvalue method with spectral collocation discretization [12]. Specifically, we expanded the eigenfunction $\mathbf{\Psi}$ into Fourier series and turned Eq. (2.1) into a matrix eigenvalue problem for the Fourier coefficients of the eigenfunction. The eigenvalues of the resulting matrix were then computed in MATLAB by the implicitly restarted Arnoldi algorithm. However, to get high accuracy eigenvalues by this matrix method is very difficult due to the large size of the resulting matrix. For example, when we expanded the eigenfunction $\mathbf{\Psi}$ into 64 Fourier modes along each dimension, the resulting matrix was dense with over 8000 × 8000 complex elements, which caused MATLAB to run out of memory on our PC with 4 GB RAM. Thus what we did was to first get a low-accuracy spectrum by this matrix method. Specifically, we took the computational domain as $-3\pi \leqslant x, y \leqslant 3\pi$, and expanded the eigenfunction $\mathbf{\Psi}$ into 41 Fourier modes along each dimension. The resulting matrix was about 3400 × 3400 and dense. Eigenvalues of this matrix were then computed in MATLAB by the Arnoldi algorithm, which took about 3.5 minutes on our PC (including the matrix-buildup time). The error of computed eigenvalues was about 10^{-2}. The discrete spectrum thus obtained contains two quadruples of complex eigenvalues, a pair of internal modes, and the zero eigenvalue. Next we refined each non-zero discrete eigenvalue to high accuracy by the SOM or MSOM. The results are displayed in Fig. 3b. The most unstable eigenvalues are found to be $\lambda_u = 0.4961 + 1.1161i$ and its complex conjugate λ_u^*, and the internal modes are $\lambda_s = 0.5854i$ and $-\lambda_s$.

To numerically compute λ_u and λ_s by the newly proposed methods, we take the computational domain as $-3\pi \leqslant x, y \leqslant 3\pi$, discretized by 64 grid points along each direction. The acceleration operator $\mathbf{M}$ is taken as (4.4). The spatial derivatives as well as $\mathbf{M}^{-1}$ are computed by the discrete Fourier transform. For λ_u, the initial condition $\mathbf{\Psi}_0$ is taken as the transpose of (v_0, w_0), where

$$v_0 = e^{-(x+\pi/2)^2-(y+\pi/2)^2} + e^{-(x-\pi/2)^2-(y-\pi/2)^2} - (1+i)[e^{-(x-\pi/2)^2-(y+\pi/2)^2} + e^{-(x+\pi/2)^2-(y-\pi/2)^2}], \tag{4.9}$$

and

$$w_0 = e^{-(x-\pi/2)^2-(y+\pi/2)^2} + e^{-(x+\pi/2)^2-(y-\pi/2)^2} - (1+i)[e^{-(x+\pi/2)^2-(y+\pi/2)^2} + e^{-(x-\pi/2)^2-(y-\pi/2)^2}]. \tag{4.10}$$

For this unstable eigenvalue, all the three methods can converge as in the previous example. With $(c, \Delta t)$ taken as (2, 2) in the OOM, and as (2, 0.4) in the MSOM, the error diagrams of these iterations are plotted in Fig. 3c. We see that the error $|\lambda_n - \lambda_{n-1}|$ drops below 10^{-10} in about 780 and 610 iterations for the OOM and MSOM, respectively (the computing times are 36 and 16 seconds in MATLAB). Thus, unlike Example 2, the OOM converges slower than the MSOM for this unstable eigenvalue. When these results are compared to those from the matrix eigenvalue method above, we see that these new methods are much faster, and the eigenvalues they give are much more accurate.

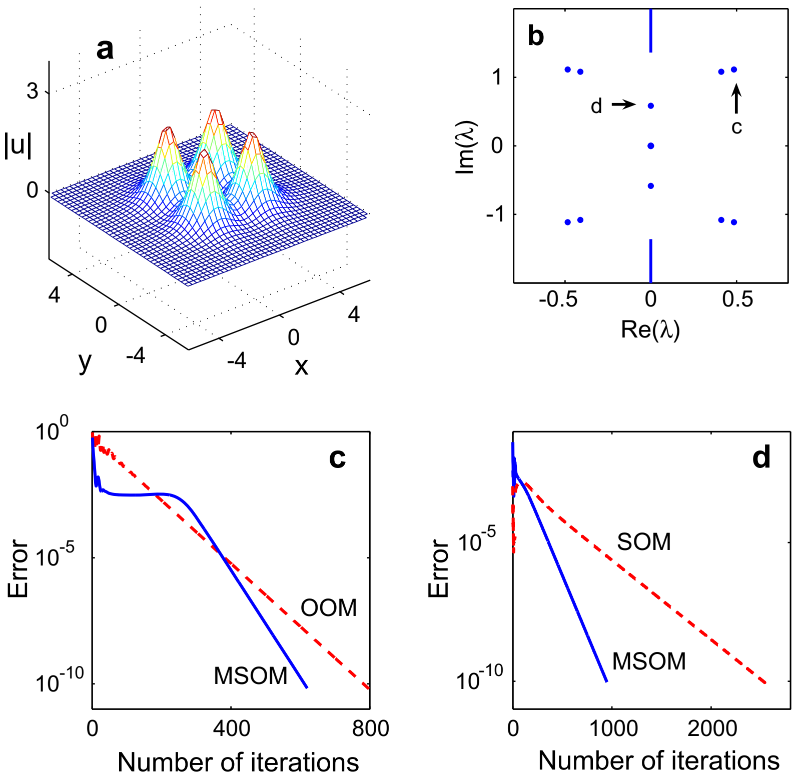

Fig. 3. (a) A lattice vortex soliton in Eq. (4.7) with $\mu = 0$ (shown is its intensity distribution); (b) eigenvalues in the stability spectrum of (a); (c) error diagrams of the OOM and MSOM for the most unstable eigenvalue $\lambda_u = 0.4961 + 1.1161i$ marked in (b); (d) error diagrams of the SOM and MSOM for the internal mode $\lambda_s = 0.5854i$ marked in (b).

For the internal mode λ_s, the OOM does not converge, while the SOM and MSOM do. For this eigenvalue, we take the initial condition as

$$v_0 = \mathrm{e}^{-(x-\pi/2)^2-(y+\pi/2)^2} + \mathrm{e}^{-(x+\pi/2)^2-(y-\pi/2)^2}, \tag{4.11}$$

and

$$w_0 = -\mathrm{e}^{-(x+\pi/2)^2-(y+\pi/2)^2} - \mathrm{e}^{-(x-\pi/2)^2-(y-\pi/2)^2}. \tag{4.12}$$

With $(c, \Delta t) = (2, 1)$ for MSOM and $(2, 1.1)$ for SOM, the error diagrams of iterations for both methods are plotted in Fig. 3d. It is seen that the errors of MSOM and SOM drop below 10^{-10} in about 950 and 2600 iterations (the computing times are 23 and 40 seconds in MATLAB). Thus, the MSOM converges faster than the SOM just like in the previous example.

From these examples as well as others we have experimented, we find that the OOM often converges to the most unstable eigenvalue, while the SOM and MSOM can converge to any eigenvalue, with the MSOM converging faster than the SOM.

5. Summary

In this paper, we proposed three iteration methods for the computation of eigenvalues and eigenfunctions in the linear stability of general solitary waves. These methods are based on iterating certain pre-conditioned time evolution equations associated with the linear stability eigenvalue problem. The first method is the original operator iteration method (2.3)–(2.9). It uses the fourth-order Runge–Kutta method to evolve the pre-conditioned linear stability operator, and it usually converges to the most unstable eigenvalue of the solitary wave. The second method is the squared-operator iteration method (3.2). It uses the Euler method to evolve the "square" of the pre-conditioned linear stability operator. This method is shown to converge to any discrete eigenvalue in the stability spectrum under mild conditions. The third method is the modified squared-operator iteration method (3.10). It is obtained by incorporating the mode elimination technique into the second method, which speeds up the convergence considerably. These methods are applied to various examples of physical interest, and are found to be efficient and easy to implement in any spatial dimension. A distinctive advantage of these methods over the traditional matrix eigenvalue method is their low use of computer memory, thus they are particularly valuable for higher spatial dimensions where the matrix eigenvalue method faces memory constraints. Another advantage of these methods is their fast speed. If a single eigenvalue is desired or needs to be traced upon parameter continuation, these methods can obtain this eigenvalue much more quickly than the matrix eigenvalue method. The disadvantages of these proposed methods are that (a) they need a reasonably good initial guess for the eigenfunction, which may not be available to the user; (b) if many eigenvalues are desired simultaneously, these methods can be awkward to use as they have to compute each eigenvalue individually. Given the above advantages and disadvantages of these proposed methods, the user can intelligently choose from them and other methods (such as the matrix eigenvalue method) according to the particular problem at hand. It should be pointed out that these proposed methods can be applied not only for the computation of stability spectra of solitary waves, but also for the computation of any eigenvalue problem of differential operators. Thus, these methods may find applications in various mathematical and physical disciplines where operator eigenvalue problems arise.

Acknowledgement

This work was supported in part by the Air Force Office of Scientific Research under Grant USAF 9550-05-1-0379.

Appendix. MATLAB code of the OOM for Example 1

In the appendix, we attach a sample MATLAB code of the OOM (2.3)–(2.9) for Example 1, which produces the unstable eigenmode and the error diagram in Fig. 1c and d. MATLAB codes of the OOM, SOM and

MSOM for Examples 2 and 3 can be similarly written, and they will be posted at the author's website when this article is published.

```
% OOM code for Example 1

Lx=20; N=128;                                       % grid parameters
x=-Lx/2:Lx/N:Lx/2-Lx/N;
kx=[0:N/2-1  -N/2:-1]*2*pi/Lx;
max_iteration=1e4; error_tolerance=1e-10;
mu=1; B=1/sqrt(1+16*mu/3);                          % soliton solution
u=sqrt(4*B*mu./(cosh(2*sqrt(mu)*x)-B));
c=2.4; DT=2.3;                                      % scheme parameters
F1=-mu-u.^2+u.^4; F2=-mu-3*u.^2+5*u.^4;
V=-0.3*(1-2*x.^2).*exp(-x.^2);                      % initial conditions
W=i*exp(-x.^2);

for nn=1:max_iteration                              % OOM iterations begin
    % first RK4 step
    LV=-i*(ifft(-kx.^2.*fft(W))+F1.*W);
    LW=-i*(ifft(-kx.^2.*fft(V))+F2.*V);
    MinvV=ifft(fft(V)./(c+kx.^2));
    MinvW=ifft(fft(W)./(c+kx.^2));
    lambda=sum(conj(MinvV).*LV+conj(MinvW).*LW)   ...
          /sum(conj(MinvV).*V+conj(MinvW).*W);
    Lambda(nn)=lambda;
    K1V=ifft(fft(LV-lambda*V)./(c+kx.^2));
    K1W=ifft(fft(LW-lambda*W)./(c+kx.^2));
    TV=V+0.5*K1V*DT;
    TW=W+0.5*K1W*DT;
    % second RK4 step
    LTV=-i*(ifft(-kx.^2.*fft(TW))+F1.*TW);
    LTW=-i*(ifft(-kx.^2.*fft(TV))+F2.*TV);
    MinvTV=ifft(fft(TV)./(c+kx.^2));
    MinvTW=ifft(fft(TW)./(c+kx.^2));
    Tlambda=sum(conj(MinvTV).*LTV+conj(MinvTW).*LTW)  ...
           /sum(conj(MinvTV).*TV+conj(MinvTW).*TW);
    K2V=ifft(fft(LTV-Tlambda*TV)./(c+kx.^2));
    K2W=ifft(fft(LTW-Tlambda*TW)./(c+kx.^2));
    TV=V+0.5*K2V*DT;
    TW=W+0.5*K2W*DT;
    % third RK4 step
    LTV=-i*(ifft(-kx.^2.*fft(TW))+F1.*TW);
    LTW=-i*(ifft(-kx.^2.*fft(TV))+F2.*TV);
    MinvTV=ifft(fft(TV)./(c+kx.^2));
    MinvTW=ifft(fft(TW)./(c+kx.^2));
    Tlambda=sum(conj(MinvTV).*LTV+conj(MinvTW).*LTW)  ...
           /sum(conj(MinvTV).*TV+conj(MinvTW).*TW);
    K3V=ifft(fft(LTV-Tlambda*TV)./(c+kx.^2));
    K3W=ifft(fft(LTW-Tlambda*TW)./(c+kx.^2));
    TV=V+K3V*DT;
    TW=W+K3W*DT;
    % fourth RK4 step
    LTV=-i*(ifft(-kx.^2.*fft(TW))+F1.*TW);
    LTW=-i*(ifft(-kx.^2.*fft(TV))+F2.*TV);
    MinvTV=ifft(fft(TV)./(c+kx.^2));
    MinvTW=ifft(fft(TW)./(c+kx.^2));
    Tlambda=sum(conj(MinvTV).*LTV+conj(MinvTW).*LTW)  ...
           /sum(conj(MinvTV).*TV+conj(MinvTW).*TW);
    K4V=ifft(fft(LTV-Tlambda*TV)./(c+kx.^2));
    K4W=ifft(fft(LTW-Tlambda*TW)./(c+kx.^2));
    V=V+(K1V+2*K2V+2*K3V+K4V)*DT/6;
    W=W+(K1W+2*K2W+2*K3W+K4W)*DT/6;

    if nn == 1
        lambda_error=1;
        else
        lambda_error=abs(Lambda(nn)-Lambda(nn-1))
    end
    if lambda_error < error_tolerance
        break
    end
end
lambda                                          % display the converged eigenvalue
semilogy(abs(Lambda(2:nn)-Lambda(1:nn-1)))      % plot the error diagram
```

References

[1] J.P. Boyd, Why Newton's method is hard for travelling waves: small denominators, KAM theory, Arnold's linear Fourier problem, non-uniqueness, constraints and erratic failure, Math. Comput. Simul. 74 (2007) 7281.

[2] J. Yang, Internal oscillations and instability characteristics of $(2+1)$ dimensional solitons in a saturable nonlinear medium, Phys. Rev. E 66 (2002) 026601.

[3] M. Mitchell, M. Segev, T.H. Coskun, D.N. Christodoulides, Theory of self-trapped spatially incoherent light beams, Phys. Rev. Lett. 79 (1997) 49904993.

[4] V.I. Petviashvili, Equation of an extraordinary soliton, Plasma Phys. 2 (1976) 469–472.

[5] M.J. Ablowitz, Z.H. Musslimani, Spectral renormalization method for computing self-localized solutions to nonlinear systems, Opt. Lett. 30 (2005) 2140–2142.

[6] T.I. Lakoba, J. Yang, A generalized Petviashvili iteration method for scalar and vector Hamiltonian equations with arbitrary form of nonlinearity, J. Comput. Phys. 226 (2007) 1668–1692.

[7] J.J. Garcia-Ripoll, V.M. Perez-Garcia, Optimizing Schrödinger functionals using Sobolev gradients: applications to quantum mechanics and nonlinear optics, SIAM J. Sci. Comput. 23 (2001) 1316.

[8] J. Yang, T.I. Lakoba, Universally-convergent squared-operator iteration methods for solitary waves in general nonlinear wave equations, Stud. Appl. Math. 118 (2007) 153–197.

[9] J. Yang, T.I. Lakoba, Accelerated imaginary-time evolution methods for the computation of solitary waves, Stud. Appl. Math. 120 (2008) 265–292.

[10] J. Yang, Vector solitons and their internal oscillations in birefringent nonlinear optical fibers, Stud. Appl. Math. 98 (1997) 61–97.

[11] N. Trefethen, Spectral Method in MATLAB, SIAM, Philadelphia, 2000.

[12] J.P. Boyd, Chebyshev and Fourier Spectral Methods, second ed., Dover Publications, 2001.

[13] G. Golub, C. Van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, 1996.

[14] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1951) 17–29.

[15] D.C. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, SIAM J. Matrix Anal. Appl. 13 (1992) 357–385.

[16] R.B. Lehoucq, D.C. Sorensen, C. Yang, ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, SIAM Publications, Philadelphia, 1998.

[17] T.I. Lakoba, J. Yang, A mode elimination technique to improve convergence of iteration methods for finding solitary waves, J. Comput. Phys. 226 (2007) 1693–1709.

[18] Y.S. Kivshar, G.P. Agrawal, Optical Solitons: From Fibers to Photonic Crystals, Academic Press, San Diego, 2003.

[19] N.G. Vakhitov, A.A. Kolokolov, Stationary solutions of the wave equation in the medium with nonlinearity saturation, Izv. Vyssh. Uchebn. Zaved. Radiofiz. 16 (1973) 1020 (Radiophys. Quantum Electron. 16 (1973) 783).

[20] F. Dalfovo, S. Giorgini, L.P. Pitaevskii, S. Stringari, Theory of Bose–Einstein condensation in trapped gases, Rev. Mod. Phys. 71 (1999) 463–512.

[21] J. Yang, Z. Musslimani, Fundamental and vortex solitons in a two-dimensional optical lattice, Opt. Lett. 28 (2003) 2094.

[22] N.K. Efremidis, J. Hudock, D.N. Christodoulides, J.W. Fleischer, O. Cohen, M. Segev, Two-dimensional optical lattice solitons, Phys. Rev. Lett. 91 (2003) 213906.

[23] A. Szameit, J. Burghoff, T. Pertsch, S. Nolte, A. Tnnermann, F. Lederer, Two-dimensional soliton in cubic fs laser written waveguide arrays in fused silica, Opt. Express 14 (2006) 6055–6062.