Journal of Computational Physics 228 (2009) 7007–7024

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Newton-conjugate-gradient methods for solitary wave computations

Jianke Yang

Department of Mathematics and Statistics, University of Vermont, 16 Colchester Avenue, Burlington, VT 05401, USA

ARTICLE INFO

Article history:
Received 10 February 2009
Received in revised form 9 June 2009
Accepted 15 June 2009
Available online 23 June 2009

MSC:
35Qxx
35Q51
65Nxx

Keywords:
Solitary waves
Newton's method
Conjugate-gradient methods

ABSTRACT

In this paper, the Newton-conjugate-gradient methods are developed for solitary wave computations. These methods are based on Newton iterations, coupled with conjugate-gradient iterations to solve the resulting linear Newton-correction equation. When the linearization operator is self-adjoint, the preconditioned conjugate-gradient method is proposed to solve this linear equation. If the linearization operator is non-self-adjoint, the preconditioned biconjugate-gradient method is proposed to solve the linear equation. The resulting methods are applied to compute both the ground states and excited states in a large number of physical systems such as the two-dimensional NLS equations with and without periodic potentials, the fifth-order KdV equation, and the fifth-order KP equation. Numerical results show that these proposed methods are faster than the other leading numerical methods, often by orders of magnitude. In addition, these methods are very robust and always converge in all the examples being tested. Furthermore, they are very easy to implement. It is also shown that the nonlinear conjugate gradient methods are not robust and inferior to the proposed methods.

1. Introduction

In studies of nonlinear wave equations in an unbounded domain, solitary waves play an important role in the solution dynamics. Indeed, an initial localized condition often evolves into a number of solitary waves and energy radiation at large times, provided that the solitary waves are stable. For integrable equations and some special non-integrable equations, solitary waves can be obtained analytically. But for most non-integrable systems, these waves defy analytical expressions and have to be computed numerically. So far, a number of numerical methods have been developed. Examples include the Newton's method [1,2], the shooting method [3], the Petviashvili-type methods [4–6], the accelerated imaginary time evolution methods [7,8], the squared-operator iteration methods [9], etc. The Newton's method is a classical iteration method. In this method, the solution is updated by solving a linear inhomogeneous operator equation, where the linear operator and the inhomogeneous term are the Jacobian (i.e. the linearization operator) and residue of the nonlinear wave equation, respectively. This linear operator equation is solved by turning it into a matrix equation through discretization, and then applying the LU or QR factorization technique [1,2]. The Newton's method has been used widely in the nonlinear wave community. However, the key step of this method, which is to solve the resulting matrix equation, can become very difficult when the matrix size is very large and not tri-diagonal (such as in two and higher dimensions). In addition, this method can encounter other difficulties in certain situations as well [10]. The shooting method is another familiar method with a long history. This method is very efficient and accurate. In addition, it can be used to compute embedded solitons for which other iterative methods generally fail [11]. But unfortunately this method works only for one-dimensional problems, or higher-dimensional problems which can be reduced to one-dimensional problems (through symmetry reduction). The Petviashvili method was

E-mail address: jyang@cems.uvm.edu

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.06.012

first proposed in the 1970s [4] and later generalized in [5,6]. It is based on the fixed-point iteration idea, but with a key improvement which is to introduce a stabilizing factor. This method became popular in recent years due to its easy implementation in arbitrary spatial dimensions as well as fast convergence in many situations. However, it only converges to the ground states of nonlinear wave equations, and would diverge for excited states [6,12]. The imaginary time evolution method is also a familiar method, especially in the physics community (see [13] for instance). But its original version is very slow, and its accelerated versions were developed only recently [7,8]. These methods are based on the idea of turning the stationary solitary wave computation problem into a time evolution problem of diffusion type, and normalize the solution by its power or amplitude at each evolution step. One important component of these methods is to introduce an acceleration operator to the time evolution equation, which would improve the convergence speed dramatically. These methods are also very easy to implement in arbitrary spatial dimensions, and their convergence is either faster than or competitive with the Petviashvili-type methods [8]. However, these methods generally can only converge to the ground states just like the Petviashvili-type methods [8]. In order to compute excited states, the squared-operator iteration methods were developed in [9]. These methods are based on the idea of time-evolving a "squared" operator equation (the power or amplitude normalization is optional). Evolution of this squared equation guarantees that these methods can converge to any solitary wave, including excited states. The acceleration operator is built inside the squared equation to improve convergence speeds. When another mode elimination technique is incorporated into these methods [14], the resulting method (called the modified squared-operator method in [9]) converges even faster. These squared-operator-type methods are also easy to implement for general nonlinear wave equations in arbitrary spatial dimensions, and they deliver satisfactory performances in many situations [9]. But there are situations where all the above methods can be quite slow, especially when the wave's propagation constant gets near the edge of the continuous spectrum so that the wave gets less localized (see Examples 3.4 and 3.5 later in the text). Thus even faster numerical methods are still called upon.

On a separate development, the conjugate-gradient method was developed in the early 1950s and has become the most prominent iterative method for solving large systems of linear equations nowadays [15–18]. Viewing the linear equation as a minimization problem of a quadratic form, this method uses conjugate directions instead of the local gradient for going downhill. The conjugate-gradient method has a number of important properties. One property is that for symmetric positive-definite matrices, this method gives the exact solution within n steps, where n is the size of the matrix [16–18]. Another property is that for symmetric positive-definite matrices, the matrix-weighted error decreases monotonically with each iteration [16,17]. The third property is that when the matrix size is large, this method often gives the solution to the required accuracy in much less than n steps, especially when a suitable preconditioning matrix is introduced [16,17]. The conjugate-gradient method was originally developed for linear equations with symmetric positive-definite matrices, but some practical applications show that this method can also solve linear equations with symmetric indefinite matrices (unless there is a breakdown due to division by zero during iterations which rarely occurs) [19]. Generalizations of the conjugate-gradient method to symmetric indefinite matrices, non-symmetric matrices and nonlinear systems have also been developed [20–25]. At the moment, no work has appeared in the literature to apply the conjugate-gradient method to solitary wave computations. It is not clear yet whether this method can be applied to solitary waves. If so, what scheme is the most efficient for this application? In addition, would this method perform better than the other leading numerical methods for solitary waves as described above?

In this paper, we apply the conjugate-gradient methods to the computation of general solitary waves (both the ground state and excited states) in nonlinear wave equations. The guiding principles in our algorithm design are fast convergence and easy implementation, which are equally weighed. We first linearize the solitary wave equation around an iterated solution and update the solution by solving a linear inhomogeneous operator equation, which resembles the idea of the Newton's method. Then, instead of solving this linear equation by direct methods as in the traditional Newton's method, we use the conjugate-gradient-type methods to solve it. If the linearization operator is self-adjoint, we use the preconditioned conjugate-gradient method to solve this linear equation. This method will be called the Newton-CG method in this paper. If the linearization operator is non-self-adjoint, we use the preconditioned biconjugate-gradient method to solve this linear equation. This method will be called the Newton-BCG method in this paper. We show that both methods converge for the ground state as well as the excited states of a wave system. In addition, they are very robust and converge in all our numerical testings with various physical wave systems and wide ranges of initial conditions (as long as the initial condition is reasonably close to the exact solution). No breakdown of these methods is ever observed (even though it is theoretically possible). The performance of these methods is demonstrated on a number of physical models such as the two-dimensional nonlinear Schrödinger (NLS) equations with and without periodic potentials, the fifth-order Kortewegde Vries (KdV) equation, and the fifth-order KadomtsevPetviashvili (KP) equation. They are found to converge much faster than the other leading numerical methods, often by orders of magnitude. In addition, these methods are very easy to implement regardless of the number of dimensions (a sample MATLAB code of the Newton-CG method will be displayed in Appendix A). Furthermore, we show that these Newton-CG/BCG methods, which are based on conjugate gradient iterations on a linear equation, are much better than the nonlinear conjugate-gradient methods which are sensitive to initial conditions. We expect that these proposed Newton-CG/BCG methods will replace the existing numerical methods and become the premier methods for computing both the ground-state and excited-state solitary waves in the days to come.

We would like to make a few remarks to put our results in a broader context. The combination of Newton-type methods (for solving nonlinear equations) and Krylov subspace methods (for solving the resulting linear Newton-correction equations) is a well known technique. In the literature, these methods are often referred to as the Newton–Krylov methods

(see [26,27] for instance). The Newton-CG and Newton-BCG methods proposed in this paper are two particular cases of the Newton–Krylov methods. Our own contributions in this paper are two-fold. One is to demonstrate that, for a wide range of solitary wave computations, one can use the simplest Krylov subspace methods, namely the conjugate-gradient method and the biconjugate-gradient method, even though the linear operator in the Newton-correction equation is generally indefinite and sometimes also non-symmetric. This message was not well recognized before. In the pursuit of algorithmic simplicity, our Newton-CG/BCG methods are the simplest (and probably also the most effective) Newton–Krylov methods for solitary wave computations. The other contribution of this paper is to demonstrate that these Newton-CG/BCG methods are more efficient than practically all the other leading numerical methods for solitary waves, often by a very wide margin. Thus these proposed methods represent a big step forward in the methodology for computing solitary waves. Given the ongoing interest for seeking solitary waves in various physical disciplines (such as nonlinear optics, Bose—Einstein condensates and water waves), these methods should prove very useful for those areas. Another remark we would like to make is that in the computations of the Helmholtz equation with a variable refraction index, the linear operator involved is also indefinite and non-symmetric, similar to the situation in this paper (see [28] and the references therein). Thus our results may be suggestive to the computations in that community as well.

2. Basic setup of the methods

We consider solitary waves in a general real-valued nonlinear wave system in arbitrary spatial dimensions, which can be written in the following form:

$$\mathbf{L}_0\mathbf{u}(\mathbf{x}) = 0. \tag{2.1}$$

Here $\mathbf{x}$ is a vector spatial variable, $\mathbf{u}(\mathbf{x})$ is a real-valued vector solitary wave solution admitted by Eq. (2.1), and $\mathbf{u} \to 0$ as $|\mathbf{x}| \to \infty$. For example, in the nonlinear Schödinger equation

$$iU_t + U_{xx} + |U|^2U = 0,$$

if one looks for solitary waves $U(x,t) = e^{i\mu t}u(x)$, where $u(x)$ is a real and localized function and μ is the propagation constant, then the equation for $u(x)$ is

$$u_{xx} - \mu u + u^3 = 0,$$

which is a special case of Eq. (2.1). Note that for complex-valued solitary waves, the equation can be rewritten in the above form with $\mathbf{u}$ containing the real and imaginary parts of the complex solution. In the above formulation, the propagation constant of the solitary wave is lumped into the operator $\mathbf{L}_0$.

Our goal is to solve solitary waves in Eq. (2.1) by iteration methods. Suppose we have an approximate solution $\mathbf{u}_n(\mathbf{x})$ which is close to the exact solution $\mathbf{u}(\mathbf{x})$. To obtain the next iteration solution $\mathbf{u}_{n+1}(\mathbf{x})$, we proceed as follows. First, we express the exact solution $\mathbf{u}(\mathbf{x})$ as

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_n(\mathbf{x}) + \mathbf{e}_n(\mathbf{x}), \tag{2.2}$$

where $\mathbf{e}_n(\mathbf{x}) \ll 1$ is the error term. Then we substitute this expression into Eq. (2.1) and expand it around $\mathbf{u}_n(\mathbf{x})$, which gives

$$\mathbf{L}_0\mathbf{u}_n + \mathbf{L}_{1n}\mathbf{e}_n = \boldsymbol{O}(\mathbf{e}_n^2). \tag{2.3}$$

Here $\mathbf{L}_{1n}$ is the linearization operator $\mathbf{L}_1$ of the solitary wave Eq. (2.1) evaluated at the approximate solution $\mathbf{u}_n(\mathbf{x})$. If we neglect the higher order term on the right hand side of Eq. (2.3), the remaining equation becomes a linear inhomogeneous equation for the error $\mathbf{e}_n$. This suggests that we update the approximate solution as

$$\mathbf{u}_{n+1}(\mathbf{x}) = \mathbf{u}_n(\mathbf{x}) + \Delta\mathbf{u}_n(\mathbf{x}), \tag{2.4}$$

where the updated amount $\Delta\mathbf{u}_n$ is computed from the linear inhomogeneous equation for $\mathbf{e}_n$, which is rearranged as

$$\mathbf{L}_{1n}\Delta\mathbf{u}_n = -\mathbf{L}_0\mathbf{u}_n. \tag{2.5}$$

We must point out that this part of the scheme is identical to that in the Newton's method [1,2]. As such, if the linear Newton-correction Eq. (2.5) is solved exactly (or to accuracy much higher than the size of $\Delta\mathbf{u}_n$), then the iterations (2.4) will converge to the exact solution $\mathbf{u}(\mathbf{x})$ quadratically. These nonlinear Newton iterations (2.4) form the outer iterations of our methods.

Our methods deviate from the Newton's method on how to solve the linear operator Eq. (2.5). In the Newton's method, Eq. (2.5) is solved by discretizing it into a matrix equation and then solved by direct methods such as LU or QR factorization [1,2]. Here we will use conjugate-gradient iterations to solve it. These linear conjugate-gradient iterations form the inner iterations of our methods. Thus our methods are loop-within-loop operations, where each Newton's iteration involves many conjugate-gradient iterations. But since both the Newton's iterations [for the nonlinear Eq. (2.1)] and conjugate-gradient iterations [for the linear Newton-correction Eq. (2.5)] converge very fast, the total number of conjugate-gradient iterations across all Newton's iterations is actually quite small, as our many examples will show later in this paper. An important feature of our setup above is that, the conjugate-gradient methods are applied to a *linear* Eq. (2.5). This contrasts the nonlinear conjugate-

gradient methods which have been developed in the literature for nonlinear optimization problems [17,25]. In the spirit of those methods, one would apply a generalized conjugate-gradient method directly to the nonlinear Eq. (2.1). We will show that our scheme above with linear conjugate–gradient methods is much more robust than the nonlinear conjugate-gradient methods, thus is the preferred way of applying conjugate-gradient ideas to solitary wave computations.

Detailed applications of the conjugate-gradient ideas for solving the linear Eq. (2.5) depend on whether the linearization operator $\mathbf{L}_1$ is self-adjoint or not. These two cases will be treated separately in the following sections.

3. The preconditioned conjugate-gradient method for self-adjoint linearization operators $\mathbf{L}_1$

In conservative wave systems, the linearization operator $\mathbf{L}_1$ in Eq. (2.5) is often self-adjoint. We consider this case in this section. The counterpart of this case in matrix equations is that the matrix is symmetric. For matrix equations, if the matrix is symmetric and positive-definite, the most efficient method is the preconditioned conjugate-gradient method, which has been described in numerous prior publications (see [16,17] for example). However, in solitary wave computations, the linear operator $\mathbf{L}_1$ is always indefinite, which corresponds to indefinite matrices in matrix equations. For symmetric indefinite matrices, the preconditioned conjugate-gradient method has a theoretical obstacle, which is that this method may break down due to division by zero during iterations. This "dark cloud" has prompted researchers to develop extended (more expensive) conjugate-gradient methods such as MINRES and SYMMLQ [20] in order to overcome this difficulty. Undeterred by this "dark cloud", we went ahead and applied the preconditioned conjugate-gradient method to Eq. (2.5) for a large number of nonlinear wave equations and various initial conditions. We found that, alas, this method always converged, and breakdown never occurred. In addition, its convergence was often so fast that it surprised us. Our later literature search found that, in Ref. [29], the author raised the question of how serious this potential breakdown of the conjugate-gradient method was for practical applications. In Ref. [19], the authors mentioned that the preconditioned conjugate-gradient method was often applied to indefinite symmetric matrices in driven microwave problems, and the breakdown rarely occurred. Our experience echoes that in [19], and shows that the breakdown of the preconditioned conjugate-gradient method does not constitute a serious concern in solitary wave computations. This is the basis on which we propose to use the preconditioned conjugate gradient method to solve Eq. (2.5).

Now we describe the preconditioned conjugate-gradient method as applied to the linear operator Eq. (2.5). To simplify notations, we drop the subscripts 'n' in Eq. (2.5). In addition, we always take the initial guess $\Delta\mathbf{u}^{(0)}$ to be zero for simplicity. Then the preconditioned conjugate-gradient method for the linear Newton-correction Eq. (2.5) is

$$
\begin{aligned}
&\Delta\mathbf{u}^{(0)} = 0,\\
&\mathbf{R}^{(0)} = -\mathbf{L}_0\mathbf{u},\\
&\mathbf{D}^{(0)} = \mathbf{M}^{-1}\mathbf{R}^{(0)},\\
&a^{(i)} = \frac{\langle \mathbf{R}^{(i)}, \mathbf{M}^{-1}\mathbf{R}^{(i)}\rangle}{\langle \mathbf{D}^{(i)}, \mathbf{L}_1\mathbf{D}^{(i)}\rangle},\\
&\Delta\mathbf{u}^{(i+1)} = \Delta\mathbf{u}^{(i)} + a^{(i)}\mathbf{D}^{(i)},\\
&\mathbf{R}^{(i+1)} = \mathbf{R}^{(i)} - a^{(i)}\mathbf{L}_1\mathbf{D}^{(i)},\\
&b^{(i+1)} = \frac{\langle \mathbf{R}^{(i+1)}, \mathbf{M}^{-1}\mathbf{R}^{(i+1)}\rangle}{\langle \mathbf{R}^{(i)}, \mathbf{M}^{-1}\mathbf{R}^{(i)}\rangle},\\
&\mathbf{D}^{(i+1)} = \mathbf{M}^{-1}\mathbf{R}^{(i+1)} + b^{(i+1)}\mathbf{D}^{(i)}.
\end{aligned}
$$

Here $i = 0, 1, 2, \ldots$ is the index of conjugate-gradient (CG) iterations, the inner product is the standard one in the square-integrable functional space:

$$\langle \mathbf{F}_1, \mathbf{F}_2\rangle = \int_{-\infty}^{\infty} \mathbf{F}_1^{\dagger}\cdot\mathbf{F}_2\, d\mathbf{x},$$

the superscript '$\dagger$' represents the Hermitian of a vector, and the operator $\mathbf{M}$ is the pre-conditioning operator which is required to be self-adjoint and positive-definite. This pre-conditioning operator is analogous to the acceleration operator in [9], and its role is to accelerate the convergence of the above conjugate gradient iterations. The operator $\mathbf{M}$ should be chosen so that it is easily invertible. In practise, it is often chosen to be the linear differential part of the operator $\mathbf{L}_0$ [9] (similar choice has also been taken in other situations, see [1,30]). These CG iterations are embedded inside the Newton iterations (2.4), and the resulting method will be called the Newton-CG method in this paper.

For solitary waves, the linearization operator $\mathbf{L}_1$ generally has both positive and negative eigenvalues, as well as the zero eigenvalue. For the ground state, $\mathbf{L}_1$ generally has one eigenvalue whose sign is opposite of all the others; for an excited state, $\mathbf{L}_1$ generally has two or more eigenvalues whose signs are opposite of all the others. Because of this, the CG iterations in the above Newton-CG method may break down since the denominator in the above $a^{(i)}$ formula may vanish. But as we have said above, our extensive testings of this method on various solitary wave equations have never encountered this breakdown (some selective examples will be shown later in this section). Even if this breakdown does occur, it can be fixed by changing the initial guess function $\mathbf{u}_0(\mathbf{x})$.

Regarding the zero eigenvalue of the linearization operator $\mathbf{L}_1$ (which exists in most solitary wave problems), it makes the solution to the linear Eq. (2.5) not unique, since its eigenfunctions can be added to a solution of (2.5) which remains a solution. If these eigenfunctions are induced by the invariances of the solitary waves (such as $\mathbf{u}_{x_j}$ when the solution $\mathbf{u}(\mathbf{x})$ is invariant with respect to a position shift in x_j), this non-uniqueness obviously is not a concern as it only leads to another solitary wave with a shifted free parameter. This is analogous to other iteration methods [8,9,12]. If the reader wishes to eliminate this non-uniqueness, there are simple techniques to do so. For instance, if the reader wants iterations to converge to a symmetric soliton with a peak at $x = 0$, he can simply take the initial guess $\mathbf{u}_0(x)$ to be symmetric in x. This way, the initial error function $\mathbf{e}_0(x)$ does not contain the position-shifting eigenmode $\mathbf{u}_x$, hence the peak will remain at $x = 0$, and no shifting will occur. What is surprising is that even if the kernel of $\mathbf{L}_1$ contains eigenfunctions which are not induced by invariances of the solitary waves, the Newton-CG method would still converge. This contrasts the other iteration methods (such as the Petviashvili method, the accelerated imaginary time evolution method, and the modified squared-operator method) which would not converge in such situations [8,9,12]. An example will be shown at the end of this section (Example 3.6). This surprising behavior of the Newton-CG method can be understood by making an analogy to the Newton's method for solving the algebraic equation

$$f(x) = (x - \alpha)^2 = 0, \tag{3.1}$$

whose root $x = \alpha$ is multi-fold. The Newton-correction equation for it is

$$f'(x_n)\Delta x_n = -f(x_n). \tag{3.2}$$

At the root $x = \alpha, f'(\alpha) = 0$, hence the kernel of $f'(\alpha)$ is non-empty, which resembles the non-empty kernel of $\mathbf{L}_1$ above. However, the Newton's method for this algebraic Eq. (3.1) clearly still converges (even though the convergence speed drops from quadratic to linear). The reason is that when x_n gets close to the root α, even though $f'(x_n)$ becomes small, the right hand side $f(x_n)$ in the correction Eq. (3.2) becomes even smaller. Thus this correction equation is actually *not* singular, hence Newton's iterations still converge. In the same spirit, when the kernel of $\mathbf{L}_1$ is non-empty, the Newton-CG method also converges. This convergence under non-empty kernels of $\mathbf{L}_1$ is one of the many advantages of the Newton-CG method over its peers. The CG iterations above are terminated when the approximate solution $\Delta\mathbf{u}_n^{(i)}$ to Eq. (2.5) has reached certain accuracy. The error of this solution can be measured by the function $\mathbf{R}^{(i)}$ in the CG iterations, which is the residue of the linear Eq. (2.5), i.e.

$$\mathbf{R}^{(i)} = -\mathbf{L}_0\mathbf{u}_n - \mathbf{L}_{1n}\Delta\mathbf{u}_n^{(i)}.$$

This error can be measured more conveniently by the $\mathbf{M}^{-1}$ weighted 2-norm of $\mathbf{R}^{(i)}$,

$$\|\mathbf{R}^{(i)}\|_M \equiv \langle \mathbf{R}^{(i)}, \mathbf{M}^{-1}\mathbf{R}^{(i)} \rangle^{1/2},$$

which appears in the CG iterations. The accuracy with which the linear Newton-correction Eq. (2.5) is solved is an important parameter in the Newton-CG method. Remember from Eq. (2.4) that the approximate soliton solution $\mathbf{u}_{n+1}$ is updated by the formula $\mathbf{u}_n + \Delta\mathbf{u}_n$. If the accuracy of $\mathbf{u}_n$ (compared to the exact soliton solution $\mathbf{u}$) is poor, then requiring too much accuracy for solving $\Delta\mathbf{u}_n$ from the linear Eq. (2.5) is a waste of effort, because it does not lead to higher accuracy in the approximate solution $\mathbf{u}_{n+1}$ (this phenomenon is called oversolving). However, when the approximate solution $\mathbf{u}_n$ gets very close to the exact soliton solution $\mathbf{u}$, higher accuracy would be necessary for solving $\Delta\mathbf{u}_n$ from Eq. (2.5) so that the rapid convergence of the Newton's method can be sustained. Thus an effective strategy for minimizing oversolving is to use the accuracy of the approximate solution $\mathbf{u}_n$ to determine *adaptively* the accuracy with which the linear Newton-correction Eq. (2.5) is solved [26]. The accuracy of $\mathbf{u}_n$ can be measured by

$$\|\mathbf{L}_0\mathbf{u}_n\|_M = \langle \mathbf{L}_0\mathbf{u}_n, \mathbf{M}^{-1}\mathbf{L}_0\mathbf{u}_n \rangle^{1/2},$$

which is the $\mathbf{M}^{-1}$ weighted 2-norm of the residue $\mathbf{L}_0\mathbf{u}_n$ of the nonlinear wave Eq. (2.1). Then a sensible stopping criterion for the CG iterations in solving the linear Eq. (2.5) is that the error of $\Delta\mathbf{u}_n^{(i)}$ is below a certain fraction of the error of the solution $\mathbf{u}_n$ itself. In this spirit, we take the stopping criterion of the CG iterations to be

$$\|\mathbf{R}^{(i)}\|_M < \epsilon_{cg}\|\mathbf{L}_0\mathbf{u}_n\|_M. \tag{3.3}$$

Here ϵ_{cg} is a small positive error tolerance parameter for CG iterations. Notice that the residue $\mathbf{L}_0\mathbf{u}_n$ is the inhomogeneous term of the linear Newton-correction Eq. (2.5). Also notice that due to our choice of the zero initial condition for CG iterations, $\mathbf{R}^{(0)} = -\mathbf{L}_0\mathbf{u}_n$. Thus the stopping criterion (3.3) for CG iterations is simply

$$\|\mathbf{R}^{(i)}\|_M < \epsilon_{cg}\|\mathbf{R}^{(0)}\|_M. \tag{3.4}$$

Regarding the choice of the error tolerance parameter ϵ_{cg}, if it is set too small, this leads to oversolving which is ineffective. On the other hand, if ϵ_{cg} is set too large, which means that the linear Eq. (2.5) is solved too inaccurately, then the Newton iterations (2.4) may not converge at all. Thus the optimal ϵ_{cg} should be neither too small nor too large. Our numerical testings show that the optimal ϵ_{cg} is generally in the range between 10^{-1} and 10^{-3}. In all numerical examples in this paper, we set $\epsilon_{cg} = 10^{-2}$. At this value of ϵ_{cg}, if the solution's accuracy is set at 10^{-10}, then the number of Newton's iterations in the Newton-CG method ranges from 5 to 8 in all our numerical examples.

In the remainder of this section, we apply this Newton-CG method to various examples of solitary wave computations, and demonstrate its performances. We also compare its performances to those of the other leading iteration methods which have been developed in the literature. The leading methods for the ground-state solitons are the accelerated imaginary time evolution method with amplitude normalization (AITEM) [8] and the Petviashvili method [4]. The leading method for excited-state solitons is the modified squared operator method (MSOM) [9]. In all our performance illustrations, the error of the numerical solution $\mathbf{u}_n$ is measured as $|\mathbf{L}_0\mathbf{u}_n|_{max}$, which is the maximum value of the nonlinear residue $|\mathbf{L}_0\mathbf{u}_n|$ in Eq. (2.1). In the error diagrams of the Newton-CG method, the error of the numerical solution $\mathbf{u}_n$ after each Newton's iteration is plotted against the total number of CG iterations across all preceding Newton's iterations (*NOT against the number of Newton iterations*). This total number of CG iterations in the Newton-CG method will be compared with the number of iterations in the other methods (AITEM, Petviashvili and MSOM). These iteration numbers are valuable information since they are computer-independent and software-independent. In addition, these numbers do not change much if finer mesh grids are used. However, one should bear in mind that the computational costs of one iteration in these different methods are different. Specifically, when spatial derivatives and $\mathbf{M}^{-1}$ are computed by the Fourier-pseudospectral method, as is done in all examples of this paper, then the ratios of one iteration's computational costs for the Newton-CG method, the Petviashvili method, the AITEM and the MSOM are approximately 1:1:1.3:3 (here one iteration in the Newton-CG method refers to one CG iteration). In other words, the costs of one CG iteration and one Petviashvili iteration are about the same, one AITEM iteration costs 30% more, and one MSOM iteration costs three times as much. These computational costs can be understood by counting the number of Fourier transforms and inverse Fourier transforms which often dominate the computations. Thus the number of iterations may not accurately reflect the efficiency of a method. A more sensible way to compare the efficiencies of different methods is to compare the CPU times they use for the same solution accuracy. This is also done in all examples of this paper. In such comparisons, all computations are performed in MATLAB (version 7.0) on a personal computer (with AMD Athlon processor, 2.4 GHz speed and 4 Gb RAM).

All examples below will show that the Newton-CG method is very robust and always converges (we have never encountered an exception). In addition, this method is faster than its peers, often by orders of magnitude. Even though a rigorous proof of its fast convergence is hardly possible, a heuristic understanding is still available. For a matrix equation $Ax = b$, if the matrix A is symmetric and positive-definite, then the error of the preconditioned conjugate gradient method (by a preconditioning matrix M) decays at least by a factor of $R_{cg} = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ with each iteration [16]. Here κ is the spectral condition number of the matrix $M^{-1}A$, i.e. $\kappa = |\lambda_{max}/\lambda_{min}|$, where λ_{max} and λ_{min} are the largest and smallest non-zero eigenvalues of $M^{-1}A$ (in magnitude). From this relation we see that the number of iterations to reach a certain relative reduction in the error is roughly proportional to $\sqrt{\kappa}$. If we extrapolate this result to the CG iterations in the Newton-CG method, then we can expect that the number of CG iterations to reach a certain error reduction is roughly proportional to $\sqrt{\kappa}$, where κ is the spectral condition number of the operator $\mathbf{M}^{-1}\mathbf{L}_1$. For the AITEM and the Petviashvili method, however, the error decays by a factor of $R = (\kappa - 1)/(\kappa + 1)$, where κ is the same as above [8,12]. Thus the numbers of AITEM and Petviashvili iterations to reach a certain error reduction are roughly proportional to κ. For the MSOM, the error decays more erratically, but this decay is roughly at the rate of $R_m = (\kappa^2 - 1)/(\kappa^2 + 1)$, since the spectral condition number of the squared operator in the MSOM is roughly κ^2, where κ is the same as above [9]. Thus the number of MSOM iterations to reach a certain error reduction is roughly proportional to κ^2. In most solitary wave computations, κ is moderate or large, thus the Newton-CG method converges faster than the AITEM, MSOM and the Petviashvili method. In situations where the spectral condition number of $\mathbf{M}^{-1}\mathbf{L}_1$ is very large (such as when the propagation constant lies near the edge of the continuous spectrum), the Newton-CG method will be much faster than its peers as Examples 3.4 and 3.5 below will show.

Example 3.1 (*Ground states of the 2D NLS equation*). The first example we consider is the computation of ground states in the familiar two-dimensional NLS equation

$$iU_t + U_{xx} + U_{yy} + |U|^2U = 0. \tag{3.5}$$

Ground states of this equation are of the form $U(x,y,t) = u(x,y)e^{i\mu t}$, where $u(x,y)$ is a positive function satisfying the equation

$$u_{xx} + u_{yy} + u^3 = \mu u. \tag{3.6}$$

The linearization operator for this equation is

$$\mathbf{L}_1 = \partial_{xx} + \partial_{yy} + 3u^2 - \mu.$$

At $\mu = 1$, the ground state is shown in Fig. 3.1(a). To compute this ground state, we have applied three iteration methods: the Petviashvili method, the AITEM, and the Newton-CG method. For both the AITEM and the Newton-CG method, the acceleration operator $\mathbf{M}$ is taken as

$$\mathbf{M} = c - \partial_{xx} - \partial_{yy}, \tag{3.7}$$

where c is a positive constant. The computational domain is taken as a square of $-15 < x, y < 15$, discretized by 256 points along each dimension. Spatial derivatives as well as $\mathbf{M}^{-1}$ are computed by the Fourier-pseudospectral method, thus the spatial accuracy of our computations is spectral [1,31]. The initial condition is taken as

$$u_0(x,y) = 2.2e^{-x^2-y^2}. \tag{3.8}$$

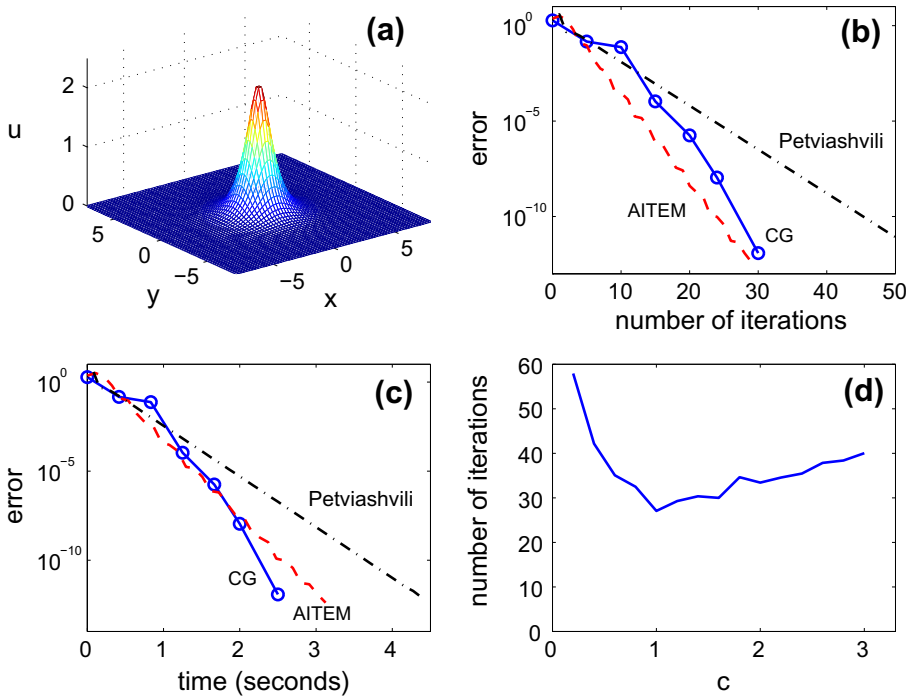


Fig. 3.1. (a) The ground state in the 2D NLS Eq. (3.6) with $\mu = 1$; (b, c) error diagrams of the Newton-CG method (marked by 'CG'), the Petviashvili method and the AITEM versus the number of iterations (b) and the CPU time (c); in the Newton-CG method, the error of the numerical solution after each Newton's iteration is plotted against the total number of CG iterations (b) or time (c) across all preceding Newton's iterations (each circle represents a Newton's iteration point); these conventions apply to all figures in this paper; (d) dependence of the total number of CG iterations on the acceleration parameter c in the Newton-CG method (solution accuracy set at 10^{-10}).

In the AITEM, the optimal scheme parameters are $c_{opt} = 1$ and $\Delta t_{opt} = 1.3$; and in the Newton-CG method, the optimal c parameter is $c_{opt} = 1$ (for this example, $c_{opt} = \mu$ in general in both methods). At these optimal scheme parameters, the error diagrams versus the number of iterations are displayed in Fig. 3.1(b). As has been mentioned before, the error diagram of the Newton-CG method plots the error of the numerical solution after each Newton's iteration against the total number of preceding CG iterations (by circle points), which is the case for all figures in this paper. These circle points are connected by straight lines for illustration purpose. One can see that for an accuracy below 10^{-10}, the Newton-CG method takes six Newton iterations. Each Newton iteration contains five CG iterations on average, which gives a total of 30 CG iterations. This total number of CG iterations is slightly more than the number of AITEM iterations, but much less than the number of Petviashvili iterations. To compare the speeds of these methods, the error diagrams versus the CPU times are displayed in Fig. 3.1(c). One can see that for an accuracy below 10^{-10}, the Newton-CG method takes about 2.4 s, which is faster than both the AITEM and Petviashvili methods. Thus, the Newton-CG method delivers the best performance among its peers on this simple example. To test the sensitivity of the Newton-CG method to the acceleration parameter c, we have tried different c values and recorded the number of CG iterations the Newton-CG method takes to reach accuracy below 10^{-10}, and the results are shown in Fig. 3.1(d). One can see that the efficiency of the Newton-CG method is not very sensitive to the c value as long as c is not very small. This insensitivity to pre-conditioning parameters is another advantage of the Newton-CG method.

Example 3.2 (*Vortex solitons of the 2D NLS equation*). The second example we consider is the computation of vortex solitons in the above 2D NLS Eq. (3.5). These vortex solitons are $U(x, y, t) = u(x, y)e^{i\mu t}$, where $u(x, y)$ is a complex function of the form $f(r)e^{i\theta}$, and (r, θ) is the polar coordinate of the (x, y) plane. These solitons are the excited states of the 2D NLS equation. At $\mu = 1$, this vortex solution is shown in Fig. 3.2(a). For these excited states, the Petviashvili method and the AITEM do not converge. Below we apply the Newton-CG method to compute this solution, and compare its performance with that of the MSOM. For this purpose, we express $u = v + iw$, where v and w are the real and imaginary parts of the function u. The equations for v and w are

$$v_{xx} + v_{yy} + (v^2 + w^2)v = \mu v, \tag{3.9}$$

$$w_{xx} + w_{yy} + (v^2 + w^2)w = \mu w. \tag{3.10}$$

In both the Newton-CG and MSOM methods, we take the acceleration operator $\mathbf{M}$ as

$$\mathbf{M} = (c - \partial_{xx} - \partial_{yy}) \quad \text{diag} \quad (1, 1), \tag{3.11}$$

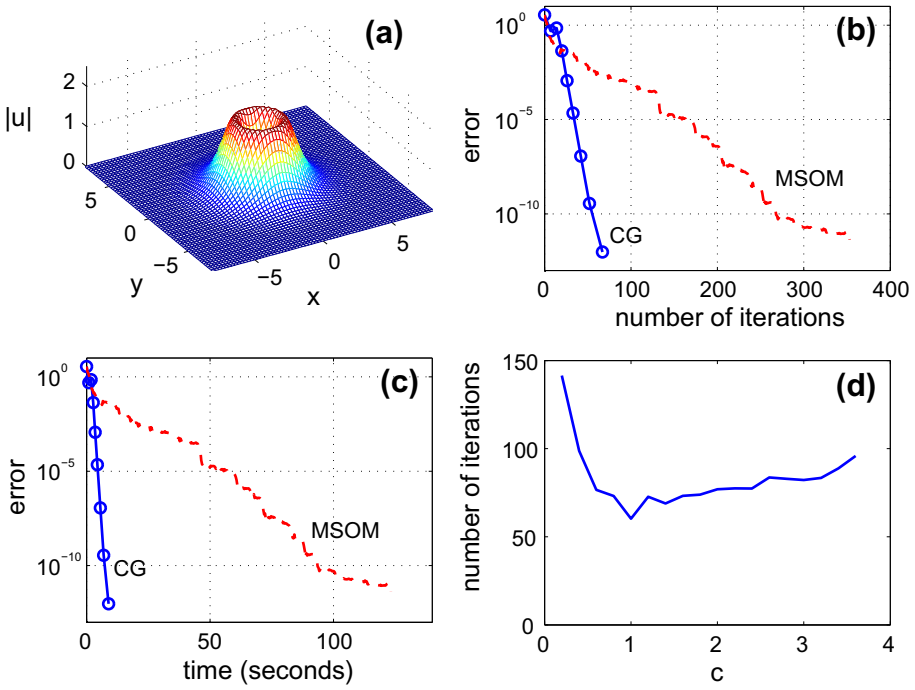


Fig. 3.2. (a) The vortex soliton $|u(x,y)|$ in the 2D NLS Eq. (3.5) with $\mu = 1$; (b, c) error diagrams of the Newton-CG method (marked by 'CG') and the MSOM versus the number of iterations (b) and the CPU time (c); each circle in the Newton-CG diagram represents a Newton's iteration point; (d) dependence of the total number of CG iterations on the acceleration parameter c in the Newton-CG method (solution accuracy set at 10^{-10}).

where c is a positive parameter. The computational domain is taken as a square of $-15 < x, y < 15$, discretized by 256 points along each dimension. The initial condition is taken as

$$u_0(x,y) = 2.5r \text{ sech } r\, e^{i\theta}. \tag{3.12}$$

In the Newton-CG method, the optimal acceleration parameter c is $c_{opt} = \mu = 1$; in the MSOM, the optimal scheme parameters are $c_{opt} = 2.5$ and $\Delta t_{opt} = 1.2$. At these optimal scheme parameters, the error diagrams of these two methods versus the number of iterations and the CPU time are displayed in Fig. 3.2(b and c), respectively. One can see from these diagrams that for the computation of this vortex soliton, the Newton-CG method is much faster than the MSOM. Specifically, to reach an accuracy below 10^{-10}, the Newton-CG method takes eight Newton iterations, which contain a total of 65 CG iterations. This total number of CG iterations is more than five times less than the number of MSOM iterations. In terms of the CPU time, the Newton-CG method takes under 8 s to reach accuracy 10^{-10}, which is more than 15 times faster than the MSOM (this speed difference is expected since one MSOM iteration costs about three times as much as one CG iteration, see earlier text in this section). Thus for this vortex soliton, the Newton-CG method is much more efficient than its peer method MSOM by orders of magnitude. To test the sensitivity of the Newton-CG method to the acceleration parameter c in this example, we have taken various c values and recorded the total number of CG iterations to reach solution accuracy 10^{-10}, and the results are shown in Fig. 3.2(d). One can see that for this excited state, the Newton-CG method is not sensitive to the acceleration parameter c either, just like Example 3.1.

Example 3.3 (*Depression and elevation waves in the fifth-order KdV equation*). Our next example is the fifth-order KdV equation

$$U_t + 6UU_x + 2U_{xxx} + U_{xxxxx} = 0, \tag{3.13}$$

which is a normalized model equation for small-amplitude gravity-capillary waves on water of finite depth when the Bond number is close to 1/3 [32]. Here $U(x,t)$ is the non-dimensionalized free-surface elevation. This equation admits depression and elevation waves with decaying oscillatory tails which bifurcate from the point of minimum phase speed. These waves are of the form $U(x,t) = u(x - vt)$, where $v < -1$ is the wave's speed, and the function $u(x)$ satisfies the equation

$$u_{xxxx} + 2u_{xx} + 3u^2 = vu, \tag{3.14}$$

as well as the boundary conditions $u(x) \to 0$ as $x \to \pm\infty$. At $v = -1.2$, the corresponding depression and elevation waves are shown in Fig. 3.3(a and b), respectively. Now we compute these waves by the Newton-CG method and its peers. The Petviashvili method and the AITEM converge for the depression wave, which is the ground state of this system. But for the ele-

vation wave which is the excited state, they both diverge [8,12], thus the MSOM will be used and compared with the Newton-CG method. In all computations, the spatial domain is taken as $-15\pi < x < 15\pi$, discretized by 512 grid points. The acceleration operator **M** in the AITEM, MSOM and the Newton-CG method are all taken as the linear part of Eq. (3.14),

$$\mathbf{M} = \partial_{xxxx} + 2\partial_{xx} - v,$$

which gives optimal or near-optimal performance. The initial condition for the depression wave is taken as

$$u_0(x) = -0.25 \quad \text{sech } 0.3x \cos x,$$

while that for the elevation wave is taken as $-u_0(x)$. For the depression wave, the error diagrams against the number of iterations for the Newton-CG and Petviashvili methods are displayed in Fig. 3.3(c). We see that for the solution accuracy of 10^{-10}, the Newton-CG method takes five Newton iterations consisting of a total of 26 CG iterations. This total number of CG iterations is half the number of Petviashvili iterations. In terms of the CPU time, the Newton-CG method takes about 0.004 s, which is twice as fast as the Petviashvili method (this is expected since the computational costs of one CG iteration and one Petviashvili iteration are about the same). The AITEM is found to be slower than the Newton-CG method but faster than the Petviashvili method. This is analogous to Example 3.1 and thus not shown. For the elevation wave, the error diagrams against the number of iterations for the Newton-CG method and the MSOM (with $\Delta t_{opt} = 0.24$) are displayed in Fig. 3.3(d). In this case, to reach accuracy 10^{-10}, the Newton-CG method takes six Newton iterations consisting of a total of 39 CG iterations. This total number of CG iterations is far less than the 95 iterations of the MSOM. In terms of the computing time, the Newton-CG method takes about 0.005 s, which is eight times as fast as the MSOM. Thus for these depression and elevation waves, the Newton-CG method is superior to its peer methods (especially for the elevation wave).

Example 3.4 (*Semi-infinite-gap solitons in the 2D NLS equation with periodic potentials*). The next example is the 2D NLS equation with periodic potentials

$$iU_t + U_{xx} + U_{yy} - V_0(\sin^2 x + \sin^2 y)U + \sigma|U|^2U = 0, \tag{3.15}$$

which models nonlinear light propagation as well as Bose–Einstein condensate's dynamics in optical lattices [33–35]. Here $\sigma = \pm 1$ corresponds to self-focusing or self-defocusing nonlinearity, and V_0 is the strength of the periodic potential. This equation has drawn a lot of attention in recent literature and has been heavily studied. This model admits a rich variety of solitary waves in the form $U(x,y,t) = u(x,y)e^{-i\mu t}$, where $u(x,y)$ satisfies the equation

$$u_{xx} + u_{yy} - V_0\left(\sin^2 x + \sin^2 y\right)u + \sigma|u|^2u = -\mu u, \tag{3.16}$$

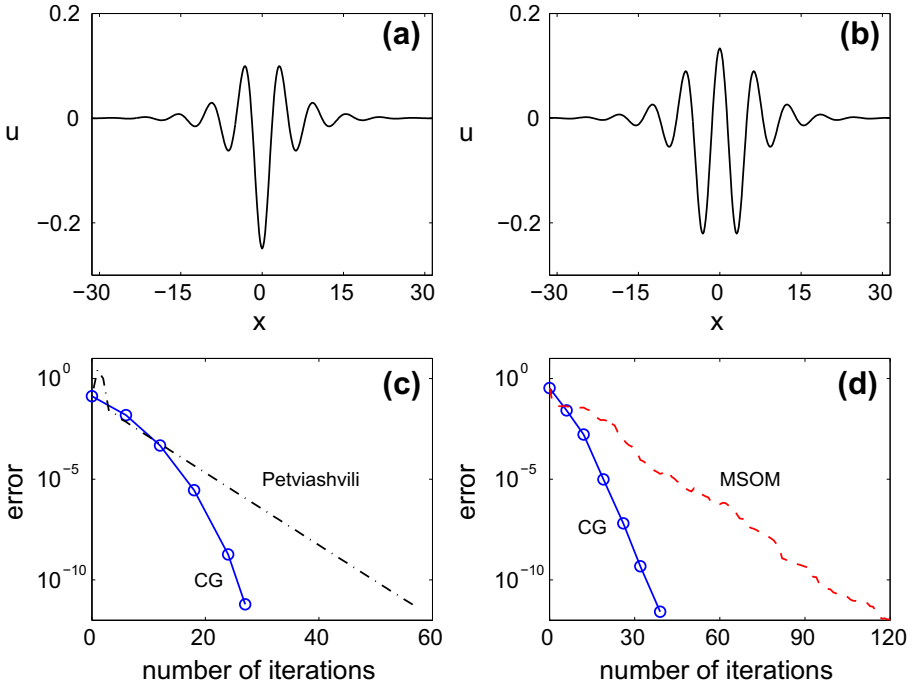
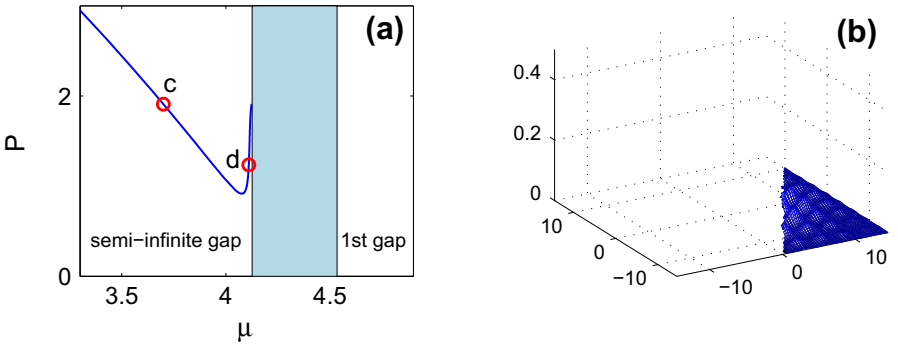


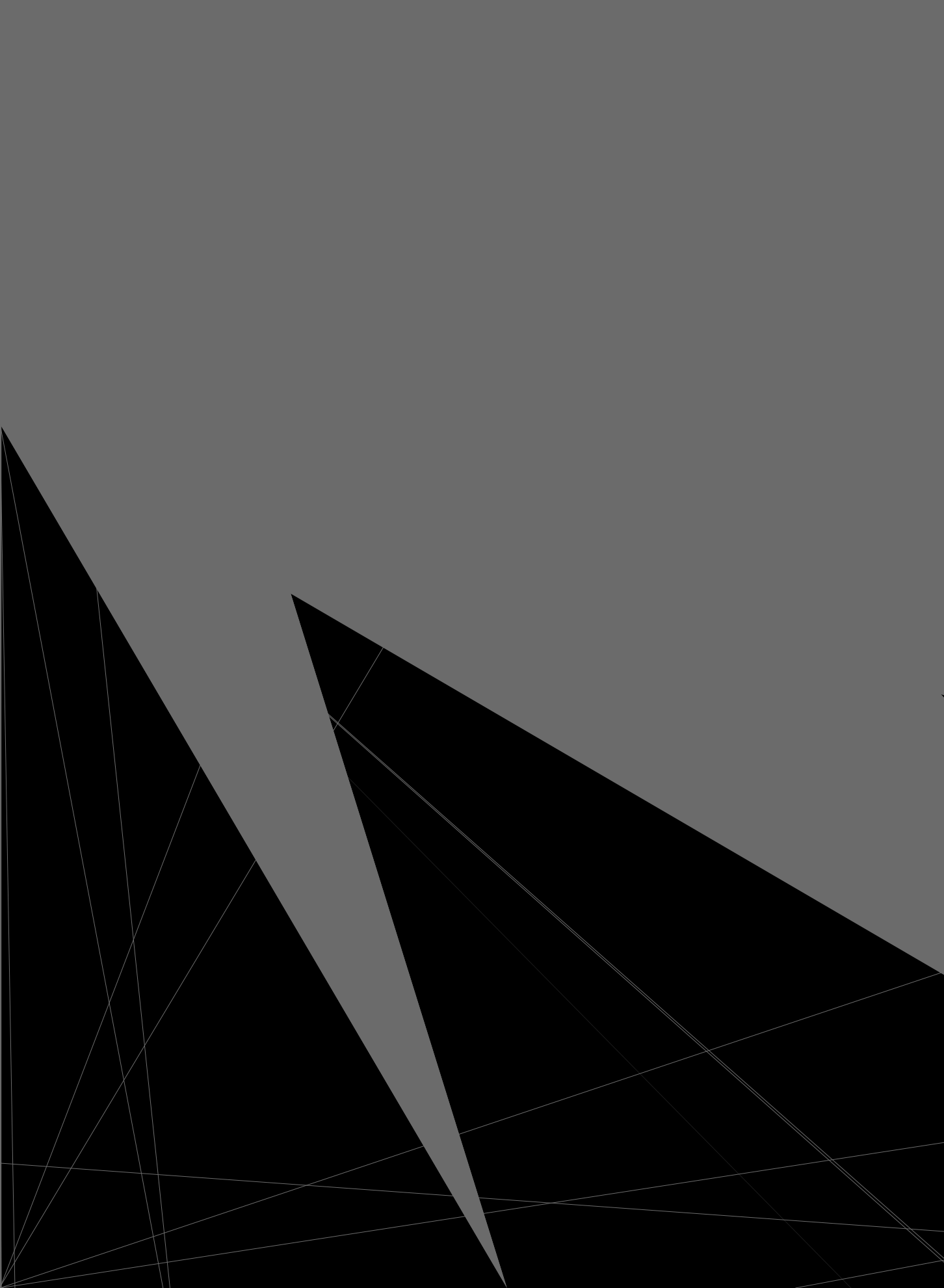
Fig. 3.3. (a, b) The depression wave (a) and elevation wave (b) in the fifth-order KdV Eq. (3.14) with $v = -1.2$; (c, d) error diagrams of the Newton-CG method (marked by 'CG') and its peers for the depression wave (c) and elevation wave (d).

and μ is the propagation constant [35]. When the nonlinearity is self-focusing ($\sigma = 1$), a family of solitary waves admitted in this model is that the solution $u(x,y)$ is positive, and has a single intensity maximum which is located at a lattice site (i.e. a potential minimum). This so-called on-site solution family resides in the semi-infinite gap of the linear spectrum, and can be considered as the ground state of this system. Defining the power of a solitary wave $u(x,y)$ as $P = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |u|^2\, dxdy$, then the power curve of this solution family at the potential strength $V_0 = 6$ is displayed in Fig. 3.4(a). At the marked point 'd' on this power curve ($\mu = 4.11$) which is near the edge of the Bloch band, the solution profile is displayed in Fig. 3.4(b). This solution has low amplitude and is quite broad. At another marked point 'c' on the power curve ($\mu = 3.7$), the solution has higher amplitude and is more localized. Now we compute these two solitons by the Newton-CG method, the Petviashvili method and the AITEM. The Petviashvili method used here is an extension of the original Petviashvili method which was proposed in [36]. In both the Newton-CG method and the AITEM, the acceleration operator **M** is taken as (3.7). For the more localized soliton at point 'c' of the power curve, the computational domain is taken as a square of $-5\pi < x, y < 5\pi$, discretized by 256 points along each dimension. The initial condition is taken as

$$u_0(x,y) = 1.15\,\mathrm{sech}2\sqrt{x^2+y^2}. \tag{3.17}$$

The optimal scheme parameters in these methods are: $c_{opt} = 3$ in the Newton-CG method; $c_{opt} = 2.4$ in the generalized Petviashvili method [36]; and $c_{opt} = 2, \Delta t_{opt} = 0.9$ in the AITEM. At these optimal scheme parameters, the error diagrams of these methods versus the number of iterations are displayed in Fig. 3.4(c). One can see that to reach accuracy 10^{-10}, the Newton-CG method takes six Newton iterations which consist of a total of 59 CG iterations. This total number of CG iterations is one third of those of the Petviashvili and AITEM iterations. In terms of the CPU time, the Newton-CG method takes less than 5 s, which is three times as fast as the Petviashvili method, and four times as fast as the AITEM. For the less localized soliton at point 'd' of the power curve, the computational domain is taken as a square of $-10\pi < x, y < 10\pi$, discretized by 256 points along each dimension. The initial condition is taken the same as (3.17), except that the amplitude is reduced from 1.15 to 0.49. For this less localized soliton, the optimal scheme parameters are $c_{opt} = 3$ for the Newton-CG method, $c_{opt} = 1.9$ for the generalized Petviashvili method, and $c_{opt} = 2, \Delta t_{opt} = 1$ for the AITEM. At these optimal scheme parameters, the error diagrams of these methods versus the number of iterations are displayed in Fig. 3.4(d). This time, the Newton-CG method is much faster than the other two methods. Specifically, to reach accuracy 10^{-10}, the Newton-CG method takes about a total of 250 CG iterations (in the seven Newton iterations), which is 12 times less than the Petviashvili iterations and 11 times less than the AITEM iterations. In terms of the CPU time, the Newton-CG method takes about 22 s, which is more than an order of magnitude faster than the Petviashvili method and the AITEM. Thus for these on-site lattice solitons, the Newton-CG method is far ahead of its peers, especially near edges of the continuous spectrum where the Newton-CG method is more than an order of magnitude faster than the others.





magnitude difference in speed is observed. In Appendix A, the sample MATLAB code for the gap soliton in Fig. 3.5(b) is displayed. This code demonstrates the simple implementation of the Newton-CG method.

Example 3.6 (*Convergence of the Newton-CG method when the kernel of* $\mathbf{L}_1$ *contains non-invariance-related eigenfunctions*). In generic cases, the kernel of the linearization operator $\mathbf{L}_1$ contains only eigenfunctions which are induced by the invariances of solitary waves which do not affect the convergence of iteration methods [8,9,12]. However, in certain non-generic cases, the kernel of $\mathbf{L}_1$ may contain eigenfunctions which are not induced by the invariances of solitary waves. In such cases, previous iteration methods (including the MSOM) would not converge [9]. However, the Newton-CG method will still quickly converge in these cases. A heuristic reason for this has been given earlier in this section. To demonstrate, we consider the 1D NLS equation with the saturable nonlinearity and a quasi-periodic potential

$$iU_t + U_{xx} - \frac{6}{1+|U|^2+I_L(x)}U = 0, \tag{3.20}$$

which models light beam propagation in a photorefractive crystal with a photonic lattice [37]. Here $I_L(x) = 3\cos^2 x(1+0.5e^{-x^8/128})$ is the intensity field of the photonic lattice with a single-site defect. This equation admits a family of defect solitons $U(x,t) = u(x,\mu)e^{-i\mu t}$ in the semi-infinite gap, where $u(x,\mu) > 0$ and satisfies the equation:

$$\mathbf{L}_0 u = u_{xx} - \frac{6}{1+u^2+I_L(x)}u + \mu u = 0. \tag{3.21}$$

The power diagram of this solution family is displayed in Fig. 3.6(a). For this solution family, the largest eigenvalue $\lambda_1(\mu)$ of the linearization operator $\mathbf{L}_1$ is always positive, and its eigenfunction is symmetric. The second largest eigenvalue $\lambda_2(\mu)$ of $\mathbf{L}_1$ (with anti-symmetric eigenfunctions) can be positive or negative depending on the μ value. The graph of $\lambda_2(\mu)$ is shown in Fig. 3.6(b). We can see that at $\mu = 1.18, \lambda_2 = 0$, thus the kernel of $\mathbf{L}_1$ here contains an anti-symmetric eigenfunction which is not invariance-induced. The defect soliton at this μ value is shown in Fig. 3.6(c), and its location on the power curve is marked by the letter 'c' in Fig. 3.6(a). It is noted that according to the Vakhitov–Kolokolov stability criterion [38,39], this defect soliton is on the boundary between unstable solitons (on the left of $\mu = 1.18$) and stable ones (on the right of $\mu = 1.18$). To compute this soliton, we take a generic initial condition

$$u_0(x) = 3e^{-0.5x^2} + 0.5\ \mathrm{sech}x\tanh x,$$

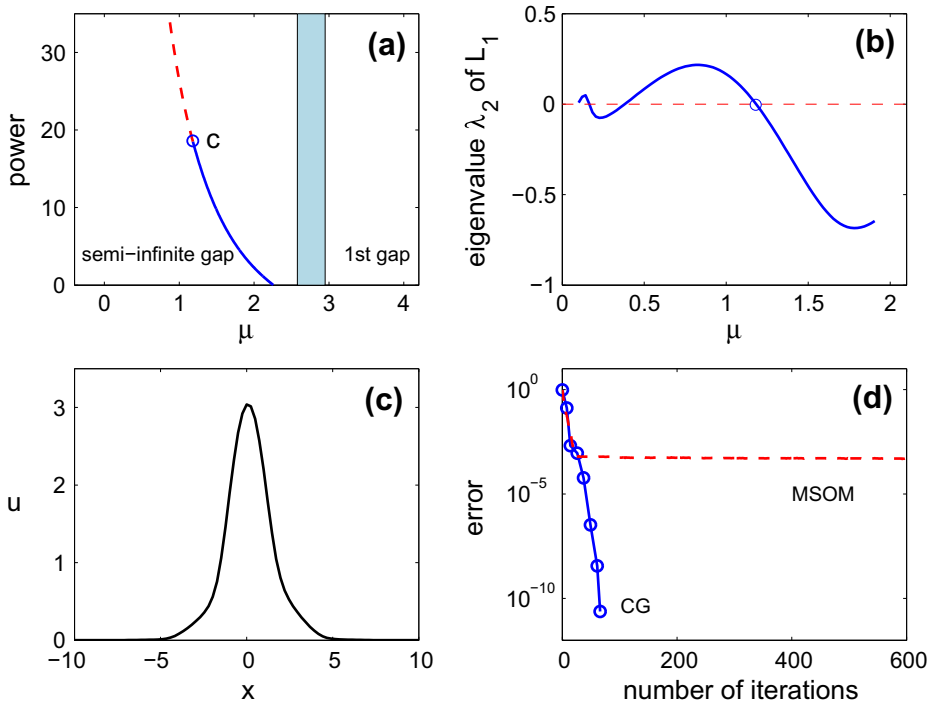


Fig. 3.6. (a) Power diagram of the defect solitons in the semi-infinite gap of Eq. (3.21); the dashed portion is linearly unstable; (b) the second largest eigenvalue $\lambda_2(\mu)$ of the linearization operator $\mathbf{L}_1$ versus μ; (c) the defect soliton at $\mu = 1.18$ where $\lambda_2 = 0$; this μ value is marked in (a, b) by a circle; (d) error diagrams of the Newton-CG method (marked by 'CG') and the MSOM for the computation of this defect soliton in (c).

which contains both symmetric and anti-symmetric components. For this initial condition, previous iteration methods such as the AITEM, MSOM and generalized Petviashvili methods do not converge due to the presence of that non-invariance-induced eigenfunction in the kernel of $\mathbf{L}_1$ [6,8,9]. To illustrate, the error diagram of the MSOM (with $c = 2, \Delta t = 0.8$) is shown in Fig. 3.6(d). It is seen that the error levels off with the number of iterations, indicating non-convergence. However, the Newton-CG method can converge very quickly. The error diagram of the CG method (with $c = 2$) is also plotted in Fig. 3.6(d). It is seen that the error drops below 10^{-10} in 65 total CG iterations (seven Newton iterations). Thus the Newton-CG method can converge regardless the kernel structure of the linearization operator $\mathbf{L}_1$. This is a notable advantage over its peers.

Besides the above six physical models, we have applied the Newton-CG method to a number of other wave systems as well, including the coupled NLS equations and the second-harmonic generation system whose solitary waves have been computed in [9] before (for the latter model, Eq. (143) in [9] is multiplied by 2 so that the resulting linearization operator $\mathbf{L}_1$ is self-adjoint and hence the Newton-CG method can apply). In addition, we have applied the Newton-CG method to compute gap vortex solitons in the 2D NLS Eq. (3.15) with periodic potentials under either self-focusing or self-defocusing nonlinearity [40]. For all the wave equations we tested, the Newton-CG method always converged as long as the initial condition is reasonably close to the exact solution (no breakdown or divergence ever happened). In addition, the Newton-CG method always converged faster or much faster than its peers. Thus the Newton-CG method proves to be a very robust and most efficient numerical method for computing both the ground-state and excited-state solitary waves in nonlinear systems if the linearization operator $\mathbf{L}_1$ is self-adjoint.

4. Comparison between the Newton-CG method and nonlinear conjugate-gradient methods

For solitary wave computations in Eq. (2.1), instead of the above Newton-CG method which is based on linear conjugate-gradient iterations to the linear Eq. (2.5), one can also apply nonlinear conjugate-gradient iterations to the original nonlinear Eq. (2.1) directly. Nonlinear conjugate-gradient methods have been developed for nonlinear optimization problems already in the literature [17,25]. Those methods can be easily extended to solitary wave computations in Eq. (2.1). The extended algorithm (with preconditioning) is described in Appendix B. In this section, we compare this nonlinear conjugate-gradient (nonlinear-CG) method with the above Newton-CG method, and demonstrate that the Newton-CG method is much more robust and also faster than the nonlinear-CG method.

We have compared the Newton-CG and nonlinear-CG methods on various examples of solitary wave computations (including those in the previous section), and the results are all similar. Thus we only display below the comparison results for the gap soliton in Fig. 3.5(b) for the 2D NLS equation with periodic potentials (3.16). In the nonlinear-CG method, the preconditioning (acceleration) operator $\mathbf{M}$ is taken as (3.7) with $c = 3$ as in the Newton-CG method. The computational domain as well as the number of grid points are taken the same as in the Newton-CG method as well. However, if the initial condition (3.19) of the Newton-CG method is used, we find that the nonlinear-CG method does not converge at all. Further numerical testing shows that the convergence or divergence of the nonlinear-CG method is very sensitive to the initial condition, even when the initial condition is very close to the exact solution. To demonstrate, let us take the initial condition as the exact solution plus a perturbation,

$$u_0(x) = u(x) + \epsilon \mathrm{sech}\sqrt{x^2 + y^2}\ \cos\ x\ \cos\ y, \tag{3.22}$$

where $u(x)$ is the exact gap soliton, and ϵ is the strength of perturbation. For this form of initial condition, we find that the nonlinear-CG method does not converge when $\epsilon = 0.001$, but does converge at a slightly different value of $\epsilon = 0.0011$. These behaviors can be seen clearly from the error diagrams of these two cases in Fig. 4.1(a and b). This sensitivity to initial conditions indicates that the nonlinear-CG method is not a robust numerical method for solitary wave computations. The Newton-CG method, on the other hand, is very robust. Indeed, for the type of initial conditions (3.22), the Newton-CG method

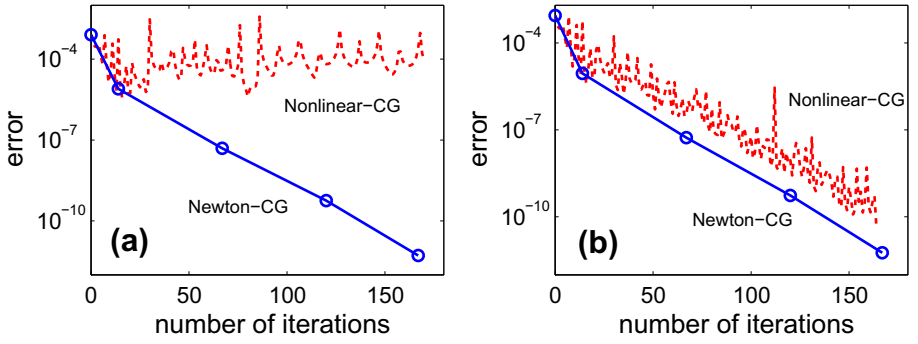


Fig. 4.1. Error diagrams of the Newton-CG method and the nonlinear-CG method for the computation of the gap soliton in Fig. 3.5(b). The initial condition is (3.22) with $\epsilon = 0.001$ in (a) and $\epsilon = 0.0011$ in (b).

converges for ϵ anywhere in the wide interval of $-0.15 < \epsilon < 0.23$. For instance, at the above two ϵ values, the error diagrams of the Newton-CG method are also plotted in Fig. 4.1(a and b) for comparison. The reason for the sensitivity of the nonlinear-CG method and the robustness of the Newton-CG method appears to be that, in the nonlinear-CG method, since the solution is frequently updated, this makes it difficult for successive search directions to maintain conjugacy, which causes the method to degrade. But in the Newton-CG method, each time Eq. (2.5) is solved by CG iterations, the linear operator $\mathbf{L}_{1n}$ and the inhomogeneous term $\mathbf{L}_0\mathbf{u}_n$ are fixed, hence the successive search directions of these conjugate-gradient iterations can remain conjugate to each other, so the benefit of the conjugate-gradient method can be fully realized. Fig. 4.1 also shows that even when the nonlinear-CG method does converge, its error diagram exhibits strong oscillations, but that of the Newton-CG method is monotone (it should be said that the inner linear conjugate-gradient iterations in the Newton-CG method for solving Eq. (2.5) also show oscillations in the residue of that equation since $\mathbf{L}_1$ is indefinite, but such oscillations are much weaker, and they usually do not cause oscillations in the error of the outer Newton iterations as plotted in Fig. 4.1). Additionally, Fig. 4.1 shows that the nonlinear-CG method takes more iterations than the Newton-CG method to reach the same accuracy. Recalling that each linear CG iteration in the Newton-CG method requires only two operator evaluations ($\mathbf{L}_1$ and $\mathbf{M}^{-1}$), while each nonlinear-CG iteration requires three operator evaluations ($\mathbf{L}_0, \mathbf{L}_1$ and $\mathbf{M}^{-1}$), we see that the Newton-CG method would be at least 50% faster than the nonlinear-CG method (if the latter method can converge). Our conclusion is that when applying conjugate-gradient ideas to solitary wave computations, the Newton-CG method we proposed in the previous section is much better than the nonlinear conjugate-gradient methods.

5. The preconditioned biconjugate-gradient method for non-self-adjoint linearization operators $\mathbf{L}_1$

For some solitary wave equations, the linearization operator $\mathbf{L}_1$ is not self-adjoint. In such cases, the conjugate-gradient iterations on Eq. (2.5) usually do not converge. In matrix computations, when the matrix is not symmetric, various extensions of the conjugate-gradient method have been developed [21–24]. Among those extended methods, we pick out the biconjugate-gradient (BCG) method for its efficiency and easy implementation [22]. In this method, two sequences of residues and search directions are updated using both the matrix and its transpose. Extending the BCG method to Eq. (2.5) and incorporating preconditioning, the preconditioned BCG iterations are

$$
\begin{aligned}
&\Delta\mathbf{u}^{(0)} = 0,\\
&\mathbf{R}^{(0)} = \tilde{\mathbf{R}}^{(0)} = -\mathbf{L}_0\mathbf{u},\\
&\mathbf{D}^{(0)} = \tilde{\mathbf{D}}^{(0)} = \mathbf{M}^{-1}\mathbf{R}^{(0)},\\
&a^{(i)} = \frac{\langle \tilde{\mathbf{R}}^{(i)}, \mathbf{M}^{-1}\mathbf{R}^{(i)}\rangle}{\langle \tilde{\mathbf{D}}^{(i)}, \mathbf{L}_1\mathbf{D}^{(i)}\rangle},\\
&\Delta\mathbf{u}^{(i+1)} = \Delta\mathbf{u}^{(i)} + a^{(i)}\mathbf{D}^{(i)},\\
&\mathbf{R}^{(i+1)} = \mathbf{R}^{(i)} - a^{(i)}\mathbf{L}_1\mathbf{D}^{(i)},\\
&\tilde{\mathbf{R}}^{(i+1)} = \tilde{\mathbf{R}}^{(i)} - a^{(i)}\mathbf{L}_1^{\dagger}\tilde{\mathbf{D}}^{(i)},\\
&b^{(i+1)} = \frac{\langle \tilde{\mathbf{R}}^{(i+1)}, \mathbf{M}^{-1}\mathbf{R}^{(i+1)}\rangle}{\langle \tilde{\mathbf{R}}^{(i)}, \mathbf{M}^{-1}\mathbf{R}^{(i)}\rangle},\\
&\mathbf{D}^{(i+1)} = \mathbf{M}^{-1}\mathbf{R}^{(i+1)} + b^{(i+1)}\mathbf{D}^{(i)},\\
&\tilde{\mathbf{D}}^{(i+1)} = \mathbf{M}^{-1}\tilde{\mathbf{R}}^{(i+1)} + b^{(i+1)}\tilde{\mathbf{D}}^{(i)}.
\end{aligned}
$$

Here $\mathbf{L}_1^{\dagger}$ is the adjoint operator (or more generally the Hermitian operator) of $\mathbf{L}_1$, and $\mathbf{M}$ is a self-adjoint and positive-definite preconditioning operator. When these BCG iterations are embedded inside the Newton iterations (2.4), the resulting method will be called the Newton-BCG method for solitary waves in Eq. (2.1).

Notice that one BCG iteration above involves four operator evaluations ($\mathbf{L}_1, \mathbf{L}_1^{\dagger}$, and two $\mathbf{M}^{-1}$), which doubles that of one conjugate-gradient iteration. If the linear operator $\mathbf{L}_1$ is self-adjoint, then these BCG iterations become the same as the conjugate-gradient iterations of Section 3, but at twice the cost per iteration.

These BCG iterations for Eq. (2.5) need to be stopped when the approximate solution $\Delta\mathbf{u}_n^{(i)}$ has reached certain accuracy. For the same reasons as explained in Section 3, the stopping criterion for these BCG iterations will be taken the same as (3.3) [or equivalently (3.4)] for conjugate gradient iterations, and the error tolerance parameter will be taken to be $\epsilon_{cg} = 10^{-2}$ as well in our examples below.

Similar to the Newton-CG method, the Newton-BCG method might also break down in theory since the updating formulae for $a^{(i)}$ and $b^{(i+1)}$ during BCG iterations may encounter zero denominators. But as with the Newton-CG method, we did not encounter this breakdown in our various solitary wave computations by the Newton-BCG method. For instance, in the example to be shown below, we tried many different initial conditions near the exact solution, and in all cases the Newton-BCG method quickly converged. Thus the Newton-BCG method proves to be a robust and efficient numerical method for computing solitary waves when the linearization operator $\mathbf{L}_1$ is non-self-adjoint.

Example 5.1 (*Depression and elevation waves in the fifth-order KP equation*). As an example of application of the Newton-BCG method, we consider the fifth-order KadomtsevPetviashvili (KP) equation

$$(U_t + 6UU_x + 2U_{xxx} + U_{xxxxx})_x + U_{yy} = 0, \tag{5.1}$$

which is a normalized model for two-dimensional small-amplitude gravity-capillary waves on water of finite depth when the Bond number is close to 1/3 [41]. Here $U(x,y,t)$ is the non-dimensionalized free-surface elevation. Looking for traveling solitary wave solutions of the form $U(x,y,t) = u(x - vt, y)$, where v is the wave's velocity, then the function $u(x,y)$ satisfies the equation

$$(u_{xxxx} + 2u_{xx} + 3u^2 - vu)_{xx} + u_{yy} = 0. \tag{5.2}$$

This equation admits two-dimensional depression and elevation waves which decay oscillatorily along the x-direction and monotonically along the y-direction when $v < -1$, and these waves bifurcate from the minimum phase speed point of $v_{min} = -1$ [41]. At $v = -1.2$, the corresponding depression and elevation waves are shown in Fig. 5.1(a and b), respectively. Note that for each value of $v < -1$, Eq. (5.2) also admits a continuous family of solutions which approach a constant as $(x,y) \to \infty$. Thus to seek solitary waves in this equation, we need to introduce a constraint which can be taken as

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(x,y)\,dxdy = 0. \tag{5.3}$$

For Eq. (5.2), the linearization operator

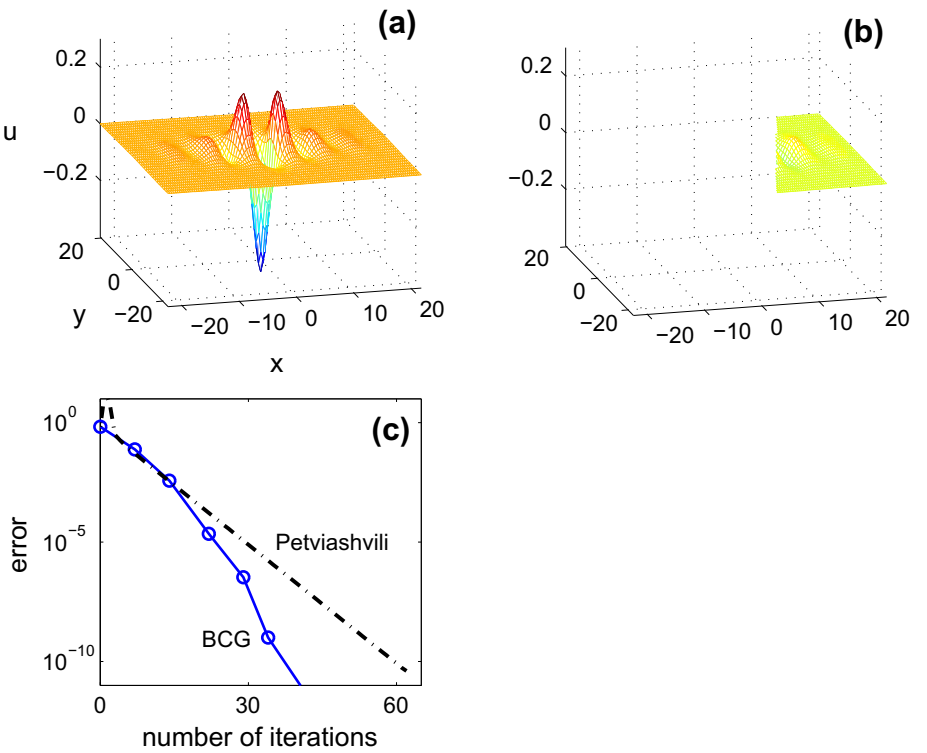
$$\mathbf{L}_1\psi = (\psi_{xxxx} + 2\psi_{xx} + 6u\psi - v\psi)_{xx} + \psi_{yy} \tag{5.4}$$

is not self-adjoint, thus we need to use the Newton-BCG method to compute its solitary waves. To obtain the depression and elevation waves in Fig. 5.1(a and b) with $v = -1.2$, we take the spatial domain as $-60\pi < x < 60\pi$, $-30\pi < y < 30\pi$, discretized by 1024 and 128 grid points along the x and y-directions, respectively. The acceleration operator $\mathbf{M}$ is taken as

$$\mathbf{M} = c - \partial_{xx}(\partial_{xxxx} + 2\partial_{xx} - v), \tag{5.5}$$

with $c = 0.0001$. The initial condition for the depression wave is taken as

$$u_0(x,y) = -0.43\ \mathrm{sech}\ 0.3\sqrt{x^2 + y^2}\cos\ x, \tag{5.6}$$



while that for the elevation wave is taken as $-u_0(x,y)$. The constraint (5.3) is imposed by setting the $(k_x,k_y)=0$ Fourier component of the solution $u_n(x,y)$ to be zero after each Newton iteration (2.4). The error diagrams for these depression and elevation waves are displayed in Fig. 5.1(c and d).

As with the previous Newton-CG method, the error diagram of the Newton-BCG method here also plots the error of the numerical solution after each Newton's iteration against the total number of preceding BCG iterations (by circle points), and these circle points are connected by straight lines for illustration purpose.

One can see that to reach accuracy 10^{-10}, the Newton-BCG method takes only a total of 39 BCG iterations (six Newton iterations) for the depression wave and 53 total BCG iterations (seven Newton iterations) for the elevation wave.

Now we compare the Newton-BCG method with its peers. For this depression wave, the Petviashvili method also converges. With the same initial condition (5.6), the error diagram of this method is also shown in Fig. 5.1(c). To reach accuracy 10^{-10}, the Petviashvili method takes 62 iterations. Noticing that one BCG iteration involves about twice the amount of computations as one Petviashvili iteration, we find that the Newton-BCG and Petviashvili methods take about the same CPU times (15 s) on this example. However, if the wave speed v is closer to the minimum phase speed $v_{min}=-1$ (where the wave is less localized), the Newton-BCG method will be faster than the Petviashvili method analogously to Example 3.4. For the elevation wave in Fig. 5.1(b), the Petviashvili method does not converge. Thus we use the MSOM on this example. The acceleration operator $\mathbf{M}$ is taken the same as Eq. (5.5), and the constraint (5.3) is imposed the same way as in the BCG method. Our numerical testings show that the MSOM is extremely slow on this example. For instance, with $c=0.01$ and $\Delta t=0.025$, the error diagram of the MSOM is shown in Fig. 5.1(d). We find that the error is still on the order of 10^{-4} even after thousands of iterations. Thus the Newton-BCG method is much faster than the MSOM on this example.

6. Summary and discussion

In this paper, we proposed conjugate-gradient-type methods for solitary wave computations. Our schemes are based on Newton outer iterations (2.4), coupled with inner conjugate-gradient iterations to solve the linear Eq. (2.5). When the linearization operator $\mathbf{L}_1$ is self-adjoint, we used the preconditioned conjugate gradient method to solve this linear Eq. (2.5). If $\mathbf{L}_1$ is non-self-adjoint, we used the preconditioned biconjugate-gradient method to solve this linear equation. The resulting Newton-CG and Newton-BCG methods were applied to compute both the ground states and excited states in various physical systems such as the two-dimensional NLS equations with and without periodic potentials, the fifth-order KdV equation, the fifth-order KP equation, and many others. The numerical results showed that these proposed methods are very robust and always converge without breakdowns. More importantly, these methods are faster than the other leading numerical methods, often by orders of magnitude. In addition, these methods are very easy to implement in arbitrary spatial dimensions. Furthermore, we showed that the nonlinear conjugate-gradient methods are not robust and thus inferior to these proposed methods.

We would like to mention that even though these proposed methods were found to be very robust and never break down in our extensive numerical testings on various examples and initial conditions, the risk of their breakdown still exists. If this breakdown does occur, it can be avoided by changing the initial guess function. Another potential risk is that these methods may not converge even if there is no breakdown. This scenario was never observed in our numerous testings of these methods, but this risk may exist since there is no theoretical guarantee for the convergence of conjugate-gradient or biconjugate-gradient iterations in the absence of breakdowns if the matrix is symmetric indefinite or non-symmetric. If one wishes for a conjugate-gradient-type method with a guaranteed convergence, a simple option is to multiply the linear Eq. (2.5) by $\mathbf{L}_{1n}^{\dagger}$ and turn it into a normal equation. The linear operator $\mathbf{L}_{1n}^{\dagger}\mathbf{L}_{1n}$ of this normal equation is self-adjoint and semi-positive-definite, thus it can be solved by the preconditioned conjugate-gradient iterations. This method will be slower than the Newton-CG or Newton-BCG method, but it does offer guaranteed convergence if the reader needs it. Another idea which was pursued in [42] is to incorporate mode elimination [14] into the nonlinear conjugate-gradient methods in order to guarantee the convergence of nonlinear conjugate-gradient methods for ground-state solitary waves. This idea was met with limited success. But it only applies to the computation of ground states. In addition, it is about twice slower than the Newton-CG/BCG methods of this paper.

Acknowledgments

The author thanks Dr. T.I. Lakoba for helpful discussions. He also thanks three anonymous referees for useful suggestions which significantly improved the clarify of this manuscript. This work was supported in part by the Air Force Office of Scientific Research.

Appendix A. MATLAB code of the newton-CG method for Example 3.5

Below is a sample MATLAB code of the Newton-CG method which computes the gap soliton of Fig. 3.5(b) in Example 3.5.

```
Lx=20*pi; Ly=20*pi; N=256; errormax=1e-10; errorCG=1e-2;
x=-Lx/2:Lx/N:Lx/2-Lx/N; kx=[0:N/2-1 -N/2:-1]*2*pi/Lx;
```

```
y=-Ly/2:Ly/N:Ly/2-Ly/N; ky=[0:N/2-1 -N/2:-1]*2*pi/Ly;
[X,Y]=meshgrid(x,y); [KX,KY]=meshgrid(kx,ky); K2=KX.^2+KY.^2;
V=-6*(sin(X).^2+sin(Y).^2); c=3; mu=4.56;
U=0.56*sech(0.5*sqrt(X.^2+Y.^2)). *cos(X).*cos(Y);
ncg=0; nnt=0;
while 1
   nnt=nnt+1;
   LOU=ifft2(-K2.*fft2(U))+(V-U.*U+mu).*U;
   errorU(nnt)=max(max(abs(LOU)));
   iterations(nnt)=ncg;
   errorU(nnt)
   if errorU(nnt) < errormax
     break
   end
   DU=0*X;
   R=-LOU; MinvR=ifft2(fft2(R)./(K2+c));
   R2=sum(sum(R.*MinvR)); R20=R2;
   D=MinvR;
   while (R2 > R20*errorCG^2)
      L1D=ifft2(-K2.*fft2(D))+(V-3*U.*U+mu).*D;
      a=R2/sum(sum(D.*L1D));
      DU=DU+a*D;
      R=R-a*L1D; MinvR=ifft2(fft2(R)./(K2+c));
      R2old=R2;
      R2=sum(sum(R.*MinvR));
      b=R2/R2old;
      D=MinvR+b*D;
      ncg=ncg+1;
   end
   U=U+DU;
end
figure(1); mesh(x,y,U); xlabel('x'); ylabel('y'); zlabel('U');
figure(2); semilogy(iterations, errorU);
xlabel('number of iterations'); ylabel('error')
```

Appendix B. Preconditioned nonlinear conjugate-gradient methods for solitary waves

The linear conjugate-gradient method was developed for solving systems of linear equations. To solve systems of nonlinear equations, nonlinear conjugate-gradient methods have been developed in the literature [17,25]. These nonlinear conjugate-gradient methods can be easily extended to compute solitary waves in Eq. (2.1). With preconditioning incorporated, the outline of the preconditioned nonlinear conjugate gradient method for Eq. (2.1) is

$$
\begin{aligned}
&\mathbf{R}_0 = -\mathbf{L}_0\mathbf{u}_0,\\
&\mathbf{D}_0 = \mathbf{M}^{-1}\mathbf{R}_0,\\
&\text{Find } a_n \text{ so that } \langle \mathbf{D}_n, \mathbf{L}_0(\mathbf{u}_n + a_n\mathbf{D}_n)\rangle = 0,\\
&\mathbf{u}_{n+1} = \mathbf{u}_n + a_n\mathbf{D}_n,\\
&\mathbf{R}_{n+1} = -\mathbf{L}_0\mathbf{u}_{n+1},\\
&b_{n+1} = \frac{\langle \mathbf{R}_{n+1}, \mathbf{M}^{-1}\mathbf{R}_{n+1}\rangle}{\langle \mathbf{R}_n, \mathbf{M}^{-1}\mathbf{R}_n\rangle},\\
&\mathbf{D}_{n+1} = \mathbf{M}^{-1}\mathbf{R}_{n+1} + b_{n+1}\mathbf{D}_n.
\end{aligned}
$$

Here the updating formula for b_{n+1} is the Fletcher-Reeves formula. An alternative formula for b_{n+1} is the Polak-Ribière formula [17]. Our numerical testings on these two formulas show similar results, thus the slightly simpler Fletcher-Reeves formula is used above.

Regarding the formula for a_n, if $\mathbf{u}_n$ is close to the exact solution $\mathbf{u}$ (i.e. the error term $a_n\mathbf{D}_n$ is small), then we can linearize the inner product for a_n around $\mathbf{u}_n$ and hence obtain an approximate formula for a_n as

$$
a_n = \frac{\langle \mathbf{D}_n, \mathbf{R}_n\rangle}{\langle \mathbf{D}_n, \mathbf{L}_{1n}\mathbf{D}_n\rangle}, \tag{B.1}
$$

where $\mathbf{L}_{1n}$ is the linearization operator of (2.1) evaluated at $\mathbf{u}_n$ (as in the main text). If $\mathbf{u}_n$ is not close to the exact solution $\mathbf{u}$, it may be necessary to obtain a_n iteratively by the Newton's method on the above inner product equation for a_n. In that case, the resulting nonlinear-CG method would be a loop-within-loop operation. But here the Newton's iteration is the inner loop, which contrasts the Newton-CG method where the Newton iteration is the outer loop.

In the comparison of Section 4, the a_n formula (B.1) was used in the nonlinear-CG method. The resulting algorithm was found to be sensitive to initial conditions, even when the initial condition is very close to the exact solution. If a_n is obtained by Newton's iterations, the resulting nonlinear-CG method would still be sensitive to initial conditions (even when they are close to the exact solution), and thus is still not a robust numerical method for solitary wave computations.

References

[1] J.P. Boyd, Chebyshev and Fourier Spectral Methods, second ed., Dover Publications, 2001.
[2] J.P. Boyd, Deleted residuals, the QR-factored Newton iteration, and other methods for formally overdetermined determinate discretizations of nonlinear eigenproblems for solitary, cnoidal, and shock waves, J. Comput. Phys. 179 (2002) 216–237.
[3] J. Yang, Internal oscillations and instability characteristics of (2 + 1) dimensional solitons in a saturable nonlinear medium, Phys. Rev. E 66 (2002) 026601.
[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, Accelerated imaginary-time evolution methods for the computation of solitary waves, Stud. Appl. Math. 120 (2008) 265–292.
[9] 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.
[10] 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.
[11] J. Yang, B.A. Malomed, D.J. Kaup, Embedded solitons in second-harmonic-generating systems, Phys. Rev. Lett. 83 (1999) 1958.
[12] D.E. Pelinovsky, Y.A. Stepanyants, Convergence of Petviashvili's iteration method for numerical approximation of stationary solutions of nonlinear wave equations, SIAM J. Numer. Anal. 42 (2004) 1110.
[13] M.L. Chiofalo, S. Succi, M.P. Tosi, Ground state of trapped interacting Bose–Einstein condensates by an explicit imaginary-time algorithm, Phys. Rev. E 62 (2000) 7438.
[14] T.I. Lakoba, J. Yang, A mode elimination technique to improve convergence of iteration methods for finding solitary waves, J. Comp. Phys. 226 (2007) 1693–1709.
[15] M.R. Hestenes, E. Stiefel, Methods of conjugate-gradients for solving linear systems, J. Res. Natl. Bur. Stand. 49 (1952) 409–436.
[16] G. Golub, C. Van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, 1996.
[17] J. Shewchuk, An Introduction to the Conjugate Gradient Method without the Agonizing Pain, Carnegie Mellon Univerisity, Technical Report CMU-CS-94-125, 1994. Available at <www.cs.cmu.edu/quake-papers/painless-conjugate-gradient.pdf>).
[18] V.S. Ryaben'kii, S.V. Tsynkov, A Theoretical Introduction to Numerical Analysis, Chapman and Hall/CRC Press, Boca Raton, 2007.
[19] H. Igarashi, T. Honma, Convergence of preconditioned conjugate gradient method applied to driven microwave problems, IEEE Trans. Magn. 39 (2003) 1705–1708.
[20] C.C. Paige, M.A. Saunders, Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12 (1975) 617–624.
[21] Y. Saad, M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[22] R. Barret, M. Berry, T.F. Chan, J. Demmel, J.M. Donato, J. Dongarra, V. Eijihout, R. Pozo, C. Romine, H. Van der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, 1994.
[23] R. Freund, N. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, Numer. Math. 60 (1991) 315–339.
[24] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 10 (1989) 36–52.
[25] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients, Comput. J. 7 (1964) 149–154.
[26] R.S. Tuminaro, H.F. Walker, J.N. Shadid, On backtracking failure in Newton-GMRES methods with a demonstration for the Navier–Stokes equations, J. Comput. Phys. 180 (2002) 549–558.
[27] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2004) 357–397.
[28] E. Turkel, Numerical difficulties solving time harmonic equations, in: A. Brandt, J. Bernholc, K. Binder (Eds.), Multiscale Computational Methods in Chemistry and Physics, IOS Press, Ohmsha, 2001, pp. 319–337.
[29] G. Markham, Conjugate gradient type methods for indefinite, asymmetric, and complex systems, IMA J. Numer. Anal. 10 (1990) 155–170.
[30] L.M. Delves, T.L. Freeman, Analysis of Global Expansion Methods: Weakly Asymptotically Diagonal Systems, Academic Press, New York, 1981.
[31] N. Trefethen, Spectral Method in MATLAB, SIAM, Philadelphia, 2000.
[32] D.C. Calvo, T.S. Yang, T.R. Akylas, On the stability of solitary waves with decaying oscillatory tails, Proc. R. Soc. London, Ser. A 456 (2000) 469–487.
[33] F. Dalfovo, S. Giorgini, L.P. Pitaevskii, S. Stringari, Theory of Bose–Einstein condensation in trapped gases, Rev. Mod. Phys. 71 (1999) 463–512.
[34] O. Morsch, M.K. Oberthaler, Dynamics of Bose–Einstein condensates in optical lattices, Rev. Mod. Phys. 78 (2006) 179.
[35] M. Skorobogatiy, J. Yang, Fundamentals of Photonic Crystal Guiding, Cambridge University Press, Cambridge, UK, 2009.
[36] Z. Musslimani, J. Yang, Self-trapping of light in a two-dimensional periodic structure, J. Opt. Soc. Am. B 21 (2004) 973–981.
[37] J. Yang, Z. Chen, Defect solitons in photonic lattices, Phys. Rev. E 73 (2006) 026609.
[38] 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).
[39] Y.S. Kivshar, G.P. Agrawal, Optical Solitons: From Fibers to Photonic Crystals, Academic Press, San Diego, 2003.
[40] J. Wang, J. Yang, "Families of vortex solitons in periodic media, Phys. Rev. A. 77 (2008) 033834.
[41] T.R. Akylas, Y. Cho, On the stability of lumps and wave collapse in water waves, Philos. Trans. R. Soc. London, Ser. A 366 (2008) 2761–2774.
[42] T.I. Lakoba, Conjugate gradient method for fundamental solitary waves, preprint.