Conjugate Gradient Method for finding fundamental solitary waves

T.I. Lakoba*

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

June 30, 2009

Abstract

The Conjugate Gradient method (CGM) is known to be the fastest generic iterative method for solving linear systems with symmetric sign definite matrices. In this paper, we modify this method so that it could find fundamental solitary waves of nonlinear Hamiltonian equations. The main obstacle that such a modified CGM overcomes is that the operator of the equation linearized about a solitary wave is not sign definite. Instead, it has a finite number of eigenvalues on the opposite side of zero than the rest of its spectrum. We present versions of the modified CGM that can find solitary waves with prescribed values of either the propagation constant or power. We also extend these methods to handle multi-component nonlinear wave equations. Convergence conditions of the proposed methods are given, and their practical implications are discussed. We demonstrate that our modified CGMs converge much faster than, say, Petviashvili's or similar methods, especially when the latter converge slowly.

Keywords: Conjugate Gradient method, Nonlinear evolution equations, Solitary waves, Iterative methods.

PACS: 03.75.Lm, 05.45.Yv, 42.65.Tg, 47.35.Fg.

*lakobati@cems.uvm.edu, 1 (802) 656-2610

1 Introduction

Solitary waves of most nonlinear evolution equations can be found only numerically. In one spatial dimension, a variety of numerical methods for finding solitary waves are available, with the shooting and Newton's methods being probably the most widely used ones. However, in two and three spatial dimensions, the shooting method is no longer applicable, and for Newton's method, one needs to employ an alternating-direction implicit (ADI) method to ensure that the operations count per iteration scales linearly (as opposed to quadratically or cubically) with the number of grid points. While the ADI-based Newton's method is time-efficient, the complexity of its programming is considerably higher than in the case of one spatial dimension. Also, the accuracy of the ADI-based Newton's method is only algebraic in the size of the spatial discretization step Δx (typically, it is $O\left((\Delta x)^2\right)$). This accuracy may be too low for some applications, since in two or three spatial dimensions, one tends to take larger discretization steps than in one dimension in order to limit the computational time and the size of stored numeric arrays. Therefore, it is desirable to have a method that would: (i) be easier to program than Newton's method; (ii) have the exponential accuracy in Δx, as spectral methods; and (iii) converge sufficiently fast under known conditions.

A number of such methods are available. The well-known iterative method proposed by Petviashvili [1] can be used to find stationary solitary waves of equations with power-law nonlinearity:

$$-Mu + u^p = 0\,, \qquad u(\mathbf{x}) \to 0 \quad \text{as} \quad |\mathbf{x}| \to \infty, \tag{1.1}$$

where u is the real-valued field of the solitary wave, M is a positive definite and self-adjoint differential operator with constant coefficients, and p is a constant. For example, the solitary wave of the nonlinear Schrödinger equation in D spatial dimensions,

$$\begin{gathered} iU_t + \nabla^2 U + |U|^2 U = 0\,, \qquad U(|\mathbf{x}| \to \infty) \to 0\,, \\ \nabla^2 \equiv \frac{\partial^2}{\partial x_1^2} + \cdots + \frac{\partial^2}{\partial x_D^2}\,, \end{gathered} \tag{1.2}$$

upon the substitution $U(\mathbf{x},t) = e^{i\mu t}u(\mathbf{x})$, where $u(\mathbf{x}) \in \mathbb{R}$, reduces to Eq. (1.1) with $p = 3$ and

$$M = \mu - \nabla^2\,. \tag{1.3}$$

The parameter $\mu > 0$ is referred to as the propagation constant of the solitary wave. Convergence conditions of the Petviashvili method for Eq. (1.1) were established in [2].

Studies of solitary waves in nonlinear photonic lattices and Bose–Einstein condensates motivated the interest in finding solitary waves of equations that have a more general form than (1.1):

$$-Mu + F(u,\mathbf{x}) = 0\,, \qquad u(\mathbf{x}) \to 0 \quad \text{as} \quad |\mathbf{x}| \to \infty, \tag{1.4}$$

where $F(u, \mathbf{x})$ is a real function. For example, solitary waves in nonlinear photonic lattices with Kerr nonlinearity satisfy the following equation of the form (1.5):

$$\nabla^2 u + V_0(\cos^2 x + \cos^2 y)\, u + u^3 \;=\; \mu u\,, \tag{1.5}$$

where the second term accounts for the effect of the periodic potential provided by the lattice. Various modifications of the Petviashvili method for obtaining solutions of (1.4) were proposed; see, e.g., [3, 4, 5]. However, convergence conditions of those modifications were not studied. In Ref. [6, 7], J. Yang and the present author proposed two alternative iterative methods which satisfied conditions (i)–(iii) stated before Eq. (1.1); in particular, the conditions of their convergence to fundamental (see below) [6] and both fundamental and non-fundamental solitary waves (also known as ground and excited states, respectively) [7] were given. In Refs. [8, 7], a technique referred there as mode elimination was proposed, which was shown to provide considerable acceleration of iterative methods when the later converged slowly to the respective solitary wave.

In this paper, we propose yet another family of numerical methods for finding fundamental solitary waves of Hamiltonian nonlinear evolution equations. This proposal is motivated as follows. Consider a linear system

$$A\mathbf{y} - \mathbf{b} = \mathbf{0}, \tag{1.6}$$

for the unknown vector $\mathbf{y}$, where A is a real symmetric matrix and $\mathbf{b}$ is a known vector. Then the methods proposed here are extensions of the well-known Conjugate Gradient method (CGM) for the linear system (1.6) to the nonlinear boundary-value problem (1.4), and they converge even faster than, e.g., the generalized Petviashvili method [6] accelerated by mode elimination [8].

The main part of this paper is organized as follows. In Section 2, we explain how the methods introduced in [6, 7, 8] for the nonlinear problem (1.4) are related to some well-known iterative methods of solving the linear system (1.6) with a real and symmetric matrix A. This comparative review will lead us to explaining the main issue that needs to be resolved in order to extend the CGM in such a way that it could find fundamental solitary waves. In Section 3, we present such an extension of the CGM for the case where the solitary wave of the single-component Eq. (1.4) has a prescribed value of the propagation constant μ. In Section 4, we present a modification of the CGM for finding fundamental waves of (1.4) with a prescribed value not of the propagation constant but of a quadratic invariant often referred to as power and defined as

$$P = \int u^2(\mathbf{x})\, d\mathbf{x}\,, \tag{1.7}$$

where the integration is over the entire spatial domain. Let us note that another method for finding solitary waves with a specified value of power was previously considered in, e.g., [9, 10, 11, 12], with its convergence conditions for (1.4) being established in [12]. We will refer to this method as the imaginary-time evolution method. In Section 5, we generalize the modified CGMs of Sections 3 and 4 to multi-component equations. In Section 6, we compare the performances of the generalized Petviashvili and imaginary-time evolution methods accelerated by the mode elimination technique [8, 7] with the performances of the respective modified CGMs from Sections 3–5. In Section 7 we summarize this work and mention alternative methods related to the original CGM that may also be employed for finding solitary waves. Appendix 1 contains proofs and discussions of some auxiliary results of Sections 3 and 4. Appendix 2 contains an alternative method to the modified CGM of Section 3. Appendix 3 contains a sample code of a modified CGM for a two-component equation considered in Section 6.

Before concluding this Introduction, we need to discuss what we will refer to as a *fundamental solitary wave*. In fact, we are not aware of a universal definition of this term. In many situations, one can intuitively identify a solitary wave as being fundamental if it satisfies two easily-verifiable conditions. First, it is to have one main peak, with smaller peaks possibly existing around the main one. Second, for envelope solitary waves (e.g., those satisfying the complex-valued Eq. (1.2)), the propagation constant of the fundamental wave must lie in the semi-infinite spectral bandgap; see Fig. 1 in Section 6. (Typically, only equations with periodic potentials or systems describing more than two linearly coupled waves propagating with different group velocities have more than one spectral bandgap.) Alternatively, for carrier (e.g., Korteweg–de Vries-type) solitary waves, the velocity, rather than the propagation constant, of a fundamental wave must lie in the semi-infinite gap. More rigorous characterization of fundamental waves is possible in special cases. For instance, when the spatial operator in the wave equation is the Laplacian (as, e.g., in (1.2) or (1.5)), the fundamental wave is known to be nonzero for all $\mathbf{x}$ (see, e.g., [13]); however, for a more general operator as, e.g., in the Kadomtsev–Petviashvili equation, the fundamental wave may have zeros [14]. For some equations (see, e.g., [15, 16]), it was rigorously shown that the fundamental solution minimizes a functional known as the action.

The property of an S-component fundamental solitary wave that *we will rely on in this paper* is that for many such waves, the linearized operator of the stationary equation has *no more* than S positive eigenvalues. To the author's knowledge, there is no general proof of this property even for a single-component Eq. (1.4), let alone for the S-component case. (See, however, a recent review of solitary wave stability in lattices [17].) Therefore, we simply *declare this property to be our working definition of a fundamental solitary wave* for the purpose of this paper. Again, we do *not* imply that any solitary wave that has the two intuitive properties described at the beginning of the previous paragraph also satisfies the above property about the positive eigenvalues of the corresponding linearized operator. Rather, we only say that it is only for the waves that do have the latter property that the numerical methods developed in this paper will be guaranteed to converge (with possible restrictions, as described in subsequent sections). For solitary waves whose corresponding linearized operators have more positive eigenvalues than the number of the wave's component, our modified CGMs are not guaranteed to converge. Alternative methods for finding such nonfundamental solitary waves are briefly discussed at the end of Section 7.

2 Comparative review of iterative methods for nonlinear problem (1.4) and for linear problem (1.6)

The reader who is not interested in such a review and wants to see the description of new algorithms proposed in this paper may skip to Section 3.

For reference purposes, let us rewrite (1.4) as

$$L^{(0)}u = 0\,. \tag{2.1}$$

For the purposes of this section it will suffice to consider the case of a single-component solitary wave. Consequently, here and in Sections 3 and 4, the solitary wave u is assumed to be real-valued. For a complex-valued u, one simply considers its real and imaginary parts as the components of vector $\mathbf{u}$; this case is treated in Section 5.

An important role in our analysis will be played by the linearized operator L of the nonlinear Eq. (2.1). Let u be the exact solution of (2.1), u_n be the approximation of that solution obtained at the nth iteration, and

$$u_n = u + \tilde{u}_n, \qquad \text{where } |\tilde{u}_n| \ll |u|. \tag{2.2}$$

Then

$$L^{(0)}u_n = L^{(0)}u + L\tilde{u}_n + O(|\tilde{u}_n|^2) = L\tilde{u}_n + O(|\tilde{u}_n|^2)\,. \tag{2.3}$$

For example, for the nonlinear equation (1.5), the linearized operator is

$$L = -M + V_0(\cos^2 x + \cos^2 y) + 3u^2\,. \tag{2.4}$$

In our discussion, the linear system (1.6) is a counterpart of the nonlinear equation (2.1) and matrix A is the counterpart of the linearized operator L. For Hamiltonian wave equations that give rise to (2.1), operator L is always self-adjoint, which is the counterpart of matrix A being real and symmetric. Having stated this correspondence between the nonlinear problem (2.1) and its linear counterpart (1.6), we will, from now on, refer to them interchangeably, because our statements will apply to both of them equally.

As we will explain below, the linearized forms of the iterative methods proposed in [1, 6, 7] for solving the nonlinear problems (1.1) and (1.4) are related to Richardson's method[1] (see, e.g., [18], p. 274) of solving the linear problem (1.6):

$$\mathbf{y}_{n+1} = \mathbf{y}_n + (A\mathbf{y}_n - \mathbf{b})\Delta\tau\,, \tag{2.5}$$

[1] also known as Picard's or fixed-point iterative method

The necessary condition for the iteration scheme (2.5) to converge to the solution, $\mathbf{y} = A^{-1}\mathbf{b}$, is that A be negative definite. Choosing the parameter $\Delta\tau$ in (2.5) so that $\Delta\tau < -2/\lambda_{\min}$, where $\lambda_{\min}$ is the minimum (i.e., most negative) eigenvalue of A, one ensures that Richardson's method for (1.6) with a negative definite A converges.

A naive counterpart of (2.5) for the nonlinear problem (2.1) is

$$u_{n+1} = u_n + L^{(0)} u_n \, \Delta\tau \, . \tag{2.6}$$

However, in most cases it will not converge to a solitary wave. Indeed, from (2.3), the linearized form of (2.6) is

$$\tilde{u}_{n+1} = \tilde{u}_n + L\tilde{u}_n \Delta\tau \, . \tag{2.7}$$

This equation is a counterpart of (2.5), and so a necessary condition for its convergence is that operator L be negative definite. However, for most fundamental single-component solitary waves, the corresponding linearized operator L has one positive eigenvalue (see the last paragraph of the Introduction[2]), which causes divergence of small deviations $\tilde{u}_n$ in (2.7) and, thereby, divergence of the iterations (2.6).

To overcome this divergence, the iterative methods of [1, 6, 7] replaced $L^{(0)} u_n$ in (2.6) by a modified expression $L^{(0,\mathrm{mod})} u_n$ such that: (i) $\big(L^{(0)} u = 0\big) \Leftrightarrow \big(L^{(0,\mathrm{mod})} u = 0\big)$ and (ii) the corresponding linearized operator $L^{(\mathrm{mod})}$ was guaranteed to be negative definite. For example, in [7],

$$L^{(0,\mathrm{mod})} u = -L \, L^{(0)} u, \tag{2.8}$$

so that $L^{(\mathrm{mod})} = -L^2$. In terms of the linear system (1.6) where A is real symmetric but not sign definite, this is equivalent to applying Richardson's method (2.5) to the so called normal equation

$$-A^2 \mathbf{y} = -A\mathbf{b}, \tag{2.9}$$

where the matrix on the left-hand side is always negative definite.

In [6], we proposed an alternative idea of constructing a negative definite $L^{(\mathrm{mod})}$. This idea will play an important role in the development of this paper, and here we will explain it as applied to the linear system (1.6). As noted above, for most single-component fundamental solitary waves, the linearized operator L has only one positive eigenvalue. Accordingly, let $\lambda^{(1)}$ be the only positive eigenvalue of matrix A in (1.6), with the corresponding eigenvector being $\mathbf{z}^{(1)}$. In the context of the nonlinear equation (2.1), the eigenvector of L corresponding to its only positive eigenvalue is related to the solution u of (2.1) [6], as we will explain later. Hence it can be considered as approximately known once the iterative solution u_n is sufficiently close to u, which we will always assume to be the case (see (2.2)).

[2] There are — rare — cases where fundamental waves have *no* positive eigenvalues: see, e.g., Example 2 in [12], where even the first excited state had no positive eigenvalues.

Now, instead of (1.6), consider an *equivalent*[3] system

$$\left(I - \gamma \mathbf{z}^{(1)} \left(\mathbf{z}^{(1)}\right)^T\right) (A\mathbf{y} - \mathbf{b}) = \mathbf{0}\,, \tag{2.10}$$

where I is the identity matrix and γ is any number greater than one. Note that the term $\mathbf{z}^{(1)} \left(\mathbf{z}^{(1)}\right)^T$ is the projection matrix onto the unstable direction $\mathbf{z}^{(1)}$. Using the spectral decomposition of a real symmetric $p \times p$ matrix A:

$$A = \sum_{i=1}^{p} \lambda^{(i)} \mathbf{z}^{(i)} \left(\mathbf{z}^{(i)}\right)^T\,, \tag{2.11}$$

where the eigenvectors $\mathbf{z}^{(i)}$ form an orthonormal set:

$$\left(\mathbf{z}^{(i)}\right)^T \mathbf{z}^{(j)} = \left\{ \begin{array}{ll} 1, & i = j \\ 0, & i \neq j, \end{array} \right. \tag{2.12}$$

it is straightforward to show that (2.10) and hence (1.6) is equivalent to

$$A^{(\text{mod})}\mathbf{y} - \left(I - \gamma \mathbf{z}^{(1)} \left(\mathbf{z}^{(1)}\right)^T\right) \mathbf{b} = \mathbf{0}, \tag{2.13a}$$

where

$$A^{(\text{mod})} \equiv \lambda^{(1)}(1-\gamma)\mathbf{z}^{(1)} \left(\mathbf{z}^{(1)}\right)^T + \sum_{i=2}^{p} \lambda^{(i)} \mathbf{z}^{(i)} \left(\mathbf{z}^{(i)}\right)^T\,. \tag{2.13b}$$

Since $\lambda^{(1)}(1-\gamma) < 0$ and $\lambda^{(i)} < 0$ for $i \geq 2$ by our assumptions about A and γ, the matrix $A^{(\text{mod})}$ in (2.13b) is negative definite, and hence Richardson's method (2.5) for it can converge.

Note that if one chooses [6]

$$\gamma = 1 + 1/(\lambda^{(1)}\Delta\tau) \tag{2.14}$$

and uses the resulting $A^{(\text{mod})}$ in Richardson's method (2.5), one thereby forces the projection of the error $\tilde{\mathbf{y}}_n \equiv \mathbf{y}_n - \mathbf{y}$ on the direction of the unstable eigenvector $\mathbf{z}^{(1)}$ to be zero at every iteration. This idea of zeroing out the projection of the iteration error on the unstable direction lies behind the original Petviashvili method [2] and its generalization for Eq. (1.4) [6]. This is also the idea that we will use later in this paper for the development of modified CGMs.

So far we have discussed how Richardson's method for (1.6) with a sign indefinite A can be made to converge. The next issue is, *how fast* it converges. This is quantified by the convergence factor R, such that the number of iterations required for the iteration error to decrease by a factor of e is (asymptotically) inversely proportional to log$R$. For the optimal value of $\Delta\tau$, the convergence factor of Richardson's method is (see, e.g., Sec. 5.2.2 in [19], or [7]):

$$R = \frac{\text{cond}(A^{(\text{mod})}) + 1}{\text{cond}(A^{(\text{mod})}) - 1}\,, \tag{2.15}$$

[3] The matrix on the left-hand side of (2.10) is nonsingular.

where the condition number is related to the eigenvalues of $A^{(\mathrm{mod})}$:

$$\mathrm{cond}(A^{(\mathrm{mod})}) = \lambda_{\min}/\lambda_{\max} \equiv |\lambda_{\min}|/|\lambda_{\max}| \tag{2.16}$$

(recall that all the eigenvalues of $A^{(\mathrm{mod})}$ are negative and so $\mathrm{cond}(A^{(\mathrm{mod})}) > 1$). For $\mathrm{cond}(A) \gg 1$, convergence to a prescribed accuracy occurs in $O(1/\log(R)) = O(\mathrm{cond}(A))$ iterations. That is, the greater the condition number, the slower the convergence of the iterative method. A common method to reduce the condition number is to use a preconditioner, which can drastically lower $|\lambda_{\min}|$ (see, e.g., Lecture 40 in [18]). The preconditioned Richardson's method is

$$\mathbf{y}_{n+1} = \mathbf{y}_n + B^{-1}(A\mathbf{y}_n - \mathbf{b})\Delta\tau\,, \tag{2.17}$$

where B is the preconditioning matrix[4]. For the Richardson-type method (2.6) with $L^{(0)}$ replaced by $L^{(0,\mathrm{mod})}$, a convenient preconditioning operator, N, has a form similar to that of M. For example, if M is as in (1.3), the convenient form for N is:

$$N = c - \nabla^2, \qquad c > 0\,, \tag{2.18}$$

where c is some constant.

Even if the condition number is lowered by reducing $|\lambda_{\min}|$ via preconditioning, the convergence of an iterative method may still be slow due to $|\lambda_{\max}|$ being close to zero. In the context of the nonlinear problem (1.4), examplified by (1.5), this occurs when the propagation constant μ is close to the edge of the spectral bandgap of the linear potential (i.e., $V_0(\cos^2 x + \cos^2 y)$ in (1.5)); see, e.g., Fig. 4(a) in [20] or Fig. 1 in Section 6 below. In [8, 7] we proposed to accelerate the iterations by eliminating the slowest-decaying eigenmode, i.e., the component of the error $\tilde{u}_n$ that corresponds to the eigenvalue $\lambda_{\max}$. This effectively removes this slowest eigenmode from the spectrum of $L^{(\mathrm{mod})}$ and thereby increases the magnitude of its maximum eigenvalue. This, in its turn, increases the convergence rate of the method via (2.16) and (2.15). The slowest mode can be eliminated in exactly the same way as the unstable mode (i.e., the mode with the positive eigenvalue $\lambda^{(1)}$) in (2.10). The slowest mode can be approximated by [8]

$$\mathbf{z}^{\mathrm{slow}} \propto \mathbf{y}_n - \mathbf{y}_{n-1}\,, \tag{2.19}$$

and a preconditioned Richardson's method is then applied not to (1.6) but to an equivalent equation (see the footnote before (2.10))

$$\left(I - \gamma\mathbf{z}^{(1)}\left(\mathbf{z}^{(1)}\right)^T - \gamma^{\mathrm{slow}}\mathbf{z}^{\mathrm{slow}}\left(\mathbf{z}^{\mathrm{slow}}\right)^T\right)(A\mathbf{y} - \mathbf{b}) = \mathbf{0}\,, \tag{2.20}$$

where γ^{slow} is computed similarly to (2.14). The explicit form of a counterpart of (2.20) for the nonlinear problem (2.1) will be given in Section 6.

[4] For example, the well-known Jacobi and Gauss–Seidel iterative methods reduce to (2.17) for appropriate choices of B.

In [6, 7] we demonstrated that when the convergence of an iterative method is slow, the slowest mode elimination technique described above can considerably (by a factor of five to ten times) accelerate the convergence. However, it is well known (see, e.g., [18], p. 341 and [19], p. 451) that the fastest generic method for a linear system (1.6) with a real symmetric and sign definite matrix A is the Conjugate Gradient method (CGM), whose algorithm is given by Eqs. (3.1) of the next section. The convergence factor of the CGM satisfies (see, e.g., [18], p. 299):

$$R \sim \frac{\sqrt{\mathrm{cond}(A)}+1}{\sqrt{\mathrm{cond}(A)}-1}\,, \tag{2.21}$$

which for matrices with large condition numbers implies considerably faster convergence than does (2.15). Therefore, if one could modify the CGM so that it would be guaranteed to converge when A (or the linearized operator L) is not negative definite but has one positive eigenvalue, then such a modified CGM is expected to be faster than a Richardson-type method accelerated by mode elimination.

In this paper we present several versions of such a modified CGM for nonlinear problem (2.1) and its generalizations and compare them with the Richardson-type methods of [6, 12] accelerated by mode elimination. We find that, as expected, the modified CGMs are the faster of these two groups of methods. As we noted after Eq. (2.14), the main idea behind all these versions of the modified CGM is to eliminate the component of the error $\tilde{u}_n$ that is aligned along certain unstable eigenmode(s) of the linearized iteration operator. Our modified CGMs are guaranteed to converge in almost all situations where a fundamental solitary wave *with a prescribed (set of) propagation constant(s)* is sought. Moreover, even for those rare situations where our guarantee is formally void, we will show a simple and practical way to still make the corresponding modified CGM converge. For the situations where a solitary wave *with a prescribed (set of) power(s)* is sought, our modified CGM is guaranteed to converge in the same parameter space where the Richardson-type method [12, 21] converges.

3 Modified CGM for solitary waves with prescribed propagation constant

In this section, we will first state the algorithm of the CGM for the linear Eq. (1.6). Then, we will briefly review the generalized Petviashvili method [6] for solitary waves and re-cast it in a form more convenient for the present analysis. This will prepare the ground for the key step of our modification of the CGM to the problem of finding fundamental solitary waves with a prescribed value of the propagation constant. Finally, we will present the correspondingly modified CGM. All analysis in this section is done for the single-component case; its generalization for multi-component solitary waves is presented in Section 5.

3.1 CGM for the linear Eq. (1.6)

The original CGM algorithm for Eq. (1.6) with a real symmetric and sign definite matrix A and starting with an initial guess $\mathbf{y}_0$, is:

$$\mathbf{r}_0 = A\mathbf{y}_0 - \mathbf{b}, \qquad \mathbf{d}_0 = \mathbf{r}_0, \tag{3.1a}$$

$$\alpha_n = -\frac{\langle \mathbf{r}_n,\, \mathbf{d}_n \rangle}{\langle A\mathbf{d}_n,\, \mathbf{d}_n \rangle}, \tag{3.1b}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \alpha_n \mathbf{d}_n, \tag{3.1c}$$

$$\mathbf{r}_{n+1} = A\mathbf{y}_{n+1} - \mathbf{b}, \tag{3.1d}$$

$$\beta_n = -\frac{\langle \mathbf{r}_{n+1},\, A\mathbf{d}_n \rangle}{\langle A\mathbf{d}_n,\, \mathbf{d}_n \rangle}, \tag{3.1e}$$

$$\mathbf{d}_{n+1} = \mathbf{r}_{n+1} + \beta_n \mathbf{d}_n\,. \tag{3.1f}$$

Here $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the natural inner product between the real-valued vectors $\mathbf{a}$ and $\mathbf{b}$. Equations (3.1a) define the initial residual vector $\mathbf{r}_0$ and the initial search direction $\mathbf{d}_0$. Equation (3.1c) updates the solution by adding to the previous solution a vector $\alpha_n \mathbf{d}_n$ along the search direction $\mathbf{d}_n$. The value of α_n is set by (3.1b) to guarantee the orthogonality of the new residual, defined by (3.1d), to the search direction $\mathbf{d}_n$:

$$\langle \mathbf{r}_{n+1},\, \mathbf{d}_n \rangle = 0\,. \tag{3.2}$$

Finally, Eq. (3.1f) updates the search direction, where β_n is set by (3.1e) so that

$$\langle \mathbf{d}_{n+1},\, A\mathbf{d}_n \rangle = 0\,. \tag{3.3}$$

The condition that A be real symmetric is inherently required for the derivation of algorithm (3.1). In particular, it guarantees that the orthogonality relations (3.2) and (3.3) between quantities at two *consecutive* iterations imply more general orthogonality relations among such quantities at *all* iterations (see, e.g., [22] or [18], p. 295):

$$\langle \mathbf{r}_{n+1},\, \mathbf{d}_j \rangle = 0\,, \quad j \le n\,, \tag{3.4}$$

$$\langle \mathbf{d}_{n+1},\, A\mathbf{d}_j \rangle = 0\,, \quad j \le n\,. \tag{3.5}$$

The orthogonality relation (3.5) with a *sign definite* A ensures that the CGM at each iteration produces a search direction that is linearly independent from all previous search directions. This implies that the error (in a certain norm) decreases with each iteration; this fact can be restated by saying that the CGM is an optimal iterative method for Eq. (1.6) with a sign definite matrix A (see, e.g., [18], p. 296).

The condition that A be sign definite is also needed to guarantee that the denominators of α_n and β_n never vanish (or, for practical purposes, never become too close to zero). Thus, applying the CGM to an equation with a sign indefinite matrix A would come without the guarantee that this method would be optimal and that it would even converge. However, it should be emphasized that the CGM *may* converge when applied to problem (1.6) with a sign indefinite matrix A (see, e.g., [22], or [18], p. 301). This is in stark contrast to Richardson's method (2.5), which, for a generic initial guess $\mathbf{y}_0$, is guaranteed to diverge in such a case.

The CGM (3.1) is straightforwardly extended to solve the nonlinear equation (2.1) whose linearized operator L is self-adjoint, by replacing $A\mathbf{y}_j - \mathbf{b}$ with $L^{(0)}u_j$ ($j = 0$ or $n+1$) in (3.1a) and (3.1d) and A with L in (3.1b) and (3.1e). In fact, for the CGM applied to a nonlinear problem, several choices of parameter β_n that are equivalent in the linear approximation, will produce different versions of the method. Two most widely used such versions are known as Fletcher–Reeves and Polak–Ribière (see, e.g., [23]). In the following we will not attempt to compare these two versions because the focus of this paper is *not* on the extension of the linear algorithm (3.1) to the nonlinear case but on the extension of that algorithm to the case when the counterpart of matrix A has one eigenvalue of the opposite sign than the rest of the spectrum.

Let us also note that, unlike the algorithm (3.1) for linear equations, its extension to nonlinear problems described in the previous paragraph may fail if the linearized operator K is negative definite but has one or more eigenvalues close to zero. This can occur if, at some iteration, the search direction $\mathbf{d}_n$ becomes aligned primarily along the eigenmode(s) corresponding to the small eigenvalue(s). Then the denominator in (3.1b) will be small and hence the increment $\alpha_n\mathbf{d}_n$ of the solution in (3.1c), large. If the nonlinear equation admits more than one solution for the considered values of its parameters, which typically occurs near a bifurcation, then a large shift, $\alpha_n\mathbf{d}_n$, of the solution may cause the iterations to converge to a different solution than initially intended, or to diverge. Such a failure of the method can be avoided simply by changing the initial guess $\mathbf{y}_0$. We will discuss this in more detail in Section 6.

3.2 Review of the generalized Petviashvili method

In [6], we proposed the following iterative method for finding the fundamental solitary wave u of the nonlinear problem (2.1):

$$u_{n+1} - u_n = \left(N^{-1}L^{(0)}u_n - \gamma\,\frac{\langle u_n, L^{(0)}u_n\rangle}{\langle u_n, Nu_n\rangle}\,u_n \right)\Delta\tau\,, \tag{3.6}$$

where

$$\langle f,\, g\rangle \equiv \int f(\mathbf{x})g(\mathbf{x})\,d\mathbf{x}$$

and the integration is over the entire spatial domain. In (3.6), a self-adjoint and positive definite differential operator N with constant coefficients *is chosen* so that u is an approximate eigenvector of $N^{-1}L$ corresponding to its largest eigenvalue $\lambda^{(1)}$:

$$N^{-1}Lu \approx \lambda^{(1)}u\,. \tag{3.7}$$

The role of the γ-term in (3.6) and the meaning of the approximate equality in (3.7) are discussed at the end of this subsection. The Sylvester law of inertia (see, e.g., [24]) guarantees that the signs of the eigenvalues of $N^{-1}L$ are the same as the signs of the respective eigenvalues of L. Then, according to our definition of a fundamental solitary wave at the end of Introduction, $\lambda^{(1)}$ may be positive and all the other eigenvalues of $N^{-1}L$ are negative.

The explicit form of N is usually taken to be similar to that of M. For example, when M is given by (1.3), N has the form (2.18) where the constant c can be computed algorithmically from u_n [6]. (For the equation (1.1) with power-law nonlinearity, $N = M$ and the equality in (3.7) is exact [2].) The constant γ in (3.6) is given by (2.14) and $\lambda^{(1)} \equiv \lambda^{(1)}_n$ is found from

$$\lambda^{(1)}_n = \langle u_n,\, Lu_n\rangle / \langle u_n,\, Nu_n\rangle\,. \tag{3.8}$$

In practice, N, γ, and $\lambda^{(1)}$ are computed until the iteration error reaches some predefined tolerance, after which their last-computed values are used for all subsequent iterations. Since N is a differential operator with constant coefficients, it has a simple representation in the Fourier space. Therefore, quantities like $N^{-1}L^{(0)}u_n$, Nu_n, etc. are easily computed using the direct and inverse Fast Fourier Transforms, which are available as built-in commands in all major computing software.

In what follows we will frequently refer to the linearized form of (3.6), which is:

$$\tilde{u}_{n+1} - \tilde{u}_n = \left(N^{-1}L\tilde{u}_n - \gamma\, \frac{\langle u,\, L\tilde{u}_n\rangle}{\langle u, Nu\rangle}\, u \right) \Delta\tau\,. \tag{3.9}$$

Note that the operator $N^{-1}L$ in the leading term of (3.9) is not self-adjoint. Therefore, we introduce a commonly employed change of variables to rewrite that equation in a form involving a self-adjoint operator. Namely, we denote

$$v = N^{1/2}u, \quad \tilde{v}_n = N^{1/2}\tilde{u}_n, \quad K = N^{-1/2}L\, N^{-1/2}\,, \tag{3.10a}$$

$$K^{(0)}v_n \equiv N^{-1/2}L^{(0)}u_n\,. \tag{3.10b}$$

Note that operator K is self-adjoint, since L is self-adjoint and N is self-adjoint and positive definite. Let us stress that K and v have been introduced only for the convenience of carrying out the subsequent analysis with self-adjoint operators. Once the derivation of the algorithm in this framework is complete, we will recast it in terms of the original variable u_n and operator L.

With the notations (3.10), Eqs. (3.7), (3.6), and (3.9) are rewritten as:

$$Kv \approx \lambda^{(1)}v\,, \tag{3.11}$$

$$v_{n+1} - v_n = \left(K^{(0)}v_n - \gamma\, \frac{\langle v_n, K^{(0)}v_n\rangle}{\langle v_n, v_n\rangle}\, v_n \right) \Delta\tau\,, \tag{3.12}$$

$$\tilde{v}_{n+1} - \tilde{v}_n = \left(K\tilde{v}_n - \gamma\, \frac{\langle v,\, K\tilde{v}_n\rangle}{\langle v, v\rangle}\, v \right) \Delta\tau\,. \tag{3.13}$$

Equations (3.12) and (3.13) are the nonlinear and linearized counterparts of the linear problem (2.10) with a symmetric matrix A. The γ-terms in these equations are needed to eliminate the component of the error $\tilde{v}_{n+1}$ along the unstable "direction" v; see (3.11) and the end of the paragraph following relation (3.7). If the equality in (3.11) (and (3.7)) is exact, which occurs only for equations with power-law nonlinearity, (1.1), this elimination is complete. This is the reason behind the success of the Petviashvili method [1], as was shown in [2]. However, for the majority of equations (1.4) with a more complicated nonlinear term, the corresponding equation (3.7) (and (3.11)) can be enforced — by the choice of N via an algorithm presented in [6] — only approximately. A practical measure of this approximation can be the cosine of the "angle" between Nu and Lu, or, equivalently, between v and Kv:

$$\cos^2(\widehat{v, Kv}) = \cos^2(\widehat{Nu, Lu}) = \frac{\langle Nu, Lu\rangle^2}{\langle Nu, Nu\rangle\langle Lu, Lu\rangle}\,. \tag{3.14}$$

For single-component equations considered in [6], this quantity was about $0.98\ldots 0.99$. Thus, the γ-terms in (3.12) and (3.13) strongly suppress, although do not completely eliminate, the component along the unstable eigenmode. This strong suppression is sufficient to turn the unstable eigenmode approximated by v into a stable one in the generalized Petviashvili method [6]. For the modified CGM described in the next subsection, this incomplete elimination of the unstable mode may lead to divergence. However, we will discuss how such divergence can be overcome.

In the remainder of this section, we will continue using Eq. (3.13) with the self-adjoint operator K for all derivations, and only at the final steps of those derivations will convert to the original variable u_n.

3.3 A modified CGM

As we noted at the end of Sec. 3.1, the straightforward nonlinear generalization of the CGM applied to Eq. (2.1) or, equivalently, to

$$K^{(0)}v = 0, \tag{3.15}$$

whose linearized operator K is self-adjoint but not sign definite, may diverge. For equations that we consider in Section 6 below, this method indeed diverges. To modify the method so that it could converge, one can use the idea stated before Eq. (2.8). Namely, replace (3.15) with an equivalent equation $K^{(0,\mathrm{mod})}v = 0$ such that the corresponding linearized operator $K^{(\mathrm{mod})}$ is self-adjoint and sign definite. In Appendix 1 we show that the expression in (3.16a) below, which mimics that on the right-hand side of (3.12), comes close to satisfying these properties: (i) it is equivalent to $K^{(0)}v$, (ii) its linearized operator $K^{(\mathrm{mod})}$ is approximately self-adjoint, and (iii) its eigenvalues that are not too close to zero are all negative. Statements (ii) and (iii) are approximate[5] rather than exact because so is relation

[5] However, as we point out in Appendix 1, in all the examples that we tried and some of which are reported in Section 6, statement (iii) holds in a more definite sense, i.e.: *all* eigenvalues of $K^{(\mathrm{mod})}$ are negative.

(3.11). We will discuss the implications of the approximate character of statement (ii) after we present the algorithm of the modified CGM.

Thus, denoting

$$K^{(0,\mathrm{mod})}v = K^{(0)}v - \Gamma\,\frac{\langle v, K^{(0)}v\rangle}{\langle v, v\rangle}\,v\,, \tag{3.16a}$$

$$\Gamma = 1 + \frac{1}{\lambda^{(1)}}\,, \tag{3.16b}$$

$$K^{(\mathrm{mod})}\tilde{v} = K\tilde{v} - \Gamma\,\frac{\langle v,\, K\tilde{v}\rangle}{\langle v, v\rangle}\,v\,, \tag{3.16c}$$

we propose the following modified CGM for finding the fundamental solitary wave of (3.15). It mimics algorithm (3.1) with the following modifications: $A\mathbf{y} - \mathbf{b}$ in (3.1a) and (3.1d) is replaced with $K^{(0,\mathrm{mod})}v$, A in (3.1b) and (3.1e) is replaced with $K^{(\mathrm{mod})}$, and vectors $\mathbf{r}$ and $\mathbf{d}$ are renamed $\bar{r}$ and $\bar{d}$. Upon the change of variables

$$\bar{r}_n = N^{1/2}r_n, \qquad \bar{d}_n = N^{1/2}d_n, \tag{3.17}$$

and (3.10), this modified CGM is:

$$r_0 = N^{-1}L^{(0,\mathrm{mod})}u_0, \qquad d_0 = r_0, \tag{3.18a}$$

$$\alpha_n = -\frac{\langle Nr_n,\, d_n\rangle}{\langle L^{(\mathrm{mod})}d_n,\, d_n\rangle}, \tag{3.18b}$$

$$u_{n+1} = u_n + \alpha_n d_n, \tag{3.18c}$$

$$r_{n+1} = N^{-1}L^{(0,\mathrm{mod})}u_{n+1} \tag{3.18d}$$

$$\beta_n = -\frac{\langle r_{n+1},\, L^{(\mathrm{mod})}d_n\rangle}{\langle L^{(\mathrm{mod})}d_n,\, d_n\rangle}, \tag{3.18e}$$

$$d_{n+1} = r_{n+1} + \beta_n d_n\,. \tag{3.18f}$$

where

$$L^{(0,\mathrm{mod})}u = L^{(0)}u - \Gamma\,\frac{\langle u, L^{(0)}u\rangle}{\langle u, Nu\rangle}\,Nu\,, \tag{3.19a}$$

$$L^{(\mathrm{mod})}d = Ld - \Gamma\,\frac{\langle u,\, Ld\rangle}{\langle u, Nu\rangle}\,Nu\,. \tag{3.19b}$$

In a practical implementation of algorithm (3.18), iterations should be carried out with the generalized Petviashvili method (3.6) until the error reaches some predefined tolerance (usually, between 1 and 10%), so that the computed parameters of operator N and the constants $\lambda^{(1)}$ and Γ may be considered as known with the corresponding accuracy. After that, the iterations should be carried out with the modified CGM (3.18), which is expected to converge considerably faster than the generalized Petviashvili method.

Let us now discuss convergence of algorithm (3.18). As noted in Section 3.1, a sufficient condition for this convergence would be negative definiteness of $K^{(\mathrm{mod})}$ in (3.16c), which, in turn, would have taken place if v were an exact eigenvector of K (see (3.11) or (3.7)). However, (3.11) and (3.7) are exact only for equations with power-law nonlinearity, (1.1). Therefore, for other nonlinear wave equations, $K^{(\mathrm{mod})}$ can be sign indefinite and then $\langle K^{(\mathrm{mod})}\bar{d}_n,\, \bar{d}_n\rangle$ (and hence $\langle L^{(\mathrm{mod})}d_n,\, d_n\rangle$ in (3.18b) and (3.18e)) can become arbitrarily close to zero. In such situations algorithm (3.18) could fail, and we will now point out when this should be expected. As we explain in Appendix 1, the slight non-self-adjointness of $K^{(\mathrm{mod})}$, which occurs due to the approximate nature of relation (3.11), can cause $\langle K^{(\mathrm{mod})}\bar{d}_n,\, \bar{d}_n\rangle$ to become too close to zero when some of the negative eigenvalues of the linearized operator K of (3.15) (or, equivalently, L of (2.1)) are close to zero. In single-component equations, this typically occurs when the solitary wave bifurcates from the edge of the continuous spectrum, while for multi-component equations this can also happen when, say, an asymmetric wave bifurcates from one with a symmetry (see, e.g., [25] and references therein).

Our numerical experiments confirm that algorithm (3.18) converges whenever the eigenvalues of L are sufficiently far from zero[6], and also that is may diverge in the opposite situation, as described above. Recall, from the last paragraph of Section 3.1, that in exactly the same situation, a CGM for a *nonlinear* problem can fail even if the quadratic form $\langle \bar{d}_n,\, K^{(\mathrm{mod})}\bar{d}_n\rangle$ were negative definite. Thus, there are two different reasons that can cause the modified CGM to fail, and they both can occur only when L has near-zero eigenvalues. Moreover, they both require that at some iteration, the search direction become primarily “aligned” with the eigenmode of L corresponding to a small eigenvalue. Therefore, a simple and practical way to avoid divergence of the algorithm is to change the initial guess u_0 in a certain manner. In Section 6 we show that this way does indeed work.

Before concluding this section, let us note that one could develop a modified version of the CGM based on a different idea than the elimination of the unstable eigenmode from the underlying operator, as in (3.16). Instead, one can employ the original operators $K^{(0)}$ and K but force all search directions $\bar{d}_n$ to be orthogonal to the unstable eigenmode of K approximated by the exact solitary wave v. Since (3.11) holds approximately, the search directions can be made only approximately orthogonal to the true unstable eigenmode of K. One can show that this causes this other modified CGM to be prone to failure under the same circumstances as the modified CGM (3.18), i.e. when L has eigenvalues that are close to zero. Our numerical experiments show that the simple trick that can help algorithm (3.18) overcome that failure, is not as effective for the other modified CGM. Moreover, the coding of the latter method is slightly more complicated than that of (3.18). For these reasons, we only present that alternative modified CGM in Appendix 2, but do not discuss it further in this paper.

[6] Note that if the solitary wave is translationally invariant, L has a zero eigenvalue corresponding to the respective translational eigenmode. As we show in Appendix 1, such a mode presents no problem for convergence of the iterations since it would simply slightly shift the solitary wave. Therefore, in the following we ignore the possible presence of such zero eigenvalues of L.

4 Modified CGM for solitary waves with prescribed power

We first review the preconditioned imaginary-time evolution method (ITEM) [12], which is a Richardson-type iterative method for finding solitary waves with a specified value of power, and then present a modified CGM that can be used for the same purpose. This modified CGM converges under the same conditions as the ITEM, but faster.

4.1 Review of the ITEM

When one seeks a solitary wave with a prescribed value of power (1.7), problem (2.1) should be redefined because the propagation constant μ, which enters $L^{(0)}$ via operator M (see (1.3)), is no longer known exactly. Instead, it is estimated at each iteration, as shown below. Thus, we rewrite (1.4) as

$$L^{(00)}u - \mu u = 0, \tag{4.1a}$$

where

$$L^{(00)}u = \nabla^2 u + F(u, \mathbf{x}), \qquad \mu = \frac{\langle f(u), L^{(00)}u \rangle}{\langle f(u), u \rangle}\,, \tag{4.1b}$$

and $f(u)$ is a function of u whose choice will be specified shortly.

The preconditioned ITEM whose convergence conditions are found in [12] is:

$$\mu_n = \frac{\langle N^{-1}u, L^{(00)}u \rangle}{\langle N^{-1}u, u \rangle}\,, \tag{4.2a}$$

$$\hat{u}_{n+1} = u_n + N^{-1}\left(L^{(00)}u_n - \mu_n u_n\right)\Delta\tau\,, \tag{4.2b}$$

$$u_{n+1} = \hat{u}_{n+1}\sqrt{\frac{P}{\langle \hat{u}_{n+1}, \hat{u}_{n+1} \rangle}}\,. \tag{4.2c}$$

Here P is the prescribed value of the power and N is a preconditioning operator of the form (2.18) where now c is an *arbitrary* (unlike in Section 3) positive number.

In what follows we will need a few facts about the linearized method (4.2) [12]. First, the condition that $\langle u_n, u_n \rangle = P = \text{const}$ entails the orthogonality between the error and the exact solution:

$$\langle \tilde{u}_n, u \rangle = O(\tilde{u}_n^2)\,. \tag{4.3}$$

Next, the linearized form of the operator in parentheses in (4.2b) is:

$$\mathcal{L}\tilde{u}_n \equiv L\tilde{u}_n - \frac{\langle N^{-1}u, L\tilde{u}_n \rangle}{\langle N^{-1}u, u \rangle}u\,. \tag{4.4}$$

(Note that operator L involves the propagation constant μ. Even though μ is not specified when we seek the solitary wave, we can still use it in analyses of iterative methods.) Using the last two equations, one can straightforwardly show that the last line, (4.2c), of the ITEM does not change the linearization of the preceding line. Equation (4.2c) is needed to ensure that the power of the solitary wave equals P *exactly* rather than in the linear approximation.

The ITEM (4.2) converges when its linearized operator, $\mathcal{L}$, has only negative eigenvalues (see the footnote at the end of Section 3). Loosely speaking (see [12] for a more precise statement), for a wide subclass of equations (1.4), this occurs when operator L has none or one positive eigenvalues. Moreover, in the latter case, the following condition has to hold:

$$dP/d\mu > 0\,. \tag{4.5}$$

Below we will use the change of variables (3.10a) and similar notations:

$$K^{(00)}v_n \equiv N^{-1/2}L^{(00)}u_n\,, \qquad \mathcal{K}^{(0)}v_N \equiv K^{(00)}v_n - \mu_n N^{-1}v_n\,, \tag{4.6}$$

where μ_n is given by (4.2a). The linearization of $\mathcal{K}^{(0)}v_n$ is an operator

$$\mathcal{K}\tilde{v}_n \equiv K\tilde{v}_n - \frac{\langle N^{-1}v, K\tilde{v}_n\rangle}{\langle N^{-1}v, N^{-1}v\rangle}N^{-1}v\,. \tag{4.7}$$

When the convergence condition of the ITEM (4.2), stated near (4.5), hold, operator $\mathcal{K}$ is [12] self-adjoint and negative definite *on the space of functions satisfying the exact form of the orthogonality relation* (4.3):

$$\langle \tilde{v}_n, N^{-1}v\rangle = 0\,. \tag{4.8}$$

4.2 A modified CGM

The idea of this modified CGM is to ensure that both the iteration error $\tilde{v}_n$ and the search direction be orthogonal, in the sense of (4.8), to the exact solution v. Then, according to the last sentence in the previous subsection, operator $\mathcal{K}$ is guaranteed to be negative definite on this space of functions, and hence the following modified CGM is guaranteed to converge under the same conditions as the ITEM (4.2):

$$\bar{r}_0 = \mathcal{K}^{(0)}v_0, \qquad \bar{d}_0 = \bar{r}_0 - \frac{1}{P}\langle N^{-1}v_0, \bar{r}_0\rangle\, v_0, \tag{4.9a}$$

$$\alpha_n = -\frac{\langle \bar{r}_n,\, \bar{d}_n\rangle}{\langle \bar{d}_n,\, \mathcal{K}\bar{d}_n\rangle}\,, \tag{4.9b}$$

$$\hat{v}_{n+1} = v_n + \alpha_n\bar{d}_n\,, \qquad v_{n+1} = \hat{v}_{n+1}\sqrt{\frac{P}{\langle \hat{v}_{n+1}, N^{-1}\hat{v}_{n+1}\rangle}}\,, \tag{4.9c}$$

$$\bar{r}_{n+1} = \mathcal{K}^{(0)}v_{n+1} \tag{4.9d}$$

$$\beta_n = -\frac{\langle \bar{r}_{n+1},\, \mathcal{K}\bar{d}_n\rangle - \frac{1}{P}\langle v_{n+1}, \mathcal{K}\bar{d}_n\rangle\langle N^{-1}v_{n+1}, \bar{r}_{n+1}\rangle}{\langle \bar{d}_n,\, \mathcal{K}\bar{d}_n\rangle}, \tag{4.9e}$$

$$\bar{d}_{n+1} = \bar{r}_{n+1} + \beta_n\bar{d}_n - \frac{1}{P}\langle N^{-1}v_{n+1},\, \bar{r}_{n+1} + \beta_n\bar{d}_n\rangle\, v_{n+1}\,. \tag{4.9f}$$

The definitions of $\bar{d}_0$ in (4.9a) and $\bar{d}_{n+1}$ in (4.9f) ensure that

$$\langle \bar{d}_n, N^{-1}v_n\rangle = 0 \tag{4.10}$$

at every iteration. Here we use the fact that our iterative solution has a fixed value of power:

$$\langle v_n, N^{-1} v_n \rangle = P \tag{4.11}$$

for all n. Since the error $\tilde{v}_n$ at the nth iteration satisfies the orthogonality condition (4.8), the first of Eqs. (4.9c) and Eq. (4.10) ensure that the error $\tilde{v}_{n+1}$ at the next iteration also satisfies (4.8). The second of Eqs. (4.9c) does not change this relation; it only forces (4.11) to be satisfied exactly rather than in the linear approximation. Equations (4.9b–d) entail the counterpart of (3.2):

$$\langle \bar{r}_{n+1},\, \bar{d}_n \rangle = O(\tilde{v}_n^3)\,, \tag{4.12}$$

and Eqs. (4.9e,f) entail a counterpart of (3.3):

$$\langle \bar{d}_{n+1},\, \mathcal{K}\bar{d}_n \rangle = O(\tilde{v}_n^3)\,. \tag{4.13}$$

We demonstrate these relations in Appendix 1.

Let us note that condition (4.10), in addition to ensuring (4.8) at each iteration, plays one other important role in this algorithm. Namely, it ensures that in the inner product on the left-hand side of (4.13), $\mathcal{K}$ is a self-adjoint operator: see the sentence before (4.8). This property of $\mathcal{K}$ entails the counterpart of (3.5), which together with negative definiteness of $\mathcal{K}$ implies that all search directions $\bar{d}_n$ are linearly independent and the error decreases as the iterations proceed. In Section 3.1 we noted that this means that the CGM is an optimal iterative method (see, e.g., [18], p. 296, and [23]).

Thus, the modified CGM (4.9) is guaranteed to converge under the same conditions as its Richardson-type counterpart (4.2). This, mathematically, is a more rigorous result than what we obtained for the modified CGM (3.18), which is guaranteed to converge in a slightly narrower parameter space than its Richardson-type counterpart (3.6).

Finally, we present algorithm (4.9) in the original variables:

$$r_0 = N^{-1}\mathcal{L}^{(0)} u_0, \qquad d_0 = r_0 - \frac{1}{P}\langle u_0, r_0 \rangle\, u_0\,, \tag{4.14a}$$

$$\alpha_n = -\frac{\langle r_n,\, N d_n \rangle}{\langle d_n,\, \mathcal{L} d_n \rangle}\,, \tag{4.14b}$$

$$\hat{u}_{n+1} = u_n + \alpha_n d_n\,, \qquad u_{n+1} = \hat{u}_{n+1}\sqrt{\frac{P}{\langle \hat{u}_{n+1}, \hat{u}_{n+1} \rangle}}\,, \tag{4.14c}$$

$$r_{n+1} = N^{-1}\mathcal{L}^{(0)} u_{n+1} \tag{4.14d}$$

$$\beta_n = -\frac{\langle r_{n+1},\, \mathcal{L} d_n \rangle - \frac{1}{P}\langle u_{n+1}, \mathcal{L} d_n \rangle \langle u_{n+1}, r_{n+1} \rangle}{\langle d_n,\, \mathcal{L} d_n \rangle}, \tag{4.14e}$$

$$d_{n+1} = r_{n+1} + \beta_n d_n - \frac{1}{P}\langle u_{n+1},\, r_{n+1} + \beta_n d_n \rangle\, u_{n+1}\,. \tag{4.14f}$$

Here $\mathcal{L}$, which is the linearization of

$$\mathcal{L}^{(0)} u_n \equiv L^{(00)} u_n - \frac{\langle N^{-1} u_n, L^{(00)} u_n \rangle}{\langle N^{-1} u_n, u_n \rangle}\, u_n\,, \tag{4.15}$$

is given in (4.4).

5 Modified CGMs for multi-component solitary waves

Here we present the multi-component versions of the modified CGMs (3.18) and (4.14) for fundamental solitary waves with prescribed values of, respectively, propagation constants and quadratic conserved quantities that are the counterparts of power (1.7). We will not give derivations of these methods because they are similar to those found in Sections 3 and 4.

We begin with stating the generalized Petviashvili method from [6] for finding a multi-component solitary wave $\mathbf{u} = [u^{(1)}, \ldots, u^{(S)}]^T$ of the equation

$$\mathbf{L^{(0)}u} = \mathbf{0} \tag{5.1}$$

whose linearized operator is $\mathbf{L}$:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \left[\mathbf{N}^{-1}\mathbf{L^{(0)}}\mathbf{u}_n - \sum_{k=1}^{S} \gamma^{(k)} \frac{\left\langle \mathbf{e}_n^{(k)},\, \mathbf{L^{(0)}}\mathbf{u}_n \right\rangle}{\left\langle \mathbf{e}_n^{(k)},\, \mathbf{N}\mathbf{e}_n^{(k)} \right\rangle}\, \mathbf{e}_n^{(k)} \right] \Delta\tau\,. \tag{5.2}$$

Here $\mathbf{N} = \text{diag}\left(N^{(1)}, \ldots, N^{(S)}\right)$ where each $N^{(k)}$ has a form similar to the linear differential part of the kth component of $\mathbf{L^{(0)}}$; e.g., for a system of nonlinear Schrödinger-type equations,

$$N^{(k)} = c^{(k)} - b^{(k)}\nabla^2, \tag{5.3}$$

and $c^{(k)}$, $b^{(k)}$ are computed by formulas found in [6]. Furthermore, $\mathbf{e}^{(k)}$ are the approximate eigenvectors of $\mathbf{N}^{-1}\mathbf{L}$:

$$\mathbf{L}\mathbf{e}^{(k)} \approx \lambda^{(k)}\mathbf{N}\mathbf{e}^{(k)}\,, \tag{5.4}$$

which are mutually orthogonal with weight $\mathbf{N}$:

$$\langle \mathbf{e}^{(k)}, \mathbf{N}\mathbf{e}^{(m)} \rangle = 0 \qquad \text{for } k \neq m. \tag{5.5}$$

Note that (5.4) is the multi-component counterpart of (3.7). In particular, the exact equality in it holds only for a special class of coupled power-law equations, which for the two-component case were found in Appendix 4 of [6]. For other coupled equations, the equality in (5.4) can only be enforced — by the choice of $\mathbf{N}$ — to hold approximately. The quantitative measure of this approximation is the cosine of the "angle" between $\mathbf{L}\mathbf{e}_k$ and $\mathbf{N}\mathbf{e}_k$ defined similarly to (3.14). The form of $\mathbf{e}^{(k)}$'s is related to the solution of (5.1) by

$$\mathbf{e}_n^{(1)} = \mathbf{u}_n, \quad \mathbf{e}^{(k)} = \left(\rho^{(k,1)}u^{(1)}, \ldots, \rho^{(k,S)}u^{(S)}\right)^T\,, \tag{5.6}$$

where $\rho^{(k,k)} = 1$ and the rest of $\rho^{(k,j)}$ are found as explained in [6]. Finally, the constants $\lambda^{(k)}$ and $\gamma^{(k)}$ in (5.2) are found similarly to (3.8) and (2.14):

$$\lambda^{(k)} = \langle \mathbf{e}^{(k)}, \mathbf{L}\mathbf{e}^{(k)} \rangle / \langle \mathbf{e}^{(k)}, \mathbf{N}\mathbf{e}^{(k)} \rangle\,, \qquad \gamma^{(k)} = 1 + 1/(\lambda^{(k)}\Delta\tau)\,. \tag{5.7}$$

The multi-component version of the modified CGM of Section 3 looks as (3.18), with the scalar functions u_n, r_n, d_n being replaced by the S-dimensional vectors $\mathbf{u}_n$, $\mathbf{r}_n$, $\mathbf{d}_n$, operator N being replaced by its matrix counterpart given before Eq. (5.3), and the operators $\mathbf{L}^{(\mathbf{0},\mathrm{mod})}$ and $\mathbf{L}^{(\mathrm{mod})}$ being given by expressions generalizing (3.19):

$$\mathbf{L}^{(\mathbf{0},\mathrm{mod})}\mathbf{u}_n \;=\; \mathbf{L}^{(\mathbf{0})}\mathbf{u}_n - \sum_{k=1}^{S}\Gamma^{(k)}\frac{\left\langle \mathbf{e}_n^{(k)},\, \mathbf{L}^{(\mathbf{0})}\mathbf{u}_n\right\rangle}{\left\langle \mathbf{e}_n^{(k)},\, \mathbf{N}\mathbf{e}_n^{(k)}\right\rangle}\,\mathbf{N}\,\mathbf{e}_n^{(k)}\,, \tag{5.8a}$$

$$\Gamma^{(k)} = 1 + \frac{1}{\lambda^{(k)}}\,, \tag{5.8b}$$

$$\mathbf{L}^{(\mathrm{mod})}\mathbf{d}_n \;=\; \mathbf{L}\mathbf{d}_n - \sum_{k=1}^{S}\Gamma^{(k)}\frac{\left\langle \mathbf{e}_n^{(k)},\, \mathbf{L}\mathbf{d}_n\right\rangle}{\left\langle \mathbf{e}_n^{(k)},\, \mathbf{N}\mathbf{e}_n^{(k)}\right\rangle}\,\mathbf{N}\,\mathbf{e}_n^{(k)}\,. \tag{5.8c}$$

A sample code of this algorithm for a two-component system considered in Section 6 is presented in Appendix 3.

Let us note that the purpose of the terms under the sum in (5.8) is to eliminate the eigenmodes $\mathbf{e}^{(k)}$ of $\mathbf{L}$ whose eigenvalues $\lambda^{(k)}$ could be positive. By our definition of the fundamental solitary wave, stated at the end of Introduction, eliminating S such eigenmodes should be sufficient to turn the positive eigenvalues of $\mathbf{L}$ into negative eigenvalues of $\mathbf{L}^{(\mathrm{mod})}$. As explained in Section 3, due to the approximate character of relations (5.4), this unstable mode elimination still may not always guarantee convergence of the multi-component modified CGM (3.18), (5.8) when $\mathbf{L}$ has small negative eigenvalues. However, a simple approach based on perturbing the initial guess, as discussed in Section 6, overcomes this divergence.

We now turn to the case where a set of quadratic conserved quantities related to the solution components' powers are prescribed. We need to introduce a number of new notations. Let us denote the s-component vector of these quantities by $\vec{Q}$, so that its kth component is

$$Q^{(k)} = \sum_{l=1}^{S} q^{(kl)}P^{(l)}, \quad P^{(l)} = \langle u^{(l)}, u^{(l)}\rangle, \qquad k = 1, \ldots, s \le S\,, \quad l = 1, \ldots, S. \tag{5.9}$$

Note that the number of these conserved quantities can be less than the number of the components of the solitary wave: $s \le S$. To emphasize this fact, we use a different vector notation for $\vec{Q}$ than for $\mathbf{u}$. The number of the components of $\vec{Q}$ equals the number of the propagation constants that one can prescribe independently. One example of this situation is the system of three waves interacting via quadratic nonlinearity [26] (see also [7]). Another example, which we discuss in more detail below to make our notations clear, is the system describing the evolution of pulses in a two-core nonlinear directional coupler [25]:

$$\left.\begin{aligned} &iU_t^{(k)} + U_{xx}^{(k)} + \left(|U^{(k)}|^2 + \kappa|U^{(k+2)}|^2\right)U^{(k)} + U^{(3-k)} = 0 \\ &iU_t^{(k+2)} + U_{xx}^{(k+2)} + \left(|U^{(k+2)}|^2 + \kappa|U^{(k)}|^2\right)U^{(k+2)} + U^{(5-k)} = 0 \end{aligned}\right. \qquad k = 1, 2\,, \tag{5.10}$$

where $U^{(k)}$ and $U^{(k+2)}$ are the orthogonal polarization components of the pulse in the kth core. Upon the substitution

$$\begin{pmatrix} U^{(1)}(x,t) \\ U^{(2)}(x,t) \\ U^{(3)}(x,t) \\ U^{(4)}(x,t) \end{pmatrix} = \begin{pmatrix} u^{(1)}(x) & 0 \\ u^{(2)}(x) & 0 \\ 0 & u^{(3)}(x) \\ 0 & u^{(4)}(x) \end{pmatrix} \begin{pmatrix} e^{i\mu^{(1)}t} \\ e^{i\mu^{(2)}t} \end{pmatrix}, \tag{5.11}$$

where $u^{(k)}$ can be chosen to be real-valued, system (5.10) reduces to:

$$\begin{pmatrix} u^{(1)}_{xx} + \left((u^{(1)})^2 + \kappa(u^{(3)})^2\right) u^{(1)} + u^{(2)} \\ u^{(2)}_{xx} + \left((u^{(2)})^2 + \kappa(u^{(4)})^2\right) u^{(2)} + u^{(1)} \\ u^{(3)}_{xx} + \left((u^{(3)})^2 + \kappa(u^{(1)})^2\right) u^{(3)} + u^{(4)} \\ u^{(4)}_{xx} + \left((u^{(4)})^2 + \kappa(u^{(2)})^2\right) u^{(4)} + u^{(3)} \end{pmatrix} - \begin{pmatrix} u^{(1)} & 0 \\ u^{(2)} & 0 \\ 0 & u^{(3)} \\ 0 & u^{(4)} \end{pmatrix} \vec{\mu} \; = \; \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{5.12}$$

where $\vec{\mu} = \left(\mu^{(1)},\, \mu^{(2)}\right)^T$. Thus, in this example, $S = 4$, $s = 2$, and the vector of quadratic conserved quantities (verified from the time-dependent equations (5.10)) is

$$\vec{Q} = \begin{pmatrix} P^{(1)} + P^{(2)} \\ P^{(3)} + P^{(4)} \end{pmatrix} \equiv \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} P^{(1)} \\ P^{(2)} \\ P^{(3)} \\ P^{(4)} \end{pmatrix}. \tag{5.13}$$

Generalizing the above example, one can see that the counterpart of Eq. (5.1) when $\vec{Q}$ rather than $\vec{\mu}$ is prescribed, is the following extension of Eq. (4.1a):

$$\mathbf{L^{(00)}u} - \mathcal{U}\,\langle \mathbf{N}^{-1}\mathcal{U},\, \mathcal{U}\rangle^{-1}\,\langle \mathbf{N}^{-1}\mathcal{U},\, \mathbf{L^{(00)}u}\rangle \;=\; \mathbf{0}\,, \tag{5.14a}$$

$$\mathcal{U} \equiv \frac{\delta \vec{Q}}{\delta \mathbf{u}}\,. \tag{5.14b}$$

For example, in (5.12), $\mathbf{L^{(00)}u}$ is the first term (the 4 × 1 vector), and $\mathcal{U}$ is the first factor of the second term (the 4 × 2 matrix) on the left-hand side.

For the convenience of subsequent notations, we will assume that the matrix $\left(q^{(kl)}\right)$ in (5.9) is put into the reduced echelon form (see any textbook on undergraduate linear algebra) and, in addition, its columns are rearranged so that

$$q^{(kl)} \;=\; \begin{cases} 0, & l < k \\ 1, & l = k\,. \end{cases} \tag{5.15}$$

For example, for the 2 × 4 matrix in (5.13) this would imply switching the second and third columns. Then for the powers of the solitary wave components whose indices equal the indices of the pivot columns of $\left(q^{(kl)}\right)$, one has from (5.9):

$$P^{(k)} = Q^{(k)} - \sum_{l=k+1}^{S} q^{(kl)} P^{(l)}\,, \qquad k = 1, \ldots, s \le S\,. \tag{5.16}$$

We will use this fact in the next paragraph.

With the convention (5.15), the multi-component version of the ITEM is:

$$\hat{\mathbf{u}}_{n+1} = \mathbf{u}_n + \mathbf{N}^{-1}\left(\mathbf{L}^{(\mathbf{00})}\mathbf{u}_n - \mathcal{U}_n \left\langle \mathbf{N}^{-1}\mathcal{U}_n,\, \mathcal{U}_n\right\rangle^{-1} \left\langle \mathbf{N}^{-1}\mathcal{U}_n,\, \mathbf{L}^{(\mathbf{00})}\mathbf{u}_n\right\rangle\right)\Delta\tau\,, \tag{5.17a}$$

$$u^{(k)}_{n+1} = \hat{u}^{(k)}_{n+1}\sqrt{\frac{Q^{(k)} - \sum_{l=k+1}^{S} q^{(kl)}\hat{P}^{(l)}_{n+1}}{\hat{P}^{(k)}_{n+1}}}\,, \qquad k = 1, \ldots, s \le S\,, \tag{5.17b}$$

where

$$\hat{P}^{(k)}_{n+1} \equiv \langle \hat{u}^{(k)}_{n+1},\, \hat{u}^{(k)}_{n+1}\rangle\,, \qquad k = 1, \ldots, S\,.$$

In analogy to the algorithm (4.2) for single-component equations, Eq. (5.17b) does not change the linearized form of (5.17a); its role is to guarantee that the s components of vector $\vec{Q}$ equal their prescribed values *exactly* rather than in the linear approximation.

To our knowledge, the multi-component ITEM (5.17) has not been presented in the literature before; however, its close "relative" (related to a family of squared-operator methods) was stated in [7]. The convergence conditions for (5.17) are [21]: (i) the $s \times s$ Jacobian $\partial\vec{Q}/\partial\vec{\mu}$ must be nonsingular, and (ii) the number of positive eigenvalues of this Jacobian must equal the number of positive eigenvalues of the linearized operator $\mathbf{L}$ of Eq. (5.1). These conditions are a counterpart of (4.5). They guarantee that the linearized operator,

$$\mathcal{L}\tilde{\mathbf{u}}_n \equiv \mathbf{L}\tilde{\mathbf{u}}_n - \mathcal{U}\left\langle \mathbf{N}^{-1}\mathcal{U},\, \mathcal{U}\right\rangle^{-1}\left\langle \mathbf{N}^{-1}\mathcal{U},\, \mathbf{L}\tilde{\mathbf{u}}_n\right\rangle \tag{5.18}$$

of the expression $\mathcal{L}^{(0)}\mathbf{u}_n$ appearing inside the parentheses in (5.17a), is negative definite on the space of vector functions satisfying an analog of the orthogonality relation (4.3):

$$\langle \mathcal{U},\, \tilde{\mathbf{u}}_n\rangle = \vec{0}\,. \tag{5.19}$$

The corresponding multi-component version of the modified CGM (4.14) is:

$$\mathbf{r}_0 = \mathbf{N}^{-1}\mathcal{L}^{(0)}\mathbf{u}_0, \qquad \mathbf{d}_0 = \mathbf{r}_0 - \mathcal{U}_0\left\langle \mathcal{U}_0, \mathcal{U}_0\right\rangle^{-1}\left\langle \mathcal{U}_0,\, \mathbf{r}_0\right\rangle\,, \tag{5.20a}$$

$$\alpha_n = -\frac{\langle \mathbf{r}_n,\, \mathbf{N}\mathbf{d}_n\rangle}{\langle \mathbf{d}_n,\, \mathcal{L}\mathbf{d}_n\rangle}\,, \tag{5.20b}$$

$$\hat{\mathbf{u}}_{n+1} = \mathbf{u}_n + \alpha_n\mathbf{d}_n\,, \qquad u^{(k)}_{n+1} = \hat{u}^{(k)}_{n+1}\sqrt{\frac{Q^{(k)} - \sum_{l=k+1}^{S} q^{(kl)}\hat{P}^{(l)}_{n+1}}{\hat{P}^{(k)}_{n+1}}}\,, \qquad k = 1, \ldots, s \le S\,, \tag{5.20c}$$

$$\mathbf{r}_{n+1} = \mathbf{N}^{-1}\mathcal{L}^{(0)}\mathbf{u}_{n+1} \tag{5.20d}$$

$$\beta_n = -\frac{\langle \mathbf{r}_{n+1},\, \mathcal{L}\mathbf{d}_n\rangle - \langle \mathcal{L}\mathbf{d}_n,\, \mathcal{U}_{n+1}\rangle\langle \mathcal{U}_{n+1}, \mathcal{U}_{n+1}\rangle^{-1}\langle \mathcal{U}_{n+1},\, \mathbf{r}_{n+1}\rangle}{\langle \mathbf{d}_n,\, \mathcal{L}\mathbf{d}_n\rangle}, \tag{5.20e}$$

$$\mathbf{d}_{n+1} = \mathbf{r}_{n+1} + \beta_n\mathbf{d}_n - \mathcal{U}_{n+1}\left\langle \mathcal{U}_{n+1}, \mathcal{U}_{n+1}\right\rangle^{-1}\left\langle \mathcal{U}_{n+1},\, \mathbf{r}_{n+1} + \beta_n\mathbf{d}_n\right\rangle\,. \tag{5.20f}$$

This method is guaranteed to converge to a fundamental solitary wave with a prescribed vector of conserved quantities $\vec{Q}$ under the conditions stated before Eq. (5.18).

6 Numerical examples

Here we compare the performance of the modified versions of the CGM presented in Sections 3–5 with the performance of the corresponding Richardson-type methods accelerated by the mode elimination (ME) technique [8, 7].

The model equations are (1.5) and its two-component extension:

$$\begin{aligned} &\nabla^2 u^{(1)} + V_0(\cos^2 x + \cos^2 y)u^{(1)} + \Big(F^{(1)}(u^{(1)})^2 + F^{(12)}(u^{(2)})^2\Big)\, u^{(1)} = \mu^{(1)} u^{(1)} \\ &\nabla^2 u^{(2)} + V_0(\cos^2 x + \cos^2 y)u^{(2)} + \Big(F^{(12)}(u^{(1)})^2 + F^{(2)}(u^{(2)})^2\Big)\, u^{(2)} = \mu^{(2)} u^{(2)} \end{aligned} \tag{6.1}$$

where the constants

$$F^{(1)} = 1, \quad F^{(2)} = 4, \quad F^{(12)} = 0.5\,.$$

Note that Eq. (1.5) is equivalent to Eqs. (2.1), (2.2) of [27], where $\mu_{[27]} = 2V_0 - \mu_{\text{this paper}}$. The values of V_0 and the propagation constant(s) or power(s) are specified below, so that each modified CGM and the corresponding Richardson-type method accelerated by ME are tested for three cases. We label these cases as mildly numerically stiff, stiffer, and stiffest, based on the observed rates of convergence of the iterative methods. (In the stiffest case the methods take the most number of iterations to converge.) A code for the most complicated of these methods — the modified CGM (3.18), (5.8) for the two-component Eqs. (6.1) — is presented in Appendix 3. (A counterpart of this code for a single-component equation is considerably simpler and can be easily written based on a code from Appendix 1 in [6]. Codes seeking solitary waves with prescribed power are also much simpler than the code in Appendix 3 below, because they do not compute quantities $\mathbf{N}$, $\mathbf{e}^{(k)}$, etc.)

We begin by describing the simulations setup for the methods where the propagation constant(s) is (are) specified. We compare the modified CGMs (3.18) (for the single equation (1.5)) and (3.18), (5.8) (for the two-component system (6.1)) with the ME-accelerated generalized Petviashvili method. The latter method is the closest competitor of the modified CGM among all the Richardson-type iterative methods that we developed earlier; see Section 2. For a single equation, it is [8]:

$$u_{n+1} = u_n + \left[N^{-1}L^{(0)}u_n - \gamma \frac{\langle u_n,\, L^{(0)}u_n\rangle}{\langle u_n,\, Nu_n\rangle}\, u_n \; - \; \gamma_n^{\text{slow}} \frac{\langle \phi_n^{\text{slow}},\, L^{(0)}u_n\rangle}{\langle \phi_n^{\text{slow}},\, N\phi_n^{\text{slow}}\rangle}\, \phi_n^{\text{slow}} \right] \Delta\tau\,; \tag{6.2a}$$

which for a multi-component system extends to:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \left[\mathbf{N}^{-1}\mathbf{L}^{(\mathbf{0})}\mathbf{u}_n - \sum_{k=1}^{S} \gamma^{(k)} \frac{\left\langle \mathbf{e}_n^{(k)},\, \mathbf{L}^{(\mathbf{0})}\mathbf{u}_n \right\rangle}{\left\langle \mathbf{e}_n^{(k)},\, \mathbf{N}\mathbf{e}_n^{(k)} \right\rangle}\, \mathbf{e}_n^{(k)} \; - \; \gamma_n^{\text{slow}} \frac{\left\langle \mathbf{\Phi}_n^{\text{slow}},\, \mathbf{L}^{(\mathbf{0})}\mathbf{u}_n \right\rangle}{\left\langle \mathbf{\Phi}_n^{\text{slow}},\, \mathbf{N}\mathbf{\Phi}_n^{\text{slow}} \right\rangle}\, \mathbf{\Phi}_n^{\text{slow}} \right] \Delta\tau\,. \tag{6.2b}$$

Here the notations are as in (5.2) above and in addition,

$$\mathbf{\Phi}_n^{\text{slow}} = \mathbf{u}_n - \mathbf{u}_{n-1}, \qquad \gamma_n^{\text{slow}} = 1 + \frac{h}{\lambda_n^{\text{slow}}\Delta\tau}, \qquad \lambda_n^{\text{slow}} = \frac{\left\langle \mathbf{\Phi}_n^{\text{slow}},\, \mathbf{L}\mathbf{\Phi}_n^{\text{slow}} \right\rangle}{\left\langle \mathbf{\Phi}_n^{\text{slow}},\, \mathbf{N}\mathbf{\Phi}_n^{\text{slow}} \right\rangle}\,, \tag{6.3}$$

where h is the fraction of the slowest-decaying mode that is subtracted at each iteration. In [8] we advocated subtracting around 70% of such a mode for robust and nearly optimal performance; accordingly, here we use $h = 0.7$ in all examples.

For all the methods listed in the previous paragraph, we designate the following sets of parameters in Eqs. (1.5) and (6.1) as corresponding to the mildly stiff, stiffer, and stiffest cases.

$$\begin{aligned} \underline{\text{For (1.5)}}: \quad \text{mildly stiff} \;&\Rightarrow\; V_0 = 4, \quad \mu = 5.03; \\ \text{stiffer} \;&\Rightarrow\; V_0 = 4, \quad \mu = 4.95; \\ \text{stiffest} \;&\Rightarrow\; V_0 = 6, \quad \mu = 7.89; \end{aligned} \tag{6.4a}$$

$$\begin{aligned} \underline{\text{For (6.1)}}: \quad \text{mildly stiff} \;&\Rightarrow\; V_0 = 4, \quad \mu^{(1)} = 5.03, \;\; \mu^{(2)} = 5.5; \\ \text{stiffer} \;&\Rightarrow\; V_0 = 4, \quad \mu^{(1)} = 4.95, \;\; \mu^{(2)} = 6.5; \\ \text{stiffest} \;&\Rightarrow\; V_0 = 6, \quad \mu^{(1)} = 7.89, \;\; \mu^{(2)} = 8.5\,. \end{aligned} \tag{6.4b}$$

Figure 1 shows schematically the power-versus-μ plots for Eq. (1.5), with the three cases of (6.4a) labeled. Figure 2 shows the solution for the stiffer case for Eqs. (6.1); note the vastly different amplitudes and widths of the two components. In the other two cases the solution looks qualitatively the same. The obtained solutions of the single Eq. (1.5) look qualitatively as the first component of the solution in Fig. 2. In general, the closer the propagation constant to the edge of the band gap, the broader and lower the corresponding solution.

For both the modified CGMs and the ME-accelerated generalized Petviashvili methods, we start the iterations by the non-accelerated generalized Petviashvili method, and when the error, defined as

$$\varepsilon_n = \sum_{k=1}^{S} \frac{\left\langle (\mathbf{L^{(0)}u}_n)^{(k)},\; (\mathbf{L^{(0)}u}_n)^{(k)} \right\rangle}{\left\langle u_n^{(k)},\; u_n^{(k)} \right\rangle}\,, \tag{6.5}$$

reaches a threshold

$$\varepsilon_{\text{acceleration threshold}} = 5 \cdot 10^{-2}, \tag{6.6}$$

we begin the acceleration by either the modified CGM or ME and carry on the iterations until the error reaches 10^{-10}. The parameters of $\mathbf{N}$, $\mathbf{e}^{(k)}$, $\lambda^{(k)}$, and $\gamma^{(k)}$ (or their single-component counterparts) stop being computed at the threshold (6.6) and the latest computed values of these parameters are used from that moment on. (As was demonstrated in [6], for Eqs. (6.1) the eigenvector $\mathbf{e}^{(2)}$ of $\mathbf{N}^{-1}\mathbf{L}$ is computed not too accurately. Yet, we show below that this does not prevent the modified CGM (3.18), (5.8) from performing quite well.)

In regards to the specific numeric value on the right-hand side of (6.6), we observed that the accelerated methods converge in all cases when this value is not too high (say, is less than 10^{-1}). If the error is substantially higher, the computed parameters of $\mathbf{N}$ etc. may be too inaccurate, so that the corresponding operator $\mathbf{K}^{(0,\text{mod})}$ would be "too far" from self-adjoint, which might then prevent the convergence of algorithms (3.18) or (5.8).

In addition, *only* for the modified CGM *in the stiffest case*, the value of the acceleration threshold should not be too low (e.g., should be higher than about $5 \cdot 10^{-3}$); otherwise, the modified CGM in this case would diverge. The explanation of this fact was sketched in the last paragraph of Section 3.1. Here we present specific details for it; for the sake of convenience, we do it for the single Eq. (1.5) and use the transformed variables (3.10), (3.17).

First, as is well known (see, e.g., [27] and references therein), when μ in (1.5) is very close to the edge of the spectral gap of the linear part of $K^{(0)}$ (i.e. in the "stiffest" case considered here), the shape of the exact solution of (1.5) is very similar to that of the eigenmode(s) of K with negative near-zero eigenvalue(s). In particular, the solution is broad, as illustrated in the top panel of Fig. 2. Next, from (3.18a,b) and an analog of (2.3) it follows that at the first CGM iteration, the step size α_0 along the search direction d_0 is

$$\alpha_0 = \frac{\langle K^{(0,\mathrm{mod})} v_0,\, K^{(0,\mathrm{mod})} v_0 \rangle}{\langle K^{(\mathrm{mod})}\, K^{(0,\mathrm{mod})} v_0,\, K^{(0,\mathrm{mod})} v_0 \rangle} \approx \frac{\langle K^{(\mathrm{mod})}\, \tilde{v}_0,\, K^{(\mathrm{mod})}\, \tilde{v}_0 \rangle}{\langle (K^{(\mathrm{mod})})^2\, \tilde{v}_0,\, K^{(\mathrm{mod})}\, \tilde{v}_0 \rangle} \gg 1\,. \qquad (6.7)$$

(Here the iteration's number is referenced from the start of the CGM algorithm.) The last strong inequality follows from two facts: (i) operator K ($= N^{-1/2} L N^{-1/2}$) has a band of negative essential spectrum that comes very close to zero (see, e.g., Fig. 4(a) in [20] for a related problem), and (ii) it is precisely a superposition of the corresponding eigenmodes that dominates the error $\tilde{v}_0$ obtained by the generalized Petviashvili method after sufficiently many iterations (because these modes decay the slowest in Richardson-type methods; see, e.g., [7]). Having $\alpha_0 \gg 1$ and hence a large first step in the CGM adds to v_0 some superposition of near-zero eigenmodes of K and thereby makes the solution at the next iteration, v_1, more broad and hence even more resembling the near-zero eigenmodes. Then the step size α_1, computed at the next iteration, becomes even greater than α_0, and the solution at the following iteration, v_2, is made even more broad. This quickly leads to divergence of the iterations. (Practically, the solution converges to a nonlocalized two-dimensional Bloch wave.) On the other hand, if the acceleration by the modified CGM starts when the error $\tilde{v}_0$ is not too small and hence still contains a significant contribution from the eigenmodes whose eigenvalues are not too close to zero, the corresponding step size α_0 is not too large, and subsequent CGM iterations are able to gradually reduce the error $\tilde{v}_n$.

The above consideration explains why the CGM iterations should start when the iteration error is not too low. In practice, however, this is not a limitation, since in numerically stiff cases, one wants to start the acceleration by the CGM sooner rather than later since this considerably reduces the computational time, as we will demonstrate below.

The initial guess in all cases that we consider is taken as

$$u_0 = 1.5 \cdot e^{-(x^2+y^2)} \cdot (1 + \epsilon_x x + \epsilon_y y), \qquad \epsilon_x = 0.1, \quad \epsilon_y = -0.2 \qquad (6.8a)$$

for Eq. (1.5) and as

$$\begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix}_0 = \begin{pmatrix} 0.8 \\ 1.5 \end{pmatrix} \cdot e^{-(x^2+y^2)} \cdot (1 + \epsilon_x x + \epsilon_y y), \qquad \epsilon_x = 0.1, \quad \epsilon_y = -0.2 \qquad (6.8b)$$

for Eqs. (6.1). The exact values of the amplitude(s) and width of the initial guess are not essential; all methods converge for a wide range of these parameters. As for the asymmetry parameters ϵ_x and ϵ_y, these can be also varied quite a bit as long as the shape if the initial guess resembles a pulse with one main peak. However, these parameters should *not* be set to zero (or too close to zero) for the modified CGMs (3.18) and (5.8) *in the stiffest case.* (That is, in the other two, less stiff, cases, the modified CGMs converge even for a symmetric initial guess.) The reason for this is similar to the reason why the acceleration threshold in (6.6) should not be chosen too small. Namely, iterations of the generalized Petviashvili method which start with a symmetric u_0, i.e., (6.7) with $\epsilon_x = \epsilon_y = 0$, produce symmetric solutions at all subsequent iterations. These solutions increasingly resemble the near-zero eigenmode of the linearized operator L. Then when the CGM starts, the first step α_0 along the search direction d_0 becomes too large, which leads to divergence of the iterations, as explained after Eq. (6.7). On the other hand, having an asymmetric initial guess leads to the iteration error being asymmetric, and such an asymmetric error can have a considerable content of eigenmodes of L whose eigenvalues are not close to zero. This reduces the step size α_0 at the first CGM iteration, and the modified CGM is able to converge. Note that from the practical perspective, starting with an asymmetric initial guess is not a limitation of the method.

While the ME-based acceleration does not require an asymmetric initial guess for convergence, we still use the same expressions (6.8) for all methods, so as to make our performance comparison uniform.

We now comment on the choices of the "fictitious time" step size $\Delta\tau$. For each simulation, we first selected its nearly optimal value for the corresponding non-accelerated generalized Petviashvili method. This takes just a couple of trials since the optimal $\Delta\tau$ is only slightly less than the maximum value of this parameter for which the generalized Petviashvili method still converges. (See, e.g., Eqs. (2.5) and (2.6) in [8] and Fig. 1(d) in [7], based on the same equations, although for another iterative method.) Once this $\Delta\tau_{\rm opt}$ has been determined, we use a slightly smaller (specifically, by 0.1) value of the step size for the method accelerated by ME; the analysis of ME [8] predicts that this should lead to a more robust and smooth peformance than using $\Delta\tau_{\rm opt}$. For example, if we find that $\Delta\tau_{\rm opt} = 0.9$ for the non-accelerated generalized Petviashvili method, we then use $\Delta\tau = 0.8$ for the ME-accelerated version of this method. For the method accelerated by a modified CGM, we also start the iterations using $\Delta\tau = \Delta\tau_{\rm opt} - 0.1$, although this does not noticeably affect its performance (since the CGM itself does not involve $\Delta\tau$).

Finally, the domain for all our numerical simulations is $[-6\pi, 6\pi] \times [-6\pi, 6\pi]$, with

	Iterative method		
Case in (6.4a)	non-accelerated (3.6)	accel. by ME (6.2a)	accel. by CGM (3.18)
mildly stiff, $\Delta\tau = 1.1$	300 iterations, 130 s	110 iterations, 60 s	60 iterations, 40 s
stiffer, $\Delta\tau = 1.1$	920 iterations, 390 s	290 iterations, 170 s	100 iterations, 60 s
stiffest, $\Delta\tau = 1.0$	3700 iterations, 1560 s	430 iterations, 240 s	200 iterations, 120 s

Table 1: Comparative performance of the generalized Petviashvili method accelerated by ME (6.2a) and the modified CGM (3.18) for the single-component Eq. (1.5).

	Iterative method		
Case in (6.4b)	non-accelerated (5.2)	accel. by ME (6.2b)	accel. by CGM (5.8)
mildly stiff, $\Delta\tau = 1.0$	330 iterations, 310 s	120 iterations, 130 s	70 iterations, 70 s
stiffer, $\Delta\tau = 1.0$	780 iterations, 740 s	200 iterations, 220 s	130 iterations, 130 s
stiffest, $\Delta\tau = 0.9$	3330 iterations, 3110 s	550 iterations, 610 s	240 iterations, 250 s

Table 2: Comparative performance of the generalized Petviashvili method accelerated by ME (6.2b) and the modified CGM (3.18), (5.8) for the two-component Eq. (6.1).

$2^8 \times 2^8$ grid points.

Tables 1 and 2 list the numbers of iterations and time (both rounded to the nearest ten) for each of the three methods (non-accelerated generalized Petviashvili method and its two versions accelerated by ME and CGM) in the three cases (6.4) for Eqs. (1.5) and (6.1). In accordance to the note above, the value of $\Delta\tau$ is listed only for the non-accelerated method. In Fig. 3 we plot the iteration error (6.5) versus the iteration number for the stiffest case for Eq. (1.5). The error evolutions shown there are representative of all the other cases, except that in the less stiff cases, the curves corresponding to the ME and CGM are less jagged.

We also verified that a straightforward extension of the CGM, as described at the end of Section 3.1, would diverge for all cases listed above and even for non-stiff cases (not listed here).

From these tables we see that, as expected, both ME- and CGM-based accelerations considerably improve the performance of the iterative method, with the stiffer the case, the more the improvement. Furthermore, the CGM-based acceleration performs better than the ME-based one by a factor of 2–2.5; again, the stiffer the case is, the more improvement the CGM provides over the ME.

We now present similar results when the power (1.7), or its multi-component counterpart (5.9), of the solution is specified. The non-accelerated methods are the ITEMs (4.2) and (5.17) applied to the single- and multi-component Eqs. (1.5) and (6.1), respectively. When the ME-based acceleration is applied, the lines updating the intermediate solutions $\hat{u}_{n+1}$

and $\hat{\mathbf{u}}_{n+1}$ of these methods become:

$$\hat{u}_{n+1} = u_n + \left[N^{-1}\left(L^{(00)}u_n - \mu_n u_n\right) - \gamma_n^{\rm slow} \frac{\langle \phi_n^{\rm slow},\, L^{(0)}u_n\rangle}{\langle \phi_n^{\rm slow},\, N\phi_n^{\rm slow}\rangle}\, \phi_n^{\rm slow} \right] \Delta\tau\,; \tag{6.9a}$$

$$\begin{aligned} \hat{\mathbf{u}}_{n+1} = \mathbf{u}_n + \Big[\mathbf{N}^{-1}\Big(\mathbf{L}^{(\mathbf{00})}\mathbf{u}_n - \mathcal{U}_n\, \langle \mathbf{N}^{-1}\mathcal{U}_n,\, \mathcal{U}_n\rangle^{-1}\, \langle \mathbf{N}^{-1}\mathcal{U}_n,\, \mathbf{L}^{(\mathbf{00})}\mathbf{u}_n\rangle \Big) & \\ - \gamma_n^{\rm slow} \frac{\langle \mathbf{\Phi}_n^{\rm slow},\, \mathbf{L}^{(\mathbf{0})}\mathbf{u}_n\rangle}{\langle \mathbf{\Phi}_n^{\rm slow},\, \mathbf{N}\mathbf{\Phi}_n^{\rm slow}\rangle}\, \mathbf{\Phi}_n^{\rm slow} \Big] \Delta\tau\,, & \end{aligned} \tag{6.9b}$$

where $\gamma_n^{\rm slow}$ and $\mathbf{\Phi}_n^{\rm slow}$ are defined as in (6.3) and $\mathbf{L}^{(\mathbf{0})}\mathbf{u}_n$ is the term in parentheses in (6.9b); the last lines remain the same as in the respective methods (4.2) and (5.17). The modified CGMs for the single- and two-component equations are (4.14) and (5.20).

The three cases of increasing numerical stiffness are chosen as,

$$\begin{aligned} \underline{\text{For (1.5)}}: \quad \text{mildly stiff} \quad &\Rightarrow \quad V_0 = 4, \quad P = 2.1\ (\mu = 5.08); \\ \text{stiffer} \quad &\Rightarrow \quad V_0 = 4, \quad P = 1.94\ (\mu = 5.01); \\ \text{stiffest} \quad &\Rightarrow \quad V_0 = 6, \quad P = 0.92\ (\mu = 7.93); \end{aligned} \tag{6.10a}$$

$$\begin{aligned} \underline{\text{For (6.1)}}: \quad \text{mildly stiff} \quad &\Rightarrow \quad V_0 = 4, \quad P^{(1)} = 1.50\ (\mu^{(1)} = 5.10), \quad P^{(2)} = 1.00\ (\mu^{(2)} = 5.92); \\ \text{stiffer} \quad &\Rightarrow \quad V_0 = 4, \quad P^{(1)} = 0.50\ (\mu^{(1)} = 4.98), \quad P^{(2)} = 1.50\ (\mu^{(2)} = 6.62); \\ \text{stiffest} \quad &\Rightarrow \quad V_0 = 6, \quad P^{(1)} = 0.49\ (\mu^{(1)} = 7.93), \quad P^{(2)} = 0.60\ (\mu^{(2)} = 8.55)\,, \end{aligned} \tag{6.10b}$$

where the values of μ, as computed within the methods, are listed for reference only. Note that for the single-component equation, we had to choose the parameters for which the solution would satisfy the stability condition (4.5) (see Fig. 1) of the iterative methods (4.2), (6.9a), and (4.14). For the two-component equation, the parameters were chosen by trial and error.

The preconditioning operator was taken to be of the form (2.18) or, in the two-component case, as

$$\mathbf{N} = \left(c - \nabla^2\right)\, \mathrm{diag}(1,\, 1) \tag{6.11}$$

with $c = 1$ (which did not necessarily result in the optimal convergence rate). The "fictitious time" step $\Delta\tau$ for the ME-based methods was chosen as $\Delta\tau = \Delta\tau_{\rm opt} - 0.1$, as described above, where $\Delta\tau_{\rm opt}$ was the optimal step size for the non-accelerated ITEM. The computational domain and the initial conditions were taken as for the methods that find solutions with prescribed values of the propagation constant. Note that here, the modified CGMs (4.14) and (5.20) are guaranteed to converge to the solution for *any* initial conditions that resemble a two-dimensional pulse. Nonetheless we used (6.8) with $\epsilon_x = 0.1$ and $\epsilon_y = -0.2$ just for uniformity of all our simulations. For the same reason, we also started the acceleration at the same threshold (6.6), even though both the ME-based methods (6.9) and the modified CGMs (4.14) and (5.20) could be employed at the very first iteration. The results

	Iterative method		
Case in (6.10a)	non-accelerated (4.2)	accel. by ME (6.9a)	accel. by CGM (4.14)
mildly stiff, $\Delta\tau = 0.9$	330 iterations, 140 s	90 iterations, 50 s	50 iterations, 40 s
stiffer, $\Delta\tau = 1.0$	1670 iterations, 710 s	160 iterations, 100 s	120 iterations, 90 s
stiffest, $\Delta\tau = 0.6$	4690 iterations, 1980 s	550 iterations, 330 s	210 iterations, 150 s

Table 3: Comparative performance of the ITEM accelerated by ME (6.9a) and the modified CGM (4.14) for the single-component Eq. (1.5).

	Iterative method		
Case in (6.10b)	non-accelerated (5.17)	accel. by ME (6.9b)	accel. by CGM (5.20)
mildly stiff, $\Delta\tau = 0.6$	300 iterations, 260 s	120 iterations, 150 s	60 iterations, 80 s
stiffer, $\Delta\tau = 0.6$	850 iterations, 740 s	220 iterations, 280 s	120 iterations, 130 s
stiffest, $\Delta\tau = 0.5$	1610 iterations, 1400 s	380 iterations, 480 s	130 iterations, 160 s

Table 4: Comparative performance of the ITEM accelerated by ME (6.9b) and the modified CGM (5.20) for the two-component Eq. (6.1).

are summarized in Tables 3 and 4. Respective error evolutions look qualitatively similar to those in Fig. 3 and hence are not shown.

Again, as expected, we see that both ME- and CGM-based accelerations considerably improve the performance of the iterative method, with the stiffer the case, the more the improvement. Furthermore, the CGM-based acceleration performs better than the ME-based one by a factor of 2–3 for the two-component equation, with the stiffer the case, the more imrovement being provided by the CGM over the ME. Interestingly, however, for the single-component equation, the modified CGM provides an improvement over the ME only for the stiffest case; for somewhat less stiff cases, the improvement (in terms of computing time) is only marginal.

7 Conclusions and discussion

In this work, we proposed modifications of the well-known Conjugate Gradient method (CGM) that are capable of finding fundamental solitary waves of single- and multi-component Hamiltonian nonlinear equations. Our modified CGMs converge much faster than previously considered ierative methods of Richardson's type, like the (generalized) Petviashvili and imaginary-time evolution methods. Moreover, the slower the convergence of the Richardson-type method, the more speedup the modified CGM provides.

The classic CGM for a linear system of equations is known to converge when the matrix of this system is sign definite. For most solitary waves, the linearized operator about them, which is a counterpart of the matrix in the previous sentence, is not sign definite. While, in theory, this does not automatically imply that a straightforward generalization of the CGM to stationary nonlinear wave equations should fail (e.g., diverge), we verified that for the equations considered in Section 6, it does indeed fail. Then, the thrust of this work was to modify the CGM in such a way that it would be guaranteed to converge even when the linearized operator has eigenvalues of either sign. More precisely, our goal was to develop such modified versions of the CGM in the case when the number of the positive eigenvalues of that operator does not exceed the number of the components of the solitary wave. According to our "definition" at the end of Introduction, this situation would hold for fundamental solitary waves.

Not all of the modified CGMs that we proposed in Sections 3 – 5 do, strictly speaking, meet that goal. However, we show below that they come as closely as theoretically possible to doing so. Moreover, as far as their practical applications, we demonstrated that *all* of our modified CGMs can be forced to converge in the same parameter regions as the earlier proposed (and slower) iterative methods. This convergence can be achieved by a simple perturbation of the initial guess, *which has a theoretical explanation* given in Section 6. We now present more detailed summaries or each of the two versions of the modified CGM.

When finding solitary waves with prescribed values of the propagation constants, we consider an equivalent nonlinear equation whose linearized operator is modified in such a way that its only positive eigenvalue is essentially turned into a negative one; see Eqs. (3.19) and (5.8). This, however, makes the modified linearized operator "slightly" non-self-adjoint, because the generalized eigenfunction(s) of the original linearized operator, as in (3.11) and (5.4), is (are) available only approximately; see the end of Section 3.1. This is a fundamental fact about all nonlinear equations except for their very narrow subclass, equations with power-law nonlinearity (1.1) (see also their two-component generalization in [6]), and hence it, in general, cannot be improved. Thus, this fact causes "slight" non-self-adjointness of the modified linearized operator. This, in turn, leads to sign indefiniteness of the quadratic form in (3.18b,e) and thereby prevents one from obtaining a rigoroius guarantee that the modified CGM would always converge. However, we explained in Section 6 and Appendix 1 that the method can diverge *only* when the linearized operator L of the stationary wave equation has small eigenvalues, and for those cases pointed out that a mere asymmetric perturbation of the initial guess will suffice to make the modified CGM to actually converge.

For finding solitary waves with prescribed values of the power (1.7) or, more generally, quadratic conserved quantities (5.9), the situation is different. There, the modified CGM is caused to converge not by modifying a linearized operator but by making the search directions satisfy a certain orthogonality relation; see (4.3) and (5.19). The linearized operator employed by the method can be shown [12] to be self-adjoint in the space of functions satisfying those orthogonality relations, but it is negative definite only under conditions stated before Eqs. (4.5) and (5.18). This restriction is intrinsic to a method seeking a solution with a prescribed power, and hence cannot be relaxed for any nonlinear wave equations.

Thus, we have justified why the modified CGMs proposed in this work come as closely as theoretically possible to guaranteeing that they would converge to fundamental solitary waves. In practice, however, these methods can always be forced to converge, as have been mentioned above and demonstrated in Section 6.

The modified CGMs are faster not only than Richardson-type methods, but also than those methods accelerated by the slowest-decaying mode elimination (ME) technique [8, 7]. Namely, in comparison to the ME-accelerated methods, the modified CGMs provide an improvement of about a factor of two to three in terms of computing time; see Tables 1–4 in Section 6 for details. Importantly, the CGMs do so in the most numerically stiff cases, when the respective non-accelerated methods would converge extremely slowly and hence the acceleration would be most desirable. In such cases, it would pay off to use the modified CGMs instead of ME, even though the latter is a little easier to program into a code. On the other hand, in non-stiff cases, i.e. when the non-accelerated methods would converge in just a few tens of iterations, both the modified CGMs and ME would provide only modest improvement in computing time; compare (2.15) and (2.21). In such cases, it may be simpler to use ME or even the non-accelerated method. Let us point out one other, practical, advantage of the ME-based acceleration over the CGMs. Namely, the fact that we were able to construct modified CGMs is critically grounded in the existence of relations (3.19), (5.8) or (4.3), (5.19), as we explained above. On the contrary, ME can be used to accelerate *any* Richardson-type iterative method, e.g., methods from [3]–[5] or the ITEM with amplitude normalization [12]. Of course, while these methods do converge in many cases, their convergence conditions are not known, and hence their use with or without ME-based acceleration would come without a guarantee that they would converge.

Finally, let us briefly mention alternative iterative methods, other than Newton's method proper, which can be used to compute solitary waves, both fundamental and non-fundamental. First, it should be noted that a number of methods had been proposed that are related to the original CGM and are applicable when matrix A in (1.6) is real symmetric but sign indefinite. These methods include the orthogonal direction algorithm (Orthodir) [28], MINRES and SYMMLQ [29], and CG-MRES [30]. They all minimize the iteration error in a somewhat different metric than does the CGM and also require *two* previously computed search directions, d_{n-1} and d_n, to compute d_{n+1}. This allows them to avoid having small or potentially zero denominators (like $\langle Ad_n,\, d_n\rangle$ in (3.1b) and (3.1e)). Consequently, they can be employed for finding not only fundamental, but also nonfundamental solitary waves of Hamiltonian equations, whose linearized operator L has arbitrarily many eigenvalues on the "wrong" side of 0. This is a clear advantage of those methods over the modified CGMs proposed here[7]. Therefore, it is appropriate to point out some advantages of the modified CGMs over the extensions of the linear algorithms of [28]–[30] to nonlinear problems. First of all, the idea of the CGM is well known and is described in many textbooks, whereas the other methods in question can be found only either in the original papers or in advanced literature directed towards experts in numerical linear algebra (e.g., none of these methods except for the CGM are described in [18, 19]). Second, programming of these methods, except for the Orthodir, is more complicated than for the CGM and in all cases, requires more operations per iteration. (Here, however, the advantage of the modified CGMs is not apparent since they also require more computational effort than the original CGM.) Third, as can be concluded from the estimates found in [31] for MINRES and in [30] for CG-MRES, the convergence rates of these methods are on the order of that of the CGM (see (2.21)) provided that the linearized operator L has only one positive eigenvalue[8]. However, when it has two or more positive eigenvalues and the minimum-modulus eigenvalues on both sides of 0 are small — which is precisely when one desires to considerably accelerate the iterations — the convergence rates of MINRES and CG-MRES can greatly decrease to a value of order (2.15). In such a case, a simpler method described in the next paragraph can be used instead. Thus, a detailed comparison of the modified CGM and the methods of [28]–[30] is an open issue.

Let us also note that in a recent numerical study [32], Yang showed that a certain combination of the CGM and Newton's method converged to both fundamental and non-fundamental solitary waves for all of the examples considered in that study. It is remarkable, and at the moment has no rigorous explanation, that the straightforward generalization of the CGM outlined at the end of Section 3.1 was found to diverge for most of the same examples for which Yang's method converges. Thus, Yang's method can be used in practice; however, it should be kept in mind that there may exist situations where it would diverge. Alternatively, if one seeks a non-fundamental solitary wave, one can "square" the equation (see (2.8) and (2.9)) and apply the original CGM to it. For linear systems, this trick has long been known (see, e.g., [18], p. 304), and some researchers used it for finding non-fundamental solitary waves [33]. Let us note, however, that "squaring" an equation leads to squaring the condition number of the corresponding linearized operator (since $\text{cond}(A^2) = (\text{cond}(A))^2$), which, according to (2.21), slows down the convergence; also, more arithmetic operatons per iteration are required for a "squared" method.

[7] Strictly speaking, our modified CGM can, in some cases like Example 1 in [8], be extended to converge for the lowest excited states, but such an extension would come at the expense of additional programming effort and hence may not be worth it.

[8] This author could not find convergence rate estimates for Orthodir and SYMMLQ.

Acknowledgement

This work at its early stage was supported in part by the National Science Foundation under grant DMS-0507429. The author thanks Dr. J. Yang for useful discussions, Mr. N. Jones for assistance with computations at the early stage of this work, and the anonymous referees for providing constructive criticism that helped improve this paper.

Appendix 1: Technical results for Sections 3 and 4

Here we will first prove statements (i)–(iii) found in the first paragraph of Section 3.3 and, in this process, also explain the choice (3.16b) for the constant Γ. Then we will prove that the fact that L may have a zero eigenvalue due to a translational eigenmode will not cause the CGM to break down. Finally, we will supply missing steps in the derivations of (4.12) and (4.13).

From (3.16a),

$$K^{(0,\mathrm{mod})}v = 0 \qquad \Rightarrow \qquad K^{(0)}v = \chi v, \quad \chi \equiv \Gamma \frac{\langle v, K^{(0)}v\rangle}{\langle v, v\rangle}\,. \tag{A1.1}$$

Substituting the second equation in (A1.1) into the definition of χ we find:

$$\chi = \Gamma \frac{\langle v, \chi v\rangle}{\langle v, v\rangle} = \chi\Gamma\,. \tag{A1.2}$$

Since $\Gamma \neq 1$ (see (3.16b)), Eq. (A1.2) implies that $\chi = 0$, which along with the second equation in (A1.1) shows that $(K^{(0)}v = 0) \Leftrightarrow (K^{(0,\mathrm{mod})}v = 0)$. This proves statement (i).

Statement (ii), i.e. that $K^{(\mathrm{mod})}$ defined in (3.16c) is approximately self-adjoint, follows by considering the following inner product for arbitrary functions f and g:

$$\begin{aligned}
\left\langle f, K^{(\mathrm{mod})}g\right\rangle &= \langle f, Kg\rangle - \Gamma\frac{\langle v, Kg\rangle}{\langle v, v\rangle}\langle f, v\rangle \\
&\approx \langle f, Kg\rangle - \Gamma\frac{\langle v, \lambda^{(1)}g\rangle\,\langle v, f\rangle}{\langle v, v\rangle} \\
&\approx \langle g, Kf\rangle - \Gamma\frac{\langle v, g\rangle\,\langle v, Kf\rangle}{\langle v, v\rangle} \\
&= \left\langle g, K^{(\mathrm{mod})}f\right\rangle\,.
\end{aligned} \tag{A1.3}$$

The two approximate equalities above are due to the approximate relation (3.11), and we have also used the fact that K is self-adjoint.

Statement (iii), i.e. that all eigenvalues of $K^{(\mathrm{mod})}$ that *are not too close to zero* are negative, is a counterpart for nonlinear equations of the statement found after the linear Eq. (2.13b). Here a quantitative measure of being "too close to zero" is related to how close the approximate relation (3.11) (or, equivalently, (3.7)) is to being exact; see (3.14) and the text around it. Below we demonstrate statement (iii) in several steps. First, due to (3.11), one eigenfunction, $\psi^{(1)}$, of $K^{(\mathrm{mod})}$ is close to v, and for it,

$$K^{(\mathrm{mod})}\psi^{(1)} \approx K^{(\mathrm{mod})}v \approx \lambda^{(1)}(1-\Gamma)v \approx \lambda^{(1)}(1-\Gamma)\psi^{(1)}\,. \qquad \text{(A1.4)}$$

When Γ is chosen according to (3.16b), the eigenvalue corresponding to $\psi^{(1)}$ is approximately -1, which is near the accumulation point of the spectrum of K [12]. It is convenient to "place" this eigenvalue inside (as opposed to at the edge of) the spectrum of $K^{(\mathrm{mod})}$ because then it does not affect the condition number of the operator and hence the convergence factor of the iterative method; see (2.16), (2.15), (2.21). Thus, we have shown that the "main culprit" of sign indefiniteness of K — the eigenvalue $\lambda^{(1)} > 0$ — has been successfully dealt with, i.e. made substantially negative.

To finish the proof of statement (iii), let us show that $K^{(\mathrm{mod})}$ can acquire small positive eigenvalue(s) only if the original linearized operator L has near-zero eigenvalues. To this end, we first note that small eigenvalues of L imply small eigenvalues of K (and vice versa). Indeed, the eigenvalues of $K = N^{-1/2}LN^{-1/2}$ are the same as those of $N^{-1}L$. According to the Sylvester law of inertia [24], eigenvalues of $N^{-1}L$ and L have the same signs. Moreover, for reasonable choices of N (i.e., for $c = O(1)$ in (2.18)), the eigenvalues of both operators are of the same order of magnitude. Next, let $\psi^{(k)},\ k \geq 2$ be eigenfunctions of K with negative eigenvalues. Since v is only an approximate eigenfunction of the self-adjoint operator K, then $\langle v, K\psi^{(k)}\rangle = \langle Kv, \psi^{(k)}\rangle$ differs slightly from zero, which causes the eigenfunctions, and hence eigenvalues, of $K^{(\mathrm{mod})}$ to differ slightly from those of K. Hence small negative eigenvalues of K can, in principle, become small positive eigenvalues of $K^{(\mathrm{mod})}$ [9]. However, those negative eigenvalues of K that are "not small" cannot be made positive eigenvalues of $K^{(\mathrm{mod})}$ by a small term $\langle v,\, K\psi^{(k)}\rangle$ in (3.16c). This concludes our demonstration of statement (iii).

Let us note that even if $K^{(\mathrm{mod})}$ has only negative eigenvalues, the quadratic form $\langle \bar{d}_n,\, K^{(\mathrm{mod})}\bar{d}_n\rangle$ may still not be negative definite because $K^{(\mathrm{mod})}$ is not self-adjoint. Since, however, $K^{(\mathrm{mod})}$ is only *"slightly"* non-self-adjoint, the sign indefiniteness of the above quadratic form can occur only when some eigenvalues of $K^{(\mathrm{mod})}$ are close to zero, as illustrated by the following 2×2 example:

$$(3\epsilon \;\; 1)\begin{pmatrix} -1 & 5\epsilon \\ 0 & -4\epsilon^2 \end{pmatrix}\begin{pmatrix} 3\epsilon \\ 1 \end{pmatrix} = 2\epsilon^2 > 0\,. \qquad \text{(A1.5)}$$

Note that, for $\epsilon \ll 1$, the vector $(3\epsilon \;\; 1)^T$ in (A1.5) is aligned primarily with the eigenvector, $(5\epsilon \;\; 1)^T$, corresponding to the smaller eigenvalue of the matrix. The practical implication of the sign indefiniteness of the quadratic form $\langle \bar{d}_n,\, K^{(\mathrm{mod})}\bar{d}_n\rangle$ is that it, and hence the

[9] This, however, did *not* appear to be the case in any of the numerical experiments conducted in [6] and in this paper. Indeed, if the corresponding $K^{(\mathrm{mod})}$ had acquired small positive eigenvalues, then the generalized Petviashvili method (3.6) would diverge (see the text near the end of Section 3.1), which was never encountered in the aforementioned numerical experiments.

denominators in (3.18b,e), can become arbitrarily close to zero. This may cause divergence of the modified CGM (3.18).

Let us now show that if operator L has an eigenmode ψ_{trans} corresponding to translational invariance of the solitary wave, this would not cause a breakdown of algorithm (3.18). As before, we will perform the analysis in transformed variables (3.10) and (3.17) and in the linear approximation with respect to $\tilde{v}_n$, $\bar{r}_n$, and $\bar{d}_n$.

The accordingly transformed translational eigenmode satisfies

$$K^{(\text{mod})}\bar{\psi}_{\text{trans}} = 0\,. \tag{A1.6}$$

Let us also write down the counterparts, respectively numbered, of the lines of algorithm (3.1) that we will refer to:

$$\bar{r}_{n+1} = K^{(0,\text{mod})}v_{n+1} = K^{(\text{mod})}\tilde{v}_{n+1}\,. \tag{A1.7d}$$

$$\bar{d}_{n+1} = \bar{r}_{n+1} + \beta_n \bar{d}_n\,. \tag{A1.7f}$$

$$\alpha_{n+1} = -\frac{\langle \bar{r}_{n+1}, \bar{d}_{n+1}\rangle}{\langle K^{(\text{mod})}\bar{d}_{n+1}, \bar{d}_{n+1}\rangle}\,. \tag{A1.7b}$$

We will show by contradiction that if $\bar{d}_n$ is not proportional to $\bar{\psi}_{\text{trans}}$ then neither will be $\bar{d}_{n+1}$. Then, clearly, since all the eigenvalues of $K^{(\text{mod})}$ other than $\bar{\psi}_{\text{trans}}$ are negative, the denominator in (A1.7b) will never vanish and hence the CGM will not break down. First, the solvability condition (Fredholm alternative) for (A1.7d) implies that

$$\bar{r}_{n+1} = \bar{\psi}_{\perp}, \qquad \text{where } \langle \bar{\psi}_{\perp}, \bar{\psi}_{\text{trans}}\rangle = 0, \tag{A1.8}$$

and hence $\beta_n \neq 0$ since $K^{(\text{mod})}\bar{d}_n \neq 0$. Second, from (3.2) and (A1.8) it follows that

$$\langle \bar{d}_n, \bar{\psi}_{\perp}\rangle = 0\,. \tag{A1.9}$$

Third, let us assume that for some n, $\bar{d}_{n+1} = a\bar{\psi}_{\text{trans}}$. Note that by (3.2) and (A1.7f), $a \neq 0$. Then from (A1.7f) it follows that

$$\bar{d}_n = \frac{a\bar{\psi}_{\text{trans}} - \bar{\psi}_{\perp}}{\beta_n} \qquad \Rightarrow \qquad \langle \bar{d}_n, \bar{\psi}_{\perp}\rangle = \frac{a}{\beta_n}\langle \bar{\psi}_{\perp}, \bar{\psi}_{\perp}\rangle \neq 0. \tag{A1.10}$$

This contradicts (A1.9), and hence $\bar{d}_{n+1}$ can never become proportional to $\bar{\psi}_{\text{trans}}$, which proves our assertion. Note that if $\tilde{v}_{n+1}$ becomes proportional to $\bar{\psi}_{\text{trans}}$, then $\bar{r}_{n+1} = 0$, in which case the iterations converge to $(v + \text{const} \cdot \bar{\psi}_{\text{trans}})$, which is just a translated solitary wave v.

Finally, let us show that (4.12) and (4.13) hold. Since, as we noted in Section 4.2, the second equation in (4.9c) does not change the linearized form of the first equation, we can use $\hat{v}_{n+1}$ instead of v_{n+1} in (4.9d). Then

$$\bar{r}_{n+1} = \bar{r}_n + \alpha_n \mathcal{K}\bar{d}_n + O(\tilde{v}_n^2)\,. \tag{A1.11}$$

Along with (4.9b), this implies (4.12). Next, the only step that requires some explanation in the derivation of (4.13) is the omission of $\langle \bar{d}_n,\, N^{-1}v_{n+1}\rangle$. By (4.10), this inner product is

$$\langle \bar{d}_n,\, N^{-1}v_{n+1}\rangle = \langle \bar{d}_n,\, N^{-1}v_n\rangle + O(\tilde{v}_n^2) \;=\; O(\tilde{v}_n^2)\,, \tag{A1.12}$$

so that its omission does not change the accuracy implied by the right-hand side of (4.13).

Appendix 2: An alternative modified CGM for Section 3

Since, for the reasons explained at the end of Section 3, we do not employ this algorithm for numerical examples presented in this paper, below we will present it only in terms of the transformed variables defined in (3.10) and (3.17).

$$\bar{r}_0 = K^{(0)}v_0, \qquad \bar{d}_0 = \bar{r}_0 - \frac{\langle \bar{r}_0, v_0\rangle}{\langle v_0, v_0\rangle}v_0, \tag{A2.1a}$$

$$\alpha_n = -\frac{\langle \bar{r}_n,\, \bar{d}_n\rangle - \dfrac{\langle v_n, K\bar{d}_n\rangle\langle \bar{r}_n,\, v_n\rangle}{\lambda^{(1)}\langle v_n,\, v_n\rangle}}{\langle \bar{d}_n,\, K\bar{d}_n\rangle}, \tag{A2.1b}$$

$$v_{n+1} = v_n + \alpha_n\bar{d}_n - \frac{\langle v_n,\, K^{(0)}v_n\rangle}{\lambda^{(1)}\langle v_n,\, v_n\rangle}v_n, \tag{A2.1c}$$

$$\bar{r}_{n+1} = K^{(0)}v_{n+1} \tag{A2.1d}$$

$$\beta_n = -\frac{\langle \bar{r}_{n+1},\, K\bar{d}_n\rangle - \dfrac{\langle v_n, K\bar{d}_n\rangle\langle \bar{r}_{n+1},\, v_{n+1}\rangle}{\langle v_{n+1},\, v_{n+1}\rangle}}{\langle \bar{d}_n,\, K\bar{d}_n\rangle}, \tag{A2.1e}$$

$$\bar{d}_{n+1} = \bar{r}_{n+1} + \beta_n\bar{d}_n - \frac{\langle v_{n+1},\, \bar{r}_{n+1} + \beta_n\bar{d}_n\rangle}{\langle v_{n+1}, v_{n+1}\rangle}v_{n+1}\,. \tag{A2.1f}$$

The definitions of $\bar{d}_0$ in (A2.1a) and $\bar{d}_{n+1}$ in (A2.1f) ensure that

$$\langle \bar{d}_n, v_n\rangle = 0 \tag{A2.2}$$

at every iteration. Equation (A2.1c) ensures that $v_n = v + \tilde{v}_n$ where

$$\langle \tilde{v}_n, v\rangle \to 0 \quad \text{as } n \text{ increases}\,. \tag{A2.3}$$

Equations (A2.1b–d) entail the counterpart of (3.2):

$$\langle \bar{r}_{n+1},\, \bar{d}_n\rangle = O(\tilde{v}_n^3)\,, \tag{A2.4}$$

and Eqs. (A2.1e,f) entail a counterpart of (3.3):

$$\langle \bar{d}_{n+1},\, K\bar{d}_n\rangle = O(\tilde{v}_n^3)\,. \tag{A2.5}$$

The last three relations can be proved similarly to relations (4.12) and (4.13).

Appendix 3: Sample code of modified CGM for a two-component solitary wave with prescribed propagation constants

We will first present the code which uses the modified CGM (3.18), (5.8) to find a solitary wave of Eqs. (6.1), then explain some of its steps, and, finally, discuss how this code could be simplified. This code can be downloaded from
`http://www.cems.uvm.edu/~lakobati/posted_papers_and_codes/code_in_Appendix3.m`.

```
% ----- Spatial and spectral domains:  ------------------------------------------
xlength=12*pi; ylength=12*pi;         % domain lengths along x and y
Nx=2^8;         Ny=Nx;                % number of points along x and y
dx=xlength/Nx;  dy=dx;                % mesh sizes along x and y
x=[-xlength/2:dx:xlength/2-dx]; y=x;  % domains along x and y
[X,Y]=meshgrid(x,y);                  % X and Y arrays of size Ny-by-Nx
kx=2*pi/xlength*[0:Nx/2-1  -Nx/2:-1]; ky=kx;  % spectral domains along x and y
[KX,KY]=meshgrid(kx,ky);              % KX and KY arrays of size Ny-by-Nx
DEL(:,:,1)=-(KX.^2+1*KY.^2); DEL(:,:,2)=DEL(:,:,1); % Fourier symbol of Laplacian
% ----- Coefficients in the equation:  ------------------------------------------
mu(1)=7.89;  mu(2)=8.5;               % mu values
W(:,:,1)=6*( (cos(X)).^2 + (cos(Y)).^2 ) - mu(1);
W(:,:,2)=6*( (cos(X)).^2 + (cos(Y)).^2 ) - mu(2); % potential minus mu
F(1)=1; F12=0.5; F(2)=4;              % nonlinearity coefficients
Dt=0.9;                                 % Delta tau
% ----- Initial condition:             ------------------------------------------
u(:,:,1)= 0.8*exp(-1*(X.^2 + Y.^2)).*(1+0.1*X-0.2*Y);;
u(:,:,2)= 1.5*exp(-1*(X.^2 + Y.^2)).*(1+0.1*X-0.2*Y);;
% ----- Loop control variables:        ------------------------------------------
normDu=1;           % initialize the error norm to start the loop
normDu_accel=0.05;  % Acceleration begins when the error reaches this threshold;
                    % parameters c, b, gamma are computed until this threshold.
accelerate=0;       % this marker indicates that the acceleration has started
counter=0;          % counter of the number of iterations
% ----- Iterate until the error reaches the prescribed tolerance  ---------------
while normDu >= 10^(-10)
  counter=counter+1;
  if normDu >= normDu_accel & accelerate == 0    % compute N, E2, gammas  -------
    %  STEP 1:  Compute parameters of N:  b(2), c(1), c(2)  ~~~~~~~~~~~~~~~~~~~~~
    DELu=real( ifft2(DEL.*fft2(u)) );    usq=u.^2;
    for k=1:2
```

```
  L0u(:,:,k)=DELu(:,:,k)+(W(:,:,k)+F(k)*usq(:,:,k)+F12*usq(:,:,3-k)).*u(:,:,k);
  u_L0u(k)=trapz(trapz( u(:,:,k).*L0u(:,:,k) ));  % <u, L0u> componentwise
end
L11u1=L0u(:,:,1)+2*F(1)*usq(:,:,1).*u(:,:,1); L12u2=2*F12*u(:,:,1).*usq(:,:,2);
L22u2=L0u(:,:,2)+2*F(2)*usq(:,:,2).*u(:,:,2); L21u1=2*F12*u(:,:,2).*usq(:,:,1);
SumjLkjEj1(:,:,1)=L11u1+L12u2-L0u(:,:,1);
SumjLkjEj1(:,:,2)=L21u1+L22u2-L0u(:,:,2);
for k=1:2
  u_SumjLkjEj1(k)=trapz(trapz( u(:,:,k).*SumjLkjEj1(:,:,k) ));
  DELu_SumjLkjEj1(k)=trapz(trapz( DELu(:,:,k).*SumjLkjEj1(:,:,k) ));
  u_u(k)=trapz(trapz( u(:,:,k).*u(:,:,k) ));
  u_DELu(k)=trapz(trapz( u(:,:,k).*DELu(:,:,k) ));
  DELu_DELu(k)=trapz(trapz( DELu(:,:,k).*DELu(:,:,k) ));
end
b(1)=1;
for k=1:2
  kappa(k)=( u_DELu(k)*DELu_SumjLkjEj1(k)-DELu_DELu(k)*u_SumjLkjEj1(k) )/...
           ( u_u(k)*DELu_SumjLkjEj1(k)-u_DELu(k)*u_SumjLkjEj1(k) );
end
b(2)=b(1)*(kappa(1)*u_u(1)-u_DELu(1)) * u_SumjLkjEj1(2) / ...
          ( (kappa(2)*u_u(2)-u_DELu(2)) * u_SumjLkjEj1(1) );
c=b.*kappa;
u_Nu=c.*u_u-b.*u_DELu;   % <u, Nu> componentwise
%  STEP 2: Compute eigenvector E2 = [rho2(1)*u1 rho2(2)*u2]^T  and  fft(N) ~~~
rho2(1)=-u_Nu(2)/u_Nu(1);   rho2(2)=1;
rho1(1)=1;  rho1(2)=1;  % these coefficients of E1 are for uniform notations
for k=1:2
  E1(:,:,k)=rho1(k)*u(:,:,k);    E2(:,:,k)=rho2(k)*u(:,:,k);
  fftN(:,:,k)=c(k)-b(k)*DEL(:,:,k);  % Fourier symbol of N componentwise
end
%  STEP 3: Compute gammas   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
E_NE(1)=sum( (rho1.^2).*u_Nu );
E_NE(2)=sum( (rho2.^2).*u_Nu );
SumjLkjEj2(:,:,1)=rho2(1)*L11u1+rho2(2)*L12u2-L0u(:,:,1);
SumjLkjEj2(:,:,2)=rho2(1)*L21u1+rho2(2)*L22u2-L0u(:,:,2);
E_LE(1)=u_SumjLkjEj1(1)+u_SumjLkjEj1(2);
E_LE(2)=sum( trapz(trapz( E2.*SumjLkjEj2 )) );
lambda=E_LE./E_NE;   gamma=1+1./(lambda*Dt);
```

```
  %  STEP 4:  Update the solution:    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
  E_L0u(1)=rho1(1)*u_L0u(1)+rho1(2)*u_L0u(2);
  E_L0u(2)=rho2(1)*u_L0u(1)+rho2(2)*u_L0u(2);
  u = u + Dt*( real( ifft2(fft2(L0u)./fftN) ) - ...
               gamma(1)*E_L0u(1)/E_NE(1)*E1 - gamma(2)*E_L0u(2)/E_NE(2)*E2 );
else  % ----- use last computed N,E1,E2,lambda,gamma and start CGM  -------------
  if accelerate == 0
    accelerate = 1;
    for k=1:2
      L0u(:,:,k)=real( ifft2(DEL(:,:,k).*fft2(u(:,:,k))) ) + ...
                 (W(:,:,k)+F(k)*u(:,:,k).^2+F12*u(:,:,3-k).^2).*u(:,:,k);
      u_L0u(k)=trapz(trapz( u(:,:,k).*L0u(:,:,k) ));
      GAMMA(k)=1+1/lambda(k);
    end
    E_L0u(1)=rho1(1)*u_L0u(1)+rho1(2)*u_L0u(2);
    E_L0u(2)=rho2(1)*u_L0u(1)+rho2(2)*u_L0u(2);
    NE1=real( ifft2(fft2(E1).*fftN) );
    NE2=real( ifft2(fft2(E2).*fftN) );
    LL0u=L0u-GAMMA(1)*E_L0u(1)/E_NE(1)*NE1-GAMMA(2)*E_L0u(2)/E_NE(2)*NE2;
    r=real( ifft2(fft2(LL0u)./fftN) );    d=r;
  end
  for k=1:2
    Ld(:,:,k)=real( ifft2(DEL(:,:,k).*fft2(d(:,:,k))) ) + ...
              (W(:,:,k)+3*F(k)*u(:,:,k).^2+F12*u(:,:,3-k).^2).*d(:,:,k)+...
              2*F12*u(:,:,k).*u(:,:,3-k).*d(:,:,3-k);
  end
  E1_Ld=sum( trapz(trapz(E1.*Ld)) );   E2_Ld=sum( trapz(trapz(E2.*Ld)) );
  LLd=Ld-GAMMA(1)*E1_Ld/E_NE(1)*NE1-GAMMA(2)*E2_Ld/E_NE(2)*NE2;
  Nr_d=sum( trapz(trapz(LL0u.*d)) );   LLd_d=sum( trapz(trapz(d.*LLd)) );
  alpha = -Nr_d/LLd_d;
  u = u+alpha*d;
  for k=1:2
    L0u(:,:,k)=real( ifft2(DEL(:,:,k).*fft2(u(:,:,k))) ) + ...
               (W(:,:,k)+F(k)*u(:,:,k).^2+F12*u(:,:,3-k).^2).*u(:,:,k);
    u_L0u(k)=trapz(trapz( u(:,:,k).*L0u(:,:,k) ));
  end
  E_L0u(1)=rho1(1)*u_L0u(1)+rho1(2)*u_L0u(2);
  E_L0u(2)=rho2(1)*u_L0u(1)+rho2(2)*u_L0u(2);
```

```
    LL0u=L0u-GAMMA(1)*E_L0u(1)/E_NE(1)*NE1-GAMMA(2)*E_L0u(2)/E_NE(2)*NE2;
    r = real( ifft2(fft2(LL0u)./fftN) );
    beta =  max(-sum( trapz(trapz(r.*LLd)) )/LLd_d, 0);
    d = r + beta*d;
  end
  normDu= norm(L0u(:,:,1))/norm(u(:,:,1))+norm(L0u(:,:,2))/norm(u(:,:,2));
  normDu_recorded(counter)=normDu;
end
figure(1); mesh(x,y,u(:,:,1)); figure(2); mesh(x,y,u(:,:,2));
figure(3); plot([1:counter],log10(normDu_recorded))
```

Step 1 inside the while-loop implements Eqs. (4.12) and (4.13) (see also (4.21)) of [6] using the notations of that paper. For example, DELu, L0u, SumjLkjEj1 in the code stand, respectively, for $\nabla^2\mathbf{u}_n$, $\mathbf{L}_0\mathbf{u}_n$, $\sum_{j=1}^{2} L_{kj}e_{j1}$; note that all these quantities are two-dimensional vectors. Notations involving the underscore denote inner products; for example, u_L0u(k) denotes $\langle u_n^{(k)}, (\mathbf{L}_0\mathbf{u})_n^{(k)}\rangle$, $k = 1, 2$, where the superscript (k) means the kth component of the two-dimensional vector.

Step 2 implements Eqs. (4.15)–(4.18) of [6]. Here rho1(k) denotes $\rho^{(1,k)}$ in the notations of Eq. (5.6) of *this* paper; note that, for the convenience of coding, the order of the superscripts is reversed compared to [6].

Step 3 implements Eqs. (4.5) of [6]. Note that α_k of [6] is denoted as $\lambda^{(k)}$ in this paper. Also note that in the code, quantities like E_NE(1) denote $\langle \mathbf{e}^{(1)}, \mathbf{N}\,\mathbf{e}^{(1)}\rangle$, i.e. here, the superscript refers to the particular eigenvector $\mathbf{e}^{(k)}$ of $\mathbf{N}^{-1}\mathbf{L}$.

Step 4 implements Eq. (4.19) of [6].

At the first iteration of the CGM, in addition to implementing Eqs. (3.18a) of this paper, one also computes $\Gamma^{(k)}$ and $\mathbf{N}\mathbf{e}^{(k)}$, $k = 1, 2$, so that these quantities are *not* computed again at subsequent iterations of the CGM. Notations LL0u and LLd stand for $\mathbf{L}^{(\mathbf{0},\mathrm{mod})}\mathbf{u}_n$ and $\mathbf{L}^{(\mathrm{mod})}\mathbf{d}_n$. The remainder of the code implements Eqs. (3.18) and (5.8) of this paper.

Note that the CGM iterations start when $\mathbf{N}$, $\mathbf{e}^{(k)}$, and $\Gamma^{(k)}$ are found rather imprecisely: accuracy (i.e., the value of normDu_accel) of 5% is used in the above code and in the examples reported in Section 6, and we verified that the code still worked when this accuracy was lowered to as much as 10%. Moreover, even if $\mathbf{N}$ etc. were computed up to a higher accuracy, the functions $\mathbf{e}^{(k)}$ could still satisfy the eigenrelations (5.4) only approximately. While this approximation is very close (99% in the least-squares sense) for $\mathbf{e}^{(1)}$, it is only about 70% for $\mathbf{e}^{(2)}$; see the second column in Table 1 of [6]. These considerations suggest that the code could still work if the $\mathbf{N}$ given by the expression above Eq. (5.3) is replaced by a simpler expression (6.11). The constant c there should be computed by a straightforward generalization of Eq. (3.11) of [6], whereby the lines in Step 1 of the above code starting with the second for-loop through the end of that step are replaced by:

```
for k=1:2
  u_u(k)=trapz(trapz( u(:,:,k).*u(:,:,k) ));       % <u, u>  componentwise
  u_DELu(k)=trapz(trapz( u(:,:,k).*DELu(:,:,k) )); % <u, DELu>  componentwise
end
DELu_DELu=sum( trapz(trapz( DELu.*DELu )) );  % <DELu, DELu> as a scalar, etc
u_SumjLkjEj1=sum( trapz(trapz( u.*SumjLkjEj1 )) );
DELu_SumjLkjEj1=sum( trapz(trapz( DELu.*SumjLkjEj1 )) );
c = ( u_SumjLkjEj1*DELu_DELu - DELu_SumjLkjEj1*sum(u_DELu) )/...
    ( u_SumjLkjEj1*sum(u_DELu) - DELu_SumjLkjEj1*sum(u_u) );
u_Nu=c*u_u-u_DELu;        % <u, Nu> componentwise
```

Note that the reason why the inner products u_u(k) and u_DELu(k) are computed for each component of $\mathbf{u}_n$ rather than for the entire vector function, as DELu_DELu etc, is that they are needed to compute the individual components of u_Nu. The latter are needed to compute $\rho^{(2,1)}$ in Step 2.

We did not test the performance of such a simplified code because the goal of this paper is the *proposal* of modified CGMs for solitary waves and *not optimization* of those methods.

References

[1] V.I. Petviashvili, "Equation for an extraordinary soliton," Sov. J. Plasma Phys. **2**, 257–258 (1976).

[2] D.E. Pelinovsky and Yu.A. Stepanyants, "Convergence of Petviashvili's iteration method for numerical approximation of stationary solutions of nonlinear wave equations," SIAM J. Numer. Anal. **42**, 1110–1127 (2004).

[3] Z.H. Musslimani and J. Yang, "Localization of light in a two-dimensional periodic structure," J. Opt. Soc. Am. B **21**, 973–981 (2004).

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

[5] Y.A. Stepanyants, I.K. Ten, and H. Tomita, "Lump solutions of 2D generalized Gardner equation," in: A.C.J. Luo, L. Dai, H.R. Hamidzadeh (Eds.), *Nonlinear Science and Complexity*, World Scientific, Hackensack, NJ, 2007, pp. 264–271.

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

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

[8] T.I. Lakoba and J. Yang, “A mode elimination technique to improve convergence of iteration methods for finding solitary waves,” J. Comp. Phys. **226**, 1693–1709 (2007).

[9] J.J. Garcia-Ripoll and V.M. Perez-Garcia, “Optimizing Schrödinger functionals using Sobolev gradients: Applications to Quantum Mechanics and Nonlinear Optics,” SIAM J. Sci. Comput. **23**, 1316–1334 (2001).

[10] W. Bao and Q. Du, “Computing the ground state solution of Bose–Einstein condensates by a normalized gradient flow,” SIAM J. Sci. Comput. **25**, 1674–1697 (2004).

[11] V.S. Shchesnovich and S.B. Cavalcanti, “Rayleigh functional for nonlinear systems,” available at `http://www.arXiv.org`, Preprint nlin.PS/0411033.

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

[13] M. Struwe, *Variational Methods: Applications to Nonlinear Partial Differential Equations and Hamiltonian Systems* (3rd ed.), Springer, Berlin, 2000. [Specifically, see the paragraph just below Theorem B.4 on page 246.]

[14] S.V. Manakov, V.E. Zakharov, L.A. Bordag, A.R. Its, and V.B. Matveev, “Two-dimensional solitons of the Kadomtsev–Petviashvili equation and their interaction,” Phys. Lett. A **63**, 205–206 (1977).

[15] C. Sulem and P.-L. Sulem, *Nonlinear Schrödinger Equations: Self-Focusing and Wave Collapse*, Springer, New York, 1999; p. 72.

[16] Y. Liu, “On the stability of solitary waves for the Ostrovsky equation,” Quart. Appl. Math. **65**, 571–589 (2007).

[17] Y. Sivan, G. Fibich, B. Ilan, and M.I. Weinstein, “Qualitative and quantitative analysis of stability and instability dynamics of positive lattice solitons,” Phys. Rev. E **78**, 046602 (2008).

[18] L.N. Trefethen and D. Bau, III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[19] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1996.

[20] J. Yang and Z. Chen, “Defect solitons in photonic lattices,” Phys. Rev. E **73**, 026609 (2006).

[21] T.I. Lakoba, "Convergence conditions for iterative methods seeking multi-component solitary waves with prescribed quadratic conserved quantities," submitted.

[22] G. Markham, "Conjugate gradient type methods for indefinite, asymmetric, and complex systems," IMA J. Numer. Anal. **10**, 155–170 (1990).

[23] J.R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Carnegie Mellon University, Technical Report: CS-94-125 (1994); available at `http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf`.

[24] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, 1991. [Specifically, see Theorems 4.5.8 and 7.6.3.]

[25] T.I. Lakoba and D.J. Kaup, "Stability of solitons in nonlinear fiber couplers with two orthogonal polarizations," Phys. Rev. E **56**, 4791–4802 (1997).

[26] Yu.N. Karamzin, A.P. Sukhorukov, and T.S. Filipchuk, "New class of coupled solutions in dispersive media with quadratic nonlinearity," Vestn. Mosk. Univ. Ser. 3: Fiz., Astron. **19**, 91–98 (1978) [in Russian]; (English translation: Moscow Univ. Phys. Bull., **33**, 73 (1978)).

[27] Z. Shi and J. Yang, "Solitary waves bifurcated from Bloch-band edges in two-dimensional periodic media," Phys. Rev. E **75**, 056602 (2007).

[28] R. Fletcher, "Conjugate gradient methods for indefinite systems," *Numerical analysis (Proc 6th Biennial Dundee Conf., 1975)*, (G.A. Watson, Ed.), Lecture Notes in Math., Vol. 506, Springer, Berlin, 1976; pp. 73–89.

[29] C.C. Paige and M.A. Saunders, "Solution of sparse indefinite systems of linear equations," SIAM J. Numer. Anal. **12**, 617–629 (1975).

[30] Y.L. Lai, W.W. Lin, and D. Pierce, "Conjugate gradient and minimal residual method for solving symmetric indefinite systems," J. Comp. Appl. Math. **84**, 243–256 (1997).

[31] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997. [Specifically, see p. 54.]

[32] J. Yang, "Conjugate-gradient methods for solitary wave computations," submitted.

[33] A.A. Sukhorukov, private communication.

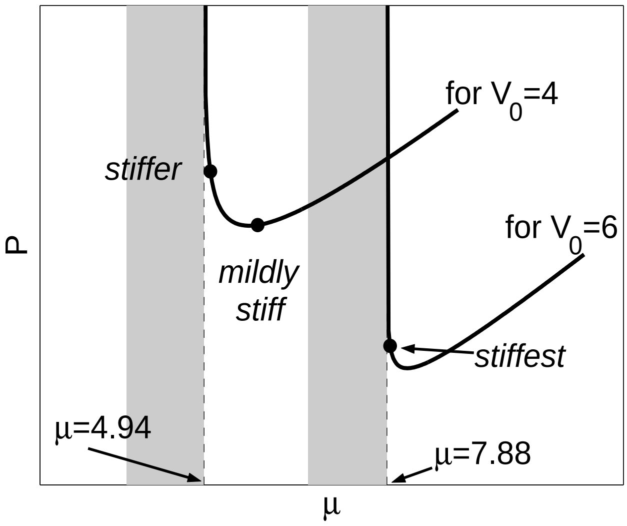

Figure 1: Schematics of the $P(\mu)$ curves for fundamental solitary waves of Eq. (1.5) with $V_0 = 4$ and $V_0 = 6$. The axes, markers, etc. are drawn not to scale. The three cases of increased numerical stiffness, listed in (6.4a), are labeled with filled circles. The shaded areas represent the first spectral bands of the linearized operator L for the two values of V_0.

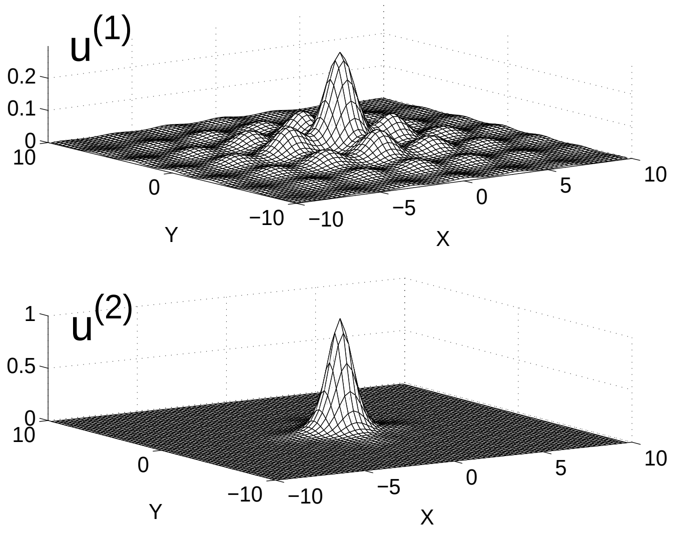

Figure 2: The two-component solution of the stiffer case in (6.4b) for Eq. (6.1). Note the different vertical scales of the two components.

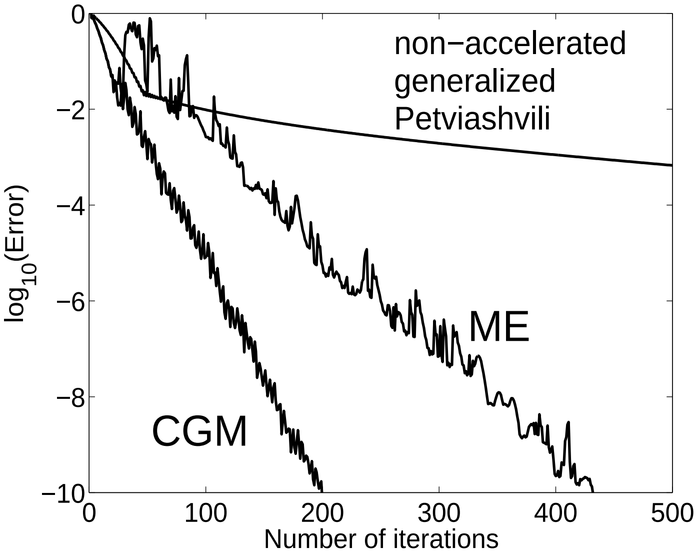

Figure 3: Evolutions of the iteration error, defined by (6.5) with $S = 1$, for the stiffest case (6.4a) for Eq. (1.5). The methods referred to in the plot are: the non-accelerated generalized Petviashvili method (3.6), its ME-accelerated form (6.2a), and the modified CGM (3.18), (3.19). The curve for the non-accelerated generalized Petviashvili method is not shown in full so as not to obscure the details of the other two curves.