Numerical approximations of localized modes in periodic potentials

James Brown*
Department of Physics and Astronomy, McMaster University, Hamilton, Ontario, Canada L8S 4M1

Supervisor: Dmitry Pelinovsky†
Department of Mathematics and Statistics, McMaster University, Hamilton, Ontario, Canada L8S 4K1
(Dated: April 16, 2010)

I. INTRODUCTION

The non-linear Schrödinger equation is used to model many physical processes including deep water waves and Bose-Einstein condensates. In general, the form of the Non-Linear Schrödinger (NLS) equation in one dimension is,

$$i\frac{\partial}{\partial t}\Phi = -\frac{\partial^2}{\partial x^2}\Phi + V(x)\,\Phi + \sigma\left|\Phi\right|^2\Phi, \tag{1.1}$$

where $V(x)$ is an external potential, and where $\sigma = 1$ is the defocusing case and $\sigma = -1$ is the focusing case[3].

The main focus of this paper will be on numerical solutions of the time-independent focusing non-linear Schrödinger (NLS) equation which can be found by substituting $\Phi = \Psi e^{-iEt}$ into Eq.1.1 with $\sigma = -1$,

$$E\Psi = -\frac{\partial^2}{\partial x^2}\Psi + V(x)\,\Psi - \left|\Psi\right|^2\Psi, \tag{1.2}$$

where E is the energy. Since we can always take Ψ to be real, the modulus can be removed and the following equation results,

$$E\Psi = -\frac{\partial^2}{\partial x^2}\Psi + V(x)\,\Psi - \Psi^3, \tag{1.3}$$

The first section will outline the three numerical methods used to analyse Eq.1.3. Following this, an analysis of one and two-pulse solutions will be presented. A single pulse is a solution that is localized with exponentially decaying tails. Two pulse solutions can be thought of as a composition of two single pulses separated well due to their exponential decay to zero.

II. SPECTRAL RENORMALIZATION METHOD

We propose three methods to solve the NLS equation outlined in Eq.1.3. They are all originated from the spectral renormalization method. This method was first proposed by Petviashvili in 1976[5] in the context of the Kadomtsev-Petviashvili equation.

The Spectral Renormalization method can be developed by first bringing all linear terms of Eq.1.3 to one side,

$$-\frac{\partial^2}{\partial x^2}\Psi + V(x)\,\Psi - E\Psi = \Psi^3, \tag{2.1}$$

Now we can combine the first two terms of Eq.2.1 as the linear operator $L = -\partial_x^2 + V(x)$, acting on smooth function Ψ that decays at infinity. If E is not in the spectrum of L then the operator $L - E$ is invertible and we can form a sequence of approximations,

$$u_{n+1} = (L - E)^{-1}\,u_n^3. \tag{2.2}$$

*Electronic address: brownj42@mcmaster.ca
†Electronic address: dmpeli@math.mcmaster.ca

Now, it is left to see if this iterative scheme will converge to a fixed solution. This can be done by performing the simple reduction $u_n = a_n \Psi$, where Ψ is the solution to Eq.2.1. The power map that needs to be satisfied is that,

$$a_{n+1} = a_n^3. \tag{2.3}$$

The three fixed point solutions to this power map are $-1, 0, 1$. The test for stability of the fixed points are to simply take the derivative of a_n^3 and check whether it satisfies the condition that its magnitude is less than one at the fixed point. When this is done, it is apparent that the only fixed point that is stable is the zero solution. This relates to the trivial $\Psi = 0$ solution and thus is of no use. Therefore, it is necessary to introduce a renormalization factor M_n in the iteration rule,

$$u_{n+1} = M_n^{\gamma} \left(L - E\right)^{-1} u_n^3, \tag{2.4}$$

where $\gamma > 0$ is a free parameter and M_n is defined as,

$$M_n = \frac{\langle u_n, \left(L - E\right) u_n \rangle}{\langle u_n, u_n^3 \rangle}. \tag{2.5}$$

Because $M_n = 1$ if $u_n = \Psi$, solutions of Eq.2.2 and Eq.2.4-2.5 coincide. This result will be used as a numerical test for convergence. It is now left to determine for what value of γ the iterations converge. To do this, it is convenient to define a Jacobian operator as,

$$H = L - E - 3\Psi^2 \tag{2.6}$$

To prove that this iterative scheme converges to the unique point Ψ, we will need to show that in the neighbourhood of Ψ, the iterative scheme of Eq.2.4-2.5 is a contraction. This follows by the Contraction Mapping Theorem. We now consider the iteration operator T and its adjoint operator T^*,

$$T = \left(L - E\right)^{-1} H, \quad T^* = H \left(L - E\right)^{-1}. \tag{2.7}$$

Operator T is not self adjoint in L^2 but is self-adjoint with respect to the weight $L - E$. If we restrict ourselves to the region where $E < \omega_0$, where ω_0 is the bottom of the spectrum of L, then the operator $L - E$ is a positive operator. In this case, the number of negative and positive eigenvalues of T and H coincide. The eigenfunction for one negative eigenvalue is known exactly as,

$$T\Psi = -2\Psi \iff T^*\Psi^3 = -2\Psi^3 \tag{2.8}$$

Let us assume in our analysis that H has exactly one negative eigenvalue, so that Eq.2.8 is the only negative eigenvalue for T. If we now linearise Eq.2.4 at the solution of Ψ with $v_n = u_n - \Psi$, then we obtain the linearised iteration map,

$$u_{n+1} = \gamma \left(1 - p\right) \alpha_n \Psi + u_n - \left(L - E\right)^{-1} H v_n. \tag{2.9}$$

Substituting $v_n = \alpha_n \Psi + w_n$, where w_n is orthogonal to Ψ^3 or, equivalently, α_n is defined by,

$$\alpha_n = \frac{\langle \Psi^3, v_n \rangle}{\langle \Psi^3, \Psi \rangle}, \tag{2.10}$$

then, Eq.2.9 splits into two uncoupled equations,

$$\begin{cases} w_{n+1} = \left(1 - T\right) w_n \\ \alpha_{n+1} = \left(3 - 2\gamma\right) \alpha_n \end{cases} \quad n \geq 1 \tag{2.11}$$

The second equation from Eq.2.11 suggests that the iteration is only a contraction if $-1 < 3 - 2\gamma < 1$ or $1 < \gamma < 2$. That being said, we would like to choose an optimal γ such that it converges as fast as possible. This can be done by making $\gamma = \dfrac{3}{2}$ for which the component α_n will converge after only one iteration. Even for $\gamma = \dfrac{3}{2}$, the first equation of system Eq.2.11 suggests that convergence is only linear if $w_1 \neq 0$.

Since T has a negative eigenvalue, operator $1 - T$ is not a contraction. However, w_n is constrained by the condition that $w_n \perp \Psi^3$ and T has no negative eigenvalues in the space induced by this constraint because $\langle \Psi, \Psi^3 \rangle \neq 0$. On the other hand, one can show that $1 - T$ is a positive operator[3], so that $(1 - T)$ is a contraction in the constrained space. As a result, we have the following convergence criterion.

If H has one negative and no zero eigenvalues, then the the iterative scheme of Eq.2.4-2.5 with $\gamma = \frac{3}{2}$ converges to solution Ψ.

The converse of this statement is also true, that is, if H has two or more negative eigenvalues, then the iterative scheme of Eq.2.4-2.5 does not converge to solution Ψ even if $\gamma = \frac{3}{2}$.

Now all that is left is to determine what a good initial guess u_0 of the iteration scheme is. For $E < \omega_0$ (that is, E is chosen in the semi-infinite gap) it is known that for symmetric potentials the fundamental solutions are even, with a single hump and exponentially decreasing tails. Thus, any function that has this similarity may converge to the solution Ψ. If the initial approximation is not close to the actual solution Ψ, then the method may not necessarily converge.

When actually performing the numerical iterations, we need to define a method in which the convergence can be measured. The easiest method would of course be a comparison of the numerical solution to a known solution. This would be defined as,

$$e_{actual}^{(n)} = \sup_{x\in\mathbb{R}} |u_n(x) - \Psi(x)| < \epsilon, \tag{2.12}$$

where Ψ is the known solution and ϵ is some error tolerance which should be taken close to machine precision. Although using e_{actual} is a good test to confirm that a method is converging, exact solutions rarely exist. So now we look to other methods to determine convergence, for instance,

$$e_u^{(n)} = \sup_{x\in\mathbb{R}} |u_{n+1}(x) - u_n(x)| < \epsilon \ \text{ and/or } \ e_M^{(n)} = |M_n - 1| < \epsilon. \tag{2.13}$$

The first convergence (e_u) condition simply states that when the iteration scheme converges, the successive approximation should be, within the error tolerance, the same as the previous iteration. The second convergence condition (e_M) applies when the method converges to the fixed point Ψ because,

$$M_n \ \rightarrow \ \frac{\langle \Psi, (L-E)\Psi \rangle}{\langle \Psi, \Psi^3 \rangle} = 1 \tag{2.14}$$

A. Finite-Difference Method

The difficulty in using the Spectral Renormalization method is the approximation in the ∂_x^2 portion of the operator L. The first way for this to be done is by using a high-order finite difference method to approximate the derivative. The iterations can easily be performed by using a matrix representation of L. The iteration scheme of Eq.2.4 ,with $\gamma = 3/2$, can then be written explicitly by,

$$u_{n+1} = M_n^{3/2} \left(\tilde{L} - EI\right)^{-1} u_n^3, \tag{2.15}$$

where I is the identity matrix and $\tilde{L}$ is now an approximation of the operator L. For efficiency, an inversion (or LU decomposition) of the operator $\left(\tilde{L} - EI\right)$ can be performed before iterations begin. The M_n is equivalent to the one described by Eq.2.5. The boundary conditions used were Dirichlet.

B. Spectral Method

This method for approximating the operator L can be derived by first transforming the focusing NLS (1.3) to Fourier space which results in,

$$E\hat{\Psi} = k^2\hat{\Psi} + \widehat{V\Psi} - \widehat{\Psi^3}. \tag{2.16}$$

If the k^2 term is brought to the left side, it can be seen that an equivalent iteration scheme can be defined as,

$$u_{n+1} = M_n^{3/2} \frac{\widehat{u_n^3} - \widehat{Vu_n}}{-E + k^2}. \tag{2.17}$$

However, with this method, there is a more stringent condition that needs to be placed on the value of E. In order for $-E + k^2$ to be invertible, E must be negative. This condition is not equivalent to the condition $E < \omega_0$ because ω_0 can be non-zero. The renormalization condition term used for this method can be performed in Fourier space using Parseval's identity. The M_n is now given by,

$$M_n = \frac{\left\langle \widehat{u_n}, \left(-E + k^2\right) \widehat{u_n} \right\rangle}{\left\langle \widehat{u_n}, \widehat{Vu_n} - \widehat{u_n^3} \right\rangle}. \tag{2.18}$$

C. Bloch-Fourier Method

The final method of approximating the operator L uses what is known in solid state physics, as *Bloch functions*. Setting $l_n(x;k) = e^{ikx} w_n(x;k)$, we can obtain eigenfunctions of the 2π-periodic operator L satisfying the boundary conditions,

$$l_m(x + 2\pi; k) = e^{2\pi i k} l_m(x;k), \tag{2.19}$$

where x is real, m is an integer, and $k \in \left[-\frac{1}{2}, \frac{1}{2}\right]$.

These Bloch functions form an orthogonal basis according to the orthogonality condition defined by,

$$\int l_n(x;k)\, \bar{l_n}(x;k')\, dx = \delta_{n,n'} \delta(k - k'), \tag{2.20}$$

where m, m' are integers, and $k, k' \in \left[-\frac{1}{2}, \frac{1}{2}\right]$. From this we can define the Bloch-Fourier transform as,

$$\tilde{\phi_n}(k) = \int \phi(x)\, \bar{l_n}(x;k)\, dx \tag{2.21}$$

where we denote $\left\{\tilde{\phi}_m(k)\right\}_{m \in \mathbb{N}}$ and where the inverse transform, or the *Bloch decomposition* is,

$$\phi(x) = \int \sum_{m \in \mathbb{N}} \tilde{\phi_n}(k)\, l_m(x;k)\, dk \tag{2.22}$$

With the above, we can now perform the iterations of Eq.2.4 as,

$$u_{n+1} = M_n^{3/2} (\xi - EI) \tag{2.23}$$

where ξ is a diagonal matrix which contains the eigenvalues for all spectral bands of the operator L. The normalization for this method is defined as,

$$M_n = \frac{\langle \widetilde{u_n}, (\xi - EI)\, \widetilde{u_n} \rangle}{\left\langle \widetilde{u_n}, \widetilde{u_n^3} \right\rangle}. \tag{2.24}$$

III. NUMERICAL SOLUTIONS TO THE TIME-INDEPENDENT FOCUSING NLS

It is always useful to test the convergence of a numerical method by first comparing it to an exact solution. Fortunately, one does exist for the focusing NLS with $V(x) = 0$ and it is given by,

$$\Psi = \frac{\sqrt{2\left|E\right|}}{\cosh\sqrt{\left|E\right|}x}. \tag{3.1}$$

The starting condition is important if we wish to analyse convergence of the numerical methods. We take the initial condition in the form,

$$u_0 = e^{-x^2}. \tag{3.2}$$

The reason that the initial approximation is not of an equivalent form as the exact solution (ie. $a\Psi$, where a is a constant) is that, all methods would have converged in one iteration and it would not have been a useful test.

When the iterations are performed, the convergence does occur (Fig.1). The convergence of the e_{actual} term to zero is slower than the convergence for the e_M error term. This means that in order to get the exact solution to machine precision, there should be more iterations performed than would be dictated by e_M. At the same time, the exact error e_{actual} is almost indistinguishable from the numerical error e_u.

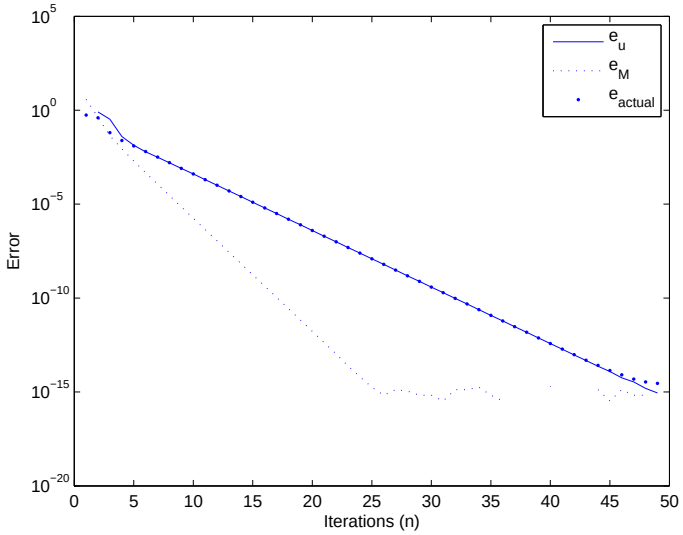

FIG. 1: The convergence of the Spectral method for the case when $V(x) = 0$ and $E = -1$.

It is apparent that when the log of the error is plotted against the iterations, the relationship is linear (Fig.1). Thus it would be reasonable to assume that the convergence is of the form,

$$\frac{\left(e^{(n+1)}\right)}{\left(e^{(n)}\right)^p} \leq \mu \tag{3.3}$$

where p is defined as the rate of convergence and μ is a positive constant. For Fig.1, $p = 1$ and $\mu < 1$ for e_{actual} or e_u, so the convergence is linear.

IV. ONE-PULSE SOLUTIONS FOR $V(x) = \sin^2(x/2)$.

We will consider one-pulse solutions where the potential is

$$V(x) = \sin^2\left(\frac{x}{2}\right). \tag{4.1}$$

For the semi-infinite gap ($E < \omega_0$), one-pulse solutions can only occur around the point x_0 where the potential is extremal, that is $V'(x_0) = 0$ and $V''(x_0) \neq 0$. For the potential of Eq.4.1, this will occur at $x = m\pi$ where m is an integer. However, only when m is an even integer will the resulting solution be stable. For m being odd, the one-pulse

solution is unstable[4]. That being said, any of the methods described in Sec.II will converge to either of these fixed solutions.

A. Stable Solutions (m is an even integer)

We now look to the stable localized modes of the semi-infinite gap. When we use the spectral method with the initial condition of Eq.3.2, we can generate the numerical solution shown in Fig2.

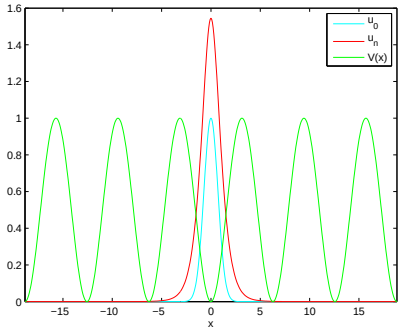

FIG. 2: The localized mode for $V(x) = \sin^2(x/2)$ and $E = -1$.

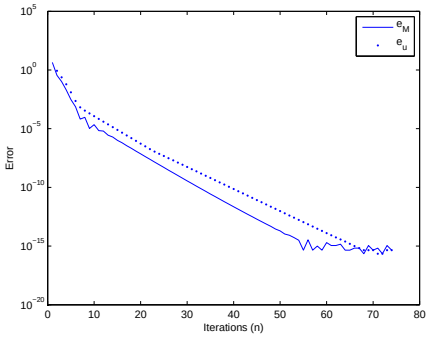

FIG. 3: The convergence of the spectral method for $V(x) = \sin^2(x/2)$ and $E = -1$.

The convergence for the solution in Fig.3 is once again linear with the $p = 0.99$. We can now analyse the same system using the Bloch-Fourier method. The solution is equivalent to the spectral method, within machine precision, where the convergence is once again linear with $p = 1.00$. For the case when $E = -0.5$, the result is more interesting. In general, the spectral method should work for any $E < 0$. However, as can be seen by Fig.4, the spectral method does not converge. This is the region where it becomes necessary to use the finite-difference or Bloch-Fourier methods.

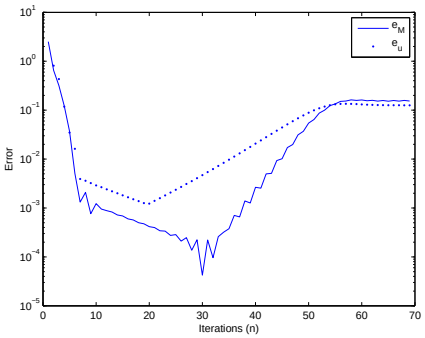

FIG. 4: The spectral method does not converge for $V(x) = \sin^2(x/2)$ and $E = -0.5$.

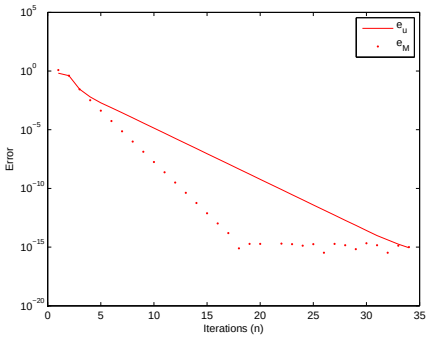

FIG. 5: The Bloch-Fourier method does converge for $V(x) = \sin^2(x/2)$ and $E = -0.5$.

When the Bloch-Fourier method is used for $E = -0.5$, the linear convergence is once again regained as can be seen in Fig.5.

Now we turn to the band gap solutions. As the spectral method will not converge for $E > 0$, we will need to use the Bloch-Fourier or finite-difference methods to obtain the band gap solutions. We can not prove analytically the convergence of the solution for this region as the operator H has a continuous band in the negative spectrum and the operator T is not a contraction. However, our computations show that the method converges and the residual of Eq.1.3 is small.

For the potential of Eq.4.1, the first band gap exists approximately in the region $0.60 < E < 0.96$. An example of a band gap solution is in Fig.6. Fig.6 depicts the numerical solution where $E = 0.85$ and is computed using the Bloch-Fourier method. The residual for this solution was calculated to be on the order of 10^{-12}. Thus, we can be confident that the numerical solution that we are calculating is correct.

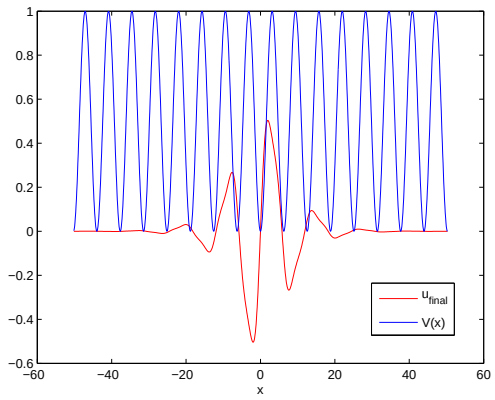

FIG. 6: The band gap solution for $V(x) = \sin^2(x/2)$ and $E = 0.85$.

As you can see in Fig.6, the localized modes are of odd parity now. Because of this, a better initial condition to use is,

$$u_0 = \sin(x)\, e^{-x^2}. \tag{4.2}$$

The localized modes bifurcate from the lower energy of the band into the band gap[4]. For both the semi-infinite gap and band gap, the bifurcation is shown in Fig.7 by plotting $||\Psi||^2_{L_2}$ vs. the energy E.

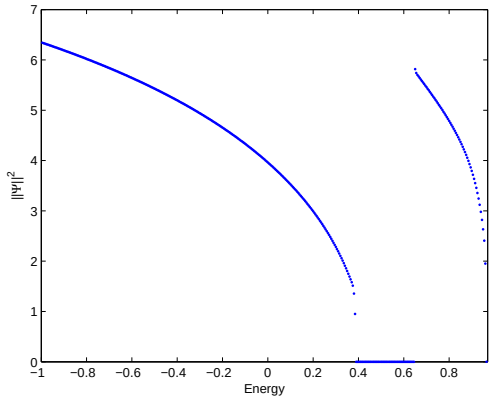

FIG. 7: The bifurcations of solutions for the band gap of system with $V(x) = \sin^2(x/2)$.

B. Unstable Solutions (m is an odd integer)

The Bloch-Fourier method converges for the unstable localized modes in the semi-infinite gap. However, as the energy increases, it may be necessary to enforce the even symmetry relative to the maximum of $V(x)$ in order to kill off the unstable directions of the iteration operator. The solution for $E = 0.2$ is shown in Fig.8. As can be seen now, the solution is such that it is no longer a single pulse but has a double-humped profile.

For the purposes of the graph, the potential used was $V(x - \pi) = \sin^2(x/2)$ so the unstable solution could be centred at 0. For $E = -1$ (Fig.9), the unstable solution has a similar form to that of the stable solution, in that it has a single hump profile, except that it has a larger L_2 norm.

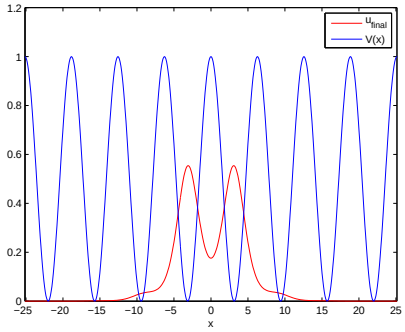

FIG. 8: The unstable localized mode for the NLS equation with $V(x-\pi) = \sin^2(x/2)$ and $E = 0.2$.

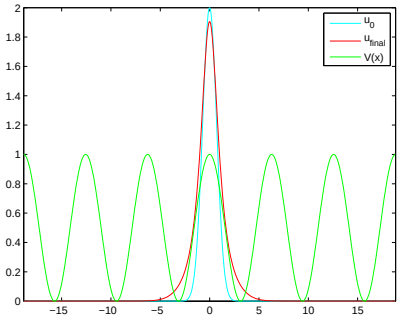

FIG. 9: The unstable localized mode for the NLS equation with $V(x-\pi) = \sin^2(x/2)$ and $E = -1.0$.

V. TWO-PULSE SOLUTIONS OF THE NLS EQUATION

So far, we have only considered single-pulse solutions of the NLS equation in the potential $V(x) = \sin^2(x/2)$. Multi-pulse solutions on the other hand can be thought of as a composition of singles pulses separated well from each other. This brings about the theory of tail-to-tail interactions[1]. For the case when there is no potential, it is known that there is an attraction between pulses that are in-phase and repulsion between pulses that are out-of-phase[2]. The periodic potential $V(x)$ produces an additional force. Thus, a bound state between the two pulses can be formed at the equilibrium of all the forces.

Knowing this, there are four different symmetric two-pulse solutions that could occur: the in-phase and out-of-phase solutions of both the stable and unstable solutions. Additionally, there could be non-symmetric two-pulse solutions consisting of bound states between a stable and unstable pulse. However, the non-symmetric two-pulse solutions are not considered here.

A. Stable-Two Pulse Solutions

In order for convergence of the stable two-pulse solution to occur, we will need to enforce a symmetry in the system. For the spectral method, this can be easily done by either taking the sine transform (for odd symmetry) or the cosine transform (for even symmetry). Each pulse brings one negative eigenvalue into the system. However, enforcing symmetry removes one negative eigenvalue and therefore reduces the system to the one that satisfies the convergence condition in Sec.II.

In-phase, stable solutions are shown on Fig.10. As you can see from Fig.10, the heights of the two pulses are equivalent to the one-pulse solution. Also, if you look closely at the solution, you will see that the centres of localizations have shifted towards each other in the process of convergence. This implies that equilibrium is achieved between the attractive tail-to-tail interaction and the repulsive force from the potential $V(x)$.

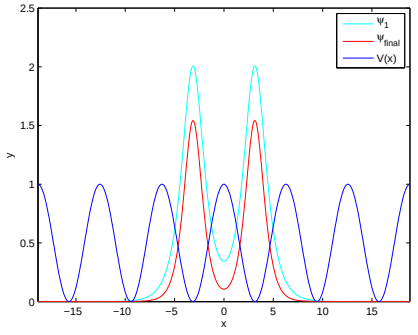

FIG. 10: The in-phase stable two-pulse solution for $V(x-\pi) = \sin^2(x/2)$ and $E = -1$.

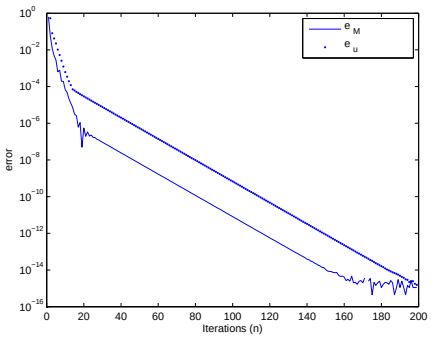

FIG. 11: Convergence of the in-phase stable two-pulse solution.

The convergence for this system of Fig.10 is shown to be linear with $p = 1.00$. However, the constant μ from Eq.3.3 is close to one, so that the convergence is slow. As you can see from Fig.11, it took about 200 iterations to reach the machine precision level.

The out-of-phase stable two-pulse solution is shown on Fig.12. The pulses are centred slightly further out from where they started. Therefore, once again, the equilibrium is achieved between the repulsive tail-to-tail interaction and the attractive force from the potential $V(x)$.

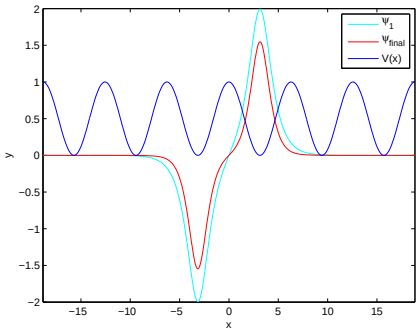

FIG. 12: The out-of-phase stable two-pulse solution with $V(x-\pi) = \sin^2(x/2)$ and $E = -1$.

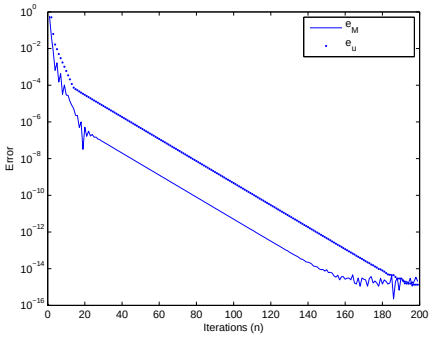

FIG. 13: Convergence of the out-of-phase stable two-pulse solution.

B. Unstable Two-Pulse Solutions

For the unstable two-pulse solutions, the tail-to-tail interactions cause another instability in the iterations of the numerical method. Even if the symmetry is enforced, the iterations will not necessarily converge to the desired solution because each pulse brings now two negative eigenvalues to operator T. For initial conditions localized near the points maximum of $V(x)$, the two pulses will in general converge to the nearest stable solutions localized near the points of minimum of $V(x)$. Thus, if the two pulses are a fair distance apart, the unstable two-pulse solution will converge to the stable two-pulse solution.

Knowing this, it is necessary to initiate the two pulses with centres either shifted inwards or outwards from each other, for out-of-phase or in-phase pulses respectively. For the solution shown in Fig.14, the initial conditions were such that both pulses where shifted outward from the points where $V'(x) = 0$ and $V''(x) < 0$ by a value of 0.028564310716943. This number was found by using a minimization algorithm where the function minimized was the e_u error. The input argument was the shift from the maximum of the potential. This technique was also implemented using the secant method in the context of the KdV equation by Chugunova and Pelinovsky in 2007[1].

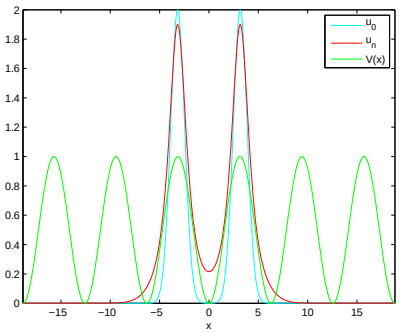

FIG. 14: The in-phase unstable two-pulse solution for $V(x) = \sin^2(x/2)$ and $E = -1$.

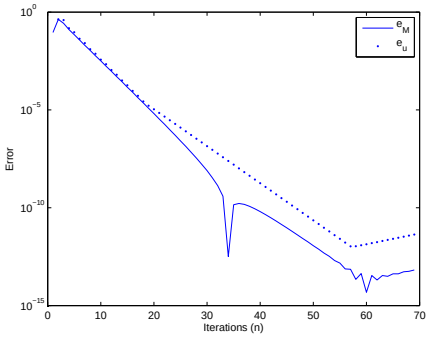

FIG. 15: Convergence of the in-phase unstable two-pulse solution.

As you can see from Fig.15, the iterations near the two-pulse solution converge initially and then begin to diverge. Depending on the starting position of the two pulses, the minimum of the error can be made as small as the machine

precision. Therefore the iterations can be truncated at this minimum error, and the numerical solution is found to be very close to the actual solution within machine precision.

For the out-of-phase two pulse solution shown in Fig.16, the initial localizations are shifted inward by 0.029670361877393. This value was once again calculated using the minimization algorithm. One thing to note is that with this carefully chosen starting point, there is no shift in the pulses when the solution converges. This suggests that this point gives the equilibrium between the tail-to-tail and the potential forces.

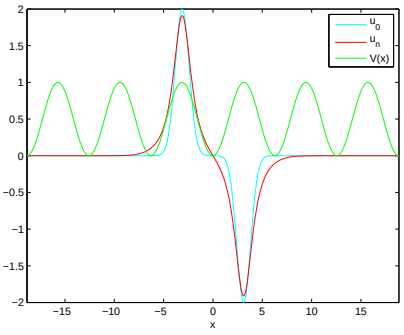

FIG. 16: The out-of-phase unstable two-pulse solution for $V(x) = \sin^2(x/2)$ with $E = -1$.

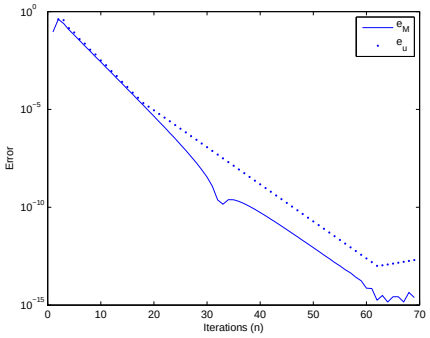

FIG. 17: Convergence of the out-of-phase unstable two-pulse solution.

Fig.17 shows the convergence of the out-of-phase unstable two-pulse solution. As with the in-phase unstable solution, the iterations initially converge and then proceed to diverge. Once again, an initial condition could theoretically be chosen such that the numerically solution could be equivalent to machine precision.

VI. CONCLUSION

We have developed three modifications of the Spectral Renormalization (Petviashvilli) method to compute numerical solutions to the NLS equation. For solutions in the semi-infinite gap, all three methods should theoretically be able to generate solutions. However, it is found that the Discrete Fourier method fails if the energy is not low enough. For the band gap solutions, the localizations can be found by either the Bloch-Fourier method or the Finite-Difference Method.

As expected in the focusing case, the one-pulse solutions bifurcate out of the left sides of the spectral bands into the band gap. In the semi-infinite gap, the solutions are found to be even. In the finite gap, the solutions are found to be odd. In the semi-infinite gap, we recover both stable and unstable solutions localized near the points x_0 where $V'(x_0) = 0$. For the band-gap, only the stable solutions are recovered.

Two-pulse solutions were numerically generated in the semi-infinite gap for both stable and unstable pulses. Only symmetric cases were discussed with both in-phase and out-of-phase two-pulse solutions. All solutions were found to exist at the equilibrium between the potential and the tail-to-tail interactions.

Appendix A: MATLAB codes

1. Finite-Difference Method

This is the MATLAB implementation of the finite-difference method.

```
M=2^4; N = 2^4; %M is the number of points per 2*pi range,
%N the number of periodic
x=linspace(-N*pi,N*pi,M*N);
h=2*N*pi/(N*M-1); h2=h*h;
V=sin(x/2).^2; %The potential
a=0.; w=0.64; %a is the centre of the initial wave, w is the energy
psi(:,1)=-exp(-(x'-a).^2).*sin(x');%the initial wave function


% The matrix with 8-th order approximation of the second derivative
L1=(9778141/(3175200*h2)+V);
Lup(1:M*N)=(-9/(5*h2));
Ldown(1:M*N)=(-9/(5*h2));
L2up(1:M*N)=18/(55*h2);
L3up(1:M*N)=-14/(165*h2);
L4up(1:M*N)=63/(2860*h2);
L5up(1:M*N)=-18/(3575*h2);
L6up(1:M*N)=2/(2145*h2);
L7up(1:M*N)=-9/(70070*h2);
L8up(1:M*N)=9/(777920*h2);
L9up(1:M*N)=-1/(1969110*h2);
L=diag(L1)+diag(Lup(1:M*N-1),1)+diag(Ldown(1:M*N-1),-1);
L=L+diag(L2up(1:M*N-2),2)+diag(L2up(1:M*N-2),-2);
L=L+diag(L3up(1:M*N-3),3)+diag(L3up(1:M*N-3),-3);
L=L+diag(L4up(1:M*N-4),4)+diag(L4up(1:M*N-4),-4);
L=L+diag(L5up(1:M*N-5),5)+diag(L5up(1:M*N-5),-5);
L=L+diag(L6up(1:M*N-6),6)+diag(L6up(1:M*N-6),-6);
L=L+diag(L7up(1:M*N-7),7)+diag(L7up(1:M*N-7),-7);
L=L+diag(L8up(1:M*N-8),8)+diag(L8up(1:M*N-8),-8);
L=L+diag(L9up(1:M*N-9),9)+diag(L9up(1:M*N-9),-9);
L(1,M*N)=Lup(1);
L(M*N,1)=Ldown(1);
L=L+diag(L2up(1:2),-M*N+2)+diag(L2up(1:2),M*N-2);
L=L+diag(L3up(1:3),-M*N+3)+diag(L3up(1:3),M*N-3);
L=L+diag(L4up(1:4),-M*N+4)+diag(L4up(1:4),M*N-4);
L=L+diag(L5up(1:5),-M*N+5)+diag(L5up(1:5),M*N-5);
L=L+diag(L6up(1:6),-M*N+6)+diag(L6up(1:6),M*N-6);
L=L+diag(L7up(1:7),-M*N+7)+diag(L7up(1:7),M*N-7);
L=L+diag(L8up(1:8),-M*N+8)+diag(L8up(1:8),M*N-8);
L=L+diag(L9up(1:9),-M*N+9)+diag(L9up(1:9),M*N-9);
err = 1; big = 1;

%Computing the inverse for efficiency
Lapp=inv(L-w*eye(N*M));

while err > 5*10^(-16) && big<20
    psi3 = psi(:,big).^3;

    %Calculate the M_n factor
    holdit1 = (psi(:,big))'*((L-w*eye(N*M))*psi(:,big));
    holdit2 = (psi(:,big))'*psi3;
    Mn(big) = (holdit1/holdit2);

    %peform the iteration
    psi(:,big+1) = (Mn(big)^(3/2)*(Lapp*psi3));

    %calculate the e_u error
```

```
    err = norm(psi(:,big+1)-psi(:,big));
    En(big) = err;

    big = big + 1;
    figure(1); plot(x',psi(:,1),'c',x',psi(:,big),'r');
    errnorm(big)=norm(abs(psi(:,big)-sqrt(-2*w)*sech(sqrt(-w)*x')),inf);
end
last=cputime

%plot the results
figure(1); plot(x',psi(:,1),'c',x',psi(:,big),'r',x,V,'y');
figure(2); semilogy(abs(Mn-1),'.r');
hold on; semilogy(En,'.c'); hold off;
legend('M_n-1','|u_{n+1}-u_n|')
figure(3); semilogy([1:big],errnorm,'y.');
%calculate the convergence rate
jk=polyfit([4:big-7],log10(En(4:big-7)),1);

time=last-start;
```

2. Spectral Method

This is the MATLAB code that implements the spectral method as defined in Section II B.

```
Nmax=2^10; %The number of points used
tot=75;  %The total number of iterations to be performed
E=1.0;  %The negative of the energy
aadd=0; %Change in starting position if necessary for convergence
a=0.+aadd;
n=0:Nmax-1;
sil=6; %The number of pi for convergence

% Setting up the x-axis k-space and potential
x=linspace(-sil*pi,sil*pi,Nmax+1);
x=x(1:length(x)-1);
k=[0:Nmax/2-1,-Nmax/2:-1];
k=k./sil;
V=sin((x-pi)/2).^2;
psi=zeros(tot,Nmax);
errorm=zeros(2,tot);
erroru=zeros(2,tot);

big=1; %number of indices needed

%Setting the initial wave function
psi(1,:)=2*exp(-(x).^2);

%Setting the values for the first iteration
psi_out=zeros(1,Nmax);
psi_hat2=real(fft(psi(1,:)));
psi_out2=real(ifft((E+k.^2).*psi_hat2));
psi3=psi(1,:).^(3)-V.*psi(1,:);

% Compute the M_n for the first iteration
H1=psi_out2*psi(1,:)';
L4=psi3*psi(1,:)';
M=(H1/L4)^(3/2);
disp(H1)
%The e_M error for the first iteration
errorm(1)=abs(M-1);

psi3=psi(1,:).^3-V.*psi(1,:);
```

```
psi_hat=real(fft(psi3));

for jk=2:tot
        psi_out=M*psi_hat./(E+(k).^2); %Compute the next u_n
        psi(jk,:)=real(ifft(psi_out));
        psi_hat2=real(fft(psi(jk,:)));
        psi_out2=real(ifft((E+k.^2).*psi_hat2));
        psi3=psi(jk,:).^3-V.*psi(jk,:);
        %Calculate M_n
        H1=real(psi_out2)*psi(jk,:)';
        L4=real(psi3*psi(jk,:)');
        M=(H1/L4)^(3/2);
        %calculate the e_u error
        erroru(big,jk)=norm(psi(jk-1,:)-psi(jk,:),inf);
        %calculate the e_M error
        errorm(big,jk)=abs(1-M);

        %For the next iteration
        psi3=psi(jk,:).^3-V.*psi(jk,:);
        psi_hat=real(fft(psi3));  %this is the cosine series,use imag to get sine
end

%plot the results
figure(1);plot(x,psi(1,:),'c',x,psi(tot,:),'r',x,V,'g')
legend('u_0','u_n','V(x)')
xlim([-6*pi,6*pi])


error=zeros(1,tot);

%for plotting purposes
xerr=1:tot-1;

%for comparison with exact solutions
for j=1:tot
    error(j)=norm(psi(j,:)-(sqrt(2*E))*sech(sqrt(E)*(x)),inf);
end

% Used to find convergence rate (Assuming u_final is approximately solution)
for j=1:tot-1
    errorj(j)=norm(psi(j,:)-psi(tot,:),inf);
end

%Plot the errors
figure(2); semilogy(xerr,errorm(big,1:tot-1),'b',xerr,erroru(big,1:tot-1),'b.')
legend('e_M','e_u')

% Find the convergence of the system
jk=polyfit(log10(errorj(big,13:tot-3)),log10(errorj(big,12:tot-4)),1)

%This line computes the residual of the answer
err=norm(-ifft(-k.^2.*fft(psi(tot,:)))+V.*psi(tot,:)-psi(tot,:).^3+E*psi(tot,:),inf)
```

3. Bloch-Fourier Method

This is the MATLAB code that implements the spectral method as defined in Section II C.

```
%M is the total number of points per periodic region
M=2^6; x=linspace(0,2*pi,M+1);
h=2*pi/M; h2=h*h; x=x(2:M+1);
%j = 1:M;
%h12=180*h*h;
```

```
%bh=60*h;

%The periodic potential used
V=sin((x)/2).^2;
%N is the total number of periodicities used
N = 16; k=linspace(-0.50,0.50,N+1);
kh=1/N; k=k(2:N+1);

% This sets up the matrix to find eigenvalues/eigenvectors (l) such that
% l=exp(ikx)wn

for init=N/2:N
    L1=(9778141/(3175200*h2)+k(init)^2+V);
    Lup(1:M)=(-9/(5*h2)-(7/8)*2i*k(init)/h);
    Ldown(1:M)=(-9/(5*h2)-(-7/8)*2i*k(init)/(h));
    L2up(1:M)=18/(55*h2)-(-7/24)*2i*k(init)/(h);
    L3up(1:M)=(-14/(165*h2))-(7/72)*2i*k(init)/(h);
    L4up(1:M)=63/(2860*h2)-(-7/264)*2i*k(init)/(h);
    L5up(1:M)=(-18/(3575*h2))-(7/1320)*2i*k(init)/(h);
    L6up(1:M)=2/(2145*h2)-(-7/10296)*2i*k(init)/(h);
    L7up(1:M)=(-9/(70070*h2))-(1/24024)*2i*k(init)/(h);
    L8up(1:M)=9/(777920*h2);
    L9up(1:M)=(-1/(1969110*h2));
    L2down(1:M)=18/(55*h2)-(7/24)*2i*k(init)/(h);
    L3down(1:M)=(-14/(165*h2))-(-7/72)*2i*k(init)/(h);
    L4down(1:M)=63/(2860*h2)-(7/264)*2i*k(init)/(h);
    L5down(1:M)=(-18/(3575*h2))-(-7/1320)*2i*k(init)/(h);
    L6down(1:M)=2/(2145*h2)-(7/10296)*2i*k(init)/(h);
    L7down(1:M)=(-9/(70070*h2))-(-1/24024)*2i*k(init)/(h);
    L8down(1:M)=9/(777920*h2);
    L9down(1:M)=(-1/(1969110*h2));
    L=diag(L1)+diag(Lup(1:M-1),1)+diag(Ldown(1:M-1),-1);
    L=L+diag(L2up(1:M-2),2)+diag(L2down(1:M-2),-2);
    L=L+diag(L3up(1:M-3),3)+diag(L3down(1:M-3),-3);
    L=L+diag(L4up(1:M-4),4)+diag(L4down(1:M-4),-4);
    L=L+diag(L5up(1:M-5),5)+diag(L5down(1:M-5),-5);
    L=L+diag(L6up(1:M-6),6)+diag(L6down(1:M-6),-6);
    L=L+diag(L7up(1:M-7),7)+diag(L7down(1:M-7),-7);
    L=L+diag(L8up(1:M-8),8)+diag(L8down(1:M-8),-8);
    L=L+diag(L9up(1:M-9),9)+diag(L9down(1:M-9),-9);
    L(1,M)=Ldown(1);
    L(M,1)=Lup(1);
    L=L+diag(L2up(1:2),-M+2)+diag(L2down(1:2),M-2);
    L=L+diag(L3up(1:3),-M+3)+diag(L3down(1:3),M-3);
    L=L+diag(L4up(1:4),-M+4)+diag(L4down(1:4),M-4);
    L=L+diag(L5up(1:5),-M+5)+diag(L5down(1:5),M-5);
    L=L+diag(L6up(1:6),-M+6)+diag(L6down(1:6),M-6);
    L=L+diag(L7up(1:7),-M+7)+diag(L7down(1:7),M-7);
    L=L+diag(L8up(1:8),-M+8)+diag(L8down(1:8),M-8);
    L=L+diag(L9up(1:9),-M+9)+diag(L9down(1:9),M-9);

    %find the eigenvalues and eigenvectors for value of k
    [Vect,D]=eig(L);
    E(:,init)=diag(D);
    wh(:,:,init) = Vect;
end

%only calculate for positive k and flip back
E(:,1:N/2-1) = E(:,N-1:-1:N/2+1);
%complex conjugate of eigenfunction for negative value of k
wh(:,:,1:N/2-1) = conj(wh(:,:,N-1:-1:N/2+1));

%This sets up the diagonal matrix of eigenvalue for all k values
EMatr = E(:,1);
```

```
    for jj = 2 : N
        EMatr = [EMatr;E(:,jj)];
    end
EMatr = diag(EMatr);

%the x-values and V for all k
x=[x-16*pi x-14*pi x-12*pi x-10*pi x-8*pi x-6*pi x-4*pi x-2*pi x];
x=[x  x+2*pi x+4*pi x+6*pi x+8*pi x+10*pi x+12*pi x+14*pi];
V=[V, V, V, V, V, V, V, V, V, V, V, V, V, V, V, V];

%place x into N*M matrix
xMatr = x'*ones(1,M);

%create the matrix of Bloch Functions
for init=1:N
    utemp = wh(:,:,init);
    for jj = 2 : N
        utemp=[utemp;wh(:,:,init)];
    end
    u(:,:,init)=exp(1i*k(init)*xMatr).*utemp;
end
uMatr = u(:,:,1);
    for jj = 2 : N
        uMatr = [uMatr,u(:,:,jj)];
    end


start=cputime;
% Normalize the Bloch Function matrix
uMatr=uMatr./sqrt(N);

%The inverse of the Bloch Function Matrix
IuMatr=uMatr';

%This loop is if you want to calculate values for more than one energy (w)
for bigger=1:1
%Set the energy and the initial shift
a=0.; w=-0.5+bigger*0.000;
%hold values of e
wm(bigger)=w;

%Initial function is odd for first band gap
if w>0.5
    psi(:,1)=2*exp(-(x).^2).*sin(x);
else
    psi(:,1)=exp(-(x).^2); %initial function even for semi-infinite gap
end
err = 1; big = 1;

%Computing the inverse of (EMatr-w*eye(M*N)) for efficiency
IEMatr=inv(EMatr-w*eye(M*N));

%Perform iterations
while err > 10^(-15) && big<50
    psi_hat =  (IuMatr*psi(:,big)); %Bloch-fourier Transform from x to k
    psi3=psi(:,big).^3;
    psi_hat3 = (IuMatr*psi3); %Bloch-Fourier Transform from x to k

    %Calculate M_n
    holdit1 = psi_hat'*(EMatr-w*eye(N*M))*psi_hat;
    holdit2 = psi_hat'*psi_hat3;
    Mn(big) = real(holdit1/holdit2);

    %calculate next u
```

```
    psi_hat_new = IEMatr*psi_hat3;
    %Bloch-fourier Transform from k to x
    psi(:,big+1) = real(Mn(big)^(3/2)*(uMatr*psi_hat_new));

    %calculate error e_u
    erru(big) = norm(psi(:,big+1)-psi(:,big),inf);
    err = norm(psi(:,big+1)-psi(:,big),inf);

    %Calculate exact error for V(x)=0
    errorNorm = norm(abs(psi(:,big)-sqrt(2)*sech(x')),inf);
    big = big + 1;
end

%Calculate L2 norm
L2(bigger)=norm(psi(:,big),2);
end

%this is where various error analysis is performed and figures are made
plot(x,psi(:,big),'r',x,V,'b'); hold on;
axis([-8*pi 8*pi 0.0 1.2])
errorNorm = norm(abs(psi(:,big)-sqrt(2)*sech(x')),inf)
for co=1:big-1
erroru(co)=norm(psi(:,co)-psi(:,big),inf);
end
jk=polyfit(log10(erroru(1,2:big-3)),log10(erroru(1,1:big-4)),1)
iter=[1:big-1];
figure(2); semilogy(iter,abs(Mn-1),'b',iter,erru,'b.');
legend('e_M','e_u')
```

---

[1] M. Chugunova and D. Pelinovsky. Two-pulse solutions in the fifth-order kdv equation: rigorous theory and numerical approximations. *Discr. cont. Dyn. Syst. B*, 8:773–800, 2007.

[2] J. P. Gordon. Interaction forces among solitons in optical fibers. *Opt. Lett.*, 8(11):596–598, 1983.

[3] Dmitry E. Pelinovsky. Localization in periodic potentials. Courseware, Math 743, McMaster University, 2009.

[4] Dmitry E. Pelinovsky, Andrey A. Sukhorukov, and Yuri S. Kivshar. Bifurcations and stability of gap solitons in periodic potentials. *Phys. Rev. E*, 70(3):036618, Sep 2004.

[5] V. I. Petviashvili. Equation of an extraordinary soliton. *Soviet Journal of Plasma Physics*, 2:469–472, June 1976.