Nonlinear Schrödinger equation with a Dirac delta potential: finite difference method[*]

Bin Cheng (程彬)[1], **Ya-Ming Chen** (陈亚铭)[2], **Chuan-Fu Xu** (徐传福)[1], **Da-Li Li** (李大力)[2] **and Xiao-Gang Deng** (邓小刚)[2]

[1] College of Computer, National University of Defense Technology Changsha 410073, China
[2] College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China

E-mail: chenym-08@163.com

Abstract

The nonlinear Schrödinger equation with a Dirac delta potential is considered in this paper. It is noted that the equation can be transformed into an equation with a drift-admitting jump. Then following the procedure proposed in Chen and Deng (2018 *Phys. Rev.* E **98** 033302), a new second-order finite difference scheme is developed, which is justified by numerical examples.

(Some figures may appear in colour only in the online journal)

1. Introduction

We consider in this paper the following nonlinear Schrödinger equation with a Dirac delta potential:

$$i\partial_t \Psi(x, t) = \left(-\frac{1}{2}\partial_x^2 + \lambda\delta(x) + g|\Psi(x, t)|^2 \right)\Psi(x, t), \quad (1)$$

where i is the imaginary unit, λ and g are constants, and $\delta(x)$ represents the Dirac delta function [1]. While the function $\Psi(x, t)$ is continuous at $x = 0$, its derivative is not. Instead, we have the following matching relation

$$\partial_x \Phi(0+, t) - \partial_x \Phi(0-, t) = 2\lambda\Phi(0, t), \quad (2)$$

which can be obtained by firstly integrating equation (1) over the interval $[-\varepsilon, \varepsilon]$, and then considering the limit $\varepsilon \to 0$.

Equation (1) provides an idealized model for short-range interactions [2–6]. Due to the jump condition (2), special techniques shall be employed to discretize equation (1) properly. There are some useful numerical methods for solving equation (1). For instance, a finite element method is developed in [7], where the Dirac delta function is incorporated directly into the discretization of the problem by using the inherent integration of the method. In [8], the evolution operator of equation (1) is approximated by means of the Lie evolution operator. Then spectral splitting methods are implemented. Viewing the problem as an interface at the point $x = 0$, a so-called explicit jump immersed interface method is proposed in [9]. A weak multi-symplectic reformulation is proposed in [10, 11] for equation (1), enabling one to get some preserving properties. One can also regularize the Dirac delta function in equation (1) by using a nonsingular function [12], resulting in an equation that can be handled by some common schemes. For instance, replacing the Dirac delta function by a so-called discrete delta function [13], a wavelet collocation method is constructed in [14].

In this paper, we provide a novel method to solve equation (1) numerically. Rather than discretizing equation (1) directly, we first show in section 2 that equation (1) can be transformed into an equation with discontinuous drift, in which case both the solution and flux are continuous. Then we can employ the method proposed in [15] to solve the transformed equation. For the purpose of efficiency, we provide in section 3 a new finite difference scheme following the derivation procedure in [15]. Numerical experiments are conducted in section 4 to validate the proposed method. Finally, we draw conclusions in section 5.

1

2. Transformation

Consider the following one-dimensional equation:

$$a\partial_t\psi(x,\,t) = [b\partial_x^2 + V(x) + g|\psi|^2]\psi(x,\,t), \qquad (3)$$

where

$$V(x) = \lambda\delta(x) + V_0(x) \qquad (4)$$

with $V_0(x)$ representing a regular potential. Here a, b, g and λ are constants. Let

$$\psi(x,\,t) = e^{c|x|}\phi(x,\,t), \qquad (5)$$

where c is a constant that will be determined later to cancel the δ function in equation (3). It is noted that

$$\partial_x\psi = e^{c|x|}[c\,\mathrm{sgn}(x)\phi + \partial_x\phi], \qquad (6)$$

where the 'sgn' stands for the sign function. Furthermore, using the relation $\partial_x\,\mathrm{sgn}(x) = 2\delta(x)$, we obtain immediately from equation (6) that

$$\partial_x^2\psi = e^{c|x|}\{c^2\phi - 2c\delta(x)\phi + 2c\partial_x[\mathrm{sgn}(x)\phi] + \partial_x^2\phi\}. \qquad (7)$$

Thus, it follows from equations (4) and (7) that

$$\begin{aligned}[b\partial_x^2 + V(x)]\psi = {}& e^{c|x|}\{[bc^2 + V_0(x)]\phi \\ & + (\lambda - 2bc)\delta(x)\phi + 2bc\partial_x[\mathrm{sgn}(x)\phi] + b\partial_x^2\phi\}. \end{aligned} \qquad (8)$$

Therefore, the choice $c = \lambda/(2b)$ cancels the term with the δ function. In this case, we know that the substitution of the transform (5) into equation (3) results in

$$\begin{aligned}a\partial_t\phi = {}& [bc^2 + V_0(x) + ge^{2c|x|}|\phi|^2]\phi \\ & + \lambda\partial_x[\mathrm{sgn}(x)\phi] + b\partial_x^2\phi. \end{aligned} \qquad (9)$$

To solve equation (9) involving the sign function, two matching conditions should be imposed at $x = 0$, i.e. the continuity of ϕ and the continuity of the flux, defined by

$$f(x,\,t) = \lambda\,\mathrm{sgn}(x)\phi + b\partial_x\phi. \qquad (10)$$

In addition, it is observed that equation (9) is very similar to the Fokker–Planck equation considered in [15], except the additional source term $[bc^2 + V_0(x) + ge^{2c|x|}|\phi|^2]\phi$. Therefore, it is no doubt that the numerical scheme developed in [15] can be applied directly here. However, the numerical scheme is only of second order for the case with drift-admitting jumps although it is designed to be fifth-order for the case with smooth drifts. In order to save some computational resource, we prefer to solve equation (9) by developing in the next section a new scheme of third order for the case with smooth drifts rather than fifth order.

3. Numerical scheme

Following the procedure in [15], we develop here a new scheme for solving equation (9). Using the method of line, we first consider how to discretize properly the derivative of the flux, i.e. $\partial_x f$ (see equation (10)). The key is to use a so-called staggered grid.

Suppose that the computational domain is truncated to be $[x_L,\,x_R]$ with $x_L < 0 < x_R$. Then we divide the domain into two subdomains, denoted by $\Omega_1 = [x_L,\,0]$ and $\Omega_2 = [0,\,x_R]$, respectively. In each subdomain Ω_i, solution points and flux points are placed uniformly, where the solution points are defined by $x_{1,j} = x_L + (j - 1/2)h_1$ $(1 \leqslant j \leqslant N_1)$ for Ω_1 and $x_{2,j} = x_d + (j - 1)h_2$ $(1 \leqslant j \leqslant N_2)$ for Ω_2. Here h_i are the spacial steps for Ω_i with $h_1 = -x_L/(N_1 - 1/2)$ and $h_2 = x_R/(N_2 - 1/2)$, where N_i are the numbers of solution points. It is noted that $x_{1,N_1} = x_{2,1} = 0$, meaning that $x = 0$ is set to be a solution point. We will see later that this setup is very important for ensuring the continuity of the solution at $x = 0$. In addition, we define the flux points by $x_{1,j+1/2} = x_L + jh_1$ $(0 \leqslant j \leqslant N_1 - 1)$ for Ω_1 and $x_{2,j+1/2} = (j - 1/2)h_2$ $(1 \leqslant j \leqslant N_2)$ for Ω_2, such that the flux points and the solution points are staggered with each other (see also figure 1 in [15]). Then it follows immediately that both the two boundary points are flux points since $x_{1,1/2} = x_L$ and $x_{2,N_2+1/2} = x_R$.

3.1. Spacial discretization

For the introduced staggered grid, we first evaluate the flux at flux points, and then compute the flux derivative at solution points.

3.1.1. Evaluation of the flux.
Here we only present the scheme for the subdomain Ω_1, while the scheme for the subdomain Ω_2 can be obtained immediately by using a mirror-symmetric property of the grid points. Additionally, let us assume $b > 0$ in equation (9) without loss of generality.

At a certain time level, suppose that the values of ϕ are known at solution points $x_{1,j}$, denoted by $\phi_{1,j}$. Then we derive a third-order interpolation scheme to obtain the values at flux points. For stability reason, we approximate the term $\lambda\,\mathrm{sgn}(x)\phi$ appearing in the flux at flux points $x_{1,j+1/2}$ by

$$\max\{\lambda,\,0\}\phi_{1,j+1/2}^- + \min\{\lambda,\,0\}\phi_{1,j+1/2}^+, \qquad (11)$$

where $\phi_{1,j+1/2}^-$ represents interpolation values by using left-biased stencils while $\phi_{1,j+1/2}^+$ are obtained by using right-biased stencils. The interpolation scheme can be described as follows. Let $\Phi_1^\pm = [\phi_{1,1/2}^\pm,\,\phi_{1,3/2}^\pm,...,\phi_{1,N_1-1/2}^\pm]^T$ and $\Phi_1 = [\phi_{1,1},\,\phi_{1,2},...,\phi_{1,N_1}]^T$. Then the right-biased scheme can be expressed as $\Phi_1^+ = I_1^+\Phi_1$, where the $N_1 \times N_1$ interpolation matrix I_1^+ reads

$$I_1^+ = \begin{pmatrix} \frac{15}{8} & -\frac{5}{4} & \frac{3}{8} & & & \\ \frac{3}{8} & \frac{3}{4} & -\frac{1}{8} & & & \\ & \ddots & \ddots & \ddots & & \\ & & & \frac{3}{8} & \frac{3}{4} & -\frac{1}{8} \\ & & & -\frac{1}{8} & \frac{3}{4} & \frac{3}{8} \end{pmatrix}. \qquad (12)$$

For the left-biased values, the scheme reads $\Phi_1^- = I_1^-[\phi_{1,1/2}^+,\,\Phi_1^T]^T$, where the $N_1 \times (N_1 + 1)$ interpolation

Table 1. Coefficients that determine the difference scheme (18) for the cases $N_1 - 1 \leqslant j \leqslant N_1 + 1$.

	$s = 1$	$s = 2$	$s = 3$
$d_{s,1}(x, y)$	$\dfrac{1}{20x + 4y}$	$\dfrac{4xy - 3y^2}{3x(x + y)(3x + y)}$	$\dfrac{2y^2}{(x + y)(x + 3y)(x + 5y)}$
$d_{s,2}(x, y)$	$-\dfrac{7x + 2y}{6x^2 + 2xy}$	$\dfrac{-12xy + 3y^2}{x(x + y)(x + 3y)}$	$-\dfrac{4x + 5y}{4xy + 4y^2}$
$d_{s,3}(x, y)$	$\dfrac{5x + 4y}{4x^2 + 4xy}$	$-\dfrac{3x^2 - 12xy}{y(x + y)(3x + y)}$	$\dfrac{2x + 7y}{2xy + 6y^2}$
$d_{s,4}(x, y)$	$-\dfrac{2x^2}{(x + y)(3x + y)(5x + y)}$	$\dfrac{3x^2 - 4xy}{3y(x + y)(x + 3y)}$	$-\dfrac{1}{4x + 20y}$

matrix is

$$I_1^- = \begin{pmatrix} 1 & & & & & \\ -\frac{1}{3} & 1 & \frac{1}{3} & & & \\ & -\frac{1}{8} & \frac{3}{4} & \frac{3}{8} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\frac{1}{8} & \frac{3}{4} & \frac{3}{8} \\ & & & & -\frac{1}{8} & \frac{3}{4} & \frac{3}{8} \end{pmatrix}. \quad (13)$$

Let $\Phi_{1,x} = [(\partial_x \phi)_{1,1/2}, (\partial_x \phi)_{1,3/2}, ..., (\partial_x \phi)_{1,N_1-1/2}]^{\mathrm{T}}$. Then we can get the difference scheme $\Phi_{1,x} = A_1 \Phi_1 / h_1$, where the $N_1 \times N_1$ difference matrix A_1 is expressed as follows:

$$A_1 = \begin{pmatrix} -2 & 3 & -1 & & & \\ -1 & 1 & & & & \\ \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & \frac{1}{24} & -\frac{9}{8} & \frac{9}{8} & -\frac{1}{24} \\ & & & & -1 & 1 \end{pmatrix}. \quad (14)$$

Finally, we obtain the approximation to the flux as

$$f_{1,j+1/2} = \max\{\lambda, 0\} \phi_{1,j+1/2}^- + \min\{\lambda, 0\} \phi_{1,j+1/2}^+ + b(\partial_x \phi)_{1,j+1/2}. \quad (15)$$

As mentioned before, the flux $f_{2,j+1/2}$ for the subdomain Ω_2 can be obtained by using the symmetry property of the grid. Moreover, we usually apply boundary conditions $f_{1,1/2} = f_{2,N_2+1/2} = 0$ in applications.

3.1.2. Compute the derivative of the flux. Theoretically, the flux is continuous at $x = 0$. Thus, we can derive a difference scheme for the whole computational domain $[x_L, x_R]$ directly. To describe the algorithm, let us introduce the flux vector $\boldsymbol{f} = [\boldsymbol{f}_1^{\mathrm{T}}, \boldsymbol{f}_2^{\mathrm{T}}]$ with

$$\boldsymbol{f}_1 = [f_{1,1/2}, f_{1,3/2}, \cdots, f_{1,N_1-1/2}]^{\mathrm{T}},$$
$$\boldsymbol{f}_2 = [f_{2,3/2}, f_{2,5/2}, \cdots, f_{2,N_2+1/2}]^{\mathrm{T}}, \quad (16)$$

and the derivative vector $\boldsymbol{f}_x = [\boldsymbol{f}_{1,x}^{\mathrm{T}}, \boldsymbol{f}_{2,x}^{\mathrm{T}}]$ with

$$\boldsymbol{f}_{1,x} = [(\partial_x f)_{1,1}, (\partial_x f)_{1,2}, \cdots, (\partial_x f)_{1,N_1}]^{\mathrm{T}},$$
$$\boldsymbol{f}_{2,x} = [(\partial_x f)_{2,2}, (\partial_x f)_{2,3}, \cdots, (\partial_x f)_{2,N_2}]^{\mathrm{T}}. \quad (17)$$

Then we express the difference scheme as $\boldsymbol{f}_x = A\boldsymbol{f}$, where the entries $a_{j,k}$ of the difference matrix A can be obtain by using the following stencils:

$$[\boldsymbol{f}_x]_j = \begin{cases} \Sigma_{k=1}^3 a_{j,k} [\boldsymbol{f}]_k, & j = 1, \\ \Sigma_{k=1}^4 a_{j,k} [\boldsymbol{f}]_{j+k-2}, & 2 \leqslant j \leqslant N_x - 1, \\ \Sigma_{k=1}^3 a_{j,k} [\boldsymbol{f}]_{N_x+k-2}, & j = N_x. \end{cases} \quad (18)$$

Here $[\boldsymbol{f}]_k$ denotes the kth entry of the vector $\boldsymbol{f}$, and $N_x = N_1 + N_2 - 1$.

By using the method of Lagrangian interpolation, it is straightforward to determine the entries of A. While the stencils locate in a single domain, we have

$$[a_{j,k}]_{1 \leqslant k \leqslant 3} = \begin{cases} \frac{1}{h_1}[-1, 1, 0], & j = 1, \\ \frac{1}{h_2}[0, -1, 1], & j = N_x, \end{cases}$$

$$[a_{j,k}]_{1 \leqslant k \leqslant 4} = \begin{cases} \frac{1}{h_1}\left[\frac{1}{24}, -\frac{9}{8}, \frac{9}{8}, -\frac{1}{24}\right], & 2 \leqslant j \leqslant N_1 - 2, \\ \frac{1}{h_2}\left[\frac{1}{24}, -\frac{9}{8}, \frac{9}{8}, -\frac{1}{24}\right], & N_1 + 2 \leqslant j \leqslant N_x - 1. \end{cases} \quad (19)$$

While the stencils across the interface point $x = 0$, i.e. $N_1 - 1 \leqslant j \leqslant N_1 + 1$, the entries can be determined as $a_{j,k} = d_{j-N_1+2,k}(h_1, h_2)$, where the expressions for $d_{j,k}(x, y)$ are presented in table 1.

3.2. Time-marching scheme

After discretizing the derivative of the flux, we obtain from equation (3) an ordinary differential system, which can be solved by a time-marching scheme. In this paper, the third-order Runge–Kutta scheme as used in [15] is employed (see equations (30) and (31) therein). If not stated otherwise, the time step is chosen to be $0.01 \min\{h_1^2, h_2^2\}$.

4. Numerical examples

In this section, we solve several problems numerically to show the validity of the aforementioned method. To measure the convergence rate, we introduce the L^2-norm error, defined by

$$L^2 \text{error} = \left[\sum_{j=1}^{N_1-1} e_{1,j}^2 h_1 + \frac{1}{2} e_{1,N_1}^2 (h_1 + h_2) + \sum_{j=2}^{N_2} e_{2,j}^2 h_2 \right]^{1/2}, \quad (20)$$
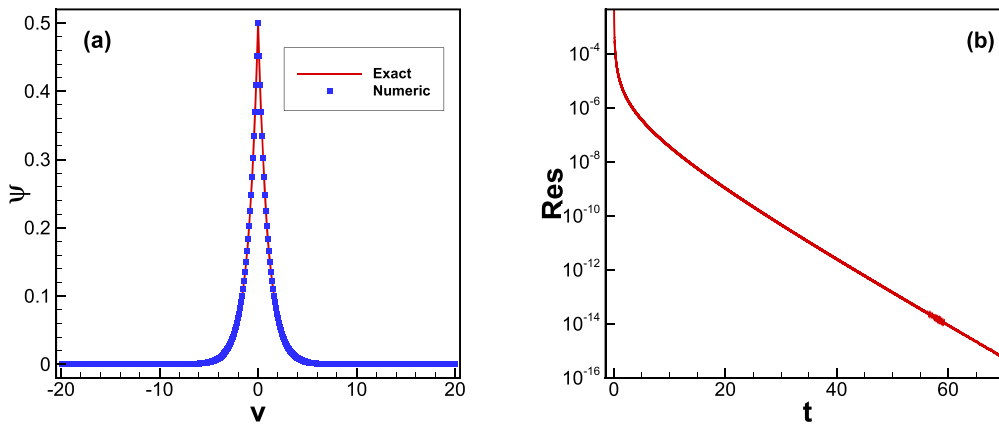
Figure 1. Numerical results for equation (23) with $g = 0$ and $\lambda = -1/2$, where $N_1 = N_2 = 200$ is used to compute the numerical solution that matches well with the analytic solution. (a) Numerical solution; (b) residue.

and the L^∞-norm error

$$L^\infty \text{error} = \max_{i,j} |e_{i,j}|, \qquad (21)$$

where $e_{i,j}$ are the errors between numerical results and analytic solutions. Then the numerical convergence rate is measured by

$$\text{rate} = -\ln(E_M/E_N)/\ln(M/N), \qquad (22)$$

where E_M and E_N are the errors corresponding to the cases with M and N solution points, respectively.

4.1. Stationary nonlinear Schrödinger equation

At first, we consider the stationary nonlinear Schrödinger equation

$$\left(-\frac{1}{2}\partial_x^2 + \lambda\delta(x) + g|\psi|^2\right)\psi = \mu\psi, \qquad (23)$$

where λ, g and μ are constants. Rather than solving equation (23) directly, we can solve the following time-dependent nonlinear Schrödinger equation

$$\Psi_t = \left(\frac{1}{2}\partial_x^2 - \lambda\delta(x) + \mu - g|\Psi|^2\right)\Psi \qquad (24)$$

until Ψ_t reaches the machine zero. Then we obtain equivalently the solution of equation (23).

It is obvious that equation (24) is in the form of the general equation considered in equation (3). Thus the algorithm presented in the previous sections can be applied directly. Here we just specify the initial condition for equation (24) to be Gaussian, i.e.

$$\Psi(x, 0) = \frac{1}{\sqrt{2\pi\tau_0}}e^{-x^2/(2\tau_0)} \qquad (25)$$

with τ_0 chosen to be 0.01. In the following two cases, we set the computational domains to be large enough, enabling us to impose the zero flux boundary condition

$$f(x_L, t) = f(x_R, t) = 0. \qquad (26)$$

Table 2. Accuracy test for equation (23) with $g = 0$ and $\lambda = -1/2$. Here, $N_1 = N_2 = N$.

N	L^2 error	Rate	L^∞ error	Rate
200	4.05E-04		4.08E-04	
400	1.03E-04	1.97	1.03E-04	1.98
800	2.58E-05	2.00	2.59E-05	1.99
1600	6.49E-06	1.99	6.50E-06	1.99

In addition, the computation is stopped until the residue, defined by

$$\text{Res}^n = \max_j |\Psi_j^n - \Psi_j^{n-1}|, \qquad (27)$$

reaches the machine zero.

4.1.1. Linear case.
When $g = 0$, equation (23) degenerates to a linear equation, which admits a bound state solution [16]

$$\psi(x) = e^{\lambda|x|} \qquad (28)$$

with $\mu = -\lambda^2/2$. Here we choose $\lambda = -1/2$ and truncate the computational domain to be $[-20, 20]$. Then we solve equation (24) and obtain the numerical solution as shown in figure 1(a). The convergence process of the numerical solution is shown in figure 1(b). The test of accuracy in table 2 demonstrates that second-order accuracy is achieved, as expected.

4.1.2. Nonlinear case.
For $g \neq 0$, the nonlinear term in equation (24) comes into play. Here we confine the study to the case with $g = 1$, which admits the analytic solution [17]

$$\psi(x) = \begin{cases} -k\,\text{cosech}(k(x + x_0)), & x < 0, \\ k\,\text{cosech}(k(x - x_0)), & x > 0 \end{cases} \qquad (29)$$

if $\mu = -(\lambda - 1/2)^2/2$. Here x_0 is determined by the equality
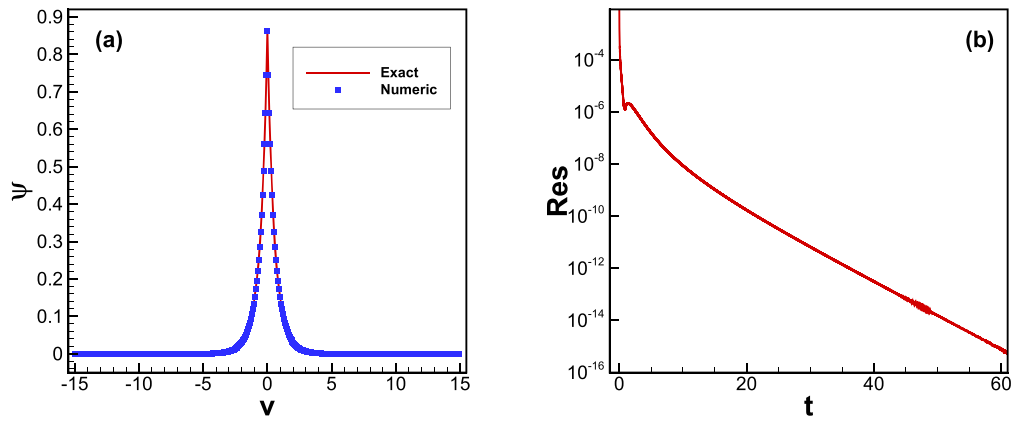
$$\tanh(kx_0) = k/\lambda, \qquad (30)$$

Figure 2. Numerical results for equation (23) with $g = 1$ and $\lambda = -1$. Here, $N_1 = N_2 = 200$. (a) Numerical solution; (b) residue.
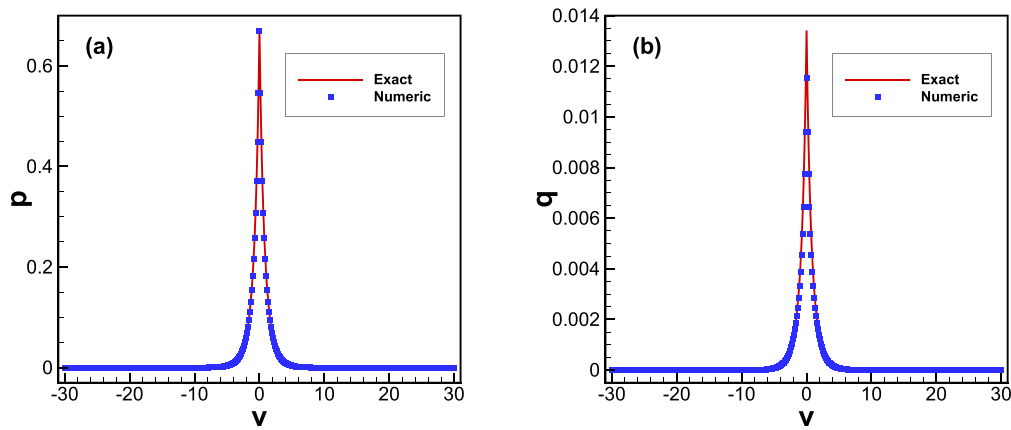

Figure 3. Numerical solution of equation (1) with $g = 1$ and $\lambda = -0.7$ at time $t = 1$. Here, $N_1 = N_2 = 200$. (a) Solution of the real part p; (b) solution of the imaginary part q.

which is a consequence of the jump condition of equation (23) at $x = 0$; see equation (2). In addition, k is set as $k = -\lambda - 1/2$ such that $\int_{-\infty}^{\infty} |\psi(x)|^2 \mathrm{d}x = 1$.

Here we set $\lambda = -1$ and truncate the computational domain to be $[-15, 15]$. As we can see from figure 2(a), the numerical results match well with the analytic solution. The convergence process of the numerical solution is shown in figure 2(b). The expected second-order convergence rate is also achieved as shown in table 3.

4.2. Time-dependent nonlinear Schrödinger equation

Now we are trying to solve the time-dependent nonlinear Schödinger equation (1) [11]. Since Ψ is a complex number for fixed x and t, we shall assume $\Psi = p(x, t) + \mathrm{i}q(x, t)$ and plug it into equation (1) to obtain a couple of equations

$$\partial_t p = \left(-\frac{1}{2}\partial_x^2 + \lambda\delta(x) + g(p^2 + q^2) \right)q, \qquad (31)$$

$$-\partial_t q = \left(-\frac{1}{2}\partial_x^2 + \lambda\delta(x) + g(p^2 + q^2) \right)p. \qquad (32)$$

Then we can use the proposed numerical method to solve this system.

Table 3. Accuracy test for equation (23) with $g = 1$ and $\lambda = -1$. Here, $N_1 = N_2 = N$.

N	L^2 error	Rate	L^∞ error	Rate
100	6.27E-03		9.84E-03	
200	1.56E-03	2.00	2.51E-03	1.96
400	3.89E-04	2.00	6.23E-04	2.01
800	9.70E-05	2.00	1.54E-04	2.01

When $g = 1$, equation (1) possesses the analytic solution

$$\Psi(x, t) = \psi(x)\exp\left(\mathrm{i}(2\lambda - 1)^2 t/8\right), \qquad (33)$$

where $\psi(x)$ is defined by equation (29). We choose $\lambda = -0.7$ and set the initial condition as $\Psi(x, 0)$ according to equation (33). The computational domain is truncated to be $[-20, 20]$. Figure 3 shows that the numerical results of the real and imaginary parts match well with the analytic solutions at time $t = 1$. In this case, second-order accuracy is still observed, as shown in table 4.

5. Conclusions

To conclude, we have proposed in this paper a transform for the one-dimensional nonlinear Schrödinger equation with a

Table 4. Accuracy test for equation (1) with $g = 1$ and $\lambda = -0.7$ at time $t = 1$. Here, $N_1 = N_2 = N$.

	p				q			
N	L^2 error	Rate	L^∞ error	Rate	L^2 error	Rate	L^∞ error	Rate
200	5.92E-04		1.22E-03		1.55E-03		1.90E-03	
400	1.48E-04	2.00	3.04E-04	2.00	3.93E-04	1.98	4.88E-04	1.96
800	3.82E-05	1.95	7.49E-05	2.02	9.94E-05	1.98	1.24E-04	1.97
1600	9.80E-06	1.96	1.85E-05	2.02	2.50E-05	1.99	3.11E-05	1.99

Dirac delta potential, such that the annoying Dirac delta function can be canceled out. For the transformed equation without the delta function, we have provided an efficient second-order finite difference method. In the numerical experiments, we have considered both stationary solutions and time-dependent solutions. All the computed results validate the new method, which is second-order.

In the future, we may consider whether the same idea is applicable for the case with double delta potentials [18–20] or delta comb potentials [21–23]. In addition, we may also consider how to improve the convergence rate of the numerical scheme.

References

[1] Belloni M and Robinett R W 2014 *Phys. Rep.* **540** 25
[2] Hakim V 1997 *Phys. Rev.* E **55** 2835
[3] Adami R and Sacchetti A 2005 *J. Phys. A: Math. Gen.* **38** 8379
[4] Bai E and Shu Q 2005 *Chin. Phys.* **14** 208
[5] Coz S L, Fukuizumi R, Fibich G, Ksherim B and Sivan Y 2008 *Phys.* D **237** 1103
[6] Billam T P, Helm J L and Gardiner S A 2012 *Phys. Rev.* A **85** 013627
[7] Holmer J, Marzuola J and Zworski M 2007 *J. Nonlin. Sci.* **17** 349
[8] Sacchetti A 2007 *J. Comput. Phys.* **227** 1483
[9] Bai J and Wang L 2015 *Appl. Math. Comput.* **254** 113
[10] Bai J, Li C and Liu X 2018 *IMA J. Numer. Anal.* **38** 399
[11] Bai J 2016 *J. Math. Anal. Appl.* **444** 721
[12] Tornberg A K and Engquist B 2003 *J. Sci. Comput.* **19** 527
[13] Peskin C S 1977 *J. Comput. Phys.* **25** 220
[14] Qian X, Fu H and Song S 2017 *Appl. Math. Comput.* **307** 1
[15] Chen Y and Deng X 2018 *Phys. Rev.* E **98** 033302
[16] Bie H D 2008 *Phys. Lett.* A **372** 4350
[17] Witthaut D, Mossmann S and Korsch H J 2005 *J. Phys. A: Math. Gen.* **38** 1777
[18] Cacciari I and Moretti P 2006 *Phys. Lett.* A **359** 396
[19] Cartarius H, Haag D, Dast D and Wunner G 2012 *J. Phys. A: Math. Theor.* **45** 444008
[20] Ahmed Z, Kumar S, Sharma M and Sharma V 2016 *Eur. J. Phys.* **37** 045406
[21] Witthaut D, Rapedius K and Korsch H J 2009 *J. Nonlin. Math. Phys.* **16** 207
[22] Erman F, Gadella M and Uncu H 2017 *Phys. Rev.* D **95** 045004
[23] Erman F, Gadella M and Uncu H 2018 *Eur. J. Phys.* **39** 035403