Nonlinear Dyn (2009) 56: 69–83
DOI 10.1007/s11071-008-9380-0

ORIGINAL PAPER

A differential quadrature algorithm for nonlinear Schrödinger equation

Alper Korkmaz · İdris Dağ

Received: 24 August 2007 / Accepted: 4 June 2008 / Published online: 16 July 2008

Abstract Numerical solutions of a nonlinear Schrödinger equation is obtained using the differential quadrature method based on polynomials for space discretization and Runge–Kutta of order four for time discretization. Five well-known test problems are studied to test the efficiency of the method. For the first two test problems, namely motion of single soliton and interaction of two solitons, numerical results are compared with earlier works. It is shown that results of other test problems agrees the theoretical results. The lowest two conserved quantities and their relative changes are computed for all test examples. In all cases, the differential quadrature Runge–Kutta combination generates numerical results with high accuracy.

Keywords Differential quadrature · Interaction of solitons · Lagrange interpolation polynomials · Nonlinear Schrödinger equation · Solitary waves

1 Introduction

Nonlinear Schrödinger equation (NLSE) is shown as a model of a wide class of physical phenomena such as propagation of optical pulses, waves in water, waves in plasmas, electromagnetic waves, and self-focusing in laser pulses. Theoretical solution of this equation is unknown for the more general initial conditions. The analytic solutions of the NLSE are given in studies [1–3], if the initial condition $U(x,0)$ vanishes for sufficiently large x. So far, commonly used numerical methods, such as finite difference spectral and finite element methods, have been applied to obtain numerical solutions of NLSE [4–8]. The quadrature discretization method has also been applied to obtain numerical solutions for the NLSE by some authors [9–12].

Differential quadrature method (DQM) is an efficient approximation to the derivative of a function at a discrete point using linear sum of a functional values at discrete points in the whole domain. Its ease to apply for getting numerical solutions of partial differential equations when compared with finite element methods makes the method preferable. Differential quadrature method is first proposed by Bellman et al. [13] being inspired by integral quadrature. The main purpose of this method is to find the weighting coefficients of the functional values at discrete points using base functions, whose derivatives are also known at the same points in the whole domain, that is,

$$U_x^{(p)}(x_i) = \sum_{j=0}^{N} w_{ij}^{(p)} U(x_j), \quad i = 0, 1, \ldots, N \tag{1}$$

A. Korkmaz (✉)
Anadolu Guzel Sanatlar Lisesi, Kütahya, Turkey
e-mail: alperkorkmaz7@gmail.com

İ. Dağ
Osmangazi University, Eskişehir, Turkey

where $U_x^{(p)}$ denotes the pth order derivative of the function U with respect to the variable x and $w_{ij}^{(p)}$ is the weighting coefficient of the pth order derivative. Many authors obtained weighting coefficients implicitly or explicitly using various test functions such as Legendre polynomials, Lagrange interpolation polynomials, spline functions, radial basis functions, harmonic functions, etc. [13–20].

Owing to giving very accurate results and easy application, DQM has been recently used to obtain numerical solutions of many engineering problems by various authors [21–26].

In this study, we shall apply the differential quadrature method based on polynomials given by [17] for space discretization and classical Runge–Kutta method of order four for time discretization to obtain numerical solutions of NLSE with cubic nonlinearity for various initial conditions. In fact, NLSE has some analytic solutions, but we have aimed to test the accuracy of the DQM by measuring the error between the analytical and numerical solutions we obtain.

2 The governing equation

In the present paper, NLSE is given by

$$iU_t = -U_{xx} - q|U|^2U, \quad (x,t) \in (-\infty,\infty) \times (0,T) \tag{2}$$

where the subscripts x and t denote differentiation, q is a real parameter and $i = \sqrt{-1}$. $U(x,t)$, which governs weakly nonlinear, strongly dispersive, and almost monochoromatic wave [27], is a complex valued-function.

In order to compute a solution on the interval $[a,b]$, artificial boundary conditions $U(a,t) = U(b,t) = 0$ are chosen to model the physical conditions that $U \to 0$ as $x \to \pm\infty$. Solutions decrease rapidly at infinity and it seems reasonable to use such an approximation. Assuming

$$U(x,t) = r(x,t) + is(x,t) \tag{3}$$

where $r(x,t)$ and $s(x,t)$ are real functions, substituting (3) into (2) leads to the associated coupled pair of real differential equation system.

$$s_t = r_{xx} + q\left(r^2 + s^2\right)r, \tag{4}$$

$$r_t = -s_{xx} - q\left(r^2 + s^2\right)s. \tag{5}$$

3 Numerical method

DQM is an approximation to a derivative of a function at a certain discrete points using the linear sum of the functional values at certain discrete points [13]. Thus, if a function $U(x)$ is sufficiently smooth over the whole domain, the first and the second order derivatives of the function $U(x)$ with respect to x at a grid point x_i are approximated by a linear sum of all the functional values in the whole domain, that is,

$$U_x(x_i) = \left.\frac{dU}{dx}\right|_{x=x_i} = \sum_{j=0}^{N} w_{ij}^{(1)} U(x_j), \quad \text{for } i = 0,1,\ldots,N, \tag{6}$$

$$U_{xx}(x_i) = \left.\frac{d^2U}{dx^2}\right|_{x=x_i} = \sum_{j=0}^{N} w_{ij}^{(2)} U(x_j), \quad \text{for } i = 0,1,\ldots,N \tag{7}$$

where $w_{ij}^{(1)}$ and $w_{ij}^{(2)}$ represent the weighting coefficients of the first and second order derivative approximations, respectively. To discretize the problem domain, we choose a uniform grid distribution, that is,

$$a = x_0 < x_1 < \cdots < x_N = b.$$

Using the Lagrange interpolation polynomials as test function, we let,

$$g_k(x) = \frac{L(x)}{(x - x_k)L^{(1)}(x_k)}, \quad k = 0,1,\ldots,N \tag{8}$$

where

$$L(x) = (x - x_0)(x - x_1)\cdots(x - x_N)$$

and

$$L^{(1)}(x_k) = \prod_{l=0, l\neq k}^{N} x_k - x_l$$

when $x \neq x_k$ and

$$g_k(x) = x^k, \quad k = 0,1,\ldots,N \tag{9}$$

otherwise since both test functions given in (8) and (9) span the problem domain. The off-diagonal weighting coefficients for the first-order derivative are determined as [17]:

$$w_{ij}^{(1)} = \frac{L^{(1)}(x_i)}{(x_i - x_j)L^{(1)}(x_j)} \quad \text{for } j \neq i. \tag{10}$$

When the base 1, from the set of base polynomials $g_k(x)=x^k, k=0,1,\dots,N$, is used as a test function, diagonal weighting coefficients are found as

$$w_{ii}^{(1)}=-\sum_{j=0,j\neq i}^{N} w_{ij}^{(1)}. \tag{11}$$

When the problem domain $[-1,1]$, the function $L(x)$ can be stated in a different way by choosing Chebyshev collocation points. In this study, we aim to modify the weighting coefficients formulation given in (10). Assuming that grid points are chosen as

$$x_i=\cos\theta_i,\quad \theta_i=i\pi/N,\ i=0,1,\dots,N \tag{12}$$

then the function $L(x)$ can be rewritten as

$$L(x)=(x-x_0)(x-x_1)\cdots(x-x_N) \tag{13}$$

where x_i is Chebyshev collocation points. After some algebra, $L(x)$ can be obtained as

$$L(x)=\left(1-x^2\right)T_N^{(1)}(x) \tag{14}$$

where $T_N^{(1)}(x)$ is the first order derivative of $T_N(x)=\cos(N\theta)$. Using the following formulations,

$$x=\cos(\theta),\qquad T_N(x)=\cos(N\theta)$$

the function $L(x)$ given in (14) can be reduced to

$$L(x)=L(\theta)=N\sin(\theta)\sin(N\theta). \tag{15}$$

Differentiating (15) successively, we obtain

$$L^{(1)}(x)=-\frac{N\cos(\theta)\sin(N\theta)+N^2\sin(\theta)\cos(N\theta)}{\sin(\theta)}. \tag{16}$$

Since $N\theta_i=i\pi$, when $\sin(\theta_i)\neq 0$, i.e., $i\neq 0, N$, (16) reduces to

$$L^{(1)}(x_i)=(-1)^{i+1}N^2. \tag{17}$$

When $\sin(\theta_i)=0$, application of L'Hospital's rule gives

$$L^{(1)}(x_0)=-2N^2, \tag{18}$$

$$L^{(1)}(x_N)=(-1)^{N+1}2N^2. \tag{19}$$

Substituting (17)–(19) into (10), we obtain the following simplified form

$$w_{ij}^{(1)}=\frac{(-1)^{i+j}\bar{c}_i}{\bar{c}_j(x_i-x_j)},\quad \text{for } j\neq i \tag{20}$$

where

$$\bar{c}_k=\begin{cases}2, & k=0,N,\\ 1, & k=1,2,\dots,N-1.\end{cases} \tag{21}$$

Using the Chebyshev collocation points, we have only modified $L(x_i)$ and $L(x_j)$ in the weighting coefficient formulation of the first order derivative given in (10).

When the problem domain is not $[1,-1]$, but $[a,b]$, a transformation is used to map the interval $[a,b]$ in the x domain onto the interval $[1,-1]$ in the ξ domain, that is,

$$\xi=1-2\frac{(x-a)}{(b-a)}. \tag{22}$$

Using the transformation given in (22), (20) can be modified as

$$w_{ij}^{(1)}=\frac{(-1)^{i+j}2\bar{c}_i}{(a-b)\bar{c}_j(x_i-x_j)},\quad \text{for } j\neq i. \tag{23}$$

As we examine (23) and (20), we see the multiplier $2/(a-b)$ in (23), which comes from the derivative of the transformation given in (22).

The weighting coefficients of the second order derivative with respect to x can be computed using the test function (8) and the base 1. Shu [17] gives the formulation as

$$w_{ij}^{(2)}=2w_{ij}^{(1)}\left(w_{ii}^{(1)}-\frac{1}{(x_i-x_j)}\right),\quad \text{for } j\neq i, \tag{24}$$

$$w_{ii}^{(2)}=-\sum_{j=0,j\neq i}^{N} w_{ij}^{(2)}. \tag{25}$$

Using the differential quadrature approximation given by (6)–(7) for space discretization of the system in (4)–(5), we get the ordinary differential equation system,

$$\left.\frac{\partial s}{\partial t}\right|_{x=x_i}=\sum_{j=0}^{N} w_{ij}^{(2)}r_j+q\left(r_i^2+s_i^2\right)r_i, \tag{26}$$

$$\left.\frac{\partial r}{\partial t}\right|_{x=x_i}=-\sum_{j=0}^{N} w_{ij}^{(2)}s_j-q\left(r_i^2+s_i^2\right)s_i \tag{27}$$

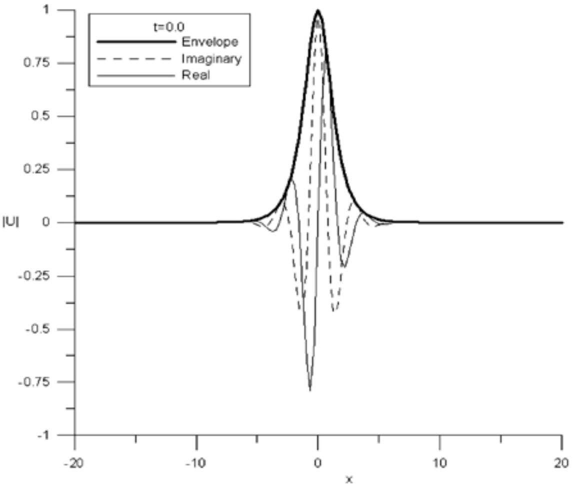


Fig. 1 Single solitary at time $t = 0$ ($N = 400$)

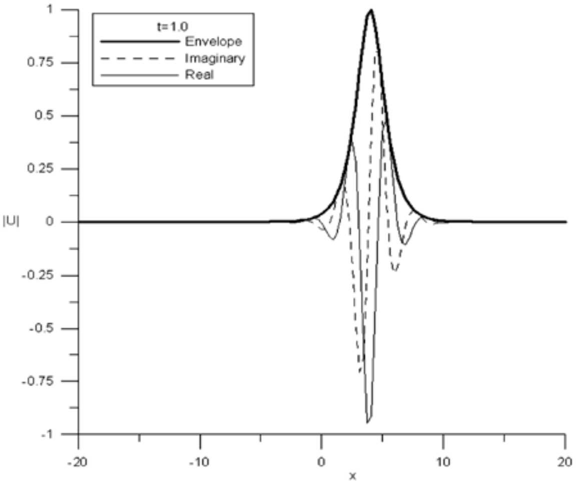


Fig. 2 Single solitary at time $t = 1.0$ ($N = 400$)

where weighting coefficients $w_{ij}^{(2)}$ are found using (24)–(25), and r_k and s_k $(k = i, j)$ denote $r(x_k)$ and $s(x_k)$, respectively.

Since $U(a,t) = U(b,t) = 0$ are chosen as boundary conditions for all test problems throughout the paper, so that $s(x_0,t) = s(x_N,t) = r(x_0,t) = r(x_N,t) = 0$. Therefore, the ODE system becomes,

$$\left.\frac{\partial s}{\partial t}\right|_{x=x_i} = \sum_{j=1}^{N-1} w_{ij}^{(2)} r_j + q\big[r_i^2 + s_i^2\big] r_i, \tag{28}$$

$$\left.\frac{\partial r}{\partial t}\right|_{x=x_i} = -\sum_{j=1}^{N-1} w_{ij}^{(2)} s_j - q\big[r_i^2 + s_i^2\big] s_i,$$

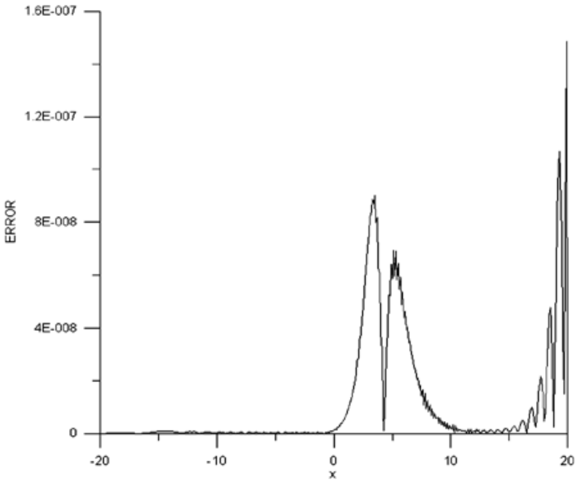


Fig. 3 Error $|U - U^N|$ at time $t = 1.0$ ($N = 400$)

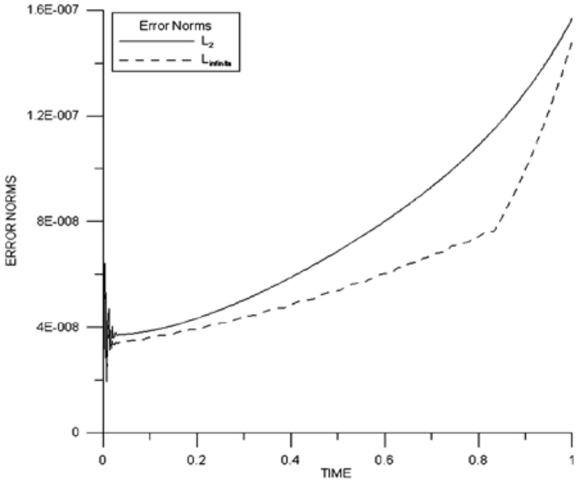


Fig. 4 Change of error norms as time increases ($N = 400$).

$$i = 1, \ldots, N-1. \tag{29}$$

Later, we solve the system given by (28) and (29) using the Runge–Kutta method of order four.

4 Numerical experiments

The differential quadrature method has been widely used to get numerical solutions of static problems since it was introduced. When the problems are independent of time, the method DQ gives very accurate results even a few grid points are used. The usage of the method DQ on solutions of dynamical problems

Table 1 Invariants and error norms for single solitary with $h = 0.1$, $\Delta t = 0.0025$, $N = 400$

Time	C_1	C_2	L_∞	L_2
0.0	2.00000000000000	7.33333333359968	0.00000×10^{-08}	0.00000×10^{-08}
0.25	1.99999999997691	7.33333333464953	4.15442×10^{-08}	4.65170×10^{-08}
0.50	1.99999999995218	7.33333333814519	5.37962×10^{-08}	6.85986×10^{-08}
0.75	1.99999999992528	7.33333334465343	7.10565×10^{-08}	9.98982×10^{-08}
1.0	1.99999999989556	7.33333335509142	1.93202×10^{-07}	1.60224×10^{-07}

Table 2 Comparison of single soliton at time $t = 1$ with some earlier results

Method	h	Δt	L_∞	$\frac{C_1-C_1^0}{C_1^0}$	$\frac{C_2-C_2^0}{C_2^0}$
PDQ (present) ($N = 400$)	0.1	0.0025	1.9320×10^{-7}	-5.2222×10^{-11}	2.9306×10^{-9}
PDQ (present) ($N = 128$)	0.3125	0.020	2.5317×10^{-5}	-1.4403×10^{-06}	-3.9419×10^{-6}
B-spline Galerkin [8]	0.05	0.005	0.0003	0.0000000	0.0000006
	0.3125	0.020	0.002	0.0000066	−0.0003417
B-spline Col. [31]	0.05	0.005	0.008	0.00000	0.00000
	0.03	0.005	0.002	0.00000	0.00000
Explicit [5]	0.05	0.000625	0.006	0.00000	0.00556
Implicit/explicit [5]	0.05	0.001	0.006	−0.00393	−0.01205
Implicit (C-N) [5]	0.05	0.005	0.006	−0.00001	−0.00557
Hopscotch [5]	0.08	0.002	0.005	0.00003	−0.01407
Split-step Fourier [5]	0.3125	0.020	0.005	0.00000	0.00005
A-L Local [5]	0.06	0.0164	0.006	0.00004	−0.00797
A-L Global [5]	0.05	0.040	0.006	0.00003	0.00550
Pseudospectral [5]	0.3125	0.00026	0.005	0.00001	−0.00003

Table 3 Error norms and rates of convergence in space at time $t = 1$

N	L_2	Order	L_∞	Order
20	1.920	–	0.860	–
40	6.21×10^{-1}	1.63	2.87×10^{-1}	1.58
60	1.38×10^{-1}	3.71	5.57×10^{-2}	4.04
80	1.91×10^{-2}	6.87	6.99×10^{-3}	7.21
100	1.61×10^{-3}	11.08	5.34×10^{-4}	11.53
120	1.14×10^{-4}	14.52	4.59×10^{-5}	13.46
140	1.09×10^{-5}	15.23	4.44×10^{-6}	15.15

Table 4 Error norms for $h = 0.3125$ at time $t = 1$

Δt	N	$L_2 \times 10^{-5}$	$L_\infty \times 10^{-5}$
0.02500	40	8.32	6.52
0.02000	50	4.86	3.25
0.01250	80	4.58	1.88
0.01000	100	4.69	1.87
0.00500	200	4.76	1.87
0.00250	400	4.76	1.87
0.00125	800	4.76	1.87

is rare. In this study, we aim to show the applicability of the method DQ on the solutions of a dynamical problem, one dimensional Schrödinger equation with cubic nonlinearity. As test problems, we examine the numerical solutions of NLSE using five different initial conditions.

In order to get a reliable solution, discrete conservation laws are an important feature in computing smooth solitons of the NLSE equation given in the form

$$iU_t = -U_{xx} - q|U|^2 U,$$

$$(x, t) \in (-\infty, \infty) \times (0, T).$$

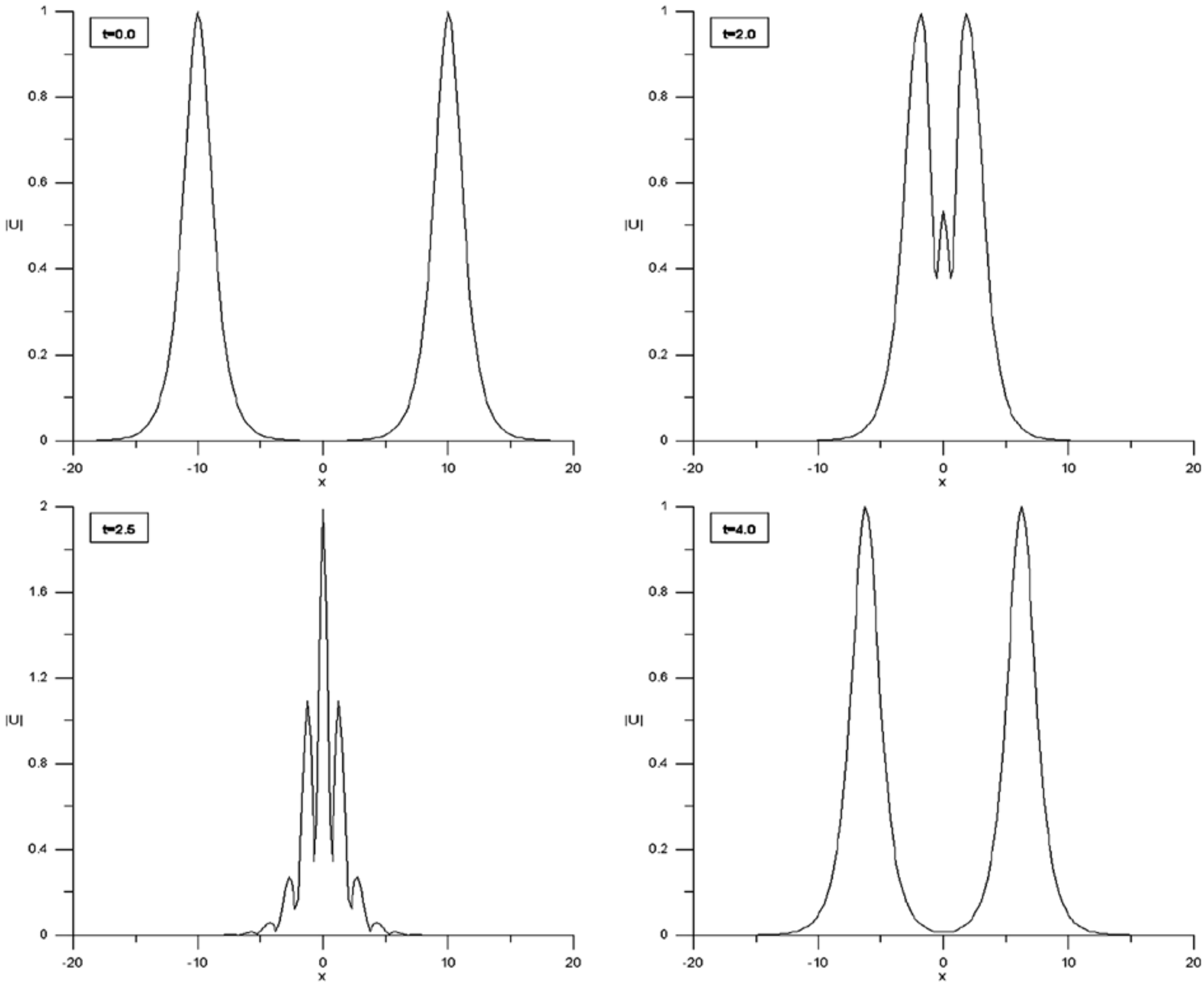


Fig. 5 Interaction of two solitons ($N = 160$)

So, the basic conservations will be satisfied in verifying efficiency of the proposed method. The conservation properties will be calculated for the lowest two invariants:

$$C_1 = \int_a^b |U|^2\,dx \approx h\sum_{j=0}^{N} \left|U_j^n\right|^2, \tag{30}$$

$$C_2 = \int_a^b \left[\left|\frac{\partial U}{\partial x}\right|^2 - \frac{1}{2}q|U|^4\right] dx$$

$$\approx h\sum_{j=0}^{N}\left(\left|\left(\frac{\partial U}{\partial x}\right)_j^n\right|^2 - \frac{1}{2}q\left|U_j^n\right|^4\right) \tag{31}$$

where U and U_j^n are the analytical solution and numerical solution at nth time step at jth node, respectively. The accuracy is measured by using the L_∞ and L_2 error norms defined by

$$L_\infty = \left\|U - U^n\right\|_\infty = \max_j \left|U_j - U_j^n\right| \tag{32}$$

and

$$L_2 = \left\|U - U^n\right\|_2 = \left[h\sum_{j=0}^{N}\left|U_j - U_j^n\right|^2\right]^{1/2}. \tag{33}$$

Relative changes of invariants are defined as $\frac{C_1 - C_1^0}{C_1^0}$ and $\frac{C_2 - C_2^0}{C_2^0}$ where C_1^0 and C_2^0 are the conserved quantities C_1 and C_2 at time $t = 0$.

We used Visual Basic on a desktop computer to obtain all numerical results. The codes to compute CPU

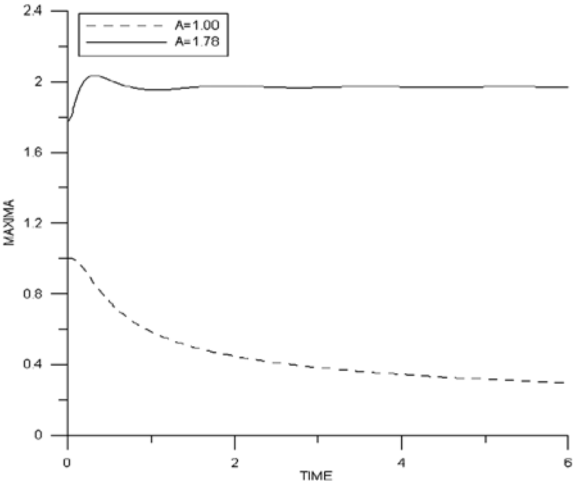


Fig. 6 Maxima for $A = 1$ and $A = 1.78$

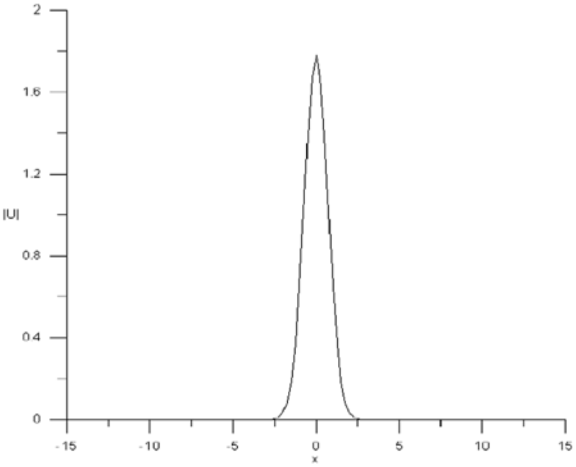


Fig. 7 $N = 360, t = 0, A = 1.78$

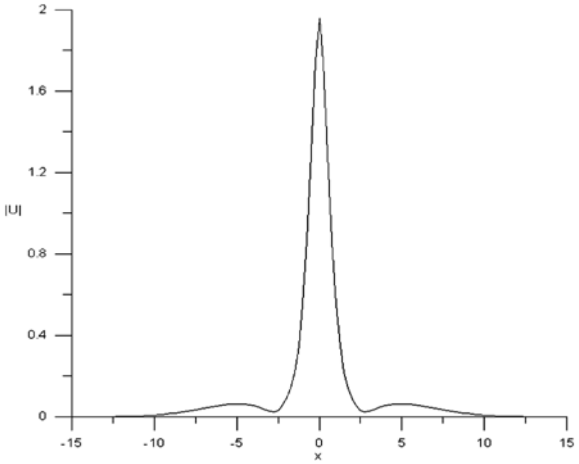


Fig. 8 $N = 360$, $t = 1$ for $A = 1.78$

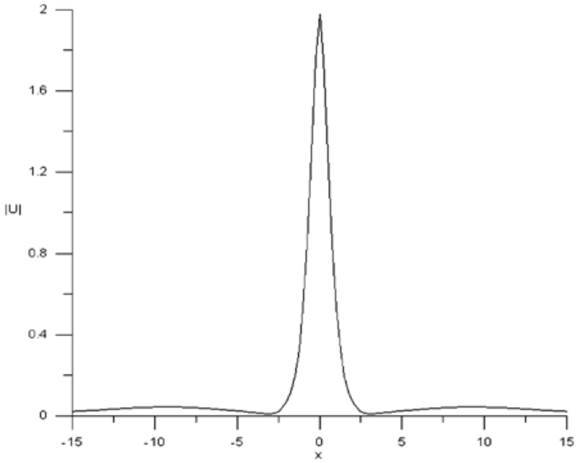


Fig. 9 $N = 360$, $t = 2$ for $A = 1.78$

times are inserted into the part that determines grids and weighting coefficients and the Runge–Kutta loop over all progress time.

Test 1. Single solitary solution

Analytic single soliton solution of the NLSE equation is

$$U(x,t) = \alpha\left(\frac{2}{q}\right)^{1/2} \exp i\left\{\frac{1}{2}Sx - \frac{1}{4}(S^2-\alpha^2)t\right\} \times \operatorname{sech}\alpha(x-St) \tag{34}$$

where S represents the speed of the soliton whose magnitude depends on α. To compare the analytical solution in (34) with those found in [5, 8, 31], computation is done with parameters $q = 2, S = 4, \alpha = 1$ over interval $-20 \le x \le 20$. When $\alpha = 1$, envelop soliton,

$$|U| = \operatorname{sech}(x-4t)$$

propagates to the right with unchanged profiles at a constant speed 4. Using these parameters and $h = 0.1, \Delta t = 0.0025$, plots of the traveling soliton are presented in Figs. 1 and 2 at time $t = 0.0$ and $t = 1.0$, respectively. In these figures, the real and imaginary components and the modules are displayed. The error between analytical and numerical solutions at time $t = 1.0$ and errors vs. time plots are shown in Figs. 3 and 4, respectively. Invariants are computed analyti-

Table 5 Conserved quantities at various times for interaction, $h = 0.25$, $\Delta t = 0.01$, $N = 160$

t	C_1	C_2	$\frac{C_1 - C_1^0}{C_1^0}$	$\frac{C_2 - C_2^0}{C_2^0}$
0.0	3.9999999896	14.6666680173	–	–
1.0	3.9999998170	14.6666622159	$-4.3157646345 \times 10^{-8}$	$-3.9555022834 \times 10^{-7}$
2.0	3.9999996426	14.6666593160	$-8.6746112626 \times 10^{-8}$	$-5.9326924173 \times 10^{-7}$
2.5	3.9999997509	14.6666033188	$-5.9680306141 \times 10^{-8}$	$-4.4112591496 \times 10^{-6}$
3.0	3.9999995003	14.6666578450	$-1.2233580504 \times 10^{-7}$	$-6.9356679566 \times 10^{-7}$
4.0	3.9999993140	14.6666571024	$-1.6890611609 \times 10^{-7}$	$-7.4419334505 \times 10^{-7}$
5.0	3.9999991036	14.6666616240	$-2.2150676370 \times 10^{-7}$	$-4.3588854792 \times 10^{-7}$

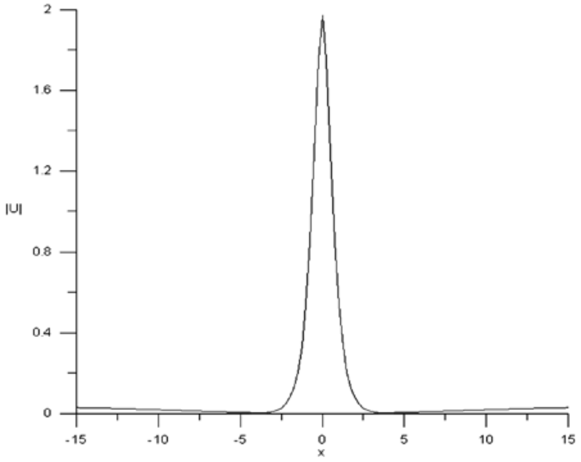


Fig. 10 $N = 360$, $t = 4$ for $A = 1.78$

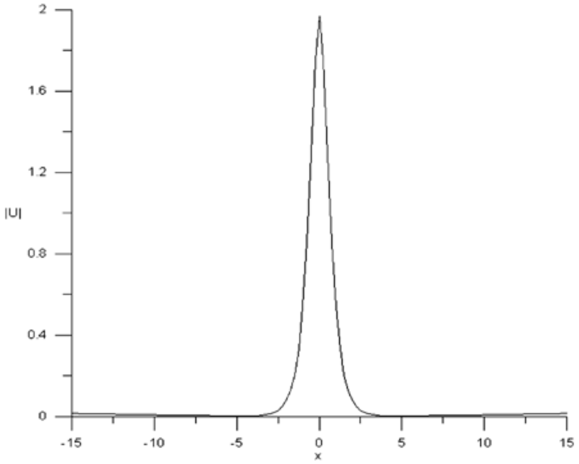


Fig. 11 $N = 360$, $t = 6$ for $A = 1.78$

cally to give $C_1 = 2$ and $C_2 = 7.333333$ by using the initial condition obtained by letting $t = 0$ in (34). Two conserved quantities defined by (30) and (31) are computed numerically by means of the trapezoidal rule and shown in Table 1. Comparison of L_∞ error norm and relative errors of lowest two conserved quantities with some earlier works are made in Table 2. CPU times are also computed for both parameters. When the parameters $h = 0.1$, $\Delta t = 0.0025$ are used CPU time is computed as 171 seconds, while for $h = 0.3125$, $\Delta t = 0.02$, it is computed as 2 seconds.

Accuracy of the proposed methods is shown by calculating the pointwise rate of convergence. Since we have observed that rate of convergence is validated for relation between the time steps and space steps $h \gg \Delta t$. We note that the observed error increases rapidly when the solitary starts hitting the right end of the domain, Fig. 3. The domain of the problem must be chosen large enough so that the boundaries do not affect the solitary wave propagation since the error 1.932×10^{-7} comes from the selection of the boundary condition $U_{\text{bo}}(20, 1) = 0$. Because the difference between the analytical solution and artificial boundary condition is

$$\left|U_{\text{bo}}(20, 1) - U(20, 1)\right| = 2.250703494 \times 10^{-7}$$

where $U_{\text{bo}}(20, 1)$ and $U(20, 1)$ denote the functional values of boundary condition at the right boundary and analytical solution at the time $t = 1.0$, respectively. Therefore, we have repeated the numerical experiments for the space interval $-20 \leq x \leq 24$. The rate of convergence for the scheme is calculated using the following formula

$$\text{rate of convergence} \approx \frac{\ln(E(N_2)/E(N_1))}{\ln(N_1/N_2)}$$

where $E(N_j)$ is either the L_∞-error or the L_2-error when using N_j subintervals. Thus, we perform some further numerical run for some space steps N and a

Table 6 Comparison of two soliton simulations with results ([8] Table 5)

Method	h	Δt	t	$\frac{C_1-C_1^0}{C_1^0}$	$\frac{C_2-C_2^0}{C_2^0}$	CPU time
PDQ (present) ($N=160$)	0.25	0.010	2.5	-5.9680×10^{-8}	-4.4112×10^{-6}	16.8125
PDQ (present) ($N=160$)	0.25	0.005	2.5	4.0901×10^{-8}	-4.0771×10^{-6}	33.9218
PDQ (present) ($N=200$)	0.20	0.005	2.5	3.0645×10^{-9}	-5.1839×10^{-7}	52.5937
B-spline Col. [31]	0.10	0.01	2.5	0.00000	0.00120	
Explicit [5]	0.13	0.0036	2.5	0.00000	0.00659	
Implicit/explicit [5]	0.05	0.0025	2.5	0.00314	0.01434	
Implicit (C-N) [5]	0.13	0.04	2.5	−0.00009	0.00619	
Hopscotch [5]	0.05	0.001	2.5	0.00003	0.00063	
Split-step Fourier [5]	0.625	0.005	2.5	0.00071	0.03595	
A-L Local [5]	0.07	0.07	2.5	0.00000	0.00156	
A-L Global [5]	0.08	0.045	2.5	−0.00012	0.00148	
Pseudospectral [5]	0.625	0.0071	2.5	0.00073	0.03247	
B-spline Galerkin [8]	0.10	0.01	1.5	0.000002	0.00065	
			2.5	−0.000003	0.00330	
			3.5	0.000002	0.00067	
			4.5	0.000002	0.00061	

Table 7 Birth of standing soliton, $h=0.25$, $\Delta t=0.01$, $N=360$

t	C_1	C_2	$\frac{C_1-C_1^0}{C_1^0}$	$\frac{C_2-C_2^0}{C_2^0}$
$A=1$				
0	1.2533141373	0.3670872118	–	–
1	1.2533141227	0.3670867577	$-1.1596062443\times10^{-8}$	$-1.2370115890\times10^{-6}$
2	1.2533141082	0.3670863096	$-2.3151771166\times10^{-8}$	$-2.4577345797\times10^{-6}$
3	1.2533140936	0.3670858986	$-3.4830872630\times10^{-8}$	$-3.5773318528\times10^{-6}$
4	1.2533140536	0.3670904874	$-6.6748995442\times10^{-8}$	$8.9232651681\times10^{-6}$
5	1.2533140662	0.3670958599	$-5.6684676730\times10^{-8}$	$2.3558790501\times10^{-5}$
6	1.2533140655	0.3670858567	$-5.7287809718\times10^{-8}$	$-3.6915579519\times10^{-6}$
Analytic	1.253 3141373	0.3670872118	–	–
$A=1.78$				
0	3.9710005126	−4.9256176213	–	–
1	3.9710005500	−4.9256219637	$9.41339561855\times10^{-9}$	$8.8160208155\times10^{-7}$
2	3.9710004927	−4.9256251610	$-5.01098551541\times10^{-9}$	$1.5307237204\times10^{-6}$
3	3.9710004009	−4.9256279282	$-2.81426298931\times10^{-8}$	$2.0925099164\times10^{-6}$
4	3.9710001997	−4.9256061790	$-7.88012207708\times10^{-8}$	$-2.3230044987\times10^{-6}$
5	3.9710002655	−4.9254514643	$-6.22284739293\times10^{-8}$	$-3.3733234533\times10^{-5}$
6	3.9710030281	−4.9252673868	$6.33467256649\times10^{-7}$	$-7.1104682934\times10^{-5}$
Analytic	3.9710005126	−4.9256176213		

Table 8 Birth of standing soliton, $h = 0.25$, $\Delta t = 0.01$, $N = 360$

t	C_1	C_2	$\frac{C_1 - C_1^0}{C_1^0}$	$\frac{C_2 - C_2^0}{C_2^0}$
$A = 1$				
0	1.2533141373	5.3803437645	–	–
1	1.2533128878	5.3803115838	$-9.9690096500 \times 10^{-7}$	$-5.9811605401 \times 10^{-6}$
2	1.2533119798	5.3802868126	$-1.7214329087 \times 10^{-6}$	$-1.0585174042 \times 10^{-5}$
3	1.2533111202	5.3802631603	$-2.4072612422 \times 10^{-6}$	$-1.4981225704 \times 10^{-5}$
4	1.2533102575	5.3802506103	$-3.0955961380 \times 10^{-6}$	$-1.7313796893 \times 10^{-5}$
5	1.2533093555	5.3802756317	$-3.8153018511 \times 10^{-6}$	$-1.2663275522 \times 10^{-5}$
6	1.2533082934	5.3807979869	$-4.6627031500 \times 10^{-6}$	$8.4422568559 \times 10^{-5}$
$A = 1.78$				
0	3.9710005126	10.9583844401	–	–
1	3.9709902443	10.9580721270	$-2.5858182791 \times 10^{-6}$	$-2.8499926231 \times 10^{-5}$
2	3.9709811225	10.9577657261	$-4.8829229216 \times 10^{-6}$	$-5.6460333804 \times 10^{-5}$
3	3.9709718317	10.9574451341	$-7.2225892840 \times 10^{-6}$	$-8.5715741782 \times 10^{-5}$
4	3.9709620841	10.9571685608	$-9.6772833347 \times 10^{-6}$	$-1.1095424812 \times 10^{-4}$
5	3.9709460531	10.9599882487	$-1.3714296963 \times 10^{-5}$	$1.4635447269 \times 10^{-4}$
6	3.9709379301	10.9641540844	$-1.5759887008 \times 10^{-5}$	$5.2650500588 \times 10^{-4}$

fixed time step smaller enough than h as $\Delta t = 0.001$. Furthermore, we make the computations about the rate of convergence under the assumption that the method is an algebraically convergent in space. In other words, we suppose that $E(N) \sim N^p$ for some $p < 0$ where $E(N)$ denotes the L_2 or L_∞-error when using N subintervals. The L_∞-error, the L_2-error, and rate of convergence, at time $t = 1$, are shown in Table 3. As a second case, we let $h = 0.3125$ and run the simulations for various values of Δt. The results are given in Table 4.

The results show that differential quadrature algorithm combined with fourth order Runge–Kutta converges rapidly in space, indicating an exponential convergence, while convergence in time remains the same.

Test 2. Soliton interaction

Interaction of two solitons: the initial condition is

$$U(x,0) = U_1(x,0) + U_2(x,0) \tag{35}$$

where

$$U_j(x,0) = \alpha_j \left(\frac{2}{q}\right)^{1/2} \exp i\left\{\frac{1}{2}S(x - x_j)\right\} \times \operatorname{sech} \alpha_j (x - x_j) \tag{36}$$

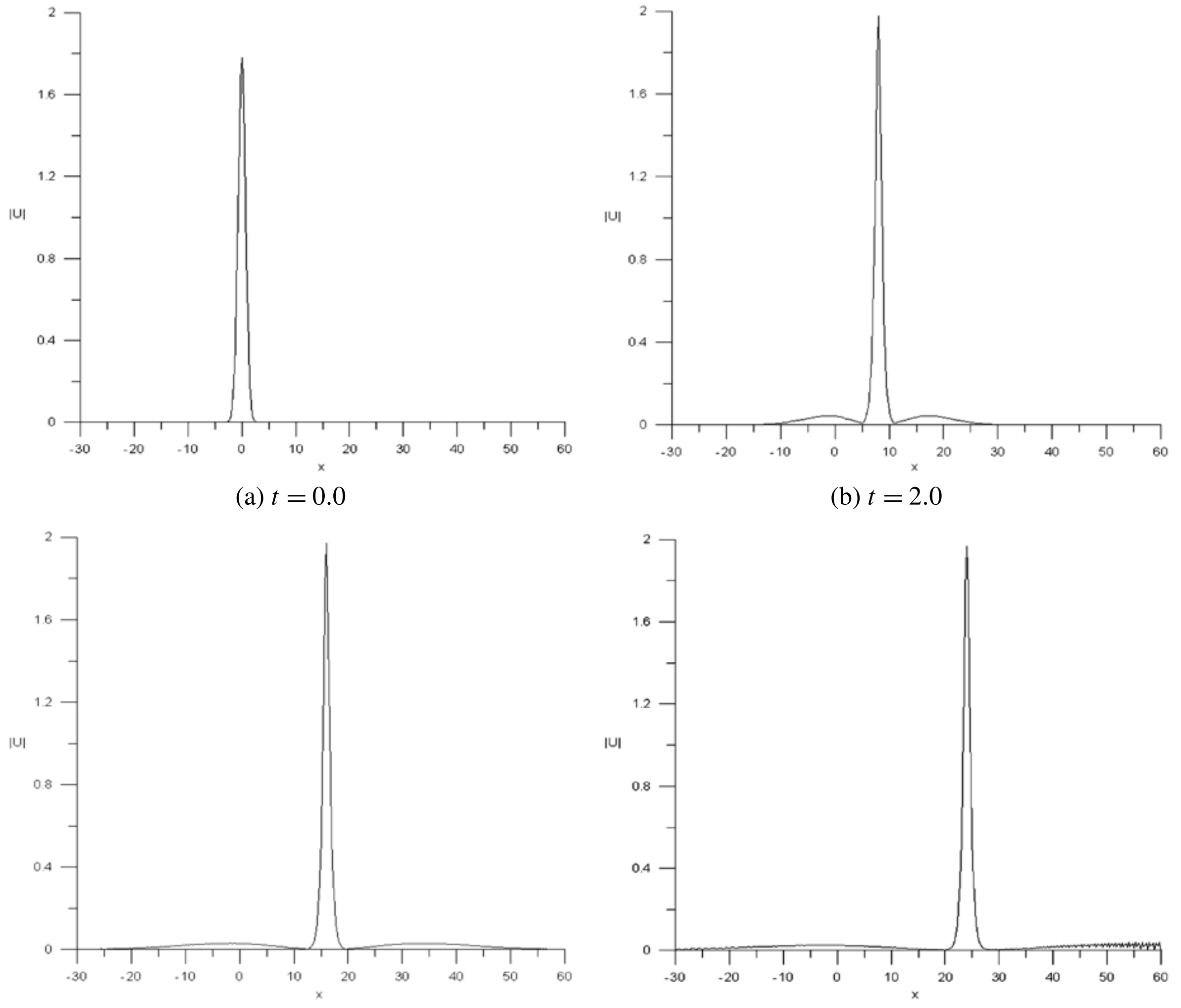
is chosen with parameters $q = 2, h = 0.25, 0.20, \Delta t = 0.01, 0.005, \alpha_1 = 1.0, S_1 = 4.0, x_1 = -10, \alpha_2 = 1.0, S_2 = -4.0, x_2 = 10$ over the region $-20 \le x \le 20$. This initial condition defines two solitons of equal magnitudes which are separated by a distant of 20 units; first one placed at $x = -10$ and second one at $x = 10$. Both of them has the velocity of 4. The simulation of interaction between two solitons is shown in Fig. 5. They are in good qualitative agreement with the behavior predicted both in theory and other numerical methods [31–34]. It is observed that two waves, traveling in opposite direction, collide and separate, but conserve initial shapes afterwards.

Conserved quantities and relative changes of conserved quantities at various times and comparison of conserved quantities and their relative changes with some earlier works are seen in Tables 5 and 6, respectively.

Test 3. Birth of standing soliton

Initial condition $U(x,0)$ with the property

$$\int_{\infty}^{\infty} U(x,0)\, dx \ge \pi$$



(a) $t = 0.0$

(b) $t = 2.0$

(c) $t = 4.0$

(d) $t = 6.0$

Fig. 12 Birth of mobile soliton of amplitude $A = 1.78$

generates a soliton, otherwise, soliton decays away. Gardner et al. [31] have studied this simulation for the following Maxwellian initial condition

$$U(x, 0) = A\exp\bigl(-x^2\bigr). \tag{37}$$

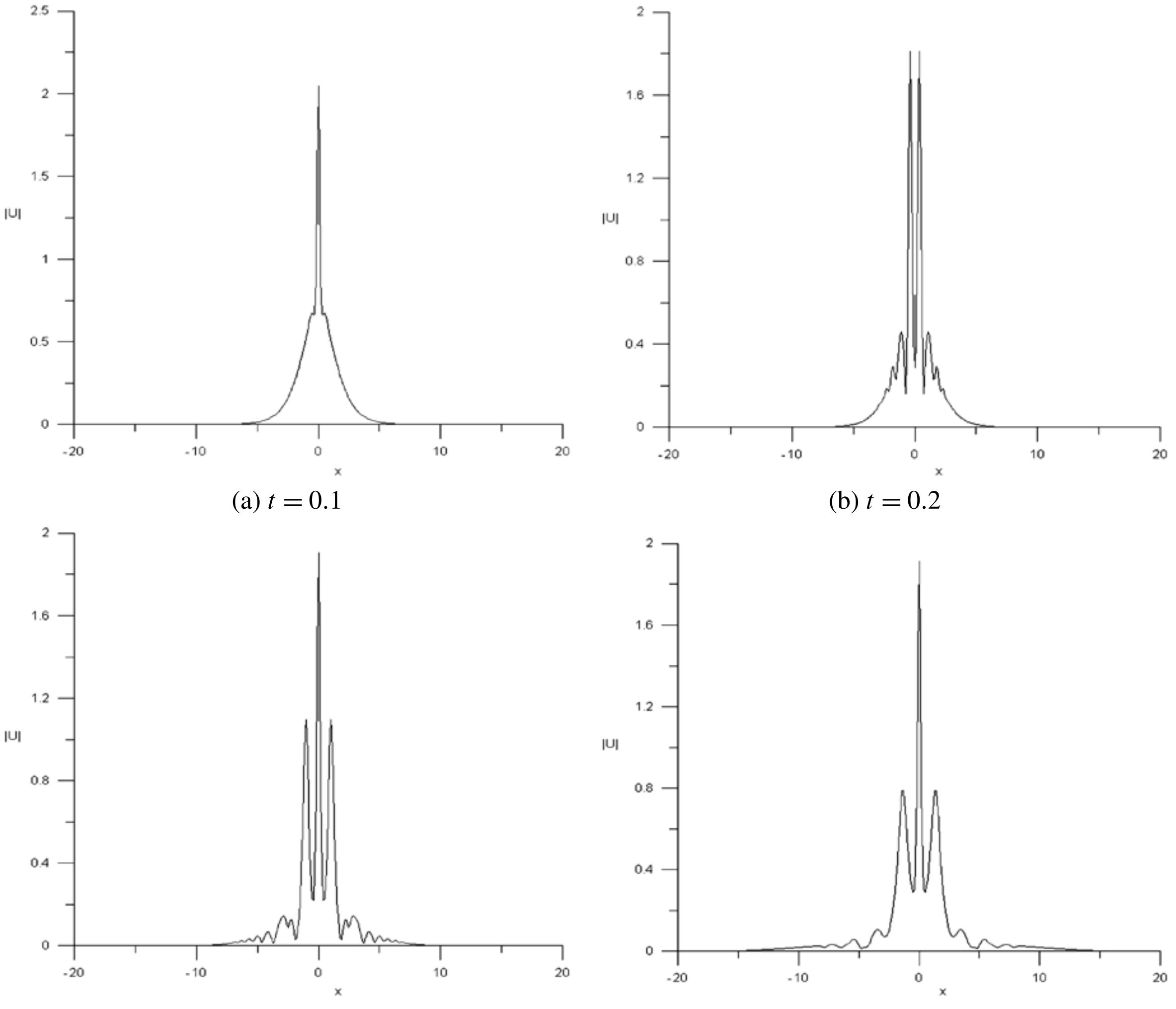
Maxwellian initial condition with parameters $h = 0.25$, $\Delta t = 0.01$, and $q = 2$ for a region $-45 \leq x \leq 45$ is used to exhibit the birth of soliton. Numerical experiment, run up to time $t = 6$, is observed in Fig. 6 for $A = 1$ and $A = 1.78$. It is seen from maxima of solution in the Fig. 6 that initial pulse fades out when $A = 1$, whereas a soliton of magnitude about 2 is observed for $A = 1.78$. This latter is simulated at different times in Figs. 7, 8, 9, 10, 11. In this case, we verify the analytical results obtained in [30], where solitons are produced only if $A = 1.78 > \sqrt{\pi} = 1.7725$. With the choice A of the less $\sqrt{\pi}$, initial soliton decays away.

Invariants C_1 and C_2 and their relative changes, whose analytic values can be found with initial condition (37) as, $C_1 = A^2\sqrt{\frac{\pi}{2}}$ and $C_2 = \frac{1}{4}A^2(2\sqrt{2} - qA^2)\sqrt{\pi}$ are recorded in Table 7. Observed invariants are conserved satisfactorily well.

Test 4. Birth of mobile soliton

In a second simulation involving the initial condition

$$U(x, 0) = A\exp\bigl(-x^2 + 2ix\bigr)$$



(a) $t = 0.1$

(b) $t = 0.2$

(c) $t = 0.4$

(d) $t = 0.6$

Fig. 13 $\lambda = 4$, $q = 32$, $N = 400$

using the parameters $h = 0.25$, $\Delta t = 0.01$ over the domain $[-30, 60]$ for the Maxwellian initial condition is used to obtain numerical results for $A = 1, 1.78$. The designed program, which simulates birth of mobile soliton of height 2 having velocity 4, is run up to time $t = 6.0$. The evaluation of the produced solution is graphed in Fig. 12 and conserved quantities and their relative errors are presented in Table 8. Both invariants for this simulation are conserved reasonably well.

Test 5. Bound state of solitons

Initial condition

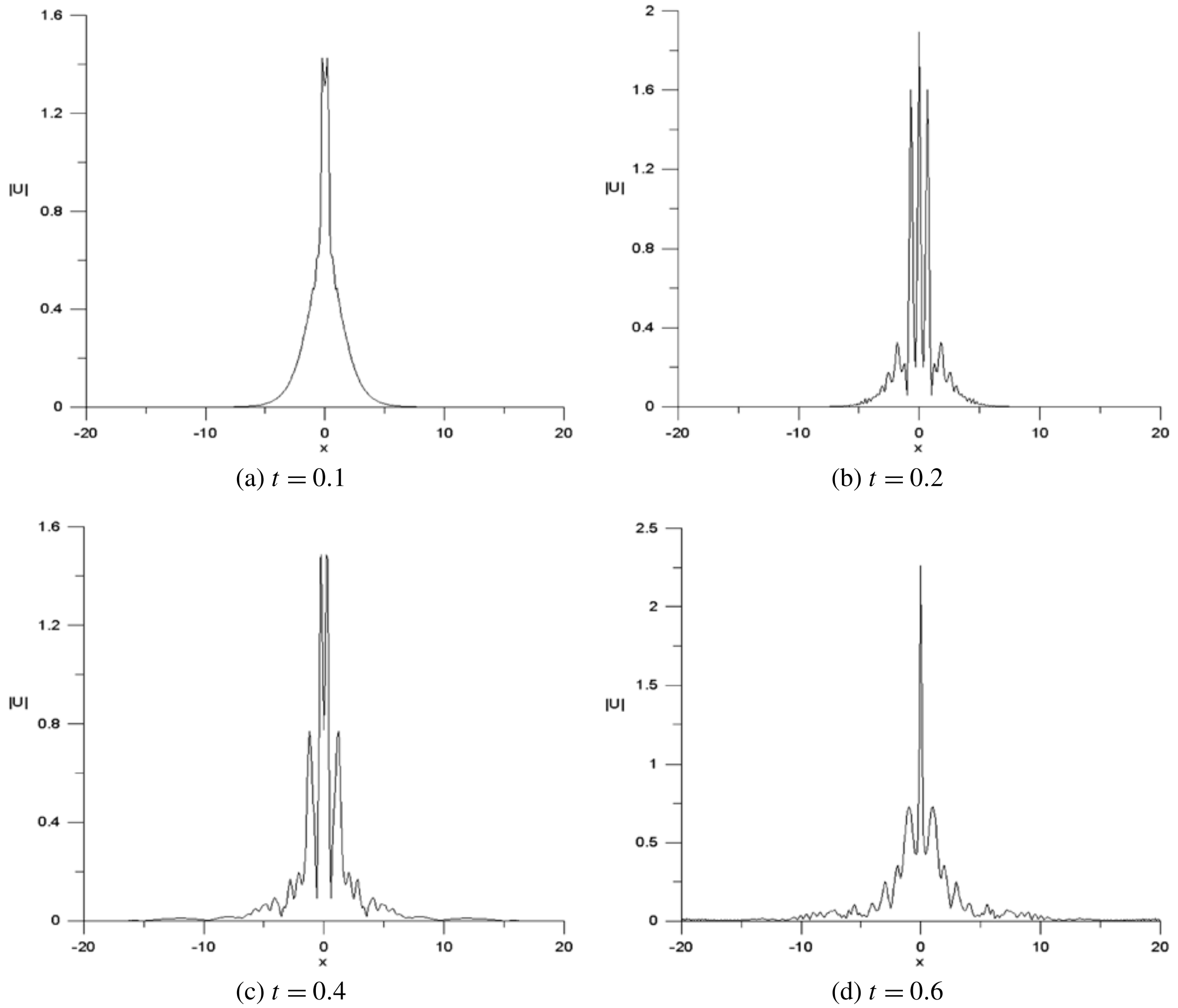
$$U(x, 0) = \operatorname{sech}(x) \qquad (38)$$

produces a bound state of λ solutions if

$$q = 2\lambda^2, \quad \lambda = 1, 2, \ldots.$$

The bound state of solitons of NLSE is given by [29]. However, the solution is not usable form if $\lambda \geq 3$. The numerical approximation of the bound state of solitons with the initial condition taking $q = 2, 8, 18$ has already been handled extensively and a comparison of the various methods has been discussed in earlier studies [28, 31, 32]. Conserved quantities with initial condition (38) can be computed analytically by

$$C_1 = 2, \qquad C_2 = \frac{2}{3}(1 - q).$$



(a) $t = 0.1$

(b) $t = 0.2$

(c) $t = 0.4$

(d) $t = 0.6$

Fig. 14 $\lambda = 5, q = 50, N = 400$

Parameters $h = 0.1$, $\Delta t = 0.01$ and $q = 32$, 50 over a region $-20 \leq x \leq 20$ are used for the sake of the comparison with study given by Gardner et al. [31, 32]. When $q = 32$, graphs are depicted in Fig. 13 at early times of simulation in which shapes of 4 bound solitons are in complete agreement with that of the paper [31]. The graph of modules of the numerical solution from the discrete set of data is also produced successfully for the $q = 50$ shown in Fig. 14.

Both invariants for both cases remain almost constant and reflect the satisfactory as illustrated in Table 9.

5 Conclusion

The aim of this paper is to test the applicability and efficiency of polynomials based differential quadrature method on nonlinear partial differential equation systems. We have discretized the NLSE using differential quadrature method in space and used fourth-order Runge–Kutta for the time integration of the resulting nonlinear ordinary differential equation system.

The first two test problems, the motion of single soliton along x-axis and interaction of solitons, are simulated with high accuracy even larger step sizes are used when compared with earlier studies. In addition, the performance of the method has been monitored by

Table 9 Boundary state solitons, $h = 0.1$, $\Delta t = 0.00$, $N = 400$

t	C_1	C_2	$\frac{C_1 - C_1^0}{C_1^0}$	$\frac{C_2 - C_2^0}{C_2^0}$
$q = 32$				
0	2.0000000000	−20.6666666664	–	–
1	2.0000017450	−20.6649294001	$8.7250556490 \times 10^{-7}$	$-8.4061271579 \times 10^{-5}$
2	2.0000036071	−20.6673625984	$1.8035919027 \times 10^{-6}$	$3.3674129069 \times 10^{-5}$
3	2.0000159310	−20.6654047853	$7.9655246194 \times 10^{-6}$	$-6.1058762810 \times 10^{-5}$
4	2.0000053412	−20.6673590062	$2.6706046216 \times 10^{-6}$	$3.3500313892 \times 10^{-5}$
5	1.9999997650	−20.6677264440	$-1.1748501993 \times 10^{-7}$	$5.1279566040 \times 10^{-5}$
$q = 50$				
0	2.0000000000	−32.6666666664	–	–
1	1.9998929254	−32.6649489885	$-5.3537263038 \times 10^{-5}$	$-5.2581975936 \times 10^{-5}$
2	1.9998903341	−32.8464649266	$-5.4832916072 \times 10^{-5}$	$5.5040283747 \times 10^{-3}$
3	1.9998229734	−32.9498388775	$-8.8513298316 \times 10^{-5}$	$8.6685370767 \times 10^{-3}$
4	1.9998837493	−32.9607896569	$-5.8125300219 \times 10^{-5}$	$9.0037650169 \times 10^{-3}$
5	1.9999552964	−33.0946218169	$-2.2351784112 \times 10^{-5}$	$1.3100667874 \times 10^{-2}$
Analytic	2.0000000000	−32.6666666664		

computing both the conserved quantities and the computational costs. We have found that the numerical results are in good agreement with the exact solutions for both the first experiment. Collision of solitons is investigated numerically and it is observed that solitons are interchanged after the collision.

As the third and the fourth experiments, birth of standing and mobile solitons are studied for two different amplitude values. Obtained results are very good agreement in theoretical ones. Reported conserved quantities and their relative changes show that the scheme is satisfactorily well.

As the last test example, the bound state of solitons are simulated successfully for two different parameters. Recorded conserved quantities are in good agreement with analytical ones, and the accuracy of the method is considerably well.

The numerical experiments reported in the paper show that the polynomial differential quadrature together with Runge–Kutta of order four algorithm gives very accurate results when compared with earlier results. Besides, the application of the method is easier than other methods such as finite element and spectral methods, but the cost of computation can be discussed when compared with finite element and finite different methods since the resulting algebraic system after discretization in space consists of N components in opposition to finite element or finite difference methods whose resulting systems include only several components.

References

1. Karpman, V.I., Krushkal, E.M.: Modulated waves in nonlinear dispersive media. Sov. Phys. JETP **28**, 277 (1969)
2. Scott, A.C., Chu, F.Y.F., Mclaughlin, D.W.: The soliton: a new concept in applied science. Proc. IEEE **61**, 1443 (1973)
3. Zakharov, V.E., Shabat, A.B.: Exact theory of two dimensional self focusing and one dimensional self waves in nonlinear media. Sov. Phys. JETP **34**, 62 (1972)
4. Delfour, M., Fortin, M., Payne, G.: Finite-difference solutions of a non-linear Schrodinger equation. J. Comput. Phys. **44**, 277–288 (1981)
5. Taha, T.R., Ablowitz, M.J.: Analytical and numerical aspects of certain nonlinear evolution equations, II: numerical, nonlinear Schrodinger equations. J. Comput. Phys. **55**, 203–230 (1984)
6. Argyris, J., Haase, M.: An engineer's guide to soliton phenomena: application of the finite element method. Comput. Methods Appl. Mech. Eng. **61**, 71–122 (1987)
7. Twizell, E.H., Bratsos, A.G., Newby, J.C.: A finite-difference method for solving the cubic Schrodinger equation. Math. Comput. Simul. **43**, 67–75 (1997)
8. Dağ, İ., A quartic B-spline finite element method for solving nonlinear Schrödinger equation. Comput. Methods Appl. Mech. Eng. **174**, 247–258 (1999)
9. Chen, H., Shizgal, B.D.: The quadrature discretization method in the solution of the Schrödinger equation. J. Chem. **24**(4), 321–343 (1998)

10. Shizgal, B.D., Chen, H.: The quadrature discretization method in the solution of the Schrödinger equation with nonclassical basis functions. J. Chem. Phys. **104**(11), 4137–4150 (1996)
11. Leung, K., Shizgal, B.D., Chen, H.: The quadrature discretization method in comparison with other numerical methods of solution of the Fokker–Planck for electron thermalization. J. Math. Chem. **24**(4), 291–319 (1998)
12. Lo, J., Shizgal, B.D.: Spectral convergence of the quadrature discretization method in the solution of the Schrödinger and Fokker–Planck equations: comparison with Sinc methods. J. Chem. Phys. **125**(19), 194108 (2006)
13. Bellman, R., Kashef, B.G., Casti, J.: Differential quadrature: a technique for the rapid solution of nonlinear differential equations. J. Comput. Phys. **10**, 40–52 (1972)
14. Quan, J.R., Chang, C.T.: New sightings in involving distributed system equations by the quadrature methods, I. Comput. Chem. Eng. **13**, 779–788 (1989)
15. Quan, J.R., Chang, C.T.: New sightings in involving distributed system equations by the quadrature methods, II. Comput. Chem. Eng. **13**, 71017–71024 (1989)
16. Bellman, R., Kashef, B., Lee, E.S., Vasudevan, R.: Differential quadrature and splines. In: Computers and Mathematics with Applications, pp. 371–376. Pergamon, Oxford (1976)
17. Shu, C., Richards, B.E.: Application of generalized differential quadrature to solve two dimensional incompressible Navier–Stokes equations. Int. J. Numer. Methods Fluids **15**, 791–798 (1992)
18. Shu, C., Wu, Y.L.: Integrated radial basis functions-based differential quadrature method and its performance. Int. J. Numer. Methods Fluids **53**, 969–984 (2007)
19. Shu, C., Xue, H.: Explicit computation of weighting coefficients in the harmonic differential quadrature. J. Sound Vib. **204**(3), 549–555 (1997)
20. Striz, A.G., Wang, X., Bert, C.W.: Harmonic differential quadrature method and applications to analysis of structural components. Acta Mech. **111**, 85–94 (1995)
21. Civalek, Ö.: Application of differential quadrature (DQ) and harmonic differential quadrature (HDQ) for buckling analysis of thin isotropic plates and elastic columns. Eng. Struct. Int. J. **26**(2), 171–186 (2004)
22. Civalek, Ö: Harmonic differential quadrature-finite differences coupled approaches for geometrically nonlinear static and dynamic analysis of rectangular plates on elastic foundation. J. Sound Vib. **294**, 966–980 (2006)
23. Malekzadeh, P., Karami, G.: Polynomial and harmonic differential quadrature methods for free vibration of variable thickness thick skew plates. Eng. Struct. **27**, 1563–1574 (2005)
24. Zhu, Y.D., Shu, C., Qiu, J., Tani, J.: Numerical simulation of natural convection between two elliptical cylinders using DQ method. Int. J. Heat Mass Transf. **47**, 797–808 (2004)
25. Lee, T.S., Hu, G.S., Shu, C.: Application of GDQ method for study of mixed convection in horizontal eccentric annuli. Int. J. Comput. Fluid Dyn. **18**(1), 71–79 (2004)
26. Zhong, H.: Spline-based differential quadrature for fourth order differential equations and its application to Kirchhoff plates. Appl. Math. Model. **28**, 353–366 (2004)
27. Whitham, G.B.: Linear and Nonlinear Waves. Wiley/Interscience, New York (1974)
28. Herbst, B.M., Morris, J.L., Mitchel, A.R.: Numerical experience with the nonlinear Schrödinger equation. J. Comput. Phys. **60**, 282–305 (1985)
29. Miles, J.W.: An envelope soliton problems. SIAM J. Appl. Math. **41**, 227–230 (1981)
30. Fornberg, B., Whitham, G.B.: A numerical and theoretical study of certain nonlinear wave phenomena. Philos. Trans. R. Soc. Lond. **289**, 373–404 (1978)
31. Gardner, L.R.T., Gardner, G.A., Zaki, S.I., Sharawi, Z.E.: B-spline finite element studies of the non-linear Schrodinger equation. Comput. Methods Appl. Mech. Eng. **108**, 303–318 (1993)
32. Gardner, L.R.T., Gardner, G.A., Zaki, S.I., Sharawi, Z.E.: A Leapfrog algorithm and stability studies for the nonlinear Schrodinger equation. Arab. J. Sci. Eng., 23–32 (1993)
33. Zacharov, V.E., Shabat, A.B.: Exact theory of two dimensional self-focusing and one dimensional self-modulation of waves in nonlinear media. Sov. Phys. JETP **34**, 62–69 (1972)
34. Tourigny, Y., Morris, J.L.: An investigation into effect of product approximation in the numerical solution of the cubic nonlinear Schrödinger equation. J. Comput. Phys. **76**, 103–130 (1988)