fluids

MDPI

Article

Soliton Solution of Schrödinger Equation Using Cubic B-Spline Galerkin Method

Azhar Iqbal [1,2,*], Nur Nadiah Abd Hamid [1] and Ahmad Izani Md. Ismail [1]

[1] School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia; nurnadiah@usm.my (N.N.A.H.); ahmad_izani@usm.my (A.I.M.I.)

[2] Mathematics and Natural Sciences, Prince Mohammad Bin Fahd University, Al Khobar 31952, Saudi Arabia

* Correspondence: azhariqbal31@gmail.com

Received: 17 April 2019; Accepted: 7 June 2019; Published: 12 June 2019

Abstract: The non-linear Schrödinger (NLS) equation has often been used as a model equation in the study of quantum states of physical systems. Numerical solution of NLS equation is obtained using cubic B-spline Galerkin method. We have applied the Crank–Nicolson scheme for time discretization and the cubic B-spline basis function for space discretization. Three numerical problems, including single soliton, interaction of two solitons and birth of standing soliton, are demonstrated to evaluate to the performance and accuracy of the method. The error norms and conservation laws are determined and found to be in good agreement with the published results. The obtained results show that the approach is feasible and accurate. The proposed method has almost second order convergence. The linear stability of the method is performed using the Von Neumann method.

Keywords: non-linear Schrödinger equation; cubic B-spline basis functions; Galerkin method

1. Introduction

The non-linear Schrödinger (NLS) equation describes how the behavior of quantum states of a physical system changes in time and space. The NLS equation can be used to describe the propagation of optical pulses and waves in water and plasmas, among other things. Due to the presence of nonlinearity and the complex nature of the problem, it is still a challenge for researchers to determine the most suitable method. Many theoretical and numerical studies have been carried out to overcome this difficulty. The B-spline Galerkin method is quite advanced and has been used by researchers to solve other complex problems.

Dag [1] presented the quadratic and cubic B-spline Galerkin finite element method for solving the Burger's equation. This method was found to give satisfactory results for the Burger's equation, particularly when continuity of the solutions was essential. Dag et al. [2] also proposed a cubic B-splines collocation method for solving the one-dimensional Burger's equation. The proposed scheme was easy to implement and did not require any inner iteration to deal with the nonlinear term of the Burger's equation. Gorgulu et al. [3] presented exponential B-splines Galerkin finite element method for the numerical solution of the advection-diffusion equation. In this study, a new algorithm was developed for the numerical solution of differential equations. This algorithm was obtained by utilizing exponential B-spline functions for the Galerkin finite element method. It was reported that the proposed method gave satisfactory results. The exponential B-splines Galerkin method for the numerical solution of the Burger's equation was also applied by Gorgulu et al. [4].

Saka et al. [5] proposed a quartic B-splines Galerkin finite element method for solving the regularized long wave equation. The performance and accuracy of the method were observed. The method has a weakness because of its large number of matrix operations. The quartic B-spline method was found to be very useful for finding the numerical solutions of the differential equation when higher continuity of solutions occurs. Aksan [6] presented the quadratic B-spline finite element method for finding the approximate solution of the nonlinear Burger's equation. The high accuracy of the proposed method was thoroughly examined. Gardner et al. [7] also applied a cubic B-spline finite element method for the numerical solution of the Burger's equation. They noticed that the proposed finite element method gave more accurate results than other approaches.

Sheng et al. [8] solved the NLS equation using a finite difference method based on quartic spline approximation and semidiscretization. The NLS equation was approximated using a finite element method by Zlotnik and Zlotnik [9]. Ersoy et al. [10] studied the numerical solution of the NLS equation using an exponential B-spline with collocation method. In that paper, they used the Crank Nicolson formulation for time integration and exponential cubic B-spline functions for space integration. Mokhtari et al. [11] solved the NLS equation using delta-shaped basis functions. Saka [12] solved the NLS using the collocation method with quintic B-spline functions. The generating function of the Hermite polynomial and the orthogonal Hermite function were introduced with ordinary Hermite polynomials by Cesarano et al. [13]. The nature of Hermite polynomials is described in the Schrödinger Equation. Analytical solutions of the NLS equation for certain initial and boundary conditions were presented by Zakharov and Shabat [14]. The numerical solution of the NLS equation was obtained by Lin [15] using a septic spline function method with the collocation method. The Galerkin finite element method for the NLS equation was applied by Dag, with quadratic B-spline functions as the weight and trial functions over finite elements [16]. The numerical solution of the NLS equation was also obtained by using the collocation method based on cubic B-spline by Gardner et al. [17]. Hu [18] presented a conservative two-grid finite element method for the numerical solution of the NLS equation and applied one Newton iteration for the linearization of the problem by using the coarse-grid solution as the initial guess. The Galerkin B-spline method is not often used for the numerical solution of the NLS equation. A review of some numerical methods for solving the NLS equation indicates that most methods use the collocation finite element method.

In this paper, we study the use of the Galerkin method with cubic B-spline function as the weight and trial functions over finite elements to solve the NLS equation. This paper is organized as follows. In Section 2, we discussed the governing equation and fundamentals of Cubic B-spline Galerkin method. In Section 3, the stability analysis of the numerical scheme is analyzed using the Von Neumann method. The numerical results and test problems are discussed in Section 4. In the last section, the conclusions are given.

2. Governing Equation and Cubic B-Spline Galerkin Method

The NLS equation is

$$iu_t + u_{xx} + q|u|^2 u = 0, \tag{1}$$

The equation in (1) is called a self-focusing NLS equation when $q > 0$ and a defocusing NLS equation when $q < 0$. The subscripts x and t denote the spatial variable and time variable respectively; q is a positive real parameter, and $i = \sqrt{-1}$. $u(x,t)$ is a complex-valued function and defined over the whole real line. The initial condition is

$$u(x,0) = f(x), \qquad x \in [a,b], \tag{2}$$

and the boundary conditions are as follows:

$$u(a,t) = u(b,t) = 0. \tag{3}$$

The solution $u(x,t)$ is decomposed into its real and imaginary parts as

$$u(x,t) = r(x,t) + is(x,t), \tag{4}$$

Substituting Equation (4) into Equation (1), the following coupled partial differential equations are obtained:

$$\begin{cases} s_t - r_{xx} - q\left(r^2 + s^2\right)r = 0, \\ r_t + s_{xx} + q\left(r^2 + s^2\right)s = 0. \end{cases} \tag{5}$$

The solution domain $[a, b]$ is divided equally into N finite elements by the nodes x_k such that $a = x_0 < x_1 < \cdots x_N = b$ and $h = \frac{b-a}{N} = x_{k+1} - x_k$. In this paper, we choose cubic B-spline function as the weight and trial function. The cubic B-splines $\varnothing_k(x)$, $(k = -1, \ldots, N+1)$ are defined over the interval $[a, b]$ as follows [1]:

$$\varnothing_k(x) \\ = \frac{1}{h^3}\begin{cases} (x - x_{k-2})^3 & x \in [x_{k-2}, x_{k-1}], \\ h^3 + 3h^2(x - x_{k-1}) + 3h(x - x_{k-1})^2 - 3(x - x_{k-1})^3 & x \in [x_{k-1}, x_k], \\ h^3 + 3h^2(x_{k+1} - x) + 3h(x_{k+1} - x)^2 - 3(x_{k+1} - x)^3 & x \in [x_k, x_{k+1}], \\ (x_{k+2} - x)^3 & x \in [x_{k+1}, x_{k+2}], \\ 0 & otherwise. \end{cases} \tag{6}$$

The approximate solution $u_N(x, t)$ to the analytical solution $u(x, t)$ for Equation (1) is written in terms of the cubic B-spline function as

$$u_N(x, t) = s_N(x, t) + r_N(x, t) = \sum_{j=-1}^{N+1} \delta_j(t)\varnothing_j(x), \tag{7}$$

where $\delta_j(t) = \rho_j(t) + \psi_j(t)$,

$$\begin{aligned} s_N(x, t) &= \sum_{j=-1}^{N+1} \rho_j(t)\varnothing_j(x), \\ r_N(x, t) &= \sum_{j=-1}^{N+1} \psi_j(t)\varnothing_j(x). \end{aligned} \tag{8}$$

The coefficients $\rho_j(t)$ and $\psi_j(t)$ are unknown time-dependent parameters that will be determined from the boundary and weighted residual conditions. Since each cubic B-spline covers 4 elements, each element $[x_k, x_{k+1}]$ is covered by 4 splines. Using the local coordinate's transformation $\eta = x - x_k$ for $0 \leq \eta \leq h$, the cubic B-spline function over the element $[x_k, x_{k+1}]$ can be written as

$$\begin{matrix} \varnothing_{k-1} \\ \varnothing_k \\ \varnothing_{k+1} \\ \varnothing_{k+2} \end{matrix} = \frac{1}{h^3}\begin{cases} (h - \eta)^3 \\ 4h^3 - 3h^2\eta + 3h(h - \eta)^2 - 3(h - \eta)^3 \\ h^3 + 3h^2\eta + 3h\eta^2 - 3\eta^3 \\ \eta^3 \end{cases}, \quad 0 \leq \eta \leq h. \tag{9}$$

Here, all other cubic B-spline functions are zero over the finite element $[x_k, x_{k+1}]$. The approximation function (Equation (8)) over the finite element can be defined in terms of the basis functions (Equation (9)) as

$$\begin{aligned} s_N(x, t) &= \sum_{j=k-1}^{k+2} \rho_j(t)\varnothing_j(x). \\ r_N(x, t) &= \sum_{j=k-1}^{k+2} \psi_j(t)\varnothing_j(x). \end{aligned} \tag{10}$$

Using the B-splines basis defined in Equation (6) and the trial function stated in Equation (8), the nodal values of s_k, r_k, s'_k and r'_k at the knots x_k are determined in terms of time-dependent element parameters ρ_j and ψ_j as:

$$\begin{aligned} s_k &= s(x_k) = \rho_{k-1} + 4\rho_k + \rho_{k+1}, \\ r_k &= r(x_k) = \psi_{k-1} + 4\psi_k + \psi_{k+1}, \\ hs'_k &= hs'(x_k) == 3(\rho_{k+1} - \rho_{k-1}), \\ hr'_k &= hr'(x_k) == 3(\psi_{k+1} - \psi_{k-1}), \\ h^2 s''_k &= h^2 s''(x_k) == 6(\rho_{k-1} - 2\rho_k + \rho_{k+1}), \\ h^2 r''_k &= h^2 r''(x_k) = 6(\ \psi_{k-1} - 2\psi_k + \psi_{k+1}), \end{aligned} \tag{11}$$

where the prime denotes differentiation with respect to x.

Applying the Galerkin approach to Equation (5) with the weight function $W(x)$, the weak form of Equation (5) over the finite element $[x_k, x_{k+1}]$ is

$$\begin{cases} \int\limits_{x_k}^{x_{k+1}} [Ws_t + W_x r_x - z_k Wr] dx = 0, \\ \int\limits_{x_k}^{x_{k+1}} [Wr_t - W_x s_x + z_k Ws] dx = 0, \end{cases} \tag{12}$$

where

$$z_k = q(r^2 + s^2). \tag{13}$$

Using the weight function W as the cubic B-spline and substituting Equations (10) into (12) with some manipulations, the following differential equations are obtained:

$$\begin{cases} \sum\limits_{j=k-1}^{k+2} \left[\left(\int\limits_0^h \varnothing_m \varnothing_j d\eta \right) \dot{\rho}_j + \left(\int\limits_0^h \varnothing'_m \varnothing'_j d\eta \right) \psi_j - \left(z_k \int\limits_0^h \varnothing_m \varnothing_j d\eta \right) \psi_j \right] = 0, \\ \sum\limits_{j=k-1}^{k+2} \left[\left(\int\limits_0^h \varnothing_m \varnothing_j d\eta \right) \dot{\psi}_j + \left(\int\limits_0^h \varnothing'_m \varnothing'_j d\eta \right) \rho_j - \left(z_k \int\limits_0^h \varnothing_m \varnothing_j d\eta \right) \rho_j \right] = 0, \end{cases} \tag{14}$$

Equation (14) can be written in matrix form as

$$\begin{cases} A^e \dot{\rho}^e + B^e \psi^e - z_k C^e \psi^e = 0, \\ A^e \dot{\psi}^e - B^e \rho^e + z_k C^e \rho^e = 0, \end{cases} \tag{15}$$

where $\rho^e = (\rho_{k-1}, \rho_k, \rho_{k+1}, \rho_{k+2})^T$ and $\psi^e = (\psi_{k-1}, \psi_k, \psi_{k+1}, \psi_{k+2})^T$ are the element parameters and dot $(\cdot)$ represents differentiation with respect to t. The element matrix A^e_{mj}, B^e_{mj} and C^e_{mj} are given by

$$A^e_{mj} = C^e_{mj} = \int\limits_0^h \varnothing_m \varnothing_{\mathrm{j}} d\eta, \quad B^e_{mj} = \int\limits_0^h \varnothing'_m \varnothing'_{\mathrm{j}} d\eta. \tag{16}$$

where $m, j = k-1, k, k+1, k+2$. The element matrix (16) is calculated as

$$A^e_{mj} = C^e_{mj} = \int_0^h \varnothing_m \varnothing_{\mathrm{j}} d\eta = \frac{h}{140}\begin{bmatrix} 20 & 129 & 60 & 1.0 \\ 129 & 1188 & 933 & 60 \\ 60 & 933 & 1188 & 129 \\ 1.0 & 60 & 129 & 20 \end{bmatrix},$$

$$B^e_{mj} = \int_0^h \varnothing'_m \varnothing'_{\mathrm{j}} d\eta = \frac{1}{10h}\begin{bmatrix} 18 & 21 & -36 & -3.0 \\ 21 & 102 & -87 & -36 \\ -36 & -87 & 102 & 21 \\ -3.0 & -36 & 21 & 18 \end{bmatrix}.$$

z_k values are calculated by writing $s = \frac{s_m + s_{m+1}}{2}$ and $r = \frac{r_m + r_{m+1}}{2}$ in Equation (13). Using the values of s_N and r_N at the points x_k, we obtain

$$z_k = q\left[\frac{(\rho_{k-1} + 5\rho_k + 5\rho_{k+1} + \rho_{k+2})^2}{4} + \frac{(\psi_{k-1} + 5\psi_k + 5\psi_{k+1} + \psi_{k+2})^2}{4}\right]. \quad (17)$$

Assembling the contributions from all of the elements, Equation (15) becomes

$$\begin{cases} A\dot{\rho} + B\psi - C(z_k)\psi = 0, \\ A\dot{\psi} - B\rho + C(z_k)\rho = 0, \end{cases} \quad (18)$$

where $\rho = (\rho_{-1}, \rho_0, \ldots, \rho_k, \rho_{k+1})^T$ and $\psi = (\psi_{-1}, \psi_0, \ldots, \psi_k, \psi_{k+1})^T$ are global element parameters and the matrices $A,\ B\ and\ C(z_k)$, which are global matrices with a generalized kth row, have the following form:

$$\begin{aligned} A &= \tfrac{h}{140}(1, 120, 1191, 2416, 1191, 120, 1), \\ B &= \tfrac{1}{10h}(-3, -72, -45, 240, -45, -72, -3), \\ C(z_k) &= \frac{h}{140}(z_{1k}, 60z_{1k} + 60z_{2k}, 129z_{1k} + 933z_{2k} + 129z_{3k},\ 20z_{1k} + 1188z_{2k} \\ &\quad + 1188z_{3k} + 20z_{4k}, 129z_{2k} + 933z_{3k} + 129z_{4k},\ 60z_{3k} + 60z_{4k}, z_{4k}). \end{aligned} \quad (19)$$

where

$$\begin{aligned} z_{1k} &= \tfrac{q}{4}\left[(\rho_{k-2} + 5\rho_{k-1} + 5\rho_k + \rho_{k+1})^2 + (\psi_{k-2} + 5\psi_{k-1} + 5\psi_k + \psi_{k+1})^2\right], \\ z_{2k} &= \tfrac{q}{4}\left[(\rho_{k-1} + 5\rho_k + 5\rho_{k+1} + \rho_{k+2})^2 + (\psi_{k-1} + 5\psi_k + 5\psi_{k+1} + \psi_{k+2})^2\right], \\ z_{3k} &= \tfrac{q}{4}\left[(\rho_k + 5\rho_{k+1} + 5\rho_{k+2} + \rho_{k+3})^2 + (\psi_k + 5\psi_{k+1} + 5\psi_{k+2} + \psi_{k+3})^2\right], \\ z_{4k} &= \tfrac{q}{4}\left[(\rho_{k+1} + 5\rho_{k+2} + 5\rho_{k+3} + \rho_{k+4})^2 + (\psi_{k+1} + 5\psi_{k+2} + 5\psi_{k+3} + \psi_{k+4})^2\right]. \end{aligned} \quad (20)$$

Discretization by the Crank-Nicolson method gives us $\rho = \left(\rho^n + \rho^{(n+1)}\right)/2$ and $\psi = \left(\psi^n + \psi^{n+1}\right)/2$. Similarly, the time discretization by finite difference scheme gives us $\dot{\rho} = \left(\rho^{n+1} - \rho^n\right)/\Delta t$ and $\dot{\psi} = (\psi^{n+1} - \psi^n)/\Delta t$. Substituting these discretizations into Equation (18), we obtain the following system:

$$\begin{cases} A\rho^{n+1} + 0.5\Delta t(B - C(z_k))\psi^{n+1} = A\rho^n - 0.5\Delta t(B - C(z_k))\psi^n, \\ A\psi^{n+1} - 0.5\Delta t(B - C(z_k)\ )\rho^{n+1} = A\psi^n + 0.5\Delta t(B - C(z_k))\rho^n. \end{cases} \quad (21)$$

Using the boundary conditions in Equation (21) yields a septadiagonal matrix. This system is solved by an appropriate method in MATLAB.

First, the initial vectors of parameters $\rho^0 = \left(\rho^0{}_0, \ldots, \rho^0{}_N\right)$ and $\psi^0 = \left(\psi^0{}_0, \ldots, \psi^0{}_N\right)$ are calculated to solve System (21) with the use of the initial condition using the relations

$$s_N(x,0) = \sum\nolimits_{j=-1}^{N+1} \varnothing_j(x)\rho_j^o \text{ and } r_N(x,0) = \sum\nolimits_{j=-1}^{N+1} \varnothing_j(x)\psi_j^o, \tag{22}$$

where all parameters ρ^0 and ψ^0 are determined. s_N and r_N require the following relations to be satisfied at points x_k:

$$\begin{gathered} s_N(x_k,0) = s(x_k,0), \\ r_N(x_k,0) = r(x_k,0), k = 0,1,\ldots,N. \end{gathered} \tag{23}$$

The initial vectors ρ^0 and ψ^0 can be calculated using the initial and boundary conditions from the following matrix equations:

$$\begin{bmatrix} 4 & 1 & 0 & 0 & & & 0 \\ 1 & 4 & 1 & 0 & & & \\ 0 & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & & & 0 & 1 & 4 & 1 \\ 0 & & & & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} \rho^0{}_0 \\ \rho^0{}_1 \\ \rho^0{}_2 \\ \vdots \\ \rho^0{}_{N-2} \\ \rho^0{}_{N-1} \end{bmatrix} = \begin{bmatrix} s^0{}_0 \\ s^0{}_1 \\ s^0{}_2 \\ \vdots \\ s^0{}_{N-3} \\ s^0{}_{N-2} \\ s^0{}_{N-1} \end{bmatrix},$$

and

$$\begin{bmatrix} 4 & 1 & 0 & 0 & & & 0 \\ 1 & 4 & 1 & 0 & & & \\ 0 & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & & & 0 & 1 & 4 & 1 \\ 0 & & & & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} \psi^0{}_0 \\ \psi^0{}_1 \\ \psi^0{}_2 \\ \vdots \\ \psi^0{}_{N-2} \\ \psi^0{}_{N-1} \end{bmatrix} = \begin{bmatrix} r^0{}_0 \\ r^0{}_1 \\ r^0{}_2 \\ \vdots \\ r^0{}_{N-3} \\ r^0{}_{N-2} \\ r^0{}_{N-1} \end{bmatrix}.$$

This system is solved using a suitable method in MATLAB. The approximate numerical solution of $s_N(x,t)$ and $r_N(x,t)$ is obtained from the ρ^n and ψ^n using Equation (21).

3. Stability Analysis

Stability analysis of the numerical scheme is carried out using the von Neumann method. The coupled system of Equation (5) is written as

$$\frac{\partial U}{\partial t} + M\frac{\partial^2 U}{\partial x^2} + G(U)U = 0, \tag{24}$$

where $U = \begin{bmatrix} s \\ r \end{bmatrix}, M = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ and $G(U) = \begin{bmatrix} 0 & -\mu_1 \\ \mu_2 & 0 \end{bmatrix}$.

To perform linear stability analysis, we further linearize the nonlinear term in Equation (24) by taking $G(U) = \mu M$. Here $\mu = \max[\mu_1, \mu_2]$. Therefore, the linearized form of Equation (24) is as follows:

$$\frac{\partial U}{\partial t} + M\frac{\partial^2 U}{\partial x^2} + \mu M U = 0. \tag{25}$$

The discretization of linear Equation (24) by the proposed scheme is given by

$$AU_m^{n+1} - MDU_m^{n+1} = AU_m^n + MDU_m^{n+1}\,, \tag{26}$$

where $D = B - \mu I$.

We use the following Fourier modes analysis for Scheme (25)

$$U_m^n = PG^n e^{imv}. \tag{27}$$

Here $i = \sqrt{-1}$, $P \in R^2$ and $G \in R^{2\times 2}$ is the amplification matrix. Equation (26) is substituted into Equation (25) to obtain the value of the amplification G. After tedious algebra, we obtain the value of G as follows

$$G = \left[\frac{A + MD}{A - MD}\right] \tag{28}$$

Matrix M is a skew symmetric matrix, and therefore both the matrices $A + MD$ and $A - MD$ have the same eigen values. Thus, the maximum value of $|G| = 1$. Hence, the linearized Scheme (25) is unconditionally stable.

4. Numerical Results and Test Problems

Three test problems, including the single soliton, interaction of two solitons and birth of standing soliton, are presented to evaluate the effectiveness and performance of the proposed method. The accuracy of the proposed method is examined using the error norms L_2andL_∞ and conservation laws defined as

$$\begin{aligned} L_2 &= \|u^{exact} - u_N\|_2 = \sqrt{h\sum_{k=0}^{N}\left|u_k{}^{exact} - (u_N)_k\right|^2}, \\ L_\infty &= \|u^{exact} - u_N\|_\infty = \max_{0\le k\le N}\left|u_k{}^{exact} - (u_N)_k\right|. \end{aligned} \tag{29}$$

Moreover, Equation (1) must satisfy the two conservation laws

$$\begin{aligned} I_1 &= \int_a^b |u|^2 dx, \\ I_2 &= \int_a^b \left(|u_x|^2 - \tfrac{1}{2}q|u|^4\right)dx. \end{aligned} \tag{30}$$

4.1. Single Soliton Solution to the NLS Equation

The analytical single soliton solution to the NLS equation is given as [2]:

$$u(x,t) = \beta\left(\sqrt{\frac{2}{q}}\right)expi\left[\frac{1}{2}Sx - \frac{1}{4}\left(S^2 - \beta^2\right)t\right]sech(\beta x - \beta St), \tag{31}$$

where S is the speed of the soliton solution whose magnitude depends on the parameter β.

The numerical solution is computed using the following parameters: $q = 2, S = 4, \beta = 1$, $x_L = -20$, and $x_R = 20$. The conservative quantities I_1 and I_2 and error norms L_∞ and L_2 are calculated. Numerical simulations were carried out at different space steps and time steps for comparison with the published results of the previous methods. The results are documented in Tables 1 and 2. The numerical simulations and the absolute numerical error are shown at different times in Figure 1.

Table 2 displays a comparison between the results obtained by the proposed method and published results. The results of the proposed method are in agreement with analytical solutions within very satisfactory limits, and the proposed method exhibits the same results as the quintic B-spline collocation method and quadratic B-spline method, as can be seen from the comparison in Table 2. We found good results even for large step sizes. It is noted that the error norms of the Galerkin cubic B-spline are lower than Explicit method [3], Implicit/explicit [3], and Split-step Fourier method [3] when we compare the results of the proposed method with those referenced in [1,4,19].

Table 1. Error norms and conservation laws for single soliton: with $h = 0.04, q = 2, S = 4, \beta = 1$.

	Δt = 0.0025				Δt = 0.005			
t	L_∞	L_2	I_1	I_2	L_∞	L_2	I_1	I_2
0.5	0.0000896	0.0000914	2.0	7.33333332	0.00018	0.00018	2.0	7.33333327
1	0.0000896	0.0000914	2.0	7.33333332	0.00018	0.00018	2.0	7.33333327
1.5	0.0000896	0.0000914	2.0	7.33333332	0.00018	0.00018	2.0	7.33333327
2	0.0000896	0.0000914	2.0	7.33333331	0.00018	0.00018	2.0	7.33333327
2.5	0.0000896	0.0000914	2.0	7.33333330	0.00018	0.00018	2.0	7.33333325
3	0.0000896	0.0000914	2.0	7.33333223	0.00018	0.00018	2.0	7.33333219
3.5	0.0000896	0.0000914	1.99999	7.33327430	0.00018	0.00018	1.99999	7.33327425

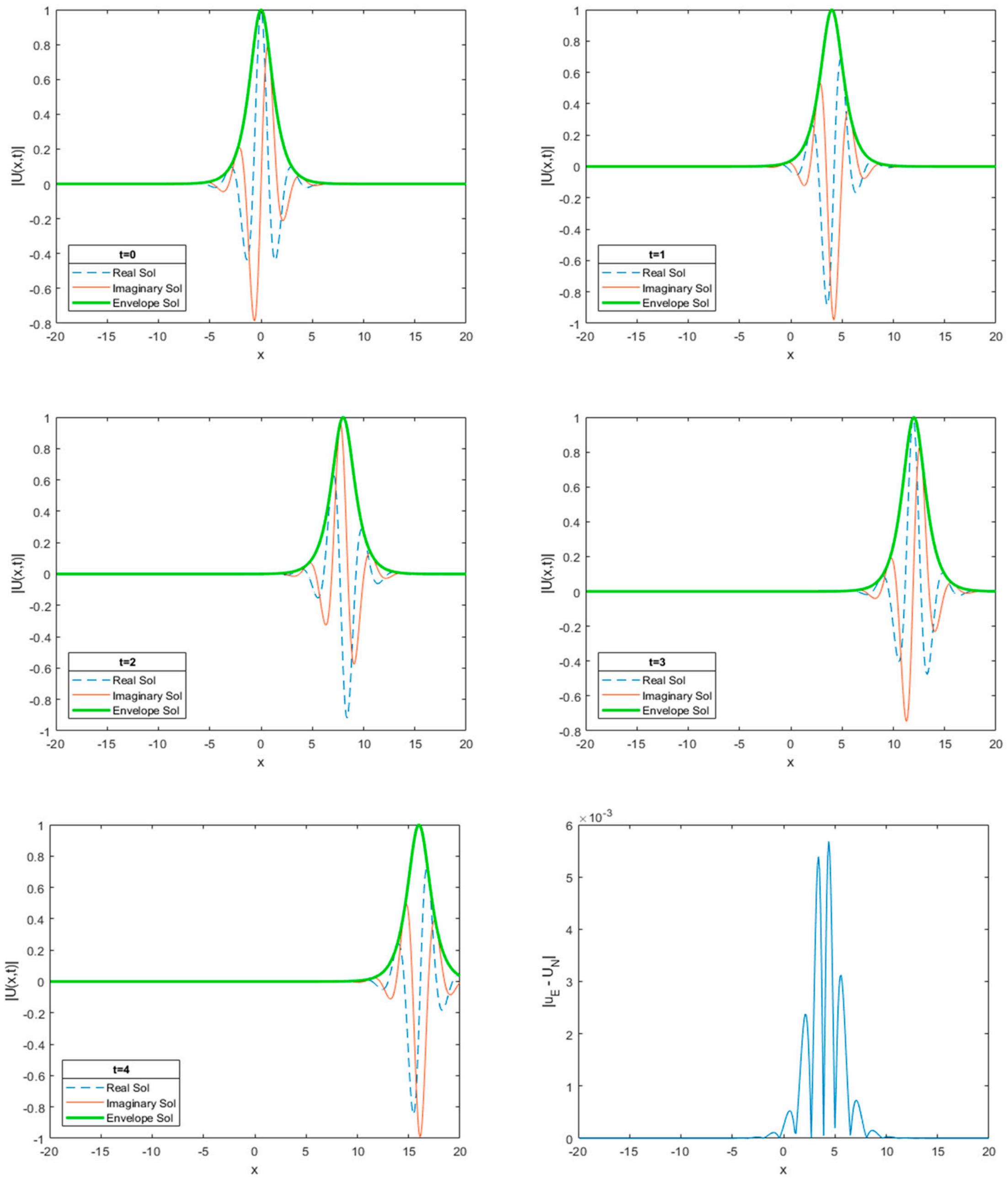

Figure 1. Single soliton solution and errors |Numerical – Analytical|.

Table 2. Comparisons of the present results with those of Taha et al. [20]., amplitude = 1 at time = 1.

h	Δt	L_∞	L_2	I_1	I_2	L_∞ [20]
0.05	0.0025	0.0001086	0.0001106	2.0	7.33333330	0.008
0.05	0.000625	0.0000272	0.0000277	2.0	7.33333333	0.006
0.05	0.001	0.0000435	0.0000443	2.0	7.33333333	0.006
0.08	0.002	0.0000628	0.0000639	2.0	7.33333330	0.005
0.3125	0.00026	0.0003174	0.0004973	2.00001	7.33332035	0.005
0.3125	0.020	0.0024417	0.0038252	2.00010	7.33320599	0.005
0.05	0.0005	0.0000217	0.0000221	2.0	7.33333333	0.008
0.3125	0.0026	0.0003174	0.0004973	2.00001	7.33332035	0.006

The rate of convergence for spatial and temporal directions is calculated using the formula [21]:

$$\text{rate of convergence} \approx log_2 \frac{error(2h, 2\Delta t)}{error(h, \Delta t)},$$

where $error(2h, 2\Delta t)$ is the error norms L_∞ and L_2 in spatial and temporal directions. The error norms L_∞, L_2 and order of convergence rate at time $t = 1$ are shown in Table 3.

Table 3. Rate of convergence in spatial and temporal directions at $t = 1$ with $h = \Delta t$.

h	L_∞	Order	L_2	Order
0.050000	0.0020492	–	0.0019146	-
0.025000	0.0005730	1.83	0.0005302	1.85
0.012500	0.0001507	1.93	0.000139	1.93
0.001625	0.0000386	1.97	0.0000356	1.96
0.003125	0.00000965	2.00	0.0000089	2.00

In Table 4, the results are documented for when the soliton amplitude is equal to 2. It was found that the results of the error norms at different times increase very slightly, as shown in Table 4. One soliton solution with amplitude $= 2$ is calculated, and simulations are shown in Figure 2.

Table 4. Error norms and conservation laws for single soliton with $h = 0.05, \Delta t = 0.0025$ $h = 0.05$, $q = 2$,$S = 4, \beta = 2$.

t	L_2	L_∞
0.5	0.0005580	0.0007595
1	0.0005580	0.0007595
1.5	0.0005580	0.0007595
2	0.0005580	0.0007595
2.5	0.0005580	0.0007595
3	0.0005580	0.0007595
3.5	0.0005580	0.0007595
4	0.0005580	0.0007595

Table 5 displays the approximate results obtained using the proposed method compared with the published methods listed in the reference for single soliton when the amplitude is 2. These results are in good agreement with the analytical solution.

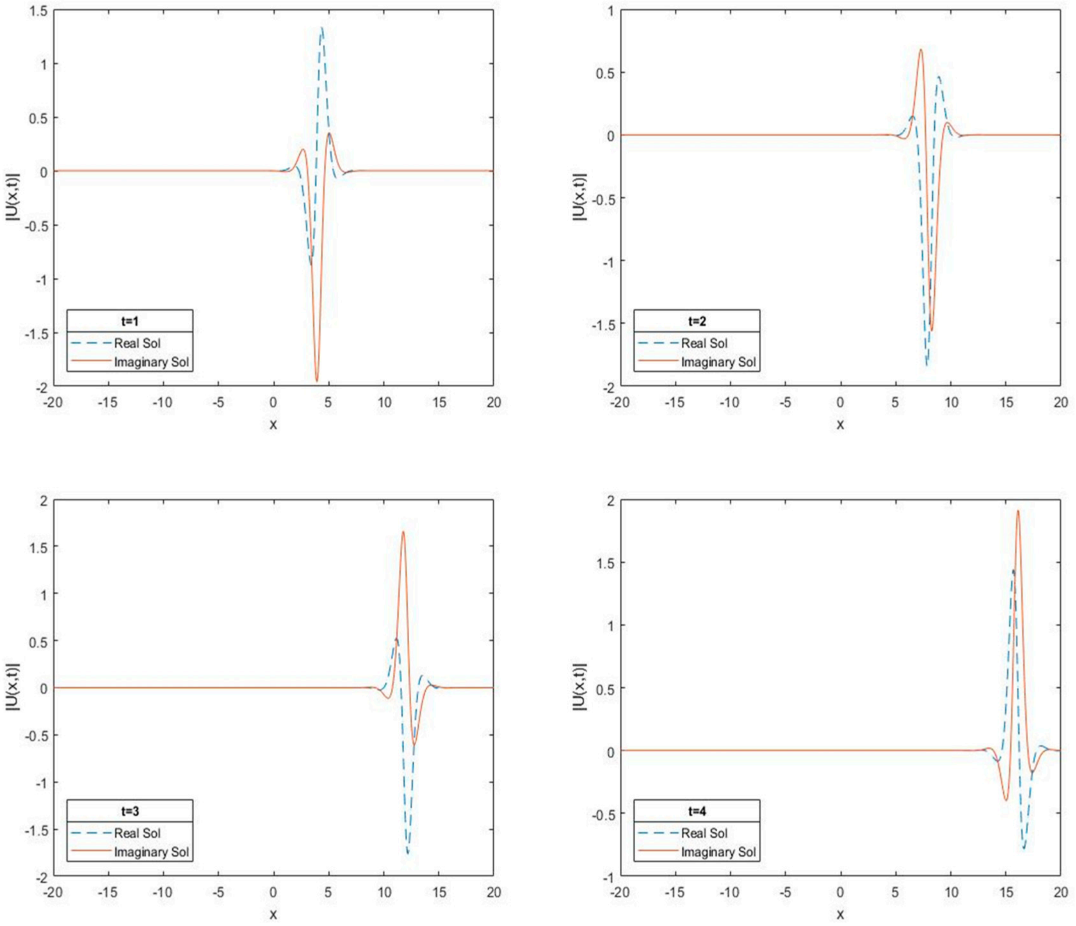

Figure 2. Single soliton solution with amplitude 2.

Table 5. Comparisons of single soliton with those of Taha et al. [20], amplitude = 2 at time = 1.

h	Δt	L_∞	L_2	L_∞ [20]
0.02	0.0025	0.0003909	0.0002859	0.0011
0.066	0.002	0.0009658	0.0007168	0.006
0.02	0.0001	0.0000156	0.0000114	0.009
0.03	0.00022	0.0000143	0.0000105	0.008
0.1	0.0025	0.0016852	0.0012732	0.0004
0.1563	0.0011	0.0010143	0.0007852	0.008

4.2. The Interaction of Two Solitons for the NLS Equation

In the second problem, we discuss the interaction of two solitons moving in opposite directions, have the same amplitude of magnitude 1 with initial conditions as follows [16,17,20]

$$\begin{gathered} u(x,0) = \sum_{k=1}^{2} u_k(x,0), \\ u_k(x,0) = \beta_k\left(\sqrt{\tfrac{2}{q}}\right) expi\left[\tfrac{1}{2}S(x-x_k)\right] sech(\beta_k(x-x_k)). \end{gathered} \tag{32}$$

where β_k, q and x_k are constants.

The numerical solution is computed by using the following parameters $x_1 = 10, x_2 = -10$, $q = 2, \beta_1 = \beta_2 = 1, S_1 = -4$ and $S_2 = 4$. The first soliton is placed at $x = 10$ and is moving to the left with speed 4, and the second soliton is placed on the other side at $x = -10$, traveling to the right with speed 4. Both solitons are moving in opposite directions, and they collide and separate. The interactions of these two solitons are visualized in Figure 3. The error norms L_∞ and L_2 are computed at different times and at different space steps and time steps. The calculated results are tabulated in Table 6.

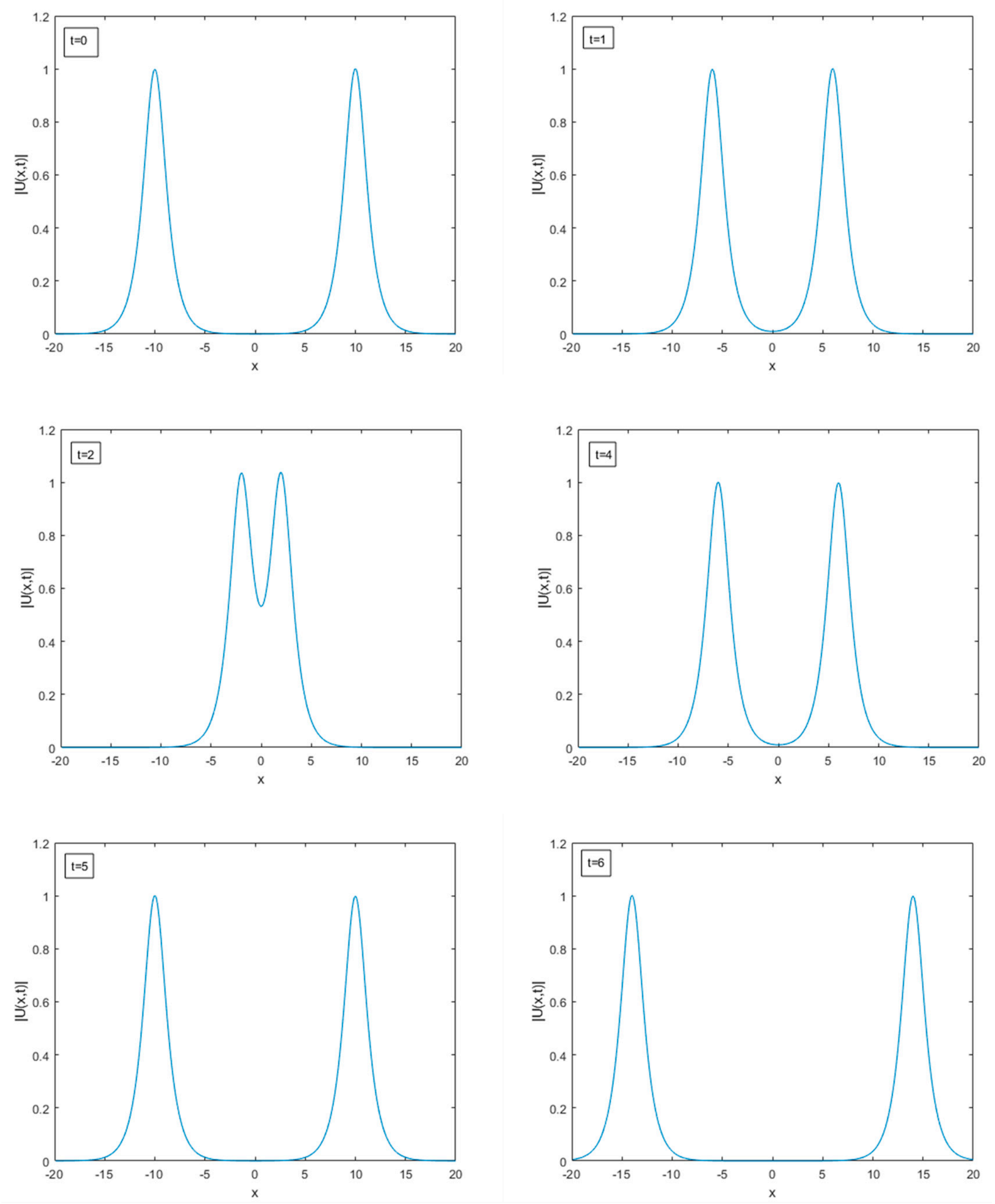

Figure 3. Interaction of two solitons, amplitude = 1 at different space and time steps.

The error norms L_∞ and L_2 are very small compared with those of Taha et al. [20], as shown in Table 7. It can be clearly seen that the error norms L_∞ and L_2 obtained by the present method are smaller than those of previous methods [20], as shown in Table 7.

Table 6. Two-solitons, amplitude = 1 at different space and time steps.

t	h	Δt	L_∞	L_2
1	0.0667	0.0025	0.0001601	0.0002367
1.5	0.133	0.01	0.0012135	0.0018299
2	0.133	0.01	0.0011711	0.0017397
3.5	0.133	0.01	0.0012135	0.0018299
4.5	0.133	0.01	0.0012144	0.0018317
0.5	0.625	0.005	0.0018529	0.0030593
1	0.625	0.005	0.0017864	0.0030560
1	0.05	0.001	0.0000486	0.0000716
1.5	0.05	0.001	0.0000485	0.0000715
3	0.5	0.0036	0.0011024	0.0017556
3	0.5	0.0025	0.0007655	0.0012192
5	0.1	0.0025	0.0002342	0.0003495

Table 7. Comparisons of two solitons with Taha et al. [20], amplitude = 1.

Galerkin Cubic B-spline (Present Method)				**Taha et al. [20]**
t	h	Δt	L_∞	L_∞
1.6	0.05	0.001	0.0000485	0.00173
1.8	0.07	0.07	0.0047374	0.00158
1	0.05	0.0025	0.0001214	0.00096
1	0.05	0.001	0.0000486	0.00141
1	0.625	0.0071	0.0025367	0.00122
1	0.130	0.0036	0.0004304	0.00141

4.3. Birth of Standing Soliton with the Maxwellian Initial Condition

If

$$\int_{-\infty}^{\infty} u(x,0) \geq \pi, \tag{33}$$

a soliton should appear over time with initial values greater than **π,** otherwise the soliton will decay away [22].

Consider the birth of soliton with the Maxwellian initial condition given by [16]:

$$u(x,0) = A\, exp\left(-x^2\right), \tag{34}$$

The values of all parameters are chosen to be $h = 0.08, \Delta t = 0.004$ and $q = 2$ for the domain $[-45, 45]$ to exhibit the birth of a soliton. The numerical simulations are shown at different times for the values of $A = 1.78$ over the domain [−45,45] in Figure 4. With $A = 1$, the soliton decays away as expected.

The conserved quantities I_1 and I_2 are computed using the Maxwellian initial condition (34). Analytical conserved quantities can be computed as:

$$I_1 = A^2\sqrt{\tfrac{\pi}{2}} = 3.97100051267043,$$
$$I_2 = \tfrac{1}{4}A^2\left(2\sqrt{2} - qA^2\right)\sqrt{\pi} = -4.92561762132093,$$

The conserved quantities I_1 and I_2 are tabulated in Table 8. The numerical results obtained are compared with the published results of Mokhtari et al. [11] and the exact solution, as shown in Table 8. The proposed method conserves I_1 to 7 decimal places and I_2 to 3 decimal places.

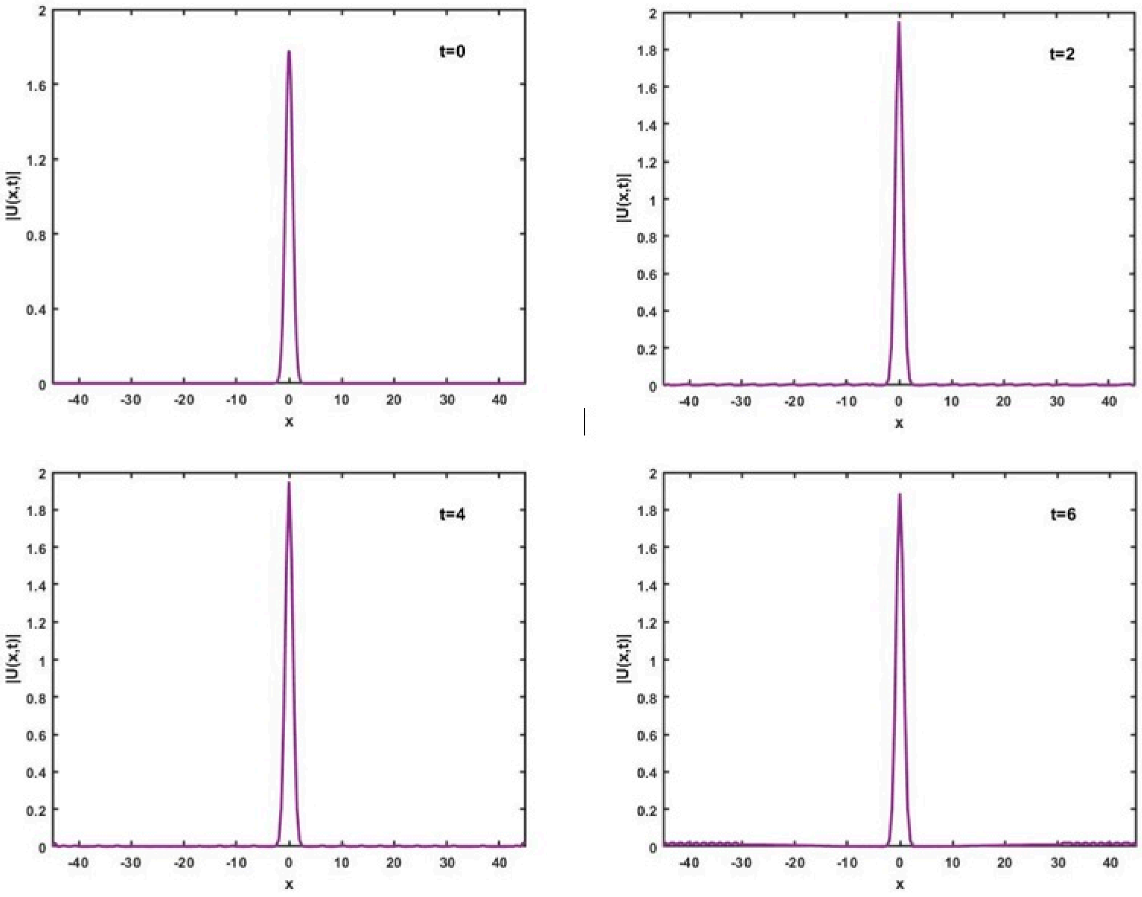

Figure 4. Birth of a standing soliton for $A = 1.78$.

Table 8. Comparison of conserved quantities of the birth of a standing soliton with Mokhtari et al. [11], $h = 0.08, \Delta t = 0.004$ and $q = 2$.

	Galerkin Cubic B-spline (Present Method)				Mokhtari et al. [11]	
t	I_1	I_2	EI_1	EI_2	EI_1	EI_2
1	3.97100052134110	−4.925146	8.67067×10^{-9}	4.7×10^{-4}	3.5×10^{-11}	4.8×10^{-10}
2	3.97100050351461	−4.925347	2.08442×10^{-8}	2.7×10^{-4}	3.8×10^{-11}	6.9×10^{-10}
3	3.97100050291892	−4.925138	2.02485×10^{-8}	4.8×10^{-4}	1.1×10^{-10}	-6.4×10^{-9}
4	3.9710005812876	−4.925325	6.86172×10^{-8}	2.9×10^{-4}	-2.0×10^{-8}	-4.5×10^{-6}
5	3.97100055661924	−4.925367	4.39488×10^{-8}	2.5×10^{-4}	-9.2×10^{-7}	-4.7×10^{-5}
6	3.97100053521529	−4.925432	2.25449×10^{-8}	1.9×10^{-4}	-6.0×10^{-6}	-9.8×10^{-5}

5. Conclusions

The approximate solution of the NLS equation was investigated using the Galerkin finite element method with a cubic B-spline shape function. Three numerical problems, including single soliton, interaction of two solitons, and birth of a standing soliton with the Maxwellian initial condition, were demonstrated to evaluate to the performance and accuracy of the method. Furthermore, we simulated the numerical solution by choosing different parameters for motion of the single soliton, the interaction of two solitons and the birth of the standing soliton. The error norms L_∞ and L_2 and conservation laws I_1 and I_2 were determined and compared with published results [11,16,17,20]. It was seen that all of our results for the problems were computed to be reasonably low, and were found to be in good agreement with the analytical solution. The present method was shown to be unconditionally stable. The method has almost a second-order convergence. In conclusion, the proposed scheme with cubic B-spline presents an acceptable result for the NLS equation.

Author Contributions: Formal analysis, A.I. and N.N.A.H.; Investigation, A.I., N.N.A.H. and A.I.M.I.; Methodology, A.I.; Supervision, N.N.A.H. and Ahmad A.I.M.I.; Writing – original draft, A.I.; Writing – review & editing, N.N.A.H. and A.I.M.I.

Funding: This work was supported by USM Short Term Grant of account number 304.PMATHS.6315077.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dag, I.; Saka, B.; Boz, A. B-spline Galerkin methods for numerical solutions of the Burgers equation. *Appl. Math. Comput.* **2005**, *166*, 506–522.
2. Dag, I.; Irk, D.; Saka, B. A numerical solution of the Burger's equation using cubic B-splines. *Appl. Math. Comput.* **2005**, *163*, 199–211.
3. Gorgulu, M.Z.; Dag, I. Galerkin method for the numerical solution of the advection-diffusion equation by using exponential B-splines. *arXiv* **2016**, arXiv:1604.04267v1.
4. Gorgulu, M.Z.; Dag, I.; Irk, D. Wave propogation by way of exponential B-spline Galerkin method. *J. Phys.* **2016**, *766*, 012031.
5. Saka, B.; Dag, I. A numerical solution of the RLW equation by Galerkin method using quartic B-splines. *Commun. Numer. Meth. Engng.* **2008**, *24*, 1339–1361. [CrossRef]
6. Aksan, E. Quadratic B-spline finite element method for numerical. *Appl. Math. Comput.* **2006**, *174*, 884–896.
7. Ali, A.; Gardner, G.; Gardner, L. A collocation solution for Burgers using cubic B-spline finite elements. *Comput. Meth. Appl.* **1992**, *100*, 325–337. [CrossRef]
8. Sheng, Q.; Khaliq, A.Q.M.; Al-Said, E.A. Solving the generalized nonlinear Schrödinger equation via quartic spline approximation. *J. Comput. Phys.* **2001**, *166*, 400–417. [CrossRef]
9. Zlotnik, A.A.; Zlotnik, I.A. Finite element method with discrete transparent boundary conditions for the time-dependent 1D schrödinger equation. *Kinet. Relat. Models* **2012**, *5*, 639–663. [CrossRef]
10. Ersoy, O.; Dag, I.; Sahin, A. Numerical investigation of the solution of Schrödinger equation with exponential cubic B-spline finite element method. *arXiv* **2016**, arXiv:1607.00166v1.
11. Mokhtari, R.; Isvand, D.; Chegini, N.G.; Salaripanah, A. Numerical solution of the Schrödinger equations. *Nonlin. Dyn.* **2013**, *74*, 77–93. [CrossRef]
12. Saka, B. A quintic B-spline finite-element method for solving the nonlinear Schrödinger equation. *Phys. Wave Phenom.* **2012**, *20*, 107–117. [CrossRef]
13. Cesarano, C.; Fornaro, C.; Fornaro, C.; Vazquez, L. Operational results in bi-orthogonal Hermite functions. *Acta Mathemat. Univ. Comenian.* **2016**, *85*, 43–68.
14. Zakharov, V.E.; Shabat, A.B. Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media. *J. Exp. Theor. Phys.* **1972**, *34*, 62–68.
15. Lin, B. Septic spline function method for nonlinear Schrödinger equations. *Appl. Analys.* **2015**, *94*, 279–293. [CrossRef]
16. Dag, I. A quadratic B-spline finite element method for solving nonlinear Schrödinger equation. *Comput. Meth. Appl. Mech. Eng.* **1999**, *174*, 247–258.
17. Gardner, L.R.T.; Gardner, G.A.; Zaki, S.I.; El Sahrawi, Z. B-spline finite element studies of the non-linear Schrödinger equation. *Comput. Methods Appl. Mech. Eng.* **1993**, *108*, 303–318. [CrossRef]
18. Hu, H. Two-grid method for two-dimensional nonlinear Schrödinger equation by finite element method. *Numeric. Methods Part. Differ. Eqs.* **2018**, *75*, 900–917.
19. Prenter, P. *Splines and Variational Methods*; Wiley: New York, NY, USA, 1975.
20. Taha, T.R.; Ablowitz, M.J. Analytical and numerical aspects of certain nonlinear evolution equations II: Numerical nonlinear Schrödinger equations. *J. Comput. Phys.* **1984**, *55*, 203–230. [CrossRef]
21. Qu, W.; Liang, Y. Stability and convergence of the Crank-Nicolson scheme for a class of variable-coefficient tempered fractional diffusion equations. *Advc. Differ. Eqs.* **2017**, *2017*, 108. [CrossRef]

22. Delfour, M.; Fortin, M.; Payne, G. Finite-difference solutions of non-linear Schrödinger equation. *J. Comput. Phys.* **1981**, *44*, 277–288. [CrossRef]