Exponential B-spline collocation method for numerical solution of the generalized regularized long wave equation

Reza Mohammadi†

Department of Mathematics, University of Neyshabur, Neyshabur 91136-899, *Iran*

(Received 2 November 2014; revised manuscript received 16 December 2014; published online 27 March 2015)

The aim of the present paper is to present a numerical algorithm for the time-dependent generalized regularized long wave equation with boundary conditions. We semi-discretize the continuous problem by means of the Crank–Nicolson finite difference method in the temporal direction and exponential B-spline collocation method in the spatial direction. The method is shown to be unconditionally stable. It is shown that the method is convergent with an order of $\mathcal{O}(k^2+h^2)$. Our scheme leads to a tri-diagonal nonlinear system. This new method has lower computational cost in comparison to the Sinc-collocation method. Finally, numerical examples demonstrate the stability and accuracy of this method.

Keywords: solitary waves, GRLW equation, exponential B-spline, collocation

PACS: 02.60.Lj, 03.65.Ge **DOI:** 10.1088/1674-1056/24/5/050206

1. Introduction

Nonlinear partial differential equations are useful in describing various phenomena in applied mathematics and physics. Analytical solutions of these equations are usually not available, especially when the nonlinear terms are involved. Since only limited classes of the equations are solved by analytical means, numerical solutions of these nonlinear partial differential equations are of practical importance (see Refs. [1]–[5]). Solitary waves are wave packets or pulses, which propagate in nonlinear dispersive media. Due to the dynamical balance between the nonlinearity and the dispersive effects, these waves retain a stable waveform. A soliton is a very special type of a solitary wave, which propagates without any change of its shape and velocity properties after collisions with other solitons. The present work is concerned with the numerical solution of the regularized long wave (RLW) equation, which was first derived by Peregrine to define undular bore development.[6] Since then, this equation has been used to model a large number of problems arising in various areas of applied science. The RLW equation is an alternative description of nonlinear dispersive waves to the more usual Korteweg–de Vries (KdV) equation. It has been found to have solitary wave solutions and govern a large number of important physical phenomena such as nonlinear transverse waves in a shallow water, ion-acoustic, magneto-hydrodynamic waves in plasma and phonon packets in nonlinear crystals, anharmonic lattices, longitudinal dispersive waves in elastic rods, pressure waves in liquid–gas bubble mixtures, rotating flow down a tube, the lossless propagation of shallow water waves, and thermally exited phonon packets in low-temperature and nonlinear crystals.[7–9] Indeed, the RLW equation is a special case of the generalized regularized long wave (GRLW) equation, which is given by

$$u_t + \gamma u_x + \varepsilon u^p u_x - \mu u_{xxt} = 0, \tag{1}$$

where p is a positive integer, $\gamma \in \mathbb{R}$, ε and μ are positive constants which require the boundary conditions $u \to 0$ as $x \to \pm\infty$. It was proposed many years ago as a model for small-amplitude long waves on the surface of water.[6,10] Here the cases $p = 1, 2$ correspond to the RLW and MRLW equations, respectively.

The solutions of this kind of equations are important in many applications. However, for more general initial conditions, the theoretical solutions for these equations are limited. Therefore, many researchers have put in a great deal of effort to compute the solutions of these equations using various numerical methods and the numerical solutions of the initial and initial-boundary problems of the nonlinear GRLW equation.

Bona *et al.*[11] investigated the Fourier spectral method for the initial value problem of the GRLW equation. The method of lines based on the discretization of the spatial derivatives by means of Fourier pseudo-spectral approximations for the GRLW equation was derived by Durán and López–Marcos.[12] Djidjeli *et al.*[13] discussed a linearized implicit pseudo-spectral method for the RLW equation. Adomian decomposition method was implemented for finding the solitary-wave solutions of GRLW equation by Kaya and El. Syed.[14,15] Zhang[16] developed an implicit second-order accurate and stable energy preserving finite difference method based on the use of central difference equations for the time and space derivatives for GRLW equation. A numerical method for the GRLW equation using a reproducing kernel function was studied by Xie *et al.*[17] Jafari *et al.*[18] used sine–cosine and tan*h* methods to obtain the solutions of the GRLW

†Corresponding author. E-mail: rez.mohammadi@gmail.com, mohammadi@neyshabur.ac.ir

© 2015 Chinese Physical Society and IOP Publishing Ltd

equation. Mokhtari and Mohammadi[19] presented a mesh-free technique based on sinc basis functions for the numerical solution of the GRLW equation. In addition, they proposed a new iterative algorithm for solving the GRLW equation.[20] Feng *et al.*[21] presented a mesh-less method for the nonlinear GRLW equation based on the moving least-squares approximation. The GRLW equation is solved numerically via the Petrov–Galerkin method by Roshan.[22] Their method uses a linear hat function as the trial function and a quintic B-spline function as the test function. Lin[23] presented a numerical method based on parametric adaptive cubic spline functions for solving the GRLW equation.

The RLW equation is an important equation in physics media describing phenomena with weak nonlinearity and dispersion waves.[24] Recently, many well-known numerical techniques, such as finite element methods based on the least square principle,[25–27] finite element methods based on the Galerkin and collocation principles,[28–36] the Petrov–Galerkin method,[37] the radial basis functions collocation method,[38,39] B-spline differential quadrature methods,[40] mesh-less methods,[41] Galerkin and collocation methods,[42–45] the spectral method[46] and non-polynomial spline method[47] are employed to solve the RLW equation.

The RLW equation possesses only three invariant constants of motion corresponding to conservation of mass, momentum, and energy given by[48]

$$\begin{aligned}\mathscr{I}_1 &= \int_a^b u\,\mathrm{d}x,\\ \mathscr{I}_2 &= \int_a^b (u^2+\mu u_x^2)\,\mathrm{d}x,\\ \mathscr{I}_3 &= \int_a^b (u^3+\mu 3u^2)\,\mathrm{d}x.\end{aligned} \tag{2}$$

The MRLW equation is another particular case of the GRLW equation for $p=2$. Gardner *et al.*[48] used a collocation method with quintic B-splines for the numerical solution of the MRLW equation. Khalifa *et al.*[49] applied a finite difference method to study solitary waves for the MRLW equation, and showed that the scheme is marginally stable. The MRLW equation is solved numerically via a collocation method using cubic B-splines finite element by Khalifa *et al.*[50] Khalifa *et al.*[51] used the Adomian decomposition method for solving the MRLW equation. Ali[52] has developed a classical radial basis function collocation method for solving the MRLW equation. Raslan[53] used the quadratic B-spline function and a central difference operator for the time derivative to develop a new algorithm based on the collocation method to solve the MRLW equation. A computational comparison study of quadratic, cubic, quartic, and quintic splines for solving the MRLW equation is presented by Raslan and Hassan.[54] Raslan and EL-Danaf[55] used the B-spline finite element method to solve the MRLW equation. The numerical scheme based on the quartic B-spline collocation method is designed for the numerical solution of the MRLW equation by Haq *et al.*[56] Achouri and Omrani[57] used the homotopy perturbation method to implement the MRLW equation with some initial conditions. The MRLW equation, with some initial conditions, is solved numerically via a variational iteration method by Labidi and Omrani.[58] Dereli[59,60] simulated traveling wave solutions of the MRLW equation using the mesh-less method based on collocation with well-known radial basis functions. Khan *et al.*[61] employed the homotopy analysis method to obtain an approximate numerical solution of the MRLW equation with some specified initial conditions. Karakoc *et al.*[62] obtained a numerical solution of the MRLW equation based on the quintic B-spline collocation method over the finite elements.

The MRLW equation also has three conservation laws related to mass, momentum, and energy[49]

$$\begin{aligned}\mathscr{I}_1 &= \int_a^b u\,\mathrm{d}x,\\ \mathscr{I}_2 &= \int_a^b (u^2+\mu u_x^2)\,\mathrm{d}x,\\ \mathscr{I}_3 &= \int_a^b (u^4-6\mu 3u_x^2)\,\mathrm{d}x.\end{aligned} \tag{3}$$

The spline function is a piecewise continuous polynomial function suitable to approximate an arbitrary function. It is known to approximate an arbitrary function better than other polynomial functions with the same degree. Spline functions are used in many applications such as interpolation, data fitting, and numerical solution of ordinary and partial differential equations. Rashidinia and Mohammadi[63,64] derived approximate expressions for the dispersion relation of the nonlinear Klein–Gordon and sine–Gordon equations in the case of strong nonlinearities using a method based on tension spline functions and finite difference approximations. Also, Mohammadi developed explicit methods using exponential spline functions and finite difference approximations for the numerical solution of the generalized Burgers'–Fisher,[65] convection–diffusion,[66] nonlinear Schrödinger,[67] and Euler–Bernoulli beam[68] equations. Chegini *et al.*[47] employed non-polynomial spline basis functions to obtain approximate solutions of the RLW equation. Bian[69] used the Crank–Nicolson (C–N) method combined with a B-spline basis set for time-dependent and time-independent calculations of the photoionization cross sections of atoms and molecules.

In this work, we use exponential B-spline collocation and Crank–Nicolson methods for the numerical study of the

GRLW equation. Our main objective is to improve the accuracy of the cubic B-spline method by utilizing some parameters, which enable us to obtain various classes of methods. Our method is a modification of the cubic B-spline method for the numerical solution of Eq. (1). Application of our method is simple. While solving initial boundary value problems in partial differential equations, the procedure is to combine a spline approximation for the space derivative with a Crank–Nicolson finite difference approximation for the time derivative. The combination of a finite difference and an exponential spline functions technique provides better accuracy than finite difference methods. Therefore, the time derivative is replaced by a finite difference representation and the space derivatives by exponential B-splines. We use the exponential B-spline basis that leads to the tri-diagonal nonlinear system which can be solved by well known algorithms such as the Newton algorithm. Numerical examples are presented, which demonstrate that the present scheme with exponential B-splines gives more accurate approximations than the scheme using cubic B-splines. Our method is a user friendly, economical and stable method for GRLW equation.

The rest of this article is organized as follows. In Section 2, the B-spline basis for the space of exponential splines is given and some interpolation results for exponential splines are stated. In Section 3, the construction of the exponential B-spline collocation method for the numerical solution of GRLW equation is described. In Section 4, the stability analysis is theoretically discussed. Using the Von Neuman method, the proposed method is shown to be unconditionally stable. Section 5 is devoted to the convergence analysis of our presented method. The method is shown to have second-order convergence. In Section 6, numerical examples are included to illustrate the practical implementation of the proposed method. The accuracy performance of the obtained numerical results are compared to the exact solution and other existing methods. The numerical results show that our proposed method is a promising approach for solving different types of these non-linear partial differential equations problems. Finally, we conclude our paper in Section 7.

2. Exponential spline functions

For an interval $[a,b]$, we divide it into n subintervals $[x_i,x_{i+1}]$ $(i=0,1,\ldots,n-1)$ by the equidistant knots $x_i=a+ih$ $(i=0,1,\ldots,n)$, where $h=(b-a)/n$.

A function $\mathscr{S}(x)$ of class $\mathscr{C}^2[a,b]$ is a piecewise non-polynomial, which interpolates a function $u(x)$ at the mesh points x_i, depends on a parameter ρ and reduces to cubic spline in $[a,b]$ as $\rho\to 0$, called an exponential spline function. For more details about exponential spline functions and their definitions, we refer readers to Refs. [66], [70], and [71] and the references therein.

Among various classes of splines, the polynomial spline has received great attention primarily because it admits a basis of B-splines which can be accurately and efficiently computed. It has been shown that an exponential spline also admits a basis of B-splines which can be defined as follows. Let $B_i(x)$ be the B-spline centered at x_i and have a finite support on the four consecutive intervals $[x_i+lh,x_i+(l+1)h]_{l=-2}^{1}$. According to Ref. [72], the $B_i(x)$ can be defined as

$$B_i(x)=\begin{cases} g_1(x-x_{i-2}), & x\in[x_{i-2},x_{i-1}],\\ g_2(x_i-x), & x\in[x_{i-1},x_i],\\ g_2(x-x_i), & x\in[x_i,x_{i+1}],\\ g_1(x_{i+2}-x), & x\in[x_{i+1},x_{i+2}],\\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where

$$g_1(x)=b_2\big(x-\frac{1}{\rho}\sinh(\rho x)\big),$$
$$g_2(x)=a_1+b_1x+c_1\,\mathrm{e}^{\rho x}+d_1\,\mathrm{e}^{-\rho x},$$

and

$$a_1=\frac{\rho h\cosh(\rho h)}{\rho h\cosh(\rho h)-\sinh(\rho h)},$$
$$b_1=\frac{\rho}{2}\left[\frac{\cosh(\rho h)\big(\cosh(\rho h)-1\big)+\sinh^2(\rho h)}{\big(\rho h\cosh(\rho h)-\sinh(\rho h)\big)\big(1-\cosh(\rho h)\big)}\right],$$
$$b_2=\frac{\rho}{2\big(\rho h\cosh(\rho h)-\sinh(\rho h)\big)},$$
$$c_1=\frac{1}{4}\left[\frac{\mathrm{e}^{-\rho h}\big(1-\cosh(\rho h)\big)+\sinh(\rho h)\big(\mathrm{e}^{-\rho h}-1\big)}{\big(\rho h\cosh(\rho h)-\sinh(\rho h)\big)\big(1-\cosh(\rho h)\big)}\right],$$
$$d_1=\frac{1}{4}\left[\frac{\mathrm{e}^{\rho h}\big(\cosh(\rho h)-1\big)+\sinh(\rho h)\big(\mathrm{e}^{\rho h}-1\big)}{\big(\rho h\cosh(\rho h)-\sinh(\rho h)\big)\big(1-\cosh(\rho h)\big)}\right].$$

Each basis function $B_i(x)$ is twice continuously differentiable.

Let $s(x)$ be the cubic spline and $\mathscr{S}(x)$ be the exponential spline. Now we consider the following results, which have been established by Pruess.[73]

Theorem 1 $\|D^i(s-\mathscr{S})\|_\infty\le\frac{26}{3}\rho^2h^{4-i}\max_{\forall k}|s_k''|$, $i=0,1,2$.

Following Ref. [74] we obtain the following theorem.

Theorem 2 $\|D^i(s-\mathscr{S})\|_\infty\le\frac{26}{3}\rho^2h^{4-i}\max_{\forall k}|\mathscr{S}_k''|, i=0,1,2$.

These theorems together with the de Boor–Hall error estimate[75] lead to the following corollary.

Corollary 1 If $u(x,t)\in C^4[a,b]$, then there exists a constant, χ_i, independent of h such that

$$\|D^i(u-\mathscr{S})\|_\infty\le\chi_ih^{4-i},\quad i=0,1,2.$$

3. Implementation of the numerical method

In this section, we consider the initial-boundary value problem of the GRLW equation

$$u_t+\gamma u_x-\mu u_{xxt}+F(u,u_x)=0,\quad a<x<b,\quad 0<t\le T, \tag{5}$$

with the initial condition

$$u(x,0)=\phi(x),\quad a\le x\le b, \tag{6}$$

and the boundary conditions

$$u(a,t)=f_0(t),\quad u(b,t)=f_1(t),\quad 0<t\le T, \tag{7}$$

where $F(u,u_x)=\varepsilon u^p u_x$, γ, ε, and μ are positive parameters, and $\phi(x)$, $f_0(t)$, and $f_1(t)$ are known functions.

The domain $[a,b]\times[0,T]$ is divided into an $n\times m$ mesh, with $h=\frac{b-a}{n}$ and $k=\frac{T}{m}$, respectively. $x_i=a+ih$, for $i=0,1,2,\ldots,n$ is the ith node. In the temporal dimension, the uniform step size k is used, so $t_j=jk$, $j=0,1,\ldots,m$, is the time level for the jth step. The quantity $U(x_i,t_j)$ represents the exact solution at (x_i,t_j) while $u(x_i,t_j)$ represents the numerical solution at (x_i,t_j).

3.1. Temporal semi-discretization

Now, we use the horizontal method of lines to discretize the time variable by means of the Crank–Nicolson method with a constant step size k, keeping the space variable x continuous. This semi-discretization yields the following system of nonlinear ordinary differential equations:

$$\begin{aligned}&\Big(\frac{u^{j+1}-u^j}{k}\Big)+\gamma\Big(\frac{(u_x)^{j+1}+(u_x)^j}{2}\Big)\\&-\mu\Big(\frac{(u_{xx})^{j+1}-(u_{xx})^j}{k}\Big)\\&+\frac{F\big(u^{j+1},(u_x)^{j+1}\big)+F\big(u^j,(u_x)^j\big)}{2}=0,\\&u^0=\phi(x),\quad a\le x\le b,\end{aligned} \tag{8}$$

with the boundary conditions

$$u^{j+1}(a)=f_0(t^{j+1}),\quad u^{j+1}(b)=f_1(t^{j+1}). \tag{9}$$

We can rewrite Eqs. (8) and (9) as

$$\begin{aligned}&u^{j+1}+\frac{k\gamma}{2}(u_x)^{j+1}-\mu(u_{xx})^{j+1}+\frac{k}{2}F\big(u^{j+1},(u_x)^{j+1}\big)\\&=G\big(x,t_j,u^j,(u_x)^j,(u_{xx})^j\big),\\&u^{j+1}(a)=f_0(t^{j+1}),\quad u^{j+1}(b)=f_1(t^{j+1}),\\&u^0=\phi(x),\quad a\le x\le b,\end{aligned} \tag{10}$$

where

$$\begin{aligned}&G\big(x,t_j,u^j,(u_x)^j,(u_{xx})^j\big)\\&=u^j-\frac{k\gamma}{2}(u_x)^j-\mu(u_{xx})^j-\frac{k}{2}F\big(u^j,(u_x)^j\big).\end{aligned}$$

Let $U(t_{j+1})$ be the exact solution and $u(t_{j+1})$ the approximate solution of problem (10) at the $(j+1)$-th time level, the local truncation error of the time semi-discretization method is given by $e_{j+1}=U(t_{j+1})-u(t_{j+1})$. The local error estimate of each time step contributes to the global error in the temporal discretization, which is defined, at t_j, as $E_j=\sum_{l=1}^{j-1}[U(t_l)-u(t_l)]$.

Lemma 1 If

$$\Big|\frac{\partial^{i+j}}{\partial x^i\partial t^j}\Big|\le\mathfrak{C},\quad (x,t)\in[a,b]\times[0,T],$$
$$0\le j\le 3,\quad 0\le i\le 4,$$

then the local truncation error associated to the semi-discrete scheme (8) satisfies

$$\|e_j\|_\infty\le\mathfrak{C}k^3, \tag{11}$$

and the global truncation error at time t_j satisfies

$$\|E_j\|_\infty\le\mathfrak{C}k^2. \tag{12}$$

Proof Because u is adequately smooth and $|u_{ttt}|\le\mathfrak{C}$, by Taylor series expansion, we have

$$\begin{aligned}u(x,t_{j+1})&=u(x,t_{j+\frac12})+\frac{k}{2}u_t(x,t_{j+\frac12})\\&\quad+\frac{k^2}{8}u_{tt}(x,t_{j+\frac12})+\frac{k^3}{48}u_{ttt}(x,t_{j+\frac12})+\mathscr{O}(k^4),\\u(x,t_j)&=u(x,t_{j+\frac12})-\frac{k}{2}u_t(x,t_{j+\frac12})+\frac{k^2}{8}u_{tt}(x,t_{j+\frac12})\\&\quad-\frac{k^3}{48}u_{ttt}(x,t_{j+\frac12})+\mathscr{O}(k^4),\\u_{xx}(x,t_{j+1})&=u_{xx}(x,t_{j+\frac12})+\frac{k}{2}u_{xxt}(x,t_{j+\frac12})\\&\quad+\frac{k^2}{8}u_{xxtt}(x,t_{j+\frac12})+\frac{k^3}{48}u_{xxttt}(x,t_{j+\frac12})+\mathscr{O}(k^4),\\u_{xx}(x,t_j)&=u_{xx}(x,t_{j+\frac12})-\frac{k}{2}u_{xxt}(x,t_{j+\frac12})\\&\quad+\frac{k^2}{8}u_{xxtt}(x,t_{j+\frac12})-\frac{k^3}{48}u_{xxttt}(x,t_{j+\frac12})+\mathscr{O}(k^4).\end{aligned}$$

With the help of the above equations, one can obtain

$$\begin{aligned}&\frac{u(x,t_{j+1})-u(x,t_j)}{k}\\&=u_t(x,t_{j+\frac12})+\frac{k^2}{24}u_{ttt}(x,t_{j+\frac12})+\mathscr{O}(k^3),\end{aligned} \tag{13}$$

and

$$\begin{aligned}&\frac{u_{xx}(x,t_{j+1})-u_{xx}(x,t_j)}{k}\\&=u_{xxt}(x,t_{j+\frac12})+\frac{k^2}{24}u_{xxttt}(x,t_{j+\frac12})+\mathscr{O}(k^3).\end{aligned} \tag{14}$$

Using Eqs. (5), (13), and (14), we have

$$\begin{aligned}&\frac{\big[u(x,t_{j+1})-u(x,t_j)\big]-\mu\big[u_{xx}(x,t_{j+1})-u_{xx}(x,t_j)\big]}{k}\\&=\big[-\gamma u_x(x,t_{j+\frac12})-F\big(u(x,t_{j+\frac12}),u_x(x,t_{j+\frac12})\big)\big]\\&\quad+\frac{k^3}{24}\big[u_{ttt}(x,t_{j+\frac12})-\mu u_{xxttt}(x,t_{j+\frac12})\big]+\mathscr{O}(k^4).\end{aligned} \tag{15}$$

Moreover, as

$$\begin{aligned}&\frac{u(x,t_{j+1})+u(x,t_j)}{2}\\ &= u(x,t_{j+\frac{1}{2}})+\frac{k^2}{8}u_{tt}(x,t_{j+\frac{1}{2}})+\mathscr{O}(k^4),\\ &\frac{u_{xx}(x,t_{j+1})+u_{xx}(x,t_j)}{2}\end{aligned} \tag{16}$$

$$= u_{xx}(x,t_{j+\frac{1}{2}})+\frac{k^2}{8}u_{xxtt}(x,t_{j+\frac{1}{2}})+\mathscr{O}(k^4), \tag{17}$$

and

$$\begin{aligned}&\left|\gamma u_{ttx}(x,t)+F\big(u_{tt}(x,t),u_{ttx}(x,t)\big)\right|\le \mathfrak{C},\\ &\text{on}\quad [a,b]\times[0,T],\end{aligned}$$

we have

$$\begin{aligned}&\frac{\gamma u_x(x,t_{j+1})+\gamma u_x(x,t_j)+F\big(u(x,t_{j+1}),u_x(x,t_{j+1})\big)+F\big(u(x,t_j),u_x(x,t_j)\big)}{2}\\ &= \gamma u_x(x,t_{j+\frac{1}{2}})+F\big(u(x,t_{j+\frac{1}{2}}),u_x(x,t_{j+\frac{1}{2}})\big)+\mathscr{O}(k^2).\end{aligned} \tag{18}$$

From Eqs. (15) and (18), we deduce that

$$\begin{aligned}&u(x,t_{j+1})-\mu u_{xx}(x,t_{j+1})+\frac{k\gamma}{2}u_x(x,t_{j+1})\\ &+\frac{k}{2}F\big(u(x,t_{j+1}),u_x(x,t_{j+1})\big)\\ &= G\big(x_i,t_j,u^j,(u_x)^j,(u_{xx})^j\big)+\mathscr{O}(k^3).\end{aligned} \tag{19}$$

Using the mean-value theorem we rewrite Eq. (19) as

$$\left[I+\frac{k}{2}L_0+\frac{k}{2}(\gamma+L_1)\frac{\partial}{\partial x}-\mu\frac{\partial^2}{\partial x^2}\right]e_{j+1}=\mathscr{O}(k^3). \tag{20}$$

Now following Ref. [76] we obtain

$$\|e_j\|_\infty\le \mathfrak{C}k^3. \tag{21}$$

Using the local error estimate up to the j-th time step given by Eq. (21), we obtain the following global error estimate at the j-th time step:

$$\begin{aligned}\|E_j\|_\infty &= \left\|\textstyle\sum_{i=1}^{j-1}e_i\right\|_\infty, && (j-1)\le\frac{T}{k},\\ &\le \|e_1\|_\infty+\|e_2\|_\infty+\cdots+\|e_{j-1}\|_\infty,\\ &\le \mathfrak{C}\big((j-1)k\big)k^2,\\ &\le \mathfrak{C}Tk^2,\\ &\equiv \mathfrak{C}k^2,\end{aligned} \tag{22}$$

which gives the second-order convergent of the method in time direction.

3.2. Spatial discretization

Let B_i $(i=-1,0,\ldots,n+1)$ be the exponential B-splines with both knots x_i and two additional knots outside the region, positioned at $x_{-1}<x_0$ and $x_n<x_{n+1}$. The set of exponential B-splines $B_{-1},B_0,\ldots,B_{n+1}$ forms a basis for functions defined over the problem domain $[a,b]$. The exponential B-splines (4) and their first and second derivatives vanish outside the interval $[x_{i-2},x_{i+2}]$.

An approximate solution $u(x,t)$ to the analytical solution $U(x,t)$ will be sought in the form of an expansion of B-splines

$$u(x,t)=\sum_{i=-1}^{n+1}\varsigma_i(t)B_i(x), \tag{23}$$

where ς_i are time-dependent parameters to be determined from exponential B-spline collocation form of Eq. (1).

At knots, nodal values and their principal first and second derivatives are obtained in terms of the time parameters using exponential functions (4) and the approximate solution $u(x,t)$ as

$$u_i^{j+1}=u(x_i,t_{j+1})=\alpha\varsigma_{i-1}^{j+1}+\varsigma_i^{j+1}+\alpha\varsigma_{i+1}^{j+1}, \tag{24}$$

$$(u_x)_i^{j+1}=\frac{\partial u}{\partial x}(x_i,t_{j+1})=\beta\big(\varsigma_{i-1}^{j+1}-\varsigma_{i+1}^{j+1}\big), \tag{25}$$

$$(u_{xx})_i^{j+1}=\frac{\partial^2 u}{\partial x^2}(x_i,t_{j+1})=\delta\big(\varsigma_{i-1}^{j+1}-2\varsigma_i^{j+1}+\varsigma_{i+1}^{j+1}\big), \tag{26}$$

where

$$\begin{aligned}\alpha&=\frac{\sinh(\rho h)-\rho h}{2\big(\rho h\cosh(\rho h)-\sinh(\rho h)\big)},\\ \beta&=\frac{\rho\big(1-\cosh(\rho h)\big)}{2\big(\rho h\cosh(\rho h)-\sinh(\rho h)\big)},\\ \delta&=\frac{\rho^2\sinh(\rho h)}{2\big(\rho h\cosh(\rho h)-\sinh(\rho h)\big)}.\end{aligned}$$

If we substitute the global approximation (23) and its necessary derivatives (24)–(26) at the knots in Eq. (10), we obtain the following difference equation with the variables ς:

$$\begin{aligned}&\left(\alpha+\frac{k\beta\gamma}{2}-\delta\mu\right)\varsigma_{i-1}^{j+1}+\big(1+2\delta\mu\big)\varsigma_i^{j+1}\\ &+\left(\alpha-\frac{k\beta\gamma}{2}-\delta\mu\right)\varsigma_{i+1}^{j+1}+\frac{k}{2}F\big((\alpha\varsigma_{i-1}^{j+1}+\varsigma_i^{j+1}+\alpha\varsigma_{i+1}^{j+1}),\\ &\beta(\varsigma_{i-1}^{j+1}-\varsigma_{i+1}^{j+1})\big)=G\big(x_i,t_j,u_i^j,(u_x)_i^j,(u_{xx})_i^j\big),\\ &i=0,1,\ldots,n,\quad j=0,1,\ldots,m.\end{aligned} \tag{27}$$

System (27) consists of $(n+1)$ equations in the unknowns ς_i^{j+1} $(i=-1,0,\ldots,n+1)$. To obtain a solution to this system we need two additional equations. From boundary conditions (7) and Eq. (24) we have

$$\varsigma_{-1}^{j+1}=\frac{1}{\alpha}\big(f_0(t_{j+1})-\varsigma_0^{j+1}\big)-\varsigma_1^{j+1},$$

and

$$\varsigma_{n+1}^{j+1} = \frac{1}{\alpha}\left(f_1(t_{j+1}) - \varsigma_n^{j+1}\right) - \varsigma_{n-1}^{j+1}.$$

Putting $i = 0, n$ in Eq. (27) and using the above notations we obtain two additional equations

$$\begin{aligned}
&\left(\delta\mu\left(2+\frac{1}{\alpha}\right) - \frac{k\beta\gamma}{2\alpha}\right)\varsigma_0^{j+1} - k\beta\gamma\varsigma_1^{j+1} \\
&+ \frac{k}{2}F\left(f_0(t_{j+1}), \frac{\beta}{\alpha}\left(f_0(t_{j+1}) - \varsigma_0^{j+1}\right) - 2\beta\varsigma_1^{j+1}\right) \\
&= G\left(x_0, t_j, u_0^j, (u_x)_0^j, (u_{xx})_0^j\right) \\
&- \left(1 + \frac{1}{2\alpha}(k\beta\gamma - 2\delta\mu)\right) f_0(t_{j+1}), \quad j = 0, 1, \ldots, m, \\
&k\beta\gamma\varsigma_{n-1}^{j+1} + \left(\delta\mu\left(2+\frac{1}{\alpha}\right) + \frac{k\beta\gamma}{2\alpha}\right)\varsigma_n^{j+1} \\
&+ \frac{k}{2}F\left(f_1(t_{j+1}), 2\beta\varsigma_{n-1}^{j+1} - \frac{\beta}{\alpha}\left(f_1(t_{j+1}) - \varsigma_n^{j+1}\right)\right) \\
&= G\left(x_n, t_j, u_n^j, (u_x)_n^j, (u_{xx})_n^j\right) \\
&- \left(1 - \frac{1}{2\alpha}(k\beta\gamma + 2\delta\mu)\right) f_1(t_{j+1}), \quad j = 0, 1, \ldots, m. \qquad (28)
\end{aligned}$$

Equations (27) and (28) can be written in the matrix form as follows:

$$\mathscr{A}\mathscr{V}^{j+1} + \frac{k}{2}\mathscr{F}^{j+1} = \mathscr{G}^j, \quad j = 0, 1, \ldots, m, \qquad (29)$$

where

$$\mathscr{V}^{j+1} = \left(\varsigma_0^{j+1}, \varsigma_1^{j+1}, \ldots, \varsigma_n^{j+1}\right)^t,$$

$$\mathscr{A} = \begin{pmatrix}
\delta\mu(2+\frac{1}{\alpha}) - \frac{k\beta\gamma}{2\alpha} & -k\beta\gamma & & & \\
\alpha + \frac{k\beta\gamma}{2} - \delta\mu & 1 + 2\delta\mu & \alpha - \frac{k\beta\gamma}{2} - \delta\mu & & \\
& \ddots & \ddots & \ddots & \\
& & \alpha + \frac{k\beta\gamma}{2} - \delta\mu & 1 + 2\delta\mu & \alpha - \frac{k\beta\gamma}{2} - \delta\mu \\
& & & k\beta\gamma & \delta\mu(2+\frac{1}{\alpha}) + \frac{k\beta\gamma}{2\alpha}
\end{pmatrix},$$

$$\mathscr{F}^{j+1} = \begin{pmatrix}
F\left(f_0(t_{j+1}), \frac{\beta}{\alpha}\left(f_0(t_{j+1}) - \varsigma_0^{j+1}\right) - 2\beta\varsigma_1^{j+1}\right) \\
F\left(\left(\alpha\varsigma_0^{j+1} + \varsigma_1^{j+1} + \alpha\varsigma_2^{j+1}\right), \beta\left(\varsigma_0^{j+1} - \varsigma_2^{j+1}\right)\right) \\
\vdots \\
F\left(\left(\alpha\varsigma_{n-2}^{j+1} + \varsigma_{n-1}^{j+1} + \alpha\varsigma_n^{j+1}\right), \beta\left(\varsigma_{n-2}^{j+1} - \varsigma_n^{j+1}\right)\right) \\
F\left(f_1(t_{j+1}), 2\beta\varsigma_{n-1}^{j+1} - \frac{\beta}{\alpha}\left(f_1(t_{j+1}) - \varsigma_n^{j+1}\right)\right)
\end{pmatrix},$$

$$\mathscr{G}^j = \begin{pmatrix}
G\left(x_0, t_j, u_0^j, (u_x)_0^j, (u_{xx})_0^j\right) - \left(1 + \frac{1}{2\alpha}(k\beta\gamma - 2\delta\mu)\right) f_0(t_{j+1}) \\
G\left(x_1, t_j, u_1^j, (u_x)_1^j, (u_{xx})_1^j\right) \\
\vdots \\
G\left(x_{n-1}, t_j, u_{n-1}^j, (u_x)_{n-1}^j, (u_{xx})_{n-1}^j\right) \\
G\left(x_n, t_j, u_n^j, (u_x)_n^j, (u_{xx})_n^j\right) - \left(1 - \frac{1}{2\alpha}(k\beta\gamma + 2\delta\mu)\right) f_1(t_{j+1})
\end{pmatrix}.$$

The above $(n+1)\times(n+1)$ nonlinear matrix system is solved using the Newton method. The starting vector is chosen to be zero.

3.3. The initial state

To obtain the solution parameters, it is necessary to find the initial vector u^0. To do this, the initial condition (6) uses the following expression:

$$u(x_i,0)=\phi(x_i),\quad i=0,\ldots,n. \tag{30}$$

After obtaining the initial vector u^0, at the next time step the time vectors $\mathscr{G}^j$ are computed from the recurrence relationship (29).

4. The stability analysis

For stability analysis, following Refs. [35], [47], [48], [50], [56], and [79] and the references therein we use the Von Neumann method. The nonlinear term $F(u,u_x)$ cannot be handled by the Fourier mode method, so we must linearize it by making $F\left(u_i^j,(u_x)_i^j\right)=\varepsilon q(u_x)_i^j$ and $F\left(u_i^{j+1},(u_x)_i^{j+1}\right)=\varepsilon q^*(u_x)_i^{j+1}$. According to the Von Neumann method, we have

$$\varsigma_i^j=\xi^j\exp(\mathrm{i}\psi\mathrm{i}h), \tag{31}$$

where ψ is the mode number, $i=\sqrt{-1}$, h is the element size, and ξ is the amplification factor of the scheme. Substituting Eq. (31) into the linearized form of spline-difference equation (27) and using

$$\begin{aligned}
u_i^j&=u(x_i,t_j)=\alpha\varsigma_{i-1}^j+\varsigma_i^j+\alpha\varsigma_{i+1}^j,\\
(u_x)_i^j&=\frac{\partial u}{\partial x}(x_i,t_j)=\beta\left(\varsigma_{i-1}^j-\varsigma_{i+1}^j\right),\\
(u_{xx})_i^j&=\frac{\partial^2 u}{\partial x^2}(x_i,t_j)=\delta\left(\varsigma_{i-1}^j-2\varsigma_i^j+\varsigma_{i+1}^j\right),
\end{aligned}$$

we obtain

$$\begin{aligned}
&\xi^{j+1}\Big[P_1\exp\left(\mathrm{i}\psi(i-1)h\right)+R\exp\left(\mathrm{i}\psi\mathrm{i}h\right)\\
&+Q_1\exp\left(\mathrm{i}\psi(i+1)h\right)\Big]\\
&=\xi^j\Big[P_2\exp\left(\mathrm{i}\psi(i-1)h\right)+R\exp\left(\mathrm{i}\psi\mathrm{i}h\right)\\
&+Q_2\exp\left(\mathrm{i}\psi(i+1)h\right)\Big],\quad i=0,1,\ldots,n,
\end{aligned} \tag{32}$$

where

$$\begin{aligned}
P_1&=2\alpha+k\beta\gamma-2\delta\mu+k\beta\varepsilon q^*,\\
R&=2(1+2\delta\mu),\\
Q_1&=2\alpha-k\beta\gamma-2\delta\mu-k\beta\varepsilon q^*,\\
P_2&=2\alpha-k\beta\gamma-2\delta\mu-k\beta\varepsilon q,\\
Q_2&=2\alpha+k\beta\gamma-2\delta\mu+k\beta\varepsilon q.
\end{aligned}$$

After simple calculations, equation (32) becomes

$$\xi=\frac{P_2\exp(-\mathrm{i}\psi h)+R+Q_2\exp(\mathrm{i}\psi h)}{P_1\exp(-\mathrm{i}\psi h)+R+Q_1\exp(\mathrm{i}\psi h)}. \tag{33}$$

Using Euler's formula, that is, $\exp(\mathrm{i}\psi h)=\cos(\psi h)+\mathrm{i}\sin(\psi h)$, equation (33) can be rewritten as

$$\xi=\frac{(P_2+Q_2)\cos(\psi h)+R+i(Q_2-P_2)\sin(\psi h)}{(P_1+Q_1)\cos(\psi h)+R+i(Q_1-P_1)\sin(\psi h)}. \tag{34}$$

This equation can be rewritten as

$$\xi=\frac{[4(\alpha-\delta\mu)\cos(\psi h)+2(1+2\delta\mu)]+\mathrm{i}[2k(\gamma+\varepsilon q)\sin(\psi h)]}{[4(\alpha-\delta\mu)\cos(\psi h)+2(1+2\delta\mu)]-\mathrm{i}[2k(\gamma+\varepsilon q^*)\sin(\psi h)]}. \tag{35}$$

For stability, we must have $|\xi|\le 1$, otherwise ξ^j in (31) would grow in an unbounded manner, that is,

$$|\xi|^2=\frac{[4(\alpha-\delta\mu)\cos(\psi h)+2(1+2\delta\mu)]^2+[2k(\gamma+\varepsilon q)\sin(\psi h)]^2}{[4(\alpha-\delta\mu)\cos(\psi h)+2(1+2\delta\mu)]^2+[2k(\gamma+\varepsilon q^*)\sin(\psi h)]^2}. \tag{36}$$

Since the numerator in Eq. (36) is smaller than the denominator, $|\xi|\le 1$, and this method is unconditionally stable, it means that there is no restriction on the grid size, i.e., on h and k, but we should choose them in such a way that the accuracy of the scheme is not degraded.

5. Error analysis and order of convergence

In this section, we prove that the exponential B-spline collocation method has uniform convergence of order two in the spatial domain for our problem. For the derivation of uniform convergence we first prove the following lemma.

Lemma 2 The exponential B-splines $\{B_{-1},B_0,\ldots,B_{n+1}\}$ defined in Eq. (4) satisfy

$$\sum_{i=-1}^{n+1}|B_i(x)|\le\frac{5}{2},\quad a\le x\le b.$$

Proof See Ref. [71] for proof.

Theorem 3 Let $u(x,t)$ be the collocation approximation from the space of exponential B-splines to the solution $U(x,t)$ of the GRLW equation problem (5)–(7). Then the uniform error estimate is given by

$$\|U(x,t)-u(x,t)\|_\infty\le\Lambda h^2,$$

where Λ is a positive constant independent of h.

Proof Let $u(x,t)$ be the unique exponential spline collocation approximate solution of Eqs. (5)–(7), which is given by

$$u(x,t)=\sum_{i=-1}^{n+1}\varsigma_i(t)B_i(x), \tag{37}$$

and let $\tilde{u}(x,t)$ be the unique exponential spline interpolation from the space of exponential B-splines to the exact solution $U(x,t)$ of Eqs. (5)–(7), which is given by

$$\tilde{u}(x,t)=\sum_{i=-1}^{n+1}\tilde{\varsigma}_i(t)B_i(x). \tag{38}$$

From Eq. (29) for $u(x,t)$ and $\tilde{u}(x,t)$ we have

$$\begin{aligned}&\mathscr{A}\mathscr{V}^{j+1}+\frac{k}{2}\mathscr{F}^{j+1}=\mathscr{G}^j+T, \quad j=0,1,\ldots,m,\\ &\mathscr{A}\tilde{\mathscr{V}}^{j+1}+\frac{k}{2}\tilde{\mathscr{F}}^{j+1}=\mathscr{G}^j, \qquad j=0,1,\ldots,m,\end{aligned} \tag{39}$$

where $T=\mathscr{O}(h^2)$, $\tilde{\mathscr{V}}^{j+1}=\left(\tilde{\varsigma}_0^{j+1},\tilde{\varsigma}_1^{j+1},\ldots,\tilde{\varsigma}_n^{j+1}\right)^t$, and

$$\tilde{\mathscr{F}}^{j+1}=\begin{pmatrix} F\left(f_0(t_{j+1}),\dfrac{\beta}{\alpha}\left(f_0(t_{j+1})-\tilde{\varsigma}_0^{j+1}\right)-2\beta\tilde{\varsigma}_1^{j+1}\right)\\ F\left(\left(\alpha\tilde{\varsigma}_0^{j+1}+\tilde{\varsigma}_1^{j+1}+\alpha\tilde{\varsigma}_2^{j+1}\right),\beta\left(\tilde{\varsigma}_0^{j+1}-\tilde{\varsigma}_2^{j+1}\right)\right)\\ \vdots\\ F\left(\left(\alpha\tilde{\varsigma}_{n-2}^{j+1}+\tilde{\varsigma}_{n-1}^{j+1}+\alpha\tilde{\varsigma}_n^{j+1}\right),\beta\left(\tilde{\varsigma}_{n-2}^{j+1}-\tilde{\varsigma}_n^{j+1}\right)\right)\\ F\left(f_1(t_{j+1}),2\beta\tilde{\varsigma}_{n-1}^{j+1}-\dfrac{\beta}{\alpha}\left(f_1(t_{j+1})-\tilde{\varsigma}_n^{j+1}\right)\right)\end{pmatrix}.$$

From Eq. (39) we obtain

$$\begin{aligned}&\mathscr{A}\left(\mathscr{V}^{j+1}-\tilde{\mathscr{V}}^{j+1}\right)+\frac{k}{2}\left(\mathscr{F}^{j+1}-\tilde{\mathscr{F}}^{j+1}\right)=T,\\ &j=0,1,\ldots,m.\end{aligned} \tag{40}$$

Now we consider the nonlinear term in the left hand side of Eq. (40). Following Ref. [77] and [78] and using the mean-value theorem, we have

$$\begin{aligned}&\mathscr{F}^{j+1}-\tilde{\mathscr{F}}^{j+1}\\ &=\left[\mathscr{M}_0\left(\frac{\partial F}{\partial u}\right)(\xi_1)+\mathscr{M}_1\left(\frac{\partial F}{\partial u_x}\right)(\xi_2)\right]\left(\mathscr{V}^{j+1}-\tilde{\mathscr{V}}^{j+1}\right),\end{aligned} \tag{41}$$

where $\xi_1,\xi_2\in(a,b)$,

$$\mathscr{M}_0=\begin{pmatrix} 0 & 0 & & & \\ \alpha & 1 & \alpha & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha & 1 & \alpha\\ & & & 0 & 0\end{pmatrix}$$

and

$$\mathscr{M}_1=\begin{pmatrix} -\dfrac{\beta}{\alpha} & -2\beta & & & \\ \beta & 0 & -\beta & & \\ & \ddots & \ddots & \ddots & \\ & & \beta & 0 & -\beta\\ & & & 2\beta & \dfrac{\beta}{\alpha}\end{pmatrix}.$$

From Eqs. (40) and (41), we have

$$\mathfrak{D}\left(\mathscr{V}^{j+1}-\tilde{\mathscr{V}}^{j+1}\right)=T, \tag{42}$$

where

$$\mathfrak{D}=\mathscr{A}+\frac{k}{2}\left[\mathscr{M}_0\left(\frac{\partial F}{\partial u}\right)(\xi_1)+\mathscr{M}_1\left(\frac{\partial F}{\partial u_x}\right)(\xi_2)\right].$$

It is easy to see that for sufficiently small h, $\mathfrak{D}$ is a monotone matrix and hence invertible, so from Eq. (42) we have

$$\left(\mathscr{V}^{j+1}-\tilde{\mathscr{V}}^{j+1}\right)=\mathfrak{D}^{-1}T. \tag{43}$$

Using the infinity norm yields

$$\|\left(\mathscr{V}^{j+1}-\tilde{\mathscr{V}}^{j+1}\right\|_\infty\le\|\mathfrak{D}^{-1}\|_\infty\|T\|_\infty. \tag{44}$$

Clearly, the row sums $\mathfrak{S}_0,\mathfrak{S}_1,\ldots,\mathfrak{S}_n$ of $\mathfrak{D}$ are

$$\begin{aligned}\mathfrak{S}_0&=\sum_{j=0}^{n}\mathfrak{D}_{0,j}=\left(2+\frac{1}{\alpha}\right)\left[\delta\mu-\frac{k\beta\gamma}{2}-\beta\frac{\partial F}{\partial u_x}\right],\\ \mathfrak{S}_i&=\sum_{j=0}^{n}\mathfrak{D}_{i,j}=(2\alpha+1)(1+\frac{\partial F}{\partial u}),\\ &\quad i=1,2,\ldots,n-1,\\ \mathfrak{S}_n&=\sum_{j=0}^{n}\mathfrak{D}_{n,j}=\left(2+\frac{1}{\alpha}\right)\left[\delta\mu+\frac{k\beta\gamma}{2}+\beta\frac{\partial F}{\partial u_x}\right].\end{aligned}$$

Moreover, from the theory of matrices, we have

$$\sum_{i=0}^{n}\mathfrak{D}_{k,i}^{-1}\mathfrak{S}_i=1, \quad k=0,1,\ldots,n, \tag{45}$$

where $\mathfrak{D}_{k,i}^{-1}$ is the (k,i)-th element of the matrix $\mathscr{D}^{-1}$. Therefore,

$$\|\mathfrak{D}^{-1}\|_\infty=\sum_{i=0}^{n}|\mathfrak{D}_{k,i}^{-1}|\le\frac{1}{\mathfrak{S}}\le\frac{1}{|\mathfrak{S}|}, \tag{46}$$

where

$$\mathfrak{S}=\min\{\mathfrak{S}_0,\mathfrak{S}_1,\ldots,\mathfrak{S}_n\}.$$

Also following Corollary 2.3, there exists a constant χ such that $\|T\|_\infty\le\chi h^2$. Now by substituting Eq. (46) into Eq. (44), we have

$$\left\|\left(\mathscr{V}^{j+1}-\tilde{\mathscr{V}}^{j+1}\right\|_\infty\le\frac{\chi h^2}{|\mathfrak{S}|}. \tag{47}$$

Moreover, as

$$u(x,t)-\tilde{u}(x,t)=\sum_{i=-1}^{n+1}\big(\varsigma_i(t)-\tilde{\varsigma}_i(t)\big)B_i(x). \tag{48}$$

Thus, combining Eqs. (47) and (48) and Lemma 2, we have

$$\begin{aligned}&\|u(x,t)-\tilde{u}(x,t)\|_\infty\\ &=\left\|\sum_{i=-1}^{n+1}\big(\varsigma_i(t)-\tilde{\varsigma}_i(t)\big)B_i(x)\right\|_\infty\\ &\le\left\|\big(\mathscr{V}^{j+1}-\tilde{\mathscr{V}}^{j+1}\big)\right\|_\infty\sum_{i=-1}^{n+1}|B_i(x)|\le\frac{5\chi h^2}{2|\mathfrak{S}|}.\end{aligned} \tag{49}$$

From Corollary 1, we deduce that

$$\|U(x,t)-\tilde{u}(x,t)\|_\infty\le\chi_0 h^4. \tag{50}$$

Generally, we have $\|U(x,t)-u(x,t)\|_\infty=\|U(x,t)-\tilde{u}(x,t)\|_\infty+\|\tilde{u}(x,t)-u(x,t)\|_\infty$. According to the above notations we deduce the stated result as

$$\|U(x,t)-u(x,t)\|_\infty\le\Lambda h^2, \tag{51}$$

where $\Lambda=\frac{5\chi}{2|\mathfrak{S}|}+\chi_0 h^2$. Combining the results, we have proved the theorem.

Theorem 4 Let $U(x,t)$ be the solution of the initial boundary value problem (5)–(7) and $u(x,t)$ be the collocation approximation to the solution $U(x,t)$ after the temporal discretization stage. Then under the hypothesis of Lemma 1 and Theorem 3 the error estimate of the totally discrete scheme is given by

$$\big\|U(x,t)-u(x,t)\big\|_\infty\le\mathfrak{C}(k^2+h^2), \tag{52}$$

where $\mathfrak{C}$ are positive constants.

6. Numerical examples

To illustrate the effectiveness of the exponential spline collocation method (29), numerical results portraying a single soliton solution and the collision of two solitons are reported for different GRLW equations. The numerical results obtained are compared with the analytical solution, each providing corroborative evidence of the findings of the other. The accuracy of the scheme is measured by using the following error norms:

$$\big\|U(x,t)-u(x,t)\big\|_2=\sqrt{h\sum_{i=1}^{n-1}\big|U(x_i,t)-u(x_i,t)\big|^2},$$

and

$$\big\|U(x,t)-u(x,t)\big\|_\infty=\max_{1\le i\le n-1}|U(x_i,t)-u(x_i,t)|,$$

where $U(x,t)$ and $u(x,t)$ represent the exact and approximate solutions, respectively.

Moreover, the quantities $\mathscr{I}_1$, $\mathscr{I}_2$, and $\mathscr{I}_3$ are applied to measure the conservation properties of the derived scheme, calculated by

$$\mathscr{I}_1=h\sum_{i=1}^{n-1}u_i^j,\quad \mathscr{I}_2=h\sum_{i=1}^{n-1}\left[(u_i^j)^2+\mu\big((u_x)_i^j\big)^2\right],$$
$$\mathscr{I}_3=h\sum_{i=1}^{n-1}\left[(u_i^j)^3+3(u_i^j)^2\right],$$

for the RLW equation and in a similar manner for the MRLW equation as

$$\mathscr{I}_1=h\sum_{i=1}^{n-1}u_i^j,\quad \mathscr{I}_2=h\sum_{i=1}^{n-1}\left[(u_i^j)^2+\mu\big((u_x)_i^j\big)^2\right],$$
$$\mathscr{I}_3=h\sum_{i=1}^{n-1}\left[(u_i^j)^4-6\mu\big((u_x)_i^j\big)^2\right].$$

The analytical solution of the GRLW equation is given by

$$u(x,t)=\sqrt[p]{\mathscr{D}\operatorname{sech}^2[\mathscr{K}\big(x-(1+\varepsilon d)t-x_0\big)]},$$

where

$$\mathscr{D}=\frac{(p+1)(p+2)d}{2},\quad \mathscr{K}=\frac{p}{2\sqrt{\mu}}\sqrt{\frac{\varepsilon d}{1+\varepsilon d}},$$

which corresponds to a solitary wave of amplitude $\sqrt[p]{\mathscr{D}}$, speed $(1+\varepsilon d)$, width $\mathscr{K}$, and initially centered at x_0. The interaction of solitons for the RLW, MRLW, and GRLW equations are examined. All experiments are performed by taking the parameters $\gamma=1$, $\mu=1$, and $\varepsilon=1$. In all computations we take $\rho=0.01$ which gives the optimum solutions. All calculations are implemented by some codes produced in Mathematica 8.0 and Matlab 7.0 softwares on an Intel (R) Core(TM) i7-2640M, 2.80 GHz CPU machine with 6 GB of memory.

6.1. Motion of single solitary waves

6.1.1. RLW equation

The RLW equation in the regions $[-40,60]$ and $[-80,120]$ for $d=0.1$ and $d=0.03$ was integrated from $t=0$ to 20 with $x_0=0$ and different values of k and h. The L_∞ and L_2 errors and invariants $\mathscr{I}_1$, $\mathscr{I}_2$, and $\mathscr{I}_3$, and the CPU times needed to reach different times are presented in Tables 1 and 2 for different times. The analytic values of the invariant quantities are

$$\mathscr{I}_1=\frac{6d}{K},\quad \mathscr{I}_2=\frac{12d^2}{K}+\frac{48\mu Kd^2}{5},\quad \mathscr{I}_3=\frac{36d^2}{K}+\frac{144d^3}{5K}.$$

Throughout the simulation, L_2 and L_∞ error norms remain less than 7.950152×10^{-5} and 4.207305×10^{-5} (for $d=0.1$), and 1.157133×10^{-5} and 4.174280×10^{-6} (for $d=0.03$). In comparison with the analytical values, it can be seen from Tables 1 and 2 that the quantities $\mathscr{I}_1$, $\mathscr{I}_2$, and $\mathscr{I}_3$ obtained by the presented method are very close to the corresponding analytical values. From these Tables, it can be seen that the conservation properties (using the exponential spline collocation method), in comparison with the analytical values for the invariants, for

$d = 0.1$ and $d = 0.03$, are very good. This degree of conservation is superior to that found with other existing methods listed (see Tables 1 and 2). A comparison with other existing methods,[19,23,27–30,32–34,36,38,39,41,47] shows that the error norms at time $t = 20$, found with the present method, are smaller than those obtained with those methods (see Tables 1 and 2). Moreover, it can be seen from Tables 1 and 2 that the exponential spline collocation method produced noticeable reductions in error when the time step is decreased. By increasing the space and the time steps, we also obtain results that are reasonably accurate, within a few seconds of CPU time. The profile of the solitary waves for the RLW equation with $d = 0.1$ obtained with the presented method, from times $t = 0$ to 20, are depicted in Fig. 1.

Table 1. Errors and invariants for single soliton of the RLW equation with $d = 0.1$, through the interval $[-40, 60]$ at time $t = 20$.

Methods	h	k	L_2	L_∞	$\mathscr{I}_1$	$\mathscr{I}_2$	$\mathscr{I}_3$	CPU ime
Exact solution			0	0	3.979908	0.81046249	2.57900743	—
Our method	0.125	0.1	2.644246(−5)	7.743492(−6)	3.979908	0.81046248	2.57900744	—
PACS[23]	0.125	0.1	4.8039(−4)	1.8296(−4)	3.979884	0.81046247	2.57900727	—
LSFE[27]	0.125	0.1	4.688(−3)	1.755(−3)	3.98203	0.80865	2.57302	—
CBSC 1[28]	0.125	0.1	2.6086(−4)	1.0299(−4)	3.979958	0.8104596	2.578999	—
CBSC 2[28]	0.125	0.1	2.2050(−4)	8.4480(−5)	3.980016	0.8104624	2.519006	—
QBSC[28]	0.125	0.1	4.3150(−5)	1.3210(−5)	3.979890	0.8104625	2.578999	—
LG[29]	0.125	0.1	2.19(−4)	8.6(−5)	3.97988	0.810465	2.57901	—
QBSG 1[30]	0.125	0.1	1.9215(−4)	7.337(−5)	3.979883	0.8104612	2.5790031	—
QBSG 2[30]	0.125	0.1	3.5489(−4)	1.2848(−4)	3.979883	0.8104616	2.5790043	—
QBSG[32]	0.125	0.1	1.92(−4)	7.3(−5)	3.97989	0.81046	2.57901	—
QBSFE[33]	0.125	0.1	3.7841(−4)	1.3993(−4)	3.97995	0.81046	2.57900	—
FEG[34]	0.125	0.1	5.11(−4)	1.98(−4)	3.98206	0.81116	2.58133	—
MQ[38]	0.125	0.1	2.0691(−4)	7.8027(−5)	3.979883	0.81046248	2.5790074	—
MQ[39]	0.125	0.1	2.0691(−4)	7.8027(−5)	3.979883	0.81046248	2.5790074	—
RBF[41]	0.125	0.1	2.2883(−4)	8.8006(−5)	3.9799497	0.8104625	2.57900744	—
NPS[47]	0.125	0.1	4.0198(−5)	1.2764(−5)	3.979908	0.81046252	2.57900744	—
GFE[36]	0.125	0.1	2.6685(−4)	9.1465(−5)	3.97972	0.81026	2.57873	—
Our method	0.78125	0.1	7.950152(−5)	4.207305(−5)	3.979912	0.81046236	2.5790073	1.762 s
	0.78125	0.05	1.011574(−5)	8.478383(−6)	3.979912	0.81046236	2.5790073	3.715 s
	0.78125	0.01	8.788790(−7)	4.974574(−7)	3.979911	0.81046239	2.5790073	—
	0.78125	0.001	7.462728(−7)	1.638867(−7)	3.979911	0.81046239	2.5790073	—
SC[19]	0.78125	0.1	1.797446(−4)	6.799314(−5)	3.979913	0.810462	2.579007	2.032 s
	0.78125	0.05	4.172751(−5)	1.953916(−5)	3.979914	0.810462	2.579007	4.062 s
	0.78125	0.01	2.704408(−6)	8.254178(−6)	—	—	—	—
	0.78125	0.001	1.742781(−6)	5.759150(−7)	—	—	—	—

Table 2. Errors and invariants for single soliton of RLW equation with $d = 0.03$.

Methods	Time	k	$L_2^{\rm a}$	$L_\infty^{\rm b}$	$\mathscr{I}_1^{\rm c}$	$\mathscr{I}_2^{\rm c}$	$\mathscr{I}_3^{\rm c}$	CPU time
Analytical			0	0	2.10704681	0.12730126	0.38880465	—
Our method	4	0.1	3.140597(−6)	1.117629(−7)	2.107037043	0.12730136	0.38880553	—
	12		9.382404(−6)	1.152157(−7)	2.1070186536	0.12730143	0.38880575	—
	20		1.157133(−5)	4.174280(−6)	2.1070354545	0.12730152	0.38880426	—
PACS[23]	4		4.2144(−4)	2.3487(−4)	2.10705516	0.12730109	0.38880398	—
	12		5.4751(−4)	2.1702(−4)	2.10651666	0.12730111	0.38880389	—
	20		5.7616(−4)	4.2242(−4)	2.10461152	0.12730154	0.38880229	—
GFE[36]	4		1.5653(−4)	1.8971(−4)	2.10842	0.12730	0.38880	—
	12		4.1011(−4)	4.1546(−4)	2.10985	0.12730	0.38880	—
	20		5.5940(−4)	4.3914(−4)	2.10902	0.12730	0.38880	—
RBF[41]	20		1.4342(−5)	4.3820(−6)	2.1094022	0.1273017	0.3888059	—
		k	$L_2^{\rm a}$	$L_\infty^{\rm d}$	$\mathscr{I}_1^{\rm e}$	$\mathscr{I}_2^{\rm e}$	$\mathscr{I}_3^{\rm e}$	CPU time
Our method	4	0.1	1.718704(−6)	2.815130(−7)	2.10820712	0.12730004	0.388803	0.325 s
	12		4.399752(−6)	6.458785(−7)	2.10856775	0.12730011	0.388803	1.002 s
	20		7.137599(−6)	1.588104(−6)	2.10697761	0.12729965	0.388803	1.787 s
SC[19]	4		2.748506(−6)	7.735109(−7)	2.109406	0.127302	0.388806	0.437 s
	12		8.230172(−6)	2.372020(−6)	2.109408	0.127302	0.388806	1.234 s
	20		1.367793(−5)	3.728636(−6)	2.109408	0.127302	0.388806	2.032 s
Our method	4	0.05	1.568117(−7)	8.242623(−8)	2.10820710	0.12730004	0.388803	0.639 s
	12		3.846237(−7)	5.301098(−7)	2.10856774	0.12730011	0.388803	2.268 s
	20		6.194377(−7)	3.808634(−7)	2.10697774	0.12729965	0.388803	3.741 s
SC[19]	4		6.967258(−7)	1.933155(−7)	2.109406	0.127302	0.388806	0.844 s
	12		2.070286(−6)	5.929557(−7)	2.109408	0.127302	0.388806	2.543 s
	20		3.427579(−6)	9.007022(−7)	2.109408	0.127302	0.388806	4.063 s

[a]Through, [b]$[-40, 60]$, [c]$n = 800$, [d]$[-80, 120]$, [e]$n = 128$.

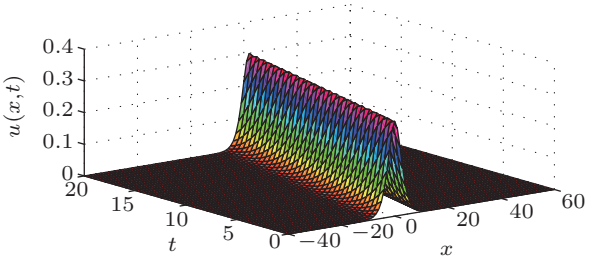


Fig. 1. (color online) Space–time graph of the single solitary wave up to $t = 20$ s, with $d = 0.1$, $k = 0.1$, and $n = 129$, for RLW equation, through the interval $[-40, 60]$.

6.1.2. MRLW equation

Now we study the motion of a single solitary wave of the MRLW equation. For the sake of comparison with other existing methods, we take $d = 1$, 0.3, and $x_0 = 40$ in the region $0 \le x \le 100$. The L_∞ and L_2 errors and invariants $\mathscr{I}_1$, $\mathscr{I}_2$, and $\mathscr{I}_3$ are obtained in Table 3 for times $t = 10$ and 20. The analytic values of the invariant quantities are

$$\mathscr{I}_1 = \frac{\pi\sqrt{d}}{K}, \quad \mathscr{I}_2 = \frac{2d}{K} + \frac{2\mu K d}{3}, \quad \mathscr{I}_3 = \frac{4d^2}{3K} - \frac{2\mu K d}{3}.$$

Throughout the simulation the L_2 and L_∞ error norms remain less than 2.3909×10^{-3} and 1.0647×10^{-3} (for $d = 1$), and 1.3378×10^{-4} and 6.1250×10^{-4} (for $d = 0.3$). A comparison with the methods in Refs. [22], [48], [50], [52], [53], [56], and [62] shows that the error norms at times $t = 10$ and 20, obtained with the present method, are smaller than those obtained with other existing methods (see Table 3). From this Table, it can be seen that the conservation properties (using the exponential spline collocation method) are very good. In comparison with the analytical values for the invariants, for $d = 1$ and $d = 0.3$, the relative changes in the invariants are very small. This degree of conservation is superior to that found with the other methods listed in Table 3. The profile of the solitary waves for the MRLW equation with $d = 0.3$, from time $t = 0$ to 20, are depicted in Fig. 2.

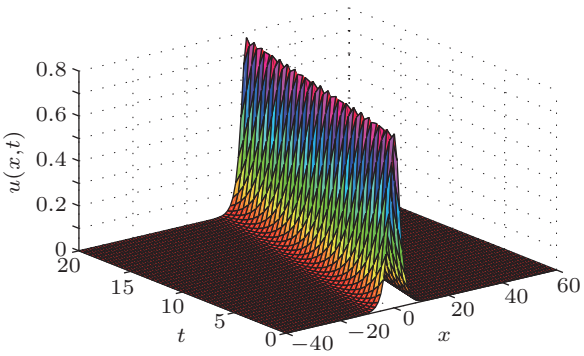


Fig. 2. (color online) Space–time graph of the single solitary wave up to $t = 20$ s, with $d = 0.3$, $k = 0.01$, and $h = 0.1$, for the MRLW equation, through the interval $[0, 100]$ for $x_0 = 40$.

Table 3. Invariants and errors for single soliton of the MRLW equation for $x_0 = 40$, $0 \le x \le 100$, and different values of d, t, h and k.

Methods	h	k	L_2	L_∞	$\mathscr{I}_1$	$\mathscr{I}_2$	$\mathscr{I}_3$
$d = 1, t = 10$	0.2	0.025					
Analytical			0	0	4.4428829	3.2998316	1.4142135
Our method			2.3909(−3)	1.0647(−3)	4.4428827	3.2998042	1.4142107
PGFE[62]			2.4155(−3)	1.0797(−3)	4.4431758	3.3003023	1.4146927
PG[22]			3.0053(−3)	1.6874(−3)	4.44288	3.29981	1.41416
CBSC-CN[48]			1.639(−2)	9.24(−3)	4.442	3.299	1.413
CBSC-PA-CN[48]			2.03(−2)	1.12(−2)	4.440	3.296	1.411
CBSC[50]			9.3019(−3)	5.4371(−3)	4.44288	3.29983	1.41420
MQ[52]			3.914(−3)	2.019(−3)	4.4428829	3.29978	1.414163
$d = 1, t = 10$	0.1	0.025					
Our method			1.9447(−4)	2.0047(−4)	4.44249	3.29980	1.41419
QBS[53]			8.644(−4)	1.2475(−3)	4.445176	3.302476	1.417411
M1[48]			1.939(−2)	9.24(−3)	4.442	9.299	1.413
M3[50]			9.3019(−3)	5.4371(−3)	4.44288	3.29983	1.41420
$d = 0.3, t = 10$	0.2	0.1					
Analytical			0	0	3.5819667	1.3450764	0.153723
Our method			1.3378(−4)	6.1250(−4)	3.5812102	1.345133	0.1538342
QBS[53]			3.3795(−4)	7.6729(−4)	3.582265	1.345182	0.1538901
$d = 0.3, t = 20$	0.1	0.01					
Our method			5.3488(−5)	3.1778(−5)	3.5819987	1.3450863	0.1537246
PGFE[62]			8.1125(−5)	3.5690(−5)	3.5820204	1.3450974	0.1537250
M3[50]			6.0688(−4)	2.9665(−4)	3.58197	1.34508	0.153723
QBS[56]			5.0892(−5)	2.2228(−5)	3.581967	1.345076	0.153723
IQ[52]			5.1498(−5)	2.2551(−5)	3.5819654	1.3450764	0.153723

6.1.3. GRLW equation

For the purpose of illustration of the presented method for solving the GRLW equation, we consider the case for $p \ge 3$ in $[-60, 60]$ for $x_0 = 10$. In this test, calculations are carried out with $d = 1.01$, $k = 0.1$, $h = 0.48$, and $p = 4, 6, 8, 10$ when the final time is $t = 10$. Errors in L_2 and L_∞ norms at several times are listed in Table 4 and compared with results in Ref. [21]. The plot of the estimated solution with $d = 0.3$ and $p = 10$,

from time $t = 0$ to 20, are shown in Fig. 3. Throughout the simulation the L_2 and L_∞ error norms remain less than 1.8×10^{-3} and 8.64×10^{-4} for the GRLW equation with different values of p. One can observe from Table 4 that the numerical results calculated with the present method are more accurate in comparison with Ref. [21].

Table 4. Values of the error norms at different values of t and p for single soliton of the GRLW equation.

Methods	Error norm	$t = 2$	$t = 4$	$t = 6$	$t = 8$	$t = 10$
$p = 4$						
Our method	L_2	2.3(−5)	4.4(−5)	6.5(−5)	8.5(−5)	1.0(−4)
	L_∞	1.6(−6)	2.7(−6)	2.5(−6)	2.6(−6)	2.4(−6)
Feng *et al.*[21]	L_2	1.6(−4)	3.0(−4)	4.0(−4)	4.6(−4)	5.0(−4)
	L_∞	4.0(−5)	7.0(−5)	9.0(−5)	9.0(−5)	1.0(−4)
$p = 6$						
Our method	L_2	4.1(−5)	8.2(−5)	1.2(−4)	1.5(−4)	1.8(−4)
	L_∞	1.2(−5)	2.1(−5)	2.0(−5)	2.3(−5)	2.9(−5)
Feng *et al.*[21]	L_2	3.2(−4)	6.1(−4)	8.4(−4)	1.0(−3)	1.1(−3)
	L_∞	1.2(−4)	2.1(−4)	2.7(−4)	3.2(−4)	3.7(−4)
$p = 8$						
Our method	L_2	3.8(−4)	8.1(−4)	1.2(−3)	1.5(−3)	1.8(−3)
	L_∞	1.1(−4)	1.8(−4)	1.7(−4)	1.9(−4)	2.4(−4)
Feng *et al.*[21]	L_2	6.3(−4)	1.1(−3)	1.6(−3)	1.9(−3)	2.2(−3)
	L_∞	2.9(−4)	4.8(−4)	6.1(−4)	7.0(−4)	7.8(−4)
$p = 10$						
Our method	L_2	8.56(−4)	1.61(−3)	1.42(−3)	1.16(−3)	1.14(−3)
	L_∞	1.02(−4)	1.70(−4)	8.64(−4)	6.31(−4)	4.61(−4)
Feng *et al.*[21]	L_2	1.09(−3)	1.97(−3)	2.64(−3)	3.18(−3)	3.65(−3)
	L_∞	5.5(−4)	8.5(−4)	1.07(−3)	1.23(−3)	1.36(−3)

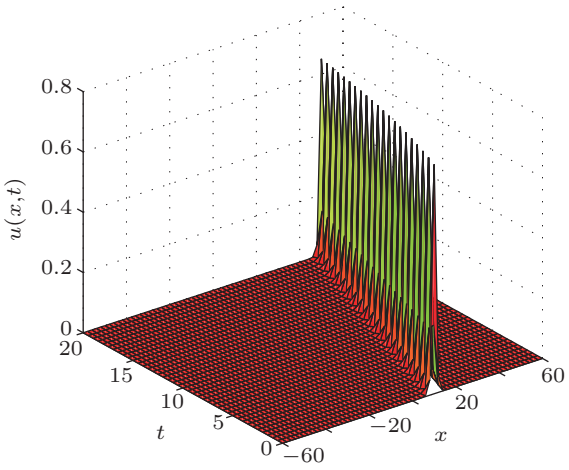


Fig. 3. (color online) Space–time graph of the single solitary wave up to $t = 20$ s, with $d = 1.01$, $p = 10$, $k = 0.1$, and $h = 0.48$, for the GRLW equation, through the interval $[-60, 60]$ for $x_0 = 10$.

6.2. Interaction of two solitary waves

The interaction of two GRLW solitary waves having different amplitudes and traveling in the same direction is illustrated. We consider the GRLW equation with the initial conditions given by the linear sum of two well separated solitary waves of various amplitudes

$$u(x,0) = \sum_{i=1}^{2} \sqrt[p]{\mathscr{D}_i \operatorname{sech}^2[\mathscr{K}_i(x - x_i)]},$$

where

$$\mathscr{D}_i = \frac{(p+1)(p+2)d_i}{2}, \quad \mathscr{K}_i = \frac{p}{2\sqrt{\mu}}\sqrt{\frac{\varepsilon d_i}{1+\varepsilon d_i}}.$$

To allow the interaction to occur, the experiment was run from $t = 0$ to 10 and in the region $-150 \le x \le 100$ for $p = 2, 4, 6, 8$ with $d_1 = 1.005$, $d_2 = 1.01$, $x_1 = 40$, $x_2 = -30$, $h = 0.625$, and $k = 0.05$. Errors in L_2 and L_∞ norms at several times are listed in Table 5 and compared with results in Ref. [21]. Figure 4 shows the interaction of two soliton solutions of the GRLW equation with $p = 8$. Throughout the simulation the L_2 and L_∞ error norms remain less than 1.6×10^{-3} and 8.4×10^{-4} for the GRLW equation with different values of p. A comparison with the method used in Ref. [21] shows that the error norms found with the present method are smaller than those obtained with that method.

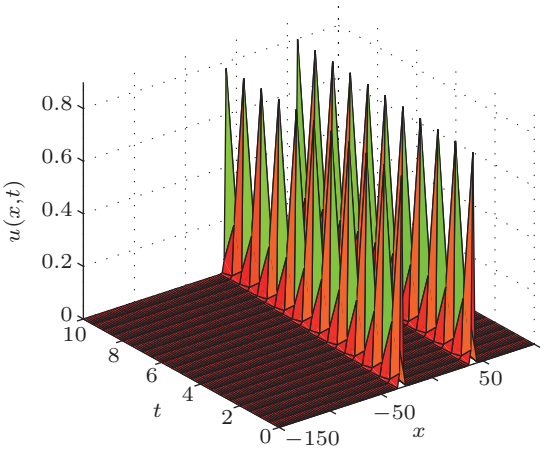


Fig. 4. (color online) Space–time graph of the single solitary wave up to $t = 10$ s, with $d_1 = 1.005$, $d_2 = 1.01$, $x_1 = 40$, $x_2 = -30$, $p = 8$, $k = 0.05$, and $h = 0.625$, for the GRLW equation, through the interval $[-150, 100]$.

Table 5. Values of the error norms at different values of t and p for interaction of two solitary solutions of the GRLW equation.

Methods	Error norm	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
$p=2$						
Our method	L_2	1.6(−4)	3.0(−4)	4.0(−4)	4.7(−4)	5.1(−4)
	L_∞	1.0(−5)	1.3(−5)	2.5(−5)	4.6(−5)	7.8(−5)
Feng *et al.*[21]	L_2	3.3(−4)	5.1(−4)	7.0(−4)	8.6(−4)	1.0(−3)
	L_∞	5.0(−5)	1.1(−4)	1.7(−4)	2.1(−4)	2.3(−4)
$p=4$						
Our method	L_2	1.9(−4)	3.6(−4)	5.2(−4)	6.8(−4)	8.3(−4)
	L_∞	2.0(−5)	3.3(−5)	4.0(−5)	5.3(−5)	7.4(−5)
Feng *et al.*[21]	L_2	8.7(−4)	1.3(−3)	1.9(−3)	2.3(−3)	2.8(−3)
	L_∞	1.2(−4)	2.7(−4)	4.1(−4)	5.2(−4)	5.6(−4)
$p=6$						
Our method	L_2	3.0(−4)	5.9(−4)	8.8(−4)	1.1(−3)	1.4(−3)
	L_∞	2.3(−5)	3.8(−5)	4.6(−5)	5.6(−5)	8.4(−4)
Feng *et al.*[21]	L_2	1.3(−3)	2.3(−3)	3.3(−3)	4.2(−3)	5.1(−3)
	L_∞	3.5(−4)	6.6(−4)	9.9(−4)	1.2(−3)	1.5(−3)
$p=8$						
Our method	L_2	3.2(−4)	6.5(−4)	1.0(−3)	1.3(−3)	1.6(−3)
	L_∞	2.4(−5)	4.5(−5)	9.4(−5)	1.4(−4)	3.5(−4)
Feng *et al.*[21]	L_2	2.0(−3)	3.8(−3)	5.5(−3)	7.2(−3)	8.9(−3)
	L_∞	7.8(−4)	1.4(−4)	2.0(−3)	2.6(−3)	3.1(−3)

7. Conclusion

A numerical technique based on a collocation method using exponential B-splines has been presented for the numerical solution of the GRLW equation. The efficiency of the method is tested for the problems of propagation of the single solitary wave, interaction of two solitary waves, and development of an undular bore. The accuracy is examined in terms of the L_∞ and L_2 error norms and the conservation quantities $\mathscr{I}_1$, $\mathscr{I}_2$, and $\mathscr{I}_3$. Stability analysis is performed by the Von Neumann method and we prove that the new method is unconditionally stable. A convergence analysis for this method is given. The observed errors and invariants are tabulated in Tables and the stability of our presented schemes is verified. These computational results show that our proposed algorithm is effective and accurate in comparison with other existing methods in the literature. The collocation method with exponential B-spline functions provides high accuracy and invariance of the conserved quantities. In addition, the CPU run time of our method is better than for that of the Sinc-collocation method. The superiority of the present method over the previous ones is clear.

References

[1] Feng Z, Wang X D and Ouyang J 2012 *Acta Phys. Sin.* **61** 230204 (in Chinese)
[2] Cai W J, Wang Y S and Song Y Z 2014 *Chin. Phys. Lett.* **31** 040201
[3] Wang X, Ma T B and Ning J G 2014 *Chin. Phys. Lett.* **31** 030201
[4] Ge H X and Cheng R J 2014 *Chin. Phys. B* **23** 040203
[5] Yu X, Ren Z G and Xu C 2014 *Chin. Phys. B* **23** 040201
[6] Peregrine D H 1966 *J. Fluid Mech.* **25** 321
[7] Lin J, Xie Z and Zhou J 2007 *Commun. Numer. Meth. Eng.* **23** 135
[8] Saka B and Dag I 2008 *Commun. Numer. Meth. Eng.* **24** 1339
[9] Abdulloev K H O, Bogalubsky H and Markhankov V G 1976 *Phys. Lett. A* **56** 427
[10] Peregrine D H 1967 *J. Fluid Mech.* **27** 815
[11] Bona J L, McKinney W R and Restrepo J M 2000 *J. Nonlinear Sci.* **10** 603
[12] Durán A and López-Marcos M A 2002 *Appl. Numer. Math.* **42** 95
[13] Djidjeli K, Price W G, Twizell E H and Cao Q 2003 *Commun. Numer. Meth. Eng.* **19** 847
[14] Kaya D and Syed E 2003 *Chaos Soliton. Fract.* **17** 869
[15] Kaya D 2004 *Appl. Math. Comput.* **149** 833
[16] Zhang L 2005 *Appl. Math. Comput.* **168** 962
[17] Xie S, Kim S, Woo G and Yi S 2008 *SIAM J. Sci. Comput.* **30** 2263
[18] Jafari H, Borhanifar A and Karimi S A 2010 *Int. J. Comput. Math.* **87** 509
[19] Mokhtari R and Mohammadi M 2010 *Comput. Phys. Commun.* **181** 1266
[20] Mohammadi M and Mokhtari R 2011 *J. Comput. Appl. Math.* **235** 4003
[21] Wang J F, Bai F N and Cheng Y M 2011 *Chin. Phys. B* **20** 030206
[22] Roshan T 2012 *Comput. Math. Appl.* **63** 943
[23] Lin B A 2014 *Stud. Appl. Math.* **132** 160
[24] Benjamin T B, Bona J L and Mahony J J 1972 *Trans. R. Soc. Lond. Ser. A* **272** 47
[25] Dağ I 2000 *Comput. Method. Appl. Mech. Eng.* **182** 205
[26] Dağ I and Özer M N 2001 *Appl. Math. Model.* **25** 221
[27] Gardner L R T, Gardner G A and Doğan A 1996 *Commun. Numer. Method. Eng.* **12** 795
[28] Dağ I, Dogan A and Saka B 2003 *Int. J. Comput. Math.* **80** 743
[29] Esen A and Kutluay S 2006 *Appl. Math. Comput.* **174** 833
[30] Dağ I, Saka B and Irk D 2006 *J. Comput. Appl. Math.* **190** 532
[31] Raslan K R 2005 *Appl. Math. Comput.* **167** 1101
[32] Saka B, Dağ I and Dogan A 2004 *Int. J. Comput. Math.* **81** 727
[33] Saka B and Dağ I 2005 *Arab J. Sci. Eng.* **30** 39
[34] Dogan A 2002 *Appl. Math. Model.* **26** 771
[35] Gardner L R T, Gardner G A and Dağ I 1995 *Commun. Numer. Meth. Eng.* **11** 59
[36] Mei L and Chen Y 2012 *Comput. Phys. Commun.* **183** 1609
[37] Dogan A 2001 *Commun. Numer. Meth. Eng.* **17** 485
[38] Islam S, Haq S and Ali A 2009 *J. Comput. Appl. Math.* **223** 997
[39] Mokhtari R and Torabi Z S 2011 *Int. J. Appl. Comput. Math.* **10** 428
[40] Korkmaz A 2010 *Numerical Solutions of Some One-dimensional Partial Differential Equations Using B-spline Differential Quadrature Methods* (Ph. D. thesis) (Eskisehir: Eskisehir Osmangazi University of Turkey)
[41] Shokri A and Dehghan M 2010 *Numer. Meth. Partial Differential Eq.* **26** 807
[42] Alexander M E and Morris J L 1979 *J. Comput. Phys.* **30** 428
[43] Dağ I, Saka B and Irk D 2006 *J. Comput. Appl. Math.* **190** 532
[44] Dağ I, Saka B and Irk D 2004 *Appl. Math. Comput.* **159** 373
[45] Soliman A A and Hussein M H 2005 *J. Appl. Math. Comput.* (Science) **161** 623
[46] Ali A H A 2008 *Nonlinear Dyn.* **51** 59
[47] Chegini N G, Salaripanah A, Mokhtari R and Isvand D 2012 *Nonlinear Dyn.* **69** 459

[48] Gardner L R T, Gardner G A, Ayoub F A and Amein N K 1997 *Arab J. Sci. Eng.* **22** 183

[49] Khalifa A K, Raslan K R and Alzubaidi H M 2007 *Appl. Math. Comput.* **189** 346

[50] Khalifa A K, Raslan K R and Alzubaidi H M 2008 *J. Comput. Appl. Math.* **212** 406

[51] Khalifa A K, Raslan K R and Alzubaidi H M 2008 *Appl. Math. Model.* **32** 2962

[52] Ali A 2009 *Mesh Free Collocation Method for Numerical Solution of Initial-boundary Value Problems using Radial Basis Functions* (Ph. D. thesis) (Ghulam Ishaq Khan Institute of Engineering Sciences and Technology)

[53] Raslan K R 2009 *Chaos Soliton. Fract.* **42** 1845

[54] Raslan K R and Hassan S M 2009 *Appl. Math. Lett.* **22** 984

[55] Raslan K R and EL-Danaf T S 2010 *J. King Saud University* **22** 161

[56] Haq F, Islam S and Tirmizi I A 2010 *Appl. Math. Model.* **34** 4151

[57] Achouri T and Omrani K 2010 *Numer. Meth. Partial Differential Eq.* **26** 399

[58] Labidi M and Omrani K 2011 *Numer. Meth. Partial Differential Eq.* **27** 478

[59] Dereli Y 2012 *Int. J. Nonlinear Sci.* **13** 28

[60] Dereli Y 2012 *Numer. Meth. Partial Differential Eq.* **28** 235

[61] Khan Y, Taghipour R, Falahian M and Nikkar A 2013 *Neural Comput. Appl.* **23** 1335

[62] Karakoc S B G, Yagmurlu N M and Ucar Y 2013 *Boundary Value Problems* **27** 1

[63] Rashidinia J and Mohammadi R 2010 *Comput. Phys. Commun.* **181** 78

[64] Rashidinia J and Mohammadi R 2011 *Numer. Algor.* **56** 129

[65] Mohammadi R 2012 *Appl. Anal.* **91** 2189

[66] Mohammadi R 2013 *Appl. Math.* **4** 933

[67] Mohammadi R 2014 *Comput. Phys. Commun.* **185** 917

[68] Mohammadi R 2014 *Appl. Math. Comput.* **241** 151

[69] Bian X B 2014 *Phys. Rev. A* **90** 033403

[70] Chandra S R S and Kumar M 2008 *Appl. Numer. Math.* **58** 1572

[71] Chandra S R S and Kumar M 2009 *Nonlinear Anal.* **71** 1579

[72] Mc Cartin B J 1991 *J. Approx. Theory* **66** 1

[73] Pruess S 1976 *J. Approx. Theory* **17** 86

[74] Mc Cartin B J 1981 *Theory, Computation, and Application of Exponential Splines* (Courant Mathematics and Computing Laboratory Research and Development Report DOE/ER/03077-171)

[75] Prenter P M 1975 *Splines and Variational Methods* (New York: John Wiley and Sons)

[76] Clavero C and Jorge J C and Lisbona F 1993 *Uniformly Convergent Schemes for Singular Perturbation Problems Combining Alternating Directions and Exponential Fitting Techniques, in: J. J. H. Miller (Ed.)* (Boole, Dublin: Applications of Advanced Computational Methods for Boundary and Interior Layers) pp. 33–52

[77] Rashidinia J, Ghasemi M and Jalilian R 2010 *Math. Sci.* **4** 87

[78] Mohammadi R 2013 *Numer. Meth. Partial Differential Eq.* **29** 1173

[79] Khalifa A K 1979 *Theory and Applications of the Collocation Method with Splines for Ordinary and Partial Differential Equations* (Ph.D. thesis) (Heriot-Watt University)

[80] Gardner L R T and Gardner G A 1990 *J. Comput. Phys.* **91** 441

[81] Soliman A A and Raslan K R 2001 *Int. J. Comput. Math.* **78** 399