

A fourth order one step method for numerical solution of good Boussinesq equation

Emre KIRLI¹ , Dursun IRK^{2,*} 

¹Department of Mathematics, Faculty of Engineering and Natural Sciences, İstanbul Bilgi University, İstanbul, Turkey

²Department of Mathematics and Computer Science, Faculty of Science and Letters, Eskişehir Osmangazi University, Eskişehir, Turkey

Received: 06.05.2021

Accepted/Published Online: 30.07.2021

Final Version: 16.09.2021

Abstract: In this paper, we investigate the numerical solution of "good" Boussinesq equation by using the quartic B-spline Galerkin method for space discretization and the fourth order one-step method for time discretization. The proposed numerical scheme is analyzed for truncation error. Four test problems are studied. The accuracy and efficiency are measured by computing error norm L_∞ and the order of convergence for the proposed method. The results of numerical experiments confirm that the proposed method has a higher accuracy.

Key words: B-spline, Galerkin method, soliton, good Boussinesq equation

1. Introduction

The good Boussinesq equation (GBqE) which describes shallow water waves propagating in the both directions is of the form:

$$u_{tt} = u_{xx} + (u^2)_{xx} - u_{xxxx}. \quad (1.1)$$

The initial and boundary conditions associated with Eq (1.1) are given by

$$u(x, 0) = f_0(x), \quad u_t(x, 0) = f_1(x) \quad x \in [a, b] \quad (1.2)$$

$$u(a, t) = \Psi_1(t), \quad u(b, t) = \Psi_2(t) \quad t \in [0, T] \quad (1.3)$$

$$u_x(a, t) = \Psi_3(t), \quad u_x(b, t) = \Psi_4(t) \quad t \in [0, T] \quad (1.4)$$

where $f_i(x)$, $i = 0, 1$ and $\Psi_j(t)$, $j = 1, 2, 3, 4$ are given continuous functions, $u = u(x, t)$ is supposed to be adequately differentiable function.

Numerous numerical methods have been applied to solve the GBqE. Some of those, Mohebbi and Asgari [1] have used a fourth order time stepping schemes with combination of discrete Fourier transform for numerical solution of GBqE. Manoranjan et al. [2] have obtained numerical solution of GBqE applying Petrov-Galerkin method. Ortega and Sema [3] have developed finite difference method (FDM) to obtain numerical solution of GBqE. Pani and Saranga [4] have applied Faedo-Galerkin method to GBqE. Zoheiry [5] has solved the same equation using an implicit finite difference scheme. Wazwaz [6] has proposed Adomian decomposition method

*Correspondence: dirk@ogu.edu.tr

2010 AMS Mathematics Subject Classification: 65D07, 65N30

to solve numerically Boussinesq equation (BqE). Ismail and Bratsos [7] have obtained numerical solutions of Bq using a FDM which is fourth order in time and the second order in space. Bratsos et al. [8] have derived a linearized numerical method to obtain numerical solution of BqE. Bratsos [9] has used FDM for numerical solution of BqE. Also, Bratsos [10, 11] has proposed third order implicit FDM for approximate solution of GBqE. Khaled and Nusier [12] have proposed Galerkin interpolation method based on sin function and Adomian decomposition method for solving GBqE. Dehghan and Salehi [13] have used combination of boundary knot method and meshless analog equation method for numerical solution of the classical Bq equation. Siddiqi and Arshed [14] have proposed collocation finite element method (FEM) to investigate numerical solution of BqE. Ismail and Mosally [15] have developed a fourth order FDM for approximate solution of GBqE. Ucar et al. [16] have applied Galerkin FEM to GBqE using cubic B-spline basis.

The goal of this study is to present a numerical method to obtain the numerical solution of GBqE by applying the Galerkin FEM based on quartic B-spline functions for the space discretization of the GBqE and a FDM which is of order four for the time discretization of the GBqE. One of the main advantages of the proposed method is that it provides extremely good results than existing studies in the literature and Crank-Nicolson method.

2. The numerical method

If we convert the GBqE into a system of coupled first-order (in time) equations by using $u_t = v$, we get the following system of PDE

$$u_t = v \quad (2.1)$$

$$v_t = u_{xx} + 2\left((u_x)^2 + u_{xx}u\right) - u_{xxxx} \quad (2.2)$$

with the boundary and initial conditions:

$$u(a, t) = \Psi_1(t), \quad u(b, t) = \Psi_2(t), \quad (2.3)$$

$$v(a, t) = \frac{\partial \Psi_1}{\partial t}(t), \quad v(b, t) = \frac{\partial \Psi_2}{\partial t}(t), \quad (2.4)$$

$$u_x(a, t) = \Psi_3(t), \quad u_x(b, t) = \Psi_4(t), \quad (2.5)$$

$$v_x(a, t) = \frac{\partial \Psi_3}{\partial t}(t), \quad v_x(b, t) = \frac{\partial \Psi_4}{\partial t}(t), \quad (2.6)$$

$$u(x, 0) = f_0(x), \quad v(x, 0) = f_1(x). \quad (2.7)$$

Let us assume that $\Omega = [a, b] \times [0, T]$ is smooth region with grid points (x_m, t_n) , where $x_m = a + mh$, $m = 0, 1, 2, \dots, N$, $t_n = n\Delta t$, $n = 0, 1, 2, \dots$. The quantities h and Δt are step size in the space and time directions, respectively.

2.1. Time discretization

For time discretization of GBqE, by applying proposed method which is fourth order one step method, Eqs. (2.1) and (2.2) are discretized in time as noted below:

$$u^{n+1} = u^n + \theta_1 u_t^{n+1} + \theta_2 u_t^n + \theta_3 u_{tt}^{n+1} + \theta_4 u_{tt}^n \quad (2.8)$$

and

$$v^{n+1} = v^n + \theta_1 v_t^{n+1} + \theta_2 v_t^n + \theta_3 v_{tt}^{n+1} + \theta_4 v_{tt}^n \tag{2.9}$$

in which $\theta_1, \theta_2, \theta_3$ and θ_4 are unknown parameters which will be defined later.

By substituting Eq. (2.1) into Eq. (2.8), computing partial derivative with respect to t in the both sides of Eq. (2.1) and employing Eq. (2.2) we obtain

$$u^{n+1} = u^n + \theta_1 v^{n+1} + \theta_2 v^n + \theta_3 (u_x^{n+1} + 2((u_x)^2)^{n+1} + 2u_{xx}^{n+1}u^{n+1} - u_{xxx}^{n+1}) + \theta_4 (u_{xx}^n + 2((u_x)^2)^n + 2u_{xx}^n u^n - u_{xxx}^n). \tag{2.10}$$

Subsequent to required arrangement, Eq. (2.10) is obtained as

$$u^{n+1} + u_{xx}^{n+1}(-\theta_3 - 2\theta_3 u^{n+1}) - 2\theta_3 u_x^{n+1} u_x^{n+1} + \theta_3 u_{xxx}^{n+1} - \theta_1 v^{n+1} = u^n + u_{xx}^n(\theta_4 + 2\theta_4 u^n) + 2\theta_4 u_x^n u_x^n - \theta_4 u_{xxx}^n + \theta_2 v^n. \tag{2.11}$$

Moreover, substituting Eq. (2.2) into Eq. (2.9) yields

$$v^{n+1} - \theta_1 (u_{xx}^{n+1} 2((u_x)^2)^{n+1} + 2u_{xx}^{n+1} u^{n+1} - u_{xxx}^{n+1}) - \theta_3 v_{tt}^{n+1} = v^n + \theta_2 (u_{xx}^n + 2((u_x)^2)^n + 2u_{xx}^n u^n - u_{xxx}^n) + \theta_4 v_{tt}^n \tag{2.12}$$

and then taking partial derivative with respect to t both sides of Eq. (2.2), we obtain

$$v_{tt} = (u_{xx})_t + 2((u_x)^2)_t + 2(u_{xx}u)_t - (u_{xxx})_t \\ v_{tt} = v_{xx} + 4u_x v_x + 2v u_{xx} + 2u v_{xx} - v_{xxx}. \tag{2.13}$$

By substituting Eq. (2.13) into Eq. (2.12), Eq. (2.12) leads to

$$u_{xx}^{n+1}(-\theta_1 - 2\theta_1 u^{n+1} - 2\theta_3 v^{n+1}) + u_x^{n+1}(-2\theta_1 u_x^{n+1} - 4\theta_3 v_x^{n+1}) + \theta_1 u_{xxx}^{n+1} + v^{n+1} + v_{xx}^{n+1}(-\theta_3 - 2\theta_3 u^{n+1}) + \theta_3 v_{xxx}^{n+1} = u_{xx}^n(\theta_2 + 2\theta_2 u^n + 2\theta_4 v^n) + u_x^n(2\theta_2 u_x^n + 4\theta_4 v_x^n) - \theta_2 u_{xxx}^n + v^n + v_{xx}^n(\theta_4 + 2\theta_4 u^n) - \theta_4 v_{xxx}^n. \tag{2.14}$$

Eqs. (2.11–2.14) can be also written as the following compact form

$$AX = B, \tag{2.15}$$

where

$$X^T = (u^{n+1} \ v^{n+1} \ u_x^{n+1} \ u_{xx}^{n+1} \ v_x^{n+1} \ u_{xxx}^{n+1} \ v_{xxx}^{n+1}), \\ A = \begin{pmatrix} 1 & -\theta_1 & -2\theta_3 u_x^{n+1} & -\theta_3 - 2\theta_3 u^{n+1} & 0 & \theta_3 & 0 \\ 0 & 1 & -2\theta_1 u_x^{n+1} - 4\theta_3 v_x^{n+1} & -\theta_1 - 2\theta_1 u^{n+1} - 2\theta_3 v^{n+1} & -\theta_3 - 2\theta_3 u^{n+1} & \theta_1 & \theta_3 \end{pmatrix}, \\ B = \begin{pmatrix} u^n + u_{xx}^n(\theta_4 + 2\theta_4 u^n) + 2\theta_4 u_x^n u_x^n - \theta_4 u_{xxx}^n + \theta_2 v^n \\ u_{xx}^n(\theta_2 + 2\theta_2 u^n + 2\theta_4 v^n) + u_x^n(2\theta_2 u_x^n + 4\theta_4 v_x^n) - \theta_2 u_{xxx}^n + v^n + v_{xx}^n(\theta_4 + 2\theta_4 u^n) - \theta_4 v_{xxx}^n \end{pmatrix}.$$

Lemma 1. Suppose that $u, v \in C^7(\Omega)$, and $\theta_1 = \theta_2 = \frac{\Delta t}{2}$ and $\theta_3 = -\theta_4 = -\frac{\Delta t^2}{12}$. Then, the numerical scheme (2.11) and (2.14) are consistent and fourth order accurate in time for the norm $\|\cdot\|_\infty$.

Proof. Let E_1 and E_2 be truncation errors of numerical scheme (2.11) and (2.14), respectively. By using the $\theta_1 = \theta_2 = \frac{\Delta t}{2}$ and $\theta_3 = -\theta_4 = -\frac{\Delta t^2}{12}$ in Eqs. (2.11)–(2.14), E_1 and E_2 are obtained as

$$E_1(u, v) = \left[\frac{u_{xx}(x, \tau) u_x(x, \tau) v_x(x, \tau)}{90} - \frac{v_{xxxxtt}(x, \tau)}{720} + \frac{v_{xxtt}(x, \tau)}{720} + \frac{u_{xx}(x, \tau) v_{xx}(x, \tau)}{90} + \frac{v_{xx}(x, \tau) (u_x(x, \tau))^2}{60} \right. \\ + \frac{v(x, \tau) v_{xxt}(x, \tau)}{120} + \frac{u(x, \tau) v_{xxtt}(x, \tau)}{360} - \frac{u_{xx}(x, \tau) v_{xxxx}(x, \tau)}{360} + \frac{u_x(x, \tau) v_{xtt}(x, \tau)}{180} \\ \left. + \frac{u_{xx}(x, \tau) u(x, \tau) v_{xx}(x, \tau)}{45} + \frac{v(x, \tau) (u_{xx}(x, \tau))^2}{180} + \frac{v_x(x, \tau) v_{xt}(x, \tau)}{60} - \frac{u_{xxxx}(x, \tau) v_{xx}(x, \tau)}{120} \right] \Delta t^5 + \dots$$

and

$$E_2(u, v) = \left[-\frac{(u_{xx}(x, \tau))^3}{36} - \frac{v_x(x, \tau) v_{xtt}(x, \tau)}{9} - \frac{(v_{xt}(x, \tau))^2}{12} - \frac{u_x(x, \tau) v_{txtt}(x, \tau)}{36} - \frac{(v_{xx}(x, \tau))^2}{18} \right. \\ - \frac{u_{xx}(x, \tau) v(x, \tau) v_{xx}(x, \tau)}{6} - \frac{7u_{xx}(x, \tau) u(x, \tau) v_{xxt}(x, \tau)}{36} - \frac{2v_{xx}(x, \tau) u_x(x, \tau) v_x(x, \tau)}{9} \\ - \frac{u(x, \tau) (u_{xx}(x, \tau))^3}{18} - \frac{(u_{xx}(x, \tau))^2 (u_x(x, \tau))^2}{18} + \frac{(u_{xx}(x, \tau))^2 u_{xxxx}(x, \tau)}{36} - \frac{7u_{xx}(x, \tau) v_{xxt}(x, \tau)}{72} \\ - \frac{u_{xx}(x, \tau) (v_x(x, \tau))^2}{18} + \frac{u_{xx}(x, \tau) v_{xxxxt}(x, \tau)}{72} - \frac{(v_{xx}(x, \tau))^2 u(x, \tau)}{9} + \frac{v_{xx}(x, \tau) v_{xxxx}(x, \tau)}{18} \\ - \frac{v_{xxt}(x, \tau) (u_x(x, \tau))^2}{6} + \frac{v_{xxt}(x, \tau) u_{xxxx}(x, \tau)}{12} - \frac{u_x(x, \tau) v_{txtt}(x, \tau)}{36} - \frac{u_{xx}(x, \tau) u_x(x, \tau) v_{xt}(x, \tau)}{18} \\ \left. - \frac{1}{72} \frac{\partial^5 v(x, \tau)}{\partial x^2 \partial t^3} u(x, \tau) + \frac{1}{120} \frac{\partial^5 v(x, \tau)}{\partial t^5} - \frac{1}{144} \frac{\partial^5 v(x, \tau)}{\partial x^2 \partial t^3} + \frac{1}{144} \frac{\partial^7 v(x, \tau)}{\partial x^4 \partial t^3} \right] \Delta t^5 + \dots$$

where $t_n < \tau < t_{n+1}$. Hence, we have :

$$\|E_1(u, v)\|_\infty \leq \Delta t^5 \sup_{(x, \zeta) \in \Omega} |\varepsilon_1(x, \zeta)|$$

and

$$\|E_2(u, v)\|_\infty \leq \Delta t^5 \sup_{(x, \zeta) \in \Omega} |\varepsilon_2(x, \zeta)|$$

Here $\varepsilon_i(x, \zeta)$, $i = 1, 2$ denote the coefficients of the Δt^5 in $E_i(u(x, \zeta), v(x, \zeta))$.

Therefore, the conclusions obtained above imply that the numerical scheme (2.11) and (2.14) are consistent and four order accurate in time.

Observe that in Eqs. (2.8) and (2.9), when $\theta_1 = \theta_2 = \frac{\Delta t}{2}$ and $\theta_3 = \theta_4 = 0$, we obtain Crank Nicolson method which is of order two in time and when $\theta_1 = \theta_2 = \frac{\Delta t}{2}$ and $\theta_3 = -\theta_4 = -\frac{\Delta t^2}{12}$, we get high order accurate which is of order four in time. Hence, our proposed method for time discretization is a high order method.

2.2. Space discretization

For the space discretization, by applying Galerkin finite element method to Eqs. (2.11) and (2.14) and then using integration by parts, the weak forms of Eqs. (2.11) and (2.14) are obtained in the following form

$$\begin{aligned} & \int_a^b w [u^{n+1} + u_{xx}^{n+1}(-\theta_3 - 2\theta_3 u^{n+1}) - 2\theta_3 u_x^{n+1} u_x^{n+1} - \theta_1 v^{n+1}] dx - \int_a^b \theta_3 w_x u_{xxx}^{n+1} dx + \\ & \theta_3 w u_{xxx}^{n+1} \Big|_a^b = \int_a^b w [u^n + u_{xx}^n(\theta_4 + 2\theta_4 u^n) + 2\theta_4 u_x^n u_x^n + \theta_2 v^n] dx + \\ & \int_a^b \theta_4 w_x u_{xxx}^n dx - \theta_4 w u_{xxx}^n \Big|_a^b \end{aligned} \tag{2.16}$$

and

$$\begin{aligned} & \int_a^b w [u_{xx}^{n+1}(-\theta_1 - 2\theta_1 u^{n+1} - 2\theta_3 v^{n+1}) + u_x^{n+1}(-2\theta_1 u_x^{n+1} - 4\theta_3 v_x^{n+1})] dx + \\ & \int_a^b w [v^{n+1} + v_{xx}^{n+1}(-\theta_3 - 2\theta_3 u^{n+1})] dx - \int_a^b w_x (\theta_1 u_{xxx}^{n+1} + \theta_3 v_{xxx}^{n+1}) dx + \\ & \theta_3 w v_{xxx}^{n+1} \Big|_a^b + \theta_1 w u_{xxx}^{n+1} \Big|_a^b = \int_a^b w [u_{xx}^n(\theta_2 + 2\theta_2 u^n + 2\theta_4 v^n) + u_x^n(2\theta_2 u_x^n + 4\theta_4 v_x^n)] dx - \\ & \int_a^b w [v^n + v_{xx}^n(\theta_4 + 2\theta_4 u^n)] dx + \int_a^b \theta_2 w_x u_{xxx}^n dx + \int_a^b \theta_4 w_x v_{xxx}^n dx - \\ & \theta_4 w v_{xxx}^n \Big|_a^b - \theta_2 w u_{xxx}^n \Big|_a^b \end{aligned} \tag{2.17}$$

where w is a weight function. The approximate solutions $U(x, t)$ and $V(x, t)$ corresponding to exact solutions $u(x, t)$ and $v(x, t)$ are expressed as linear combination of quartic B-spline functions as:

$$U(x, t) = \sum_{m=-2}^{N+1} \delta_m(t) Q_m(x) \quad V(x, t) = \sum_{m=-2}^{N+1} \sigma_m(t) Q_m(x) \tag{2.18}$$

in which δ_m and σ_m $m = -2, -1, 0, \dots, N + 1$ are unknown time dependent parameters which are going to be calculated by using boundary and discretized form of Eq. (1.1). The quartic B-spline for $m = -2, -1, 0, \dots, N + 1$ is defined as the following;

$$Q_m(x) = \frac{1}{h^4} \begin{cases} (z_{m-2})^4, & x \in [x_{m-2}, x_{m-1}) \\ (z_{m-2})^4 - 5(z_{m-1})^4, & x \in [x_{m-1}, x_m), \\ (z_{m-2})^4 - 5(z_{m-1})^4 + 10(z_m)^4, & x \in [x_m, x_{m+1}), \\ (z_{m+3})^4 - 5(z_{m+2})^4, & x \in [x_{m+1}, x_{m+2}), \\ (z_{m+3})^4, & x \in [x_{m+2}, x_{m+3}), \\ 0, & \text{otherwise} \end{cases} \tag{2.19}$$

where $z_m = x - x_m$. The set of quartic B-spline $\{Q_{-2}, Q_{-1}, \dots, Q_{N+1}\}$ forms a basis over the interval $[a, b]$. The approximate functions given in Eq. (2.18) over the subelement $[x_m, x_{m+1}]$ in terms of quartic B-splines are defined as follows:

$$U(x, t) = \sum_{j=m-2}^{m+2} \delta_j(t) Q_j(x) \quad V(x, t) = \sum_{j=m-2}^{m+2} \sigma_j(t) Q_j(x). \tag{2.20}$$

Employing quartic B-spline functions (2.19) and approximate functions (2.20), the approximate solutions U , V , and their first, second and third order derivatives over the element $[x_m, x_{m+1}]$ are obtained as the following.

$$\begin{aligned}
 U(x_m) &= \delta_{m-1} + 11\delta_m + 11\delta_{m+1} + \delta_{m+2}, \\
 V(x_m) &= \sigma_{m-1} + 11\sigma_m + 11\sigma_{m+1} + \sigma_{m+2}, \\
 U'(x_m) &= \frac{4}{h}(-\delta_{m-1} - 3\delta_m + 3\delta_{m+1} + \delta_{m+2}), \\
 V'(x_m) &= \frac{4}{h}(-\sigma_{m-1} - 3\sigma_m + 3\sigma_{m+1} + \sigma_{m+2}), \\
 U''(x_m) &= \frac{12}{h^2}(\delta_{m-1} - \delta_m - \delta_{m+1} + \delta_{m+2}), \\
 V''(x_m) &= \frac{12}{h^2}(\sigma_{m-1} - \sigma_m - \sigma_{m+1} + \sigma_{m+2}), \\
 U'''(x_m) &= \frac{24}{h^3}(-\delta_{m-1} + 3\delta_m - 3\delta_{m+1} + \delta_{m+2}), \\
 V'''(x_m) &= \frac{24}{h^3}(-\sigma_{m-1} + 3\sigma_m - 3\sigma_{m+1} + \sigma_{m+2}).
 \end{aligned}$$

By taking quartic B-spline shape functions instead of weight function w , and substituting (2.20) into Eqs. (2.16) and (2.17), the Eqs. (2.16)–(2.17) are rewritten as

$$\begin{aligned}
 &\sum_{j=m-2}^{m+2} \left\{ \int_{x_m}^{x_{m+1}} Q_i Q_j dx - \theta_3 \int_{x_m}^{x_{m+1}} Q_i Q_j'' dx - 2\theta_3 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \delta_r^{n+1} \right) Q_j'' dx \right\} \delta_j^{n+1} + \\
 &\sum_{j=m-2}^{m+2} \left\{ -2\theta_3 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r' \delta_r^{n+1} \right) Q_j' dx \right\} \delta_j^{n+1} + \sum_{j=m-2}^{m+2} \left\{ -\theta_3 \int_{x_m}^{x_{m+1}} Q_i' Q_j''' dx \right\} \delta_j^{n+1} + \\
 &\sum_{j=m-2}^{m+2} \left\{ \theta_3 Q_i Q_j''' \Big|_{x_m}^{x_{m+1}} \right\} \delta_j^{n+1} + \sum_{j=m-2}^{m+2} \left\{ -\theta_1 \int_{x_m}^{x_{m+1}} Q_i Q_j dx \right\} \sigma_j^{n+1} - \\
 &\sum_{j=m-2}^{m+2} \left\{ \int_{x_m}^{x_{m+1}} Q_i Q_j dx + \theta_4 \int_{x_m}^{x_{m+1}} Q_i Q_j'' dx + 2\theta_4 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \delta_r^n \right) Q_j'' dx \right\} \delta_j^n - \\
 &\sum_{j=m-2}^{m+2} \left\{ 2\theta_4 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-1}^{m+2} Q_r' \delta_r^n \right) Q_j' dx \right\} \delta_j^n - \sum_{j=m-2}^{m+2} \left\{ \theta_4 \int_{x_m}^{x_{m+1}} Q_i' Q_j''' dx \right\} \delta_j^n - \\
 &\sum_{j=m-2}^{m+2} \left\{ -\theta_4 Q_i Q_j''' \Big|_{x_m}^{x_{m+1}} \right\} \delta_j^n - \sum_{j=m-2}^{m+2} \left\{ \theta_2 \int_{x_m}^{x_{m+1}} Q_i Q_j dx \right\} \sigma_j^n \\
 &i = m - 2, m - 1, m, m + 1, m + 2,
 \end{aligned} \tag{2.21}$$

and

$$\begin{aligned}
 & \sum_{j=m-2}^{m+2} \left\{ -\theta_1 \int_{x_m}^{x_{m+1}} Q_i Q_j'' dx - 2\theta_1 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \delta_r^{n+1} \right) Q_j'' dx \right\} \delta_j^{n+1} - \\
 & \sum_{j=m-2}^{m+2} \left\{ 2\theta_3 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \sigma_r^{n+1} \right) Q_j'' dx + 4\theta_3 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r' \sigma_r^{n+1} \right) Q_j' dx \right\} \delta_j^{n+1} + \\
 & \sum_{j=m-2}^{m+2} \left\{ -2\theta_1 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r' \delta_r^{n+1} \right) Q_j' dx - \theta_1 \int_{x_m}^{x_{m+1}} Q_i' Q_j''' dx \right\} \delta_j^{n+1} + \\
 & \sum_{j=m-2}^{m+2} \left\{ \theta_1 Q_i Q_j''' \Big|_{x_m}^{x_{m+1}} \right\} \delta_j^{n+1} + \sum_{j=m-2}^{m+2} \left\{ \int_{x_m}^{x_{m+1}} Q_i Q_j dx - \theta_3 \int_{x_m}^{x_{m+1}} Q_i Q_j'' dx \right\} \sigma_j^{n+1} + \\
 & \sum_{j=m-2}^{m+2} \left\{ -2\theta_3 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \delta_r^{n+1} \right) Q_j'' dx - \theta_3 \int_{x_m}^{x_{m+1}} Q_i' Q_j''' dx \right\} \sigma_j^{n+1} + \\
 & \sum_{j=m-2}^{m+2} \left\{ \theta_3 Q_i Q_j''' \Big|_{x_m}^{x_{m+1}} \right\} \sigma_j^{n+1} - \sum_{j=m-2}^{m+2} \left\{ 2\theta_4 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \sigma_r^n \right) Q_j'' dx \right\} \delta_j^n - \\
 & \sum_{j=m-2}^{m+2} \left\{ \theta_2 \int_{x_m}^{x_{m+1}} Q_i Q_j'' dx + 2\theta_2 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \delta_r^n \right) Q_j'' dx \right\} \delta_j^n - \\
 & \sum_{j=m-2}^{m+2} \left\{ 2\theta_2 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r' \delta_r^n \right) Q_j' dx + 4\theta_4 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r' \sigma_r^n \right) Q_j' dx \right\} \delta_j^n - \\
 & \sum_{j=m-2}^{m+2} \left\{ \theta_2 \int_{x_m}^{x_{m+1}} Q_i' Q_j''' dx \right\} \delta_j^n - \sum_{j=m-2}^{m+2} \left\{ -\theta_2 Q_i Q_j''' \Big|_{x_m}^{x_{m+1}} \right\} \delta_j^n - \\
 & \sum_{j=m-2}^{m+2} \left\{ \int_{x_m}^{x_{m+1}} Q_i Q_j dx + \theta_4 \int_{x_m}^{x_{m+1}} Q_i Q_j'' dx + 2\theta_4 \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \delta_r^n \right) Q_j'' dx \right\} \sigma_j^n - \\
 & \sum_{j=m-2}^{m+2} \left\{ \theta_4 \int_{x_m}^{x_{m+1}} Q_i' Q_j''' dx \right\} \sigma_j^n - \sum_{j=m-2}^{m+2} \left\{ -\theta_4 Q_i Q_j''' \Big|_{x_m}^{x_{m+1}} \right\} \sigma_j^n \\
 & i = m - 2, m - 1, m, m + 1, m + 2,
 \end{aligned} \tag{2.22}$$

for $m = 0, 1, \dots, N - 1$. The (2.21) and (2.22) are also defined in matrix form as

$$\begin{aligned}
 & [A^e - \theta_3 B^e - 2\theta_3 C^e - 2\theta_3 D^e - \theta_3 E^e + \theta_3 F^e] (\delta^e)^{n+1} - \theta_1 A^e (\sigma^e)^{n+1} - \\
 & [A^e + \theta_4 B^e + 2\theta_4 C^e + 2\theta_4 D^e + \theta_4 E^e - \theta_4 F^e] (\delta^e)^n - \theta_2 A^e (\sigma^e)^n
 \end{aligned} \tag{2.23}$$

and

$$\begin{aligned}
 & [-\theta_1 B^e - 2\theta_1 C^e - 2\theta_3 \check{C}^e - 4\theta_3 \check{D}^e - 2\theta_1 D^e - \theta_1 E^e + \theta_1 F^e] (\delta^e)^{n+1} + \\
 & [A^e - \theta_3 B^e - 2\theta_3 C^e - \theta_3 E^e + \theta_3 F^e] (\sigma^e)^{n+1} - \\
 & [\theta_2 B^e + 2\theta_4 \check{C}^e + 2\theta_2 C^e + 2\theta_2 D^e + 4\theta_4 \check{D}^e + \theta_2 E^e - \theta_2 F^e] (\delta^e)^n - \\
 & [A^e + \theta_4 B^e + 2\theta_4 C^e + \theta_4 E^e - \theta_4 F^e] (\sigma^e)^n
 \end{aligned} \tag{2.24}$$

where $i, j = m - 2, m - 1, m, m + 1, m + 2$ the element matrices and element parameters are defined as follows:

$$\begin{aligned}
 A_{ij}^e &= \int_{x_m}^{x_{m+1}} Q_i Q_j dx, \quad B_{ij}^e = \int_{x_m}^{x_{m+1}} Q_i Q_j'' dx, \quad F_{ij}^e = Q_i Q_j''' \Big|_{x_m}^{x_{m+1}}, \quad E_{ij}^e = \int_{x_m}^{x_{m+1}} Q_i' Q_j''' dx \\
 C_{ij}^e(\delta^{n+1}) &= \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \delta_r^{n+1} \right) Q_j' dx, \\
 \check{C}_{ij}^e(\sigma^{n+1}) &= \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r \sigma_r^{n+1} \right) Q_j'' dx, \\
 D_{ij}^e(\delta^{n+1}) &= \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r' \delta_r^{n+1} \right) Q_j' dx, \\
 \check{D}_{ij}^e(\sigma^{n+1}) &= \int_{x_m}^{x_{m+1}} Q_i \left(\sum_{r=m-2}^{m+2} Q_r' \sigma_r^{n+1} \right) Q_j' dx, \\
 \delta^e &= (\delta_{m-2}, \delta_{m-1}, \delta_m, \delta_{m+1}, \delta_{m+2})^T, \quad \sigma^e = (\sigma_{m-2}, \sigma_{m-1}, \sigma_m, \sigma_{m+1}, \sigma_{m+2})^T
 \end{aligned}$$

After combining the element matrices over all elements $[x_m, x_{m+1}]$, the new matrix forms of (2.23) and (2.24) are obtained as:

$$\begin{aligned}
 &[\mathbf{A} - \theta_3 \mathbf{B} - 2\theta_3 \mathbf{C} - 2\theta_3 \mathbf{D} - \theta_3 \mathbf{E} + \theta_3 \mathbf{F}] (\delta)^{n+1} - \theta_1 \mathbf{A} (\sigma)^{n+1} = \\
 &[\mathbf{A} + \theta_4 \mathbf{B} + 2\theta_4 \mathbf{C} + 2\theta_4 \mathbf{D} + \theta_4 \mathbf{E} - \theta_4 \mathbf{F}] (\delta)^n + \theta_2 \mathbf{A} (\sigma)^n
 \end{aligned} \tag{2.25}$$

and

$$\begin{aligned}
 &[-\theta_1 \mathbf{B} - 2\theta_1 \mathbf{C} - 2\theta_3 \check{\mathbf{C}} - 4\theta_3 \check{\mathbf{D}} - 2\theta_1 \mathbf{D} - \theta_1 \mathbf{E} + \theta_1 \mathbf{F}] (\delta)^{n+1} + \\
 &[\mathbf{A} - \theta_3 \mathbf{B} - 2\theta_3 \mathbf{C} - \theta_3 \mathbf{E} + \theta_3 \mathbf{F}] (\sigma)^{n+1} = \\
 &[\theta_2 \mathbf{B} + 2\theta_4 \check{\mathbf{C}} + 2\theta_2 \mathbf{C} + 2\theta_2 \mathbf{D} + 4\theta_4 \check{\mathbf{D}} + \theta_2 \mathbf{E} - \theta_2 \mathbf{F}] (\delta)^n + \\
 &[\mathbf{A} + \theta_4 \mathbf{B} + 2\theta_4 \mathbf{C} + \theta_4 \mathbf{E} - \theta_4 \mathbf{F}] (\sigma)^n.
 \end{aligned} \tag{2.26}$$

Here $\delta^{n+1} = (\delta_{-2}^{n+1}, \delta_{-1}^{n+1}, \delta_0^{n+1}, \dots, \delta_{N+1}^{n+1})^T$ and $\sigma^{n+1} = (\sigma_{-2}^{n+1}, \sigma_{-1}^{n+1}, \sigma_0^{n+1}, \dots, \sigma_{N+1}^{n+1})^T$ are unknown time dependent parameters. The set of equations given in Eqs. (2.25) and (2.26) comprises of $2N + 8$ equations in $2N + 8$ unknown paramters. Before begining the procedure, it is required that boundary conditions are adapted into system. For that purpose, first and last equations are deleted from the systems (2.25) and (2.26), and then the terms δ_{-2}, σ_{-2} and $\delta_{N+1}, \sigma_{N+1}$ are eliminated from the remaining systems (2.25) and (2.26) by employing boundary conditions (2.3) and (2.4). Hence, the new matrix system is obtained in type of $(2N + 4)$ equations and $(2N + 4)$ unknowns. After initial vectors δ^0 and σ^0 are calculated by using intial and boundary conditions. The unknown vectors $\delta = (\delta_{-2}^{n+1}, \delta_{-1}^{n+1}, \delta_0^{n+1}, \dots, \delta_{N+1}^{n+1})^T$ and $\sigma = (\sigma_{-2}^{n+1}, \sigma_{-1}^{n+1}, \sigma_0^{n+1}, \dots, \sigma_{N+1}^{n+1})^T$ for $n = 0, 1, 2, \dots$ are found repeatedly by using the following inner iteration algorithm:

- Step 1: Set *error* = 1 and counter=1
 - Step 2: Set $\delta_m^* = \delta_m^{n+1}$ and $\sigma_m^* = \sigma_m^{n+1}$ in $\mathbf{C}(\delta^{n+1}), \check{\mathbf{C}}(\sigma^{n+1}), \mathbf{D}(\delta^{n+1}), \check{\mathbf{D}}(\sigma^{n+1})$ and take $\delta_m^* = \delta_m^n$ and $\sigma_m^* = \sigma_m^n$.
 - Step 3: While *error* $\geq 10^{-10}$ and counter ≤ 5 do Steps 4–5,
 - Step 4: Find U_m^{n+1} and V_m^{n+1}
 - Step 5: Take *error* = $\max_m \{|U_m^{n+1} - U_m^*|, |V_m^{n+1} - V_m^*|\}$, $\delta_m^* = \delta_m^{n+1}$ and $\sigma_m^* = \sigma_m^{n+1}$
- Stop and go to next time step.

2.3. Initial step

In order to begin iterative computation, the initial parameter vectors δ^0 and σ^0 must be determined. Using the following initial conditions

$$\begin{aligned} U_{xx}(x_0, 0) &= \frac{12}{h^2}(\delta_{-1} - \delta_0 - \delta_1 + \delta_2) = f_0''(x_0), \\ U_x(x_0, 0) &= \frac{4}{h}(-\delta_{-1} - 3\delta_0 + 3\delta_1 + \delta_2) = f_0'(x_0), \\ U(x_m, 0) &= \delta_{m-1} + 11\delta_m + 11\delta_{m+1} + \delta_{m+2} = f_0(x_m) \quad m = 0, 1, 2, \dots, N \\ U_x(x_N, 0) &= \frac{4}{h}(-\delta_{N-1} - 3\delta_N + 3\delta_{N+1} + \delta_{N+2}) = f_0'(x_N) \end{aligned} \quad (2.27)$$

gives us a system of linear equation which is $(N + 4)$ equation in $(N + 4)$ unknown parameters. With a similar method, the following initial conditions for approximate solution V

$$\begin{aligned} V_{xx}(x_0, 0) &= \frac{12}{h^2}(\sigma_{-1} - \sigma_0 - \sigma_1 + \sigma_2) = f_1''(x_0), \\ V_x(x_0, 0) &= \frac{4}{h}(-\sigma_{-1} - 3\sigma_0 + 3\sigma_1 + \sigma_2) = f_1'(x_0), \\ V(x_m, 0) &= \sigma_{m-1} + 11\sigma_m + 11\sigma_{m+1} + \sigma_{m+2} = f_1(x_m) \quad m = 0, 1, 2, \dots, N \\ V_x(x_N, 0) &= \frac{4}{h}(-\sigma_{N-1} - 3\sigma_N + 3\sigma_{N+1} + \sigma_{N+2}) = f_1'(x_N) \end{aligned} \quad (2.28)$$

gives a $(N + 4)$ equation in $(N + 4)$ unknowns. Thus, the initial vectors δ^0 and σ^0 are computed.

3. Numerical experiments

In this section, we implement proposed method for the numerical solution of the GBqE. The accuracy of the proposed method is tested by employing maximum error L_∞

$$L_\infty = \|u - U_N\|_\infty = \max_j |u_j - U_j|.$$

In literature [15], the conservation constant for GBqE is given by

$$I = \int_{\Omega} u(x, t) dx = \text{constant}.$$

In the simulations, invariant I is monitored to check the conservation of the proposed method and the trapezoidal rule is employed to compute the above integral. The order of convergence is calculated by the formula:

$$\text{order} = \frac{\log \left| \frac{(L_\infty)_{\Delta t_i}}{(L_\infty)_{\Delta t_{i+1}}} \right|}{\log \left| \frac{\Delta t_i}{\Delta t_{i+1}} \right|}.$$

Here $(L_\infty)_{\Delta t_i}$ is the error norm L_∞ for the time step Δt_i .

3.1. Single soliton wave test problem

Consider the GBqE given in Eq. (1.1) with initial conditions

$$\begin{aligned} f_0(x) = u(x, 0) &= -A \operatorname{sech}^2 \left(\sqrt{\frac{A}{6}} (x - \bar{x}_0) \right), \\ f_1(x) = u_t(x, 0) &= -2Ac \sqrt{\frac{A}{6}} \operatorname{sech}^2 \left(\sqrt{\frac{A}{6}} (x - \bar{x}_0) \right) \tanh \left(\sqrt{\frac{A}{6}} (x - \bar{x}_0) \right). \end{aligned}$$

The exact solution of this test problem is given by

$$u(x, t) = -A \operatorname{sech}^2 \left(\sqrt{\frac{A}{6}} (x - ct - \bar{x}_0) \right) - (b + 1/2),$$

and boundary conditions can be found with the help of exact solutions. In the exact solution, $c = \sqrt{1 - 2A/3}$ is the speed of the soliton wave and A is the amplitude of the soliton wave.

A comparison of error norms L_∞ is presented in Table 1 for $x \in [-80, 100]$, $A = 0.369$, $b = \frac{-1}{2}$, $h = 0.5, 0.3, 0.1$, $\bar{x}_0 = 0$, and $\Delta t = 0.002$ at time $t = 30$. It is clearly seen from the Table 1 that the results obtained by the fourth order method in time with quartic B-spline Galerkin method are more accurate than those obtained by some earlier papers.

Table 1. The maximum error norm for GBq with $\Delta t = 0.002$ at $t = 30$.

h	[7]	[16]	Proposed method
0.5	0.005501	0.003274×10^{-3}	0.008465×10^{-5}
0.3	0.001959	0.000452×10^{-3}	0.003613×10^{-6}
0.1	0.000126	0.024854×10^{-3}	0.007094×10^{-8}

Figure 1 shows the absolute values of the single solitons at $t = 0, 10, 20$ and 30 for $x \in [-80, 100]$, $A = 0.369$, $b = \frac{-1}{2}$, $h = 0.1$, $\bar{x}_0 = 0$, and $\Delta t = 0.002$. It can be seen from the figure that single soliton moves the right almost with unchanged in forms.

The error norm L_∞ and rate of convergence for both Crank Nicolson and proposed method are listed in Table 2 for the space interval $[-40, 40]$, $A = 0.5$, $\bar{x}_0 = 0$, space interval $h = 0.1$, and various time steps $\Delta t = 5, 2, 1, 0.5, 0.2, 0.1$ at time $t = 10$. According to the results in Tables 1 and 2, the proposed method is considerable good in comparison with other methods. Additionally, from Table 2, it can be seen that order of proposed method is almost 4. The distribution of absolute error is plotted in Figures 2 and 3 for the proposed and Crank Nicolson methods.

3.2. Interaction of two soliton waves

The collision problem of two soliton waves is studied by using the following initial conditions

$$\begin{aligned} u(x, 0) &= u_1(x, 0) + u_2(x, 0), \\ v(x, 0) &= v_1(x, 0) + v_2(x, 0), \end{aligned}$$

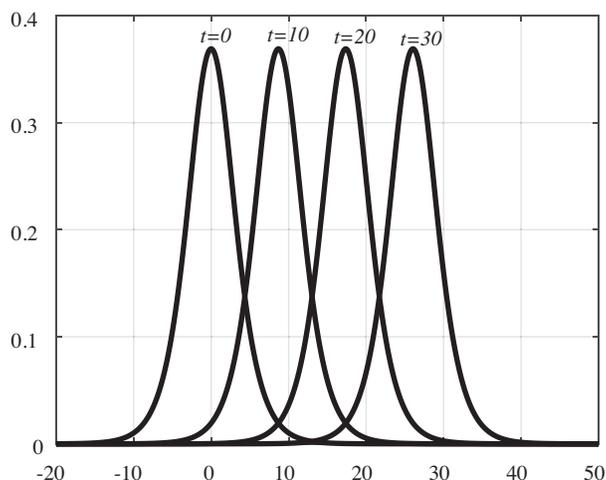


Figure 1. Absolute values of the single solitons at $t = 0, 10, 20,$ and 30 .

Table 2. The error norm and order of convergence with $h = 0.1$ at $t = 10$.

Δt	Crank Nicolson method		Proposed method	
	L_∞	Order	L_∞	Order
5	1.78×10^{-1}	1.21	2.95×10^{-2}	3.46
2	5.88×10^{-2}	1.79	1.39×10^{-3}	3.89
1	1.70×10^{-2}	2.07	9.39×10^{-5}	3.95
0.5	4.07×10^{-3}	2.01	6.06×10^{-6}	3.99
0.2	6.45×10^{-4}	2.00	1.57×10^{-7}	4.00
0.1	1.61×10^{-4}		9.80×10^{-9}	

where

$$\begin{aligned}
 u_i(x, 0) &= -A_i \operatorname{sech}^2 \left[\sqrt{\frac{A_i}{6}} (x - x_i^0) \right], \\
 v_i(x, 0) &= -2A_i c_i \sqrt{\frac{A_i}{6}} \operatorname{sech}^2 \left[\sqrt{\frac{A_i}{6}} (x - x_i^0) \right] \tanh \left[\sqrt{\frac{A_i}{6}} (x - x_i^0) \right], \\
 c_i &= \pm \left(1 - \frac{2A_i}{3} \right)^{\frac{1}{2}}, \quad i = 1, 2.
 \end{aligned}$$

The computations are carried out by choosing the parameters $x_1^0 = -x_2^0 = -50$, $A_1 = A_2 = 0.369$, $c_1 = -c_2 = \sqrt{1 - 2A/3}$, $h = 0.1$, and $\Delta t = 0.01$. These parameters provide two separated soliton waves which are initially located at $x_1^0 = -50$ and $x_2^0 = 50$, respectively. The program is run up to time $t = 120$ over the interval $x \in [-100, 100]$. The interaction of two soliton waves is shown in Figure 4. It can be seen from the figure that the waves collide and seem in the form of a single wave at $t \simeq 60$. After the collision, the waves leave each other and regain their initial shape and amplitude. So the collision is elastic. Moreover, when two waves interact, the joint amplitude is greater than the sum of amplitudes of waves; this is in good agreement with those given in Refs. [2, 5, 7, 8, 11, 15, 16].

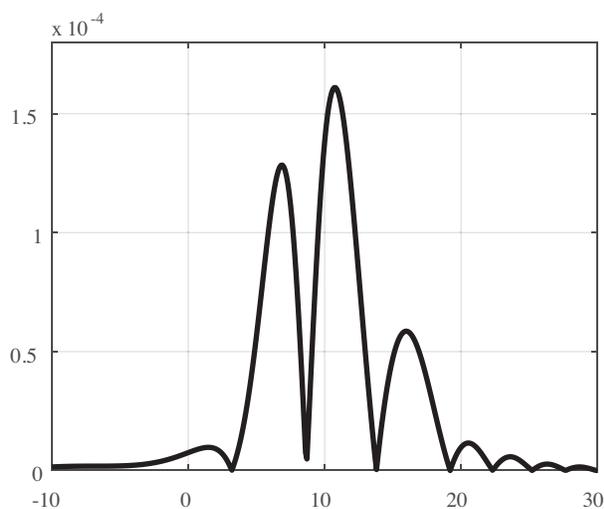


Figure 2. Absolute error for Crank Nicolson method.

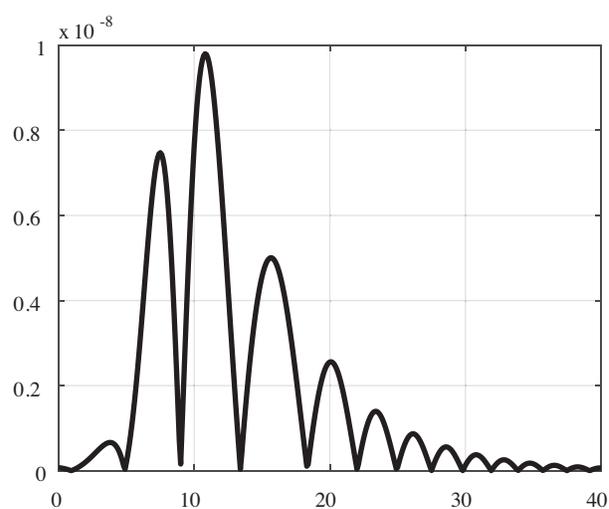


Figure 3. Absolute error for proposed method.

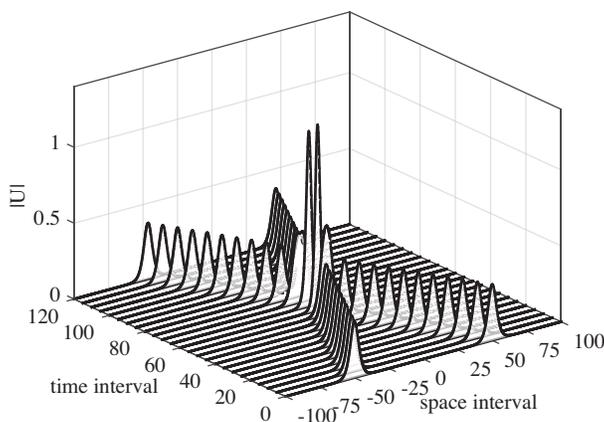


Figure 4. Interaction of two solitons.

The numerical values of conservation constant at several times are presented in the Table 3, which shows that the proposed method is more accurate than Crank Nicolson method. Numerical values of conservation constant over the time interval for both methods are also drawn in Figures 5 and Figure 6. It is seen from the figures and Table 3 that less error is obtained when the high order method is used.

3.3. Overtaking soliton interaction

We consider the interaction of two solitons, moving in the same direction with the initial conditions

$$\begin{aligned}
 u(x, 0) &= u_1(x, 0) + u_2(x, 0), \\
 v(x, 0) &= v_1(x, 0) + v_2(x, 0),
 \end{aligned}$$

Table 3. The comparison of conservation constant.

Time	Crank Nicolson method	Proposed method
$t = 0$	-5.951806448	-5.951806448
$t = 20$	-5.951806452	-5.951806448
$t = 40$	-5.951806449	-5.951806449
$t = 60$	-5.951806837	-5.951806449
$t = 80$	-5.951806682	-5.951806449
$t = 100$	-5.951806058	-5.951806450
$t = 120$	-5.951806344	-5.951806450
Exact	-5.951806447	-5.951806447

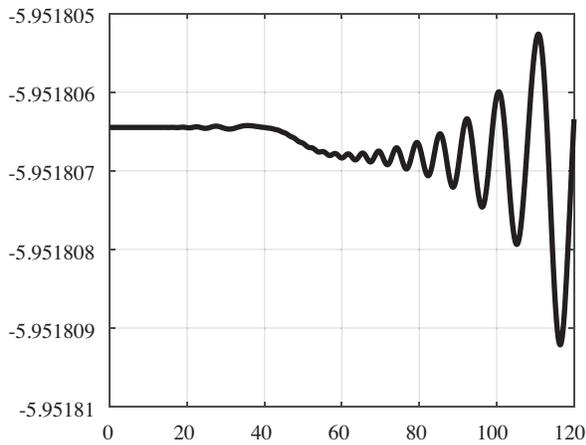


Figure 5. Conservation constant for Crank Nicolson method.

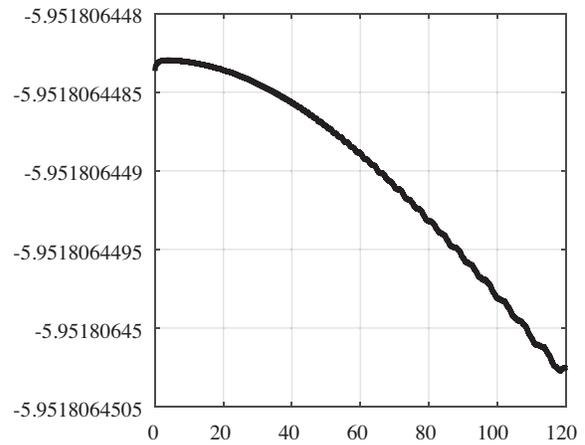


Figure 6. Conservation constant for the proposed method.

where

$$\begin{aligned}
 u_i(x, 0) &= -A_i \operatorname{sech}^2 \left[\sqrt{\frac{A_i}{6}} (x - x_i^0) \right], \\
 v_i(x, 0) &= -2A_i c_i \sqrt{\frac{A_i}{6}} \operatorname{sech}^2 \left[\sqrt{\frac{A_i}{6}} (x - x_i^0) \right] \tanh \left[\sqrt{\frac{A_i}{6}} (x - x_i^0) \right], \\
 c_i &= \pm \left(1 - \frac{2A_i}{3} \right)^{\frac{1}{2}}, \quad i = 1, 2.
 \end{aligned}$$

This numerical test problem is conducted with the parameters $A_1 = 0.3$, $A_2 = 1$, $x_1^0 = -80$, $x_2 = -50$, $h = 0.1$, and $\Delta t = 0.01$ over the solution domain $[-100, 100]$. The initial conditions along with the above parameters represent two solitons which are initially located at $x_1^0 = -80$ and $x_2^0 = -50$, respectively, moving in the same direction with different velocities $c_1 = 0.8944271910$ and $c_2 = 0.5773502692$. The program is run over the time interval $[0, 180]$ to allow the interaction to take place. Then, two solitons are propagated to the right with velocities c_1, c_2 . Figure 7 shows the interactions of two solitons and as observed from the figure, the interaction has occurred and the faster wave interacted and separated from the slower wave and left it behind, which is consistent with the results given in [15]. Conservation constant for Crank Nicolson and proposed methods over the time interval are shown in Figures 8 and 9, respectively. The conservation quantities

during interaction scenario are reported in Table 4 which shows that the proposed method and Crank Nicolson method can preserves conserved quantities.

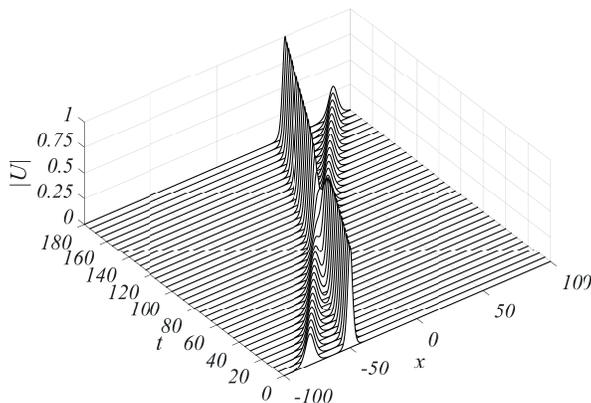


Figure 7. Overtaking soliton interaction.

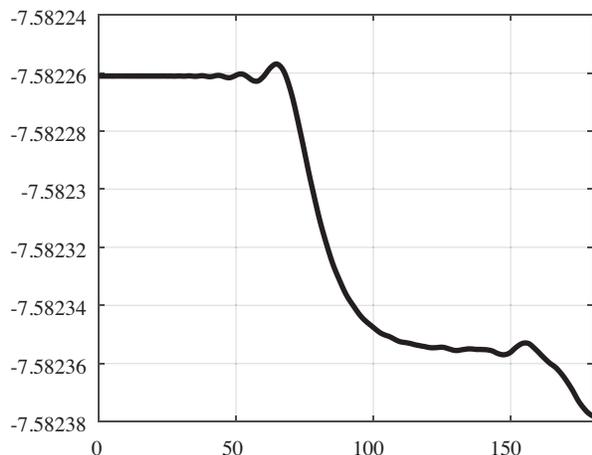


Figure 8. Conservation constant for Crank Nicolson method.

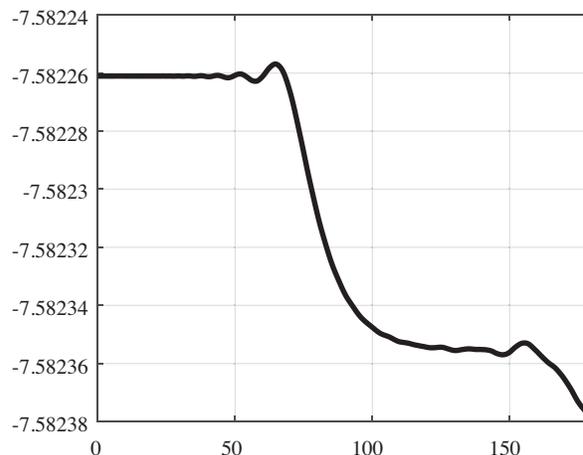


Figure 9. Conservation constant for proposed method.

3.4. Birth of the solitons

In this problem, we study the birth of solitons by using the initial conditions

$$u(x, 0) = -A \operatorname{sech}^2 \left[\sqrt{\frac{A}{6}} (x - \bar{x}_0) \right],$$

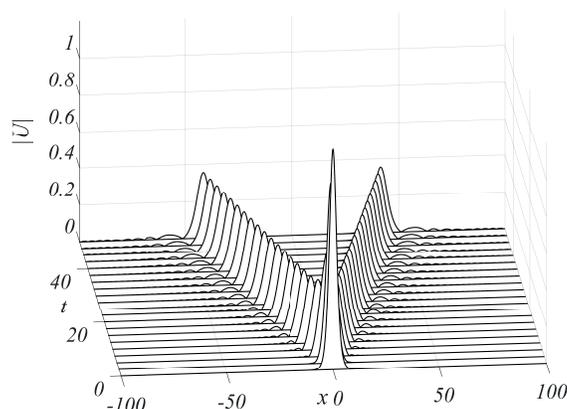
$$v(x, 0) = 0.$$

The computations are carried out by choosing the parameters as amplitude $A = 1.2$, $\bar{x}_0 = 0$, space step $h = 0.1$ and time step $\Delta t = 0.01$. The program is run up to time $t = 50$ over the interval $x \in [-100, 100]$ and then the birth of solitons is displayed in Figure 10 for the proposed method. It is noticed from the figure that as time progresses, the initial profile of the wave is divided into two solitons, moving in different directions with almost

Table 4. The comparison of conservation constant.

Time	Crank Nicolson method			Proposed method			
$t = 0$	-7.582261054	$t = 100$	-7.582347590	$t = 0$	-7.582261054	$t = 100$	-7.582345361
$t = 10$	-7.582261051	$t = 110$	-7.582352547	$t = 10$	-7.582261051	$t = 110$	-7.582349973
$t = 20$	-7.582261048	$t = 120$	-7.582354430	$t = 20$	-7.582261050	$t = 120$	-7.582352120
$t = 30$	-7.582261043	$t = 130$	-7.582355517	$t = 30$	-7.582261036	$t = 130$	-7.582352930
$t = 40$	-7.582261246	$t = 140$	-7.582355198	$t = 40$	-7.582261212	$t = 140$	-7.582351798
$t = 50$	-7.582260762	$t = 150$	-7.582356097	$t = 50$	-7.582260686	$t = 150$	-7.582353424
$t = 60$	-7.582261164	$t = 160$	-7.582355967	$t = 60$	-7.582261033	$t = 160$	-7.582350911
$t = 70$	-7.582266580	$t = 170$	-7.582365475	$t = 70$	-7.582266495	$t = 170$	-7.582366639
$t = 80$	-7.582308860	$t = 180$	-7.582378091	$t = 80$	-7.582307762	$t = 180$	-7.582375289
$t = 90$	-7.582336118			$t = 90$	-7.582334121		
Exact	-7.582261059		-7.582261059		-7.582261059		-7.582261059

equal amplitudes. The numerical values of conservation constant and the larger amplitude of the solitons at several times for both methods are given in Table 5 and the numerical values of conservation constant over the time interval for both methods are also drawn in Figures 11 and 12. When the tables and figures are examined, it is seen that similar results are obtained for both methods.

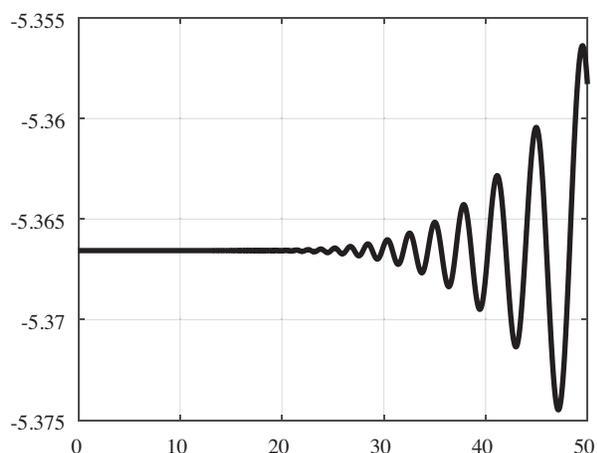
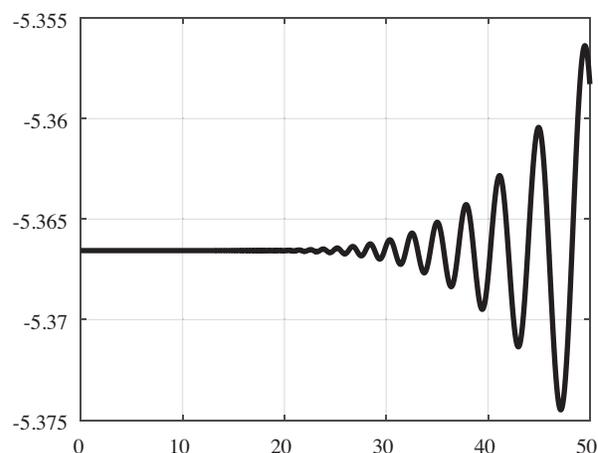
**Figure 10.** Birth of solitons.

4. Conclusion

In this paper, a fourth order one step method for time discretization has been developed to obtain the numerical solutions of GBqE. Truncation error analysis shows that our numerical schemes are consistent and fourth order accurate in time. Four numerical experiments related to single soliton, collision of two solitons moving in opposite directions, interaction of two solitons moving in the same direction and birth of the solitons have been considered as the test problems. To show how well and accurate the proposed method produces results, we have computed the error norm L_∞ for the first test problem and conservation constants for the three test

Table 5. The comparison of conservation constant.

Time	Crank Nicolson method		Proposed method	
	Amplitude	Invariants	Amplitude	Invariants
$t = 0$	1.2000000000	-5.366563146	1.2000000000	-5.366563146
$t = 10$	0.3678583685	-5.366563146	0.3678582023	-5.366563145
$t = 20$	0.3609667912	-5.366576500	0.3609663194	-5.366576256
$t = 30$	0.3579634254	-5.366314984	0.3579627074	-5.366311372
$t = 40$	0.3560696820	-5.367873404	0.3560693781	-5.367867649
$t = 50$	0.3546107137	-5.358279995	0.3546083743	-5.358283946
Exact		-5.366563146		-5.366563146

**Figure 11.** Conservation constant for Crank Nicolson method.**Figure 12.** Conservation constant for proposed method.

problems. The numerical outcomes also are compared with those obtained by different methods and technique. The obtained results demonstrate that the proposed method has considerable accurate and gives numerically reliable results for the numerical solution of the GBqE.

References

- [1] Mohebbi A, Asgari Z. Efficient numerical algorithms for the solution of “good” Boussinesq equation in water wave propagation. *Computer Physics Communications* 2011; 182: 2464-2470. doi: 10.1016/j.cpc.2011.07.004
- [2] Manoranjan VS, Mitchell AR, Morris JL. Numerical solutions of the good Boussinesq equation. *SIAM Journal on Scientific and Statistical Computing* 1984; 5(4): 946-957. doi: 10.1137/0905065
- [3] Ortega T, Sanz-Sema JM. Nonlinear stability and convergence of finite difference methods for the “good” Boussinesq equation. *Numerische Mathematik* 1990; 58: 215-229. doi: 10.1007/BF01385620
- [4] Pani AK, Saranga H. Finite element Galerkin method for the “good” Boussinesq equation. *Nonlinear Analysis: Theory, Methods and Applications* 1997; 29: 937-956. doi: 10.1016/S0362-546X(96)00093-4
- [5] El-Zoheiry H. Numerical investigation for the solitary waves interaction of the “good” Boussinesq equation. *Applied Numerical Mathematics* 2003; 45: 161-173. doi: 10.1016/S0168-9274(02)00187-3

- [6] Wazwaz AM. Construction of soliton solutions and periodic solutions of the Boussinesq equation by the modified decomposition method. *Chaos, Solitons and Fractals* 2001; 12: 549-1556. doi: 10.1016/S0960-0779(00)00133-8
- [7] Ismail MS, Bratsos AG. A predictor- corrector scheme for the numerical solutions of the Boussinesq equation. *Journal of Applied Mathematics and Computing* 2003; 13: 11-27. doi: 10.1007/BF02936071
- [8] Bratsos AG, Tsitouras Ch, Natsis DG. Linearized numerical schemes for the Boussinesq equation. *Applied Numerical Analysis and Computational Mathematics* 2005; 2: 34-53. doi: 10.1002/anac.200410021
- [9] Bratsos AG. The solution of the Boussinesq equation using the method of lines. *Computer Methods in Applied Mechanics and Engineering* 1998; 157(1-2): 33-44. doi: 10.1016/S0045-7825(97)00211-9
- [10] Bratsos AG. A second order numerical scheme for the solution of the one-dimensional Boussinesq equation. *Numerical Algorithms* 2007; 46: 45-58. doi: 10.1007/s11075-007-9126-y
- [11] Bratsos AG. Solitary-wave propagation and interactions for the “good” Boussinesq equation. *International Journal of Computer Mathematics* 2008; 85: 1431-1440. doi: 10.1080/00207160701509740
- [12] Al-Khaled K, Nusier AS. Numerical investigation for solitary solutions of the Boussinesq equation. *Applied Mathematics E - Notes* 2008; 8: 159-170
- [13] Dehghan M, Salehi RA. Meshless based numerical technique for traveling solitary wave solution of Boussinesq equation. *Applied Mathematical Modelling* 2012; 36(5): 1939-1956. doi: 10.1016/j.apm.2011.07.075
- [14] Siddiqi SS, Arshed S. Quintic B-spline for the numerical solution of the good Boussinesq equation. *Journal of the Egyptian Mathematical Society* 2014; 22(2): 209-213. doi: 10.1016/j.joems.2013.06.015
- [15] Ismail MS, Mosally F. A fourth order finite difference method for the good Boussinesq equation. *Abstract and Applied Analysis* 2014; Article ID 323260, 10 pages. doi: 10.1155/2014/323260
- [16] Ucar Y, Esen A, Karaagac B. Numerical solutions of Boussinesq equation using Galerkin finite element method. *Numerical Methods for Partial Differential Equations* 2020; 3782: 1612-1630. doi: 10.1002/num.22600