ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 12(6), DOI: 10.17485/ijst/2019/v12i6/141953, February 2019 ISSN (Online) : 0974-5645

New Cubic B-spline Approximations for Solving Non-linear Third-order Korteweg-de Vries Equation

Muhammad Abbas[1]*, Muhammad Kashif Iqbal[2], Bushra Zafar[3] and Shazalina Binti Mat Zin[4]

[1]Department of Mathematics, University of Sargodha, Sargodha, Pakistan; muhammad.abbas@uos.edu.pk

[2]Department of Mathematics, Government College University, Faisalabad, Pakistan; kashifiqbal@gcuf.edu.pk

[3]Department of Computer Science, Government College University, Faisalabad, Pakistan; bushrazafar@gcuf.edu.pk

[4]Institute of Engineering Mathematics, Universiti Malaysia Perlis, Arau, Perlis; shazalina@unimap.edu.my

Abstract

Objectives: In this work, the approximate solution of non-linear third order Korteweg-de Vries equation has been studied. **Methods:** The proposed numerical technique engages finite difference formulation for temporal discretization, whereas, the discretization in space direction is achieved by means of a new cubic B-spline approximation. **Findings:** In order to corroborate this effort, three test problems have been considered and the computational outcomes are compared with the current methods. It is found that the proposed scheme involves straight forward computations and operates superior to the existing methods. **Novelty/Improvements:** The proposed numerical scheme is novel for Korteweg-de Vries equation and has never been employed for this purpose before.

Keywords: Cubic B-spline Collocation Method, Cubic B-spline Functions, Finite Difference Formulation, Korteweg-de Vries Equation

1. Introduction

The third order non-linear Korteweg-de Vries (KdV) equation occurs in many physical applications such as non-linear plasma waves which exhibit certain dissipative effects[1], propagation of waves[2] and propagation of bores in shallow water waves[3]. The KdV equation is given by

$$u_t + \alpha u u_x + \beta u_{xx} + \gamma u_{xxx} = 0, x \in [a,b], t \in [0,T], \quad (1)$$

with conditions

$$\begin{aligned} &u(x,0) = g(x), \\ &u(a,t) = \phi_1(t), u(b,t) = \phi_2(t), u_x(b,t) = \phi_3(t), \end{aligned} \quad (2)$$

where, $u = u(x,t)$, α, β, γ are constants and $g(x), \phi_1(t), \phi_2(t), \phi_3(t)$ are known functions.

In recent years, the KdV equation has gained a considerable research attraction due to its numerous applications in real life phenomena. Especially, the traveling wave solution has been considered extensively. Kutluay et al.[4] employed integral methods with heat balance to study the small time solutions to KdV equation. The numerical solution to third order KdV equation was discussed by Bahadir[5] using exponential finite difference scheme. Ozer and Kultuay[6] proposed a numerical technique for solving KdV type equations. The authors in[7] employed the method of lines for small times solution of KdV equation. Dehghan and Shokri[8] proposed a numerical method based on multi-quadratic radial basis functions for solving KdV equation. Dag and Dereli[9] explored the numerical solution of KdV equation by

*Author for correspondence

means of radial basis functions. A mesh free method based on radial basis functions was presented by Khattak and Tirmizi[10] for approximate solution of KdV equation. Xiao et al.[11] investigated the numerical solution to KdV equation using multi-quadric quasi-interpolation operator. Sarboland and Aminataei[12] proposed a numerical scheme based on integrated radial basis functions and multi-quadric quasi-interpolation operator for solving of KdV equation. Rashid et al.[13] solved Hirota-Satsuma coupled KdV equation by Fourier Pseudo-spectral method.

The spline functions are used extensively to solve the initial and boundary value problems. These functions preserve a smoothness at the nodes and have the ability to provide the numerical solution in the entire domain with great accuracy. Irk et al.[14] employed quadratic polynomial splines for small time solution to KdV equation. The second degree B-spline functions together with Galerkin finite-element method were used by Aksan and Ozdes[15] for solving one dimensional KdV equation. Saka[16] employed differential quadrature method for solving KdV equation. Canivar et al.[17] studied the numerical solution of KdV equation by means of third degree B-spline functions. Yu et al.[18] proposed blended basis splines for numerical solution of KdV equation. The spline finite-element and collocation methods have been discussed by Micula and Micula[19] for solving KdV-Burger equation. Ersoy and Dag[20] proposed exponential cubic basis splines for numerical solution of KdV equation. The modified exponential B-spline collocation method has been proposed by Raslan et al.[21] for numerical solution of one dimensional KdV equation. Lakestani[22] presented a numerical scheme based on finite difference method and B-spline functions for solving third order non-linear KdV equation. Dong[23] developed a new hybrid discontinuous Galerkin approach for numerical solution of KdV equation.

In this work, the numerical solution of non-linear KdV equation has been considered. The usual finite difference scheme[24] and new Cubic B-Spline (CBS) approximations[25,26] have been used for temporal and spatial discretization respectively.

The roadways of this study is: In section 2, we shall discuss some preliminaries of ordinary CBS interpolation. The numerical method is presented in section 3 and experimental outcomes are given in section 4.

2. Cubic B-spline Functions

We uniformly partition the spatial domain $[a,b]$ into $n+1$ equidistant knots as $x_i = x_0 + ih, i = 0(1)n$ with $h = \frac{1}{n}(b-a)$. The p^{th} B-spline function of degree r, order $r+1$, is defined as[27]

For $r=0$

$$B_{0,p}(x) = \begin{cases} 1, & \text{if } x \in [x_p, x_{p+1}] \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

For $r>0$ and $x \in [x_p, x_{p+1+r}]$

$$B_{r,p}(x) = \frac{(x-x_p)}{(x_{p+r}-x_p)} B_{r-1,p}(x) + \left(1 - \frac{(x-x_{p+1})}{(x_{p+1+r}-x_{p+1})}\right) B_{r-1,p+1}(x). \tag{4}$$

Using(4), the typical CBS functions are defined as[28]

$$B_p(x) = \frac{1}{6h^3}\begin{cases} (x-x_{p-2})^3, & x \in [x_{p-2}, x_{p-1}] \\ h^3 + 3h^2(x-x_{p-1}) + 3h(x-x_{p-1})^2 - 3(x-x_{p-1})^3, & x \in [x_{p-1}, x_p] \\ h^3 + 3h^2(x_{p+1}-x) + 3h(x_{p+1}-x)^2 - 3(x_{p+1}-x)^3, & x \in [x_p, x_{p+1}] \\ (x_{p+1}-x)^3, & x \in [x_{p+1}, x_{p+2}] \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where, $p = -1(1)n+1$. For a sufficiently smooth function $u(x,t)$ there always exists a unique third degree spline $U(x,t)$, which satisfies the prescribed interpolating conditions such that

$$U(x,t) = \sum_{p=-1}^{n+1} c_p(t) B_p(x), \tag{6}$$

where, $c_p(t)$'s are, time dependent real constants, yet to be calculated. For simplicity, we express the CBS approximations $U(x_i)$, $U'(x_i)$, $U''(x_i)$ and $U'''(x_i)$ by U_i, m_i, M_i and T_i respectively. The third degree basis spline functions (5) together with (6) yield the following relations

$$U_i = \sum_{p=i-1}^{i+1} c_p B_p(x) = \frac{1}{6}(c_{i-1} + 4c_i + c_{i+1}) \tag{7}$$

$$m_i = \sum_{p=i-1}^{i+1} c_p B_p'(x) = \frac{1}{2h}(-c_{i-1} + c_{i+1}) \tag{8}$$

Moreover, for second and third order derivatives, we shall use the following new CBS approximations[25,26]

$$M_i = \frac{1}{12h^2}\begin{cases} 14c_{-1} - 33c_0 + 28c_1 - 14c_2 + 6c_3 - c_4, & \text{for } i = 0 \\ c_{i-2} + 8c_{i-1} - 18c_i + 8c_{i+1} + c_{i+2}, & \text{for } i = 1(1)n-1 \\ -c_{n-4} + 6c_{n-3} - 14c_{n-2} + 28c_{n-1} - 33c_n + 14c_{n+1} & \text{for } i = n \end{cases} \tag{9}$$

$$T_i = \frac{1}{360h^3}\begin{cases} 10\left(-68c_{-1} + 249c_0 - 351c_1 + 238c_2 - 78c_3 + 9c_4 + c_5\right), & \text{for } i = 0 \\ 5\left(-31c_{-1} + 48c_0 + 45c_1 - 112c_2 + 51c_3 + 0c_4 - c_5\right), & \text{for } i = 1 \\ 10c_{-1} - 213c_0 + 378c_1 + 55c_2 - 450c_3 + 225c_4 - 2c_5 - 3c_6, & \text{for } i = 2 \\ 2c_{i-4} - 2c_{i-3} - 183c_{i-2} + 338c_{i-1} + 85c_i - 462c_{i+1} + 227c_{i+2} - 2c_{i+3} - 3c_{i+4}, & \text{for } i = 3(1)n-3 \\ 2\left(c_{n-6} - c_{n-5} - 90c_{n-4} + 160c_{n-3} + 65c_{n-2} - 261c_{n-1} + 136c_n - 10c_{n+1}\right), & \text{for } i = n-2 \\ 5\left(c_{n-5} + 0c_{n-4} - 51c_{n-3} + 112c_{n-2} - 45c_{n-1} - 48c_n + 31c_{n+1}\right), & \text{for } i = n-1 \\ 10\left(-c_{n-5} - 9c_{n-4} + 78c_{n-3} - 238c_{n-2} + 351c_{n-1} - 249c_n + 68c_{n+1}\right) & \text{for } i = n \end{cases} \tag{10}$$

3. Description of the Numerical Method

In this section, we present the numerical scheme for solving non-linear KdV equation. Applying usual finite difference method and θ weighted scheme, the problem is discretized in time direction as

$$\frac{u^{j+1} - u^j}{\Delta t} + \theta\left[\alpha\left(uu_x\right)^{j+1} + \beta u_{xx}^{j+1} + \gamma u_{xxx}^{j+1}\right] + (1-\theta)\left[\alpha u^j u_x^j + \beta u_{xx}^j + \gamma u_{xxx}^j\right] = 0. \tag{11}$$

where, Δt is the step size in time direction, $0 \le \theta \le 1$ and u_i^{j+1} is used to denote $u\left(x_i, t_j + \Delta t\right)$. The non-linear term $\left(uu_x\right)^{j+1}$ is linearized as[29,30]

$$\left(uu_x\right)^{j+1} = u^{j+1}u_x^j + u^j u_x^{j+1} - u^j u_x^j. \tag{12}$$

Substituting (12) into (11), we get

$$\frac{u^{j+1} - u^j}{\Delta t} + \theta\left[\alpha(u^{j+1}u_x^j + u^j u_x^{j+1} - u^j u_x^j) + \beta u_{xx}^{j+1} + \gamma u_{xxx}^{j+1}\right] + (1-\theta)\left[\alpha u^j u_x^j + \beta u_{xx}^j + \gamma u_{xxx}^j\right] = 0. \tag{13}$$

For $\theta = \frac{1}{2}$, the relation (13) can be rearranged as

$$\left[\frac{2}{\Delta t} + \alpha u_x^j\right]u^{j+1} + \alpha u^j u_x^{j+1} + \beta u_{xx}^{j+1} + \gamma u_{xxx}^{j+1} = \frac{2}{\Delta t}u^j - \beta u_{xx}^j - \gamma u_{xxx}^j. \tag{14}$$

Substituting the approximation for u and its derivatives at the knot x_i, equation (14) takes the following form

$$w_i^j U_i^{j+1} + y_i^j m_i^{j+1} + \beta M_i^{j+1} + \gamma T_i^{j+1} = z_i^j, \tag{15}$$

where, $w_i^j = \frac{2}{\Delta t} + \alpha m_i^j$, $y_i^j = \alpha U_i^j$ and

$$z_i^j = \frac{2}{\Delta t}U_i^j - \beta M_i^j - \gamma T_i^j.$$

Using (7)–(10) in (15), for $i = 0,1,2,3,\cdots,n-1$, we obtain the following linear equations involving $n+3$ unknowns.

$$\begin{aligned} &\frac{w_0^j}{6}\left(c_{-1}^{j+1} + 4c_0^{j+1} + c_1^{j+1}\right) + \frac{y_0^j}{2h}\left(-c_{-1}^{j+1} + c_1^{j+1}\right) \\ &+ \frac{\beta}{12h^2}\begin{pmatrix} 14c_{-1}^{j+1} - 33c_0^{j+1} + 28c_1^{j+1} - 14c_2^{j+1} \\ +6c_3^{j+1} - c_4^{j+1} \end{pmatrix} \\ &+ \frac{\gamma}{36h^3}\left(-68c_{-1}^{j+1} + 249c_0^{j+1} - 351c_1^{j+1} + 238c_2^{j+1} - 78c_3^{j+1} + 9c_4^{j+1} + c_5^{j+1}\right) = z_0^j, \end{aligned} \tag{16}$$

$$\begin{aligned} &\frac{w_1^j}{6}\left(c_0^{j+1} + 4c_1^{j+1} + c_2^{j+1}\right) + \frac{y_1^j}{2h}\left(-c_0^{j+1} + c_2^{j+1}\right) \\ &+ \frac{\beta}{12h^2}\left(c_{-1}^{j+1} + 8c_0^{j+1} - 18c_1^{j+1} + 8c_2^{j+1} + c_3^{j+1}\right) \\ &+ \frac{\gamma}{72h^3}\left(-31c_{-1}^{j+1} + 48c_0^{j+1} + 45c_1^{j+1} - 112c_2^{j+1} + 51c_3^{j+1} + 0c_4^{j+1} - c_5^{j+1}\right) = z_1^j, \end{aligned} \tag{17}$$

$$\begin{aligned} &\frac{w_2^j}{6}\left(c_1^{j+1} + 4c_2^{j+1} + c_3^{j+1}\right) + \frac{y_2^j}{2h}\left(-c_1^{j+1} + c_3^{j+1}\right) \\ &+ \frac{\beta}{12h^2}\left(c_0^{j+1} + 8c_1^{j+1} - 18c_2^{j+1} + 8c_3^{j+1} + c_4^{j+1}\right) \\ &+ \frac{\gamma}{360h^3}\left(10c_{-1}^{j+1} - 213c_0^{j+1} + 378c_1^{j+1} + 55c_2^{j+1} - 450c_3^{j+1} + 225c_4^{j+1} - 2c_5^{j+1} - 3c_6^{j+1}\right) = z_2^j, \end{aligned} \tag{18}$$

$$\begin{aligned} &\frac{w_i^j}{6}\left(c_{i-1}^{j+1} + 4c_i^{j+1} + c_{i+1}^{j+1}\right) + \frac{y_i^j}{2h}\left(-c_{i-1}^{j+1} + c_{i+1}^{j+1}\right) \\ &+ \frac{\beta}{12h^2}\left(c_{i-2}^{j+1} + 8c_{i-1}^{j+1} - 18c_i^{j+1} + 8c_{i+1}^{j+1} + c_{i+2}^{j+1}\right) \\ &+ \frac{\gamma}{360h^3}\left(2c_{i-4} - 2c_{i-3} - 183c_{i-2} + 338c_{i-1} + 85c_i - 462c_{i+1} + 227c_{i+2} - 2c_{i+3} - 3c_{i+4}\right) = z_i^j, \\ &\qquad i = 3,4,5,\ldots,n-3, \end{aligned} \tag{19}$$

$$\begin{aligned} &\frac{w_{n-2}^j}{6}\left(c_{n-3}^{j+1} + 4c_{n-2}^{j+1} + c_{n-1}^{j+1}\right) + \frac{y_{n-2}^j}{2h}\left(-c_{n-3}^{j+1} + c_{n-1}^{j+1}\right) \\ &+ \frac{\beta}{12h^2}\left(c_{n-4}^{j+1} + 8c_{n-3}^{j+1} - 18c_{n-2}^{j+1} + 8c_{n-1}^{j+1} + c_n^{j+1}\right) \\ &+ \frac{\gamma}{180h^3}\left(c_{n-6}^{j+1} - c_{n-5}^{j+1} - 90c_{n-4}^{j+1} + 160c_{n-3}^{j+1} + 65c_{n-2}^{j+1} - 261c_{n-1}^{j+1} + 136c_n^{j+1} - 10c_{n+1}^{j+1}\right) \square z_{n-2}^j, \end{aligned} \tag{20}$$

$$\begin{aligned} &\frac{w_{n-1}^j}{6}\left(c_{n-2}^{j+1} + 4c_{n-1}^{j+1} + c_n^{j+1}\right) + \frac{y_{n-1}^j}{2h}\left(-c_{n-2}^{j+1} + c_n^{j+1}\right) \\ &+ \frac{\beta}{12h^2}\left(c_{n-3}^{j+1} + 8c_{n-2}^{j+1} - 18c_{n-1}^{j+1} + 8c_n^{j+1} + c_{n+1}^{j+1}\right) \\ &+ \frac{\gamma}{72h^3}\left(c_{n-5}^{j+1} + 0c_{n-4}^{j+1} - 51c_{n-3}^{j+1} + 112c_{n-2}^{j+1} - 45c_{n-1}^{j+1} - 48c_n^{j+1} + 31c_{n+1}^{j+1}\right) = z_{n-1}^j. \end{aligned} \tag{21}$$

Three more equations are obtained from the boundary conditions (2) as

$$U_0^{j+1} = \phi_1\left(t_{j+1}\right) \tag{22}$$

$$U_n^{j+1} = \phi_2\left(t_{j+1}\right) \tag{23}$$

$$m_n^{j+1} = \phi_3\left(t_{j+1}\right) \tag{24}$$

The set of equations (16)–(24) can be written in matrix from as

$$AC^{j+1} = B, \tag{25}$$

where A denotes the coefficient matrix of order $n+3$, B is column matrix of order $n+3$ and $C^{j+1} = \left[c_{-1}^{j+1}\ c_0^{j+1}\ c_1^{j+1} \cdots c_{n+1}^{j+1}\right]^T$ is the set of control points at the $(j+1)^{th}$ time level.

Before starting any computation using (25), we obtain the following three equations from initial condition (2)

$$m_0^0 = g'(x_0), \tag{26}$$

$$U_i^0 = g(x_i),\ i = 0(1)n, \tag{27}$$

$$m_n^0 = g'(x_n). \tag{28}$$

Using (7)–(8), we get

$$-c_{i-1}^0 + c_{i+1}^0 = 2h\,g'(x_0), \tag{29}$$

$$c_{i-1}^0 + 4c_i^0 + c_{i+1}^0 = 6g(x_i),\ i = 0(1)n, \tag{30}$$

$$-c_{n-1}^0 + c_{n+1}^0 = 2h\,g'(x_n). \tag{31}$$

The above system can be expressed in matrix form as

$$AC^0 = B. \tag{32}$$

The unknown column vector C^0 is determined by well-known Thomas algorithm. The numerical computations are executed in *Mathematica 9*.

4. Numerical Results

In this section, the approximate solution to (1)–(2) is presented. The accuracy and validity of the proposed numerical method is tested by three error norms L_∞, L_2 and Root Mean Square (RMS), which are calculated as

$$L_\infty = \max_i |U_i - u_i|,\ L_2 = \sqrt{\sum_{i=0}^{n}(U_i - u_i)^2},\ \text{RMS} = \sqrt{\frac{\sum_{i=0}^{n}(U_i - u_i)^2}{n+1}}$$

where, U_i and u_i represent the numerical and exact solutions at the i^{th} knot respectively. The approximate results are compared with Multi-Quadratic Radial Basis Functions (MQRBF)[8], Multi-Quadric (MQ) and Inverse Multi-Quadric (IMQ) radial basis functions method[10], Multi-Quadric Quasi-Interpolation (MQQI) approach[11] and integrated multi-quadric quasi-interpolation (IMQQI) method[12].

Example 1:

Consider the following KdV equation[8,10–12]

$$u_t + 6uu_x + u_{xxx} = 0,\ x \in [a,b],\ t \in [0,T]$$

$$u(x,0) = \frac{\lambda}{2}\operatorname{sech}^2\left(\frac{\sqrt{\lambda}}{2}x - \mu\right)$$

The exact solution is $u(x,t) = \frac{\lambda}{2}\operatorname{sech}^2\left(\frac{\sqrt{\lambda}}{2}(x - \lambda t) - \mu\right)$.

The error norms L_∞, L_2 and RMS are listed in Tables 1–3, when $n = 200$ and $\Delta t = 0.01$. It is revealed that the proposed numerical scheme produces more reliable and accurate results as compared to MQRBF[8], MQ[10], IMQ[10], MQQI[11] and IMQQI[12]. Figure 1 shows a very close agreement of the numerical solution with closed form solution for $t = 1,3,5$. Three dimensional plots of exact and approximate solutions are shown in Figures 2 and 3. The absolute computational error using $n = 200$, $\Delta t = 0.01$ is displayed in Figure 4.

Table 1. Absolute numerical error for Example 1, when $0 \le x \le 40$, $0 \le t \le 5$, $\lambda = 0.5$, $\mu = 7$

t	MQ[10] $\Delta t = 0.001$	IMQ[10] $\Delta t = 0.001$	MQQI[11] $\Delta t = 0.001$	Proposed method $\Delta t = 0.01$
1	1.79×10^{-5}	6.96×10^{-5}	1.53×10^{-3}	8.63×10^{-6}
2	3.01×10^{-5}	1.96×10^{-4}	2.87×10^{-3}	1.11×10^{-5}
3	3.98×10^{-5}	3.83×10^{-3}	4.14×10^{-3}	1.26×10^{-5}
4	4.78×10^{-5}	5.91×10^{-3}	5.39×10^{-3}	1.36×10^{-5}
5	5.46×10^{-5}	8.37×10^{-3}	6.81×10^{-3}	1.45×10^{-5}

Table 2. Error norms for Example 1, when $0 \le x \le 40,\ 0 \le t \le 5,\ \lambda = 0.5,\ \mu = 7$

t		L_∞	L_2	RMS
1	Proposed method IMQQI[12]	8.63×10^{-6} 1.67×10^{-5}	3.22×10^{-5} 6.00×10^{-4}	2.27×10^{-6} 4.23×10^{-5}
2	Proposed method IMQQI[12]	1.11×10^{-5} 2.38×10^{-4}	4.33×10^{-5} 9.22×10^{-4}	3.06×10^{-6} 6.51×10^{-5}
3	Proposed method IMQQI[12]	1.26×10^{-5} 2.38×10^{-4}	4.94×10^{-5} 1.13×10^{-4}	3.48×10^{-6} 8.00×10^{-5}
4	Proposed method IMQQI[12]	1.36×10^{-5} 3.14×10^{-4}	5.33×10^{-5} 1.29×10^{-3}	3.76×10^{-6} 9.12×10^{-5}
5	Proposed method IMQQI[12]	1.45×10^{-5} 3.41×10^{-4}	5.64×10^{-5} 1.42×10^{-3}	3.98×10^{-6} 1.00×10^{-4}

Table 3. Error norms for Example 1, when $30 \le x \le 80,\ 0 \le t \le 10,\ \lambda = 0.14,\ \mu = 10$

t		Δt	L_∞	L_2	RMS
1	Proposed method MQRBF[8]	0.01 0.001	2.00×10^{-7} 6.89×10^{-6}	7.24×10^{-7} 2.14×10^{-5}	5.11×10^{-8} 1.35×10^{-6}
2	Proposed method MQRBF[8]	0.01 0.001	4.43×10^{-7} 8.60×10^{-6}	1.84×10^{-6} 3.50×10^{-5}	1.30×10^{-7} 2.21×10^{-6}
3	Proposed method MQRBF[8]	0.01 0.001	5.84×10^{-7} 8.40×10^{-6}	2.60×10^{-6} 4.10×10^{-5}	1.83×10^{-7} 2.59×10^{-6}
4	Proposed method MQRBF[8]	0.01 0.001	6.84×10^{-7} 9.21×10^{-6}	3.14×10^{-6} 4.28×10^{-5}	2.21×10^{-7} 2.70×10^{-6}
5	Proposed method MQRBF[8]	0.01 0.001	7.87×10^{-7} 8.56×10^{-6}	3.71×10^{-6} 4.55×10^{-5}	2.61×10 2.87×10^{-6}

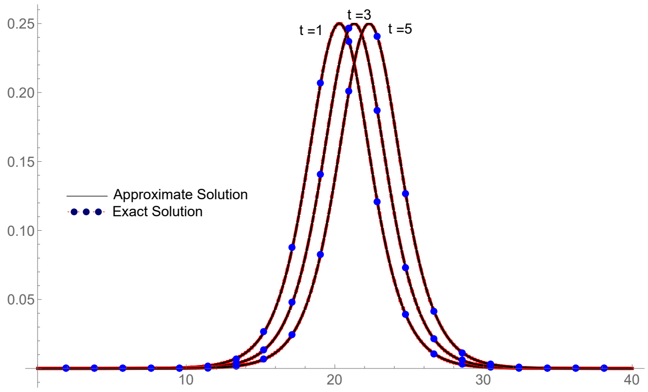

Figure 1. Numerical and exact solution for Example 1, when $t = 1,3,5$ and $n = 200$, $\Delta t = 0.01$, $0 \le x \le 40$, $\lambda = 0.5$, $\mu = 7$.

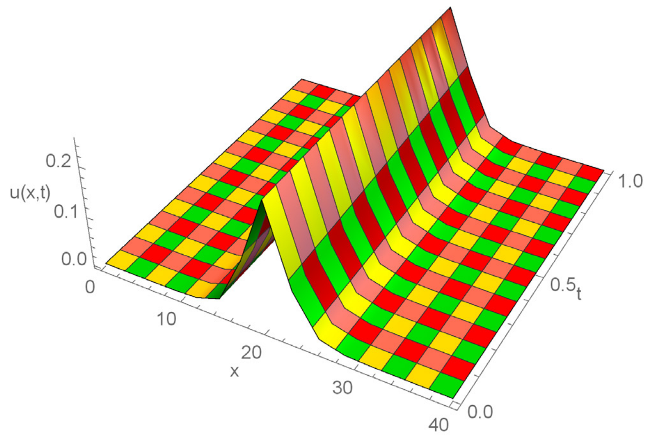

Figure 2. Exact solution for Example 1, when $0 \le x \le 40$, $0 \le t \le 1$, $\lambda = 0.5$, $\mu = 7$.

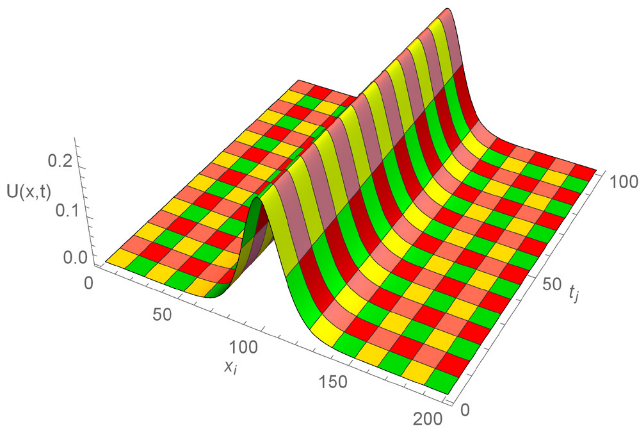

Figure 3. Approximate solution for Example 1, with $0 \le x \le 40$, $0 \le t \le 1$, $\lambda = 0.5$, $\mu = 7$, $n = 200$, $\Delta t = 0.01$.

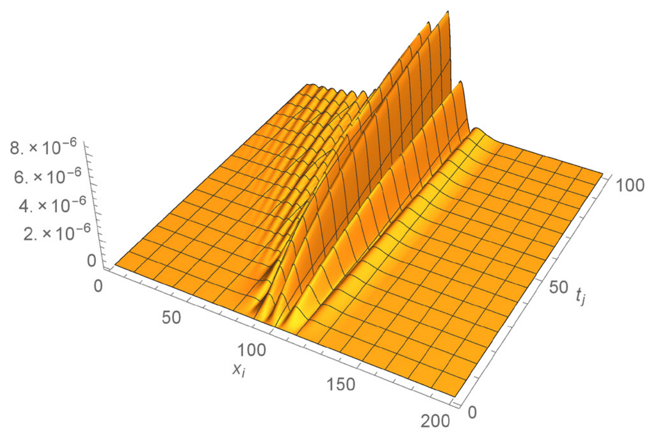

Figure 4. Absolute error for Example 1, with $0 \le x \le 40$, $0 \le T \le 1$, $\lambda = 0.5$, $\mu = 7$, $n = 200$, $\Delta t = 0.01$.

Example 2:

Consider the following KdV equation[23]

$$u_t + 6uu_x + u_{xxx} = 0,\ x \in [a,b], t \in [0,T],$$
$$u(x,0) = 2\ \text{sech}^2(x+4).$$

The exact solution is $u(x,t) = 2\,\text{sech}^2(x - 4t + 4)$. The computational error norms L_∞, L_2 and RMS are listed in Table 4 when n = 200 and Δt = 0.01. Figure 5 shows the approximate and exact solution at $t = 0.2, 0.4, 0.6, 0.8, 1$. The three dimensional plots of analytical and approximate solutions are displayed in Figures 6 and 7. The absolute computational error is portrayed in Figure 8 using n = 200 and Δt = 0.01.

Table 4. Error norms for Example 2, when $-10 \le x \le 0$, $0 \le t \le 1$, $\lambda = 0.5$

t	L_∞	L_2	RMS
0.2	3.39×10^{-5}	1.82×10^{-4}	1.29×10^{-5}
0.4	3.49×10^{-5}	2.40×10^{-4}	1.69×10^{-5}
0.6	5.12×10^{-5}	3.41×10^{-4}	2.40×10^{-5}
0.8	8.21×10^{-5}	4.76×10^{-4}	3.36×10^{-5}
1.0	8.09×10^{-5}	4.30×10^{-4}	3.03×10^{-5}

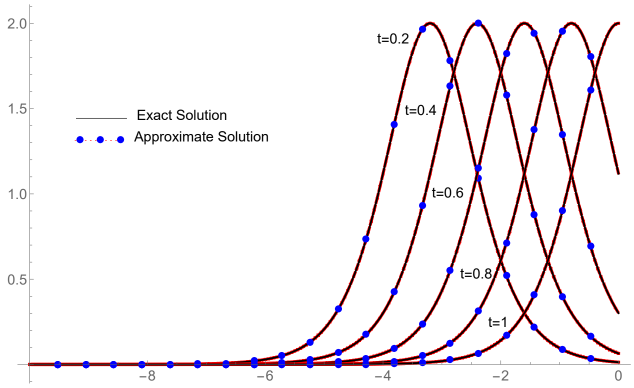

Figure 5. Numerical and exact solution for Example 2, when $n = 200,\ \Delta t = 0.01,\ -10 \le x \le 0$.

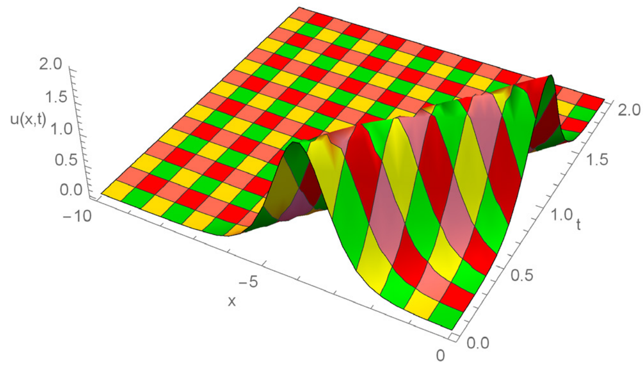

Figure 6. Exact solution for Example 2, when $-10 \le x \le 0,\ 0 \le t \le 2$.

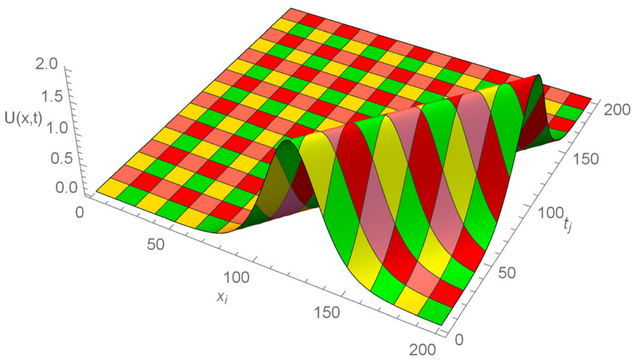

Figure 7. Approximate solution for Example 2, with $-10 \le x \le 0,\ 0 \le t \le 2,\ n = 200,\ \Delta t = 0.01$.

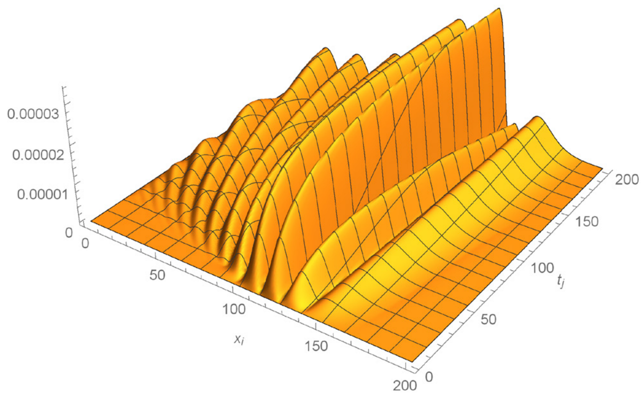

Figure 8. Absolute error for Example 2, with $-10 \le x \le 0,\ 0 \le t \le 0.2,\ n = 200,\ \Delta t = 0.001$.

5. Conclusion

In this paper, numerical solution of non-linear third order KdV equation has been explored. We conclude the outcomes of this research as:

1. The presented algorithm is based on usual finite difference scheme and CBS collocation method.
2. The proposed technique is novel for third order nonlinear KdV equation.
3. Usual finite difference scheme has been employed for temporal discretization.
4. The new CBS approximations have been used to interpolate the solution in space direction.
5. Due to straightforward and simple application, it outperforms the MQRBF[8], MQ[10], IMQ[10], MQQI[11] and IMQQI[12] approaches.

6. Acknowledgments

This study was fully supported by Travel Grant of No. 305.28/TG/R&D/HEC/2018/29239 from Higher Education Commission, Islamabad to present research paper at ICCEMS 2018. The first author does acknowledge University of Sargodha, Sargodha, Pakistan for providing ex-Pakistan leave. The authors are also indebted to the anonymous reviewers for their helpful, valuable comments and suggestions in the improvement of this manuscript.

7. References

1. Grad H, Hu PN. Unified shock profile in a plasma. The Physics of Fluids. 1967; 10(12):2596–602. https://doi.org/10.1063/1.1762081

2. Johnson R., A non-linear equation incorporating damping and dispersion. Journal of Fluid Mechanics. 1970; 42(1): 49–60. https://doi.org/10.1017/S0022112070001064
3. Johnson R., Shallow water waves on a viscous fluid—the undular bore. The Physics of Fluids. 1972; 15(10):1693–9. https://doi.org/10.1063/1.1693764
4. Kutluay S, Bahadir A, Özdeş A. A small time solution for the Korteweg–de Vries equation. Applied Mathematics and Computation. 2000; 107(2–3):203–10. https://doi.org/10.1016/S0096-3003(98)10119-4
5. Bahadır AR. Exponential finite-difference method applied to Korteweg–de Vries equation for small times. Applied Mathematics and Computation. 2005; 160(3):675–82. https://doi.org/10.1016/j.amc.2003.11.025
6. Özer S, KutluayS. An analytical–numerical method for solving the Korteweg–de Vries equation. Applied Mathematics and Computation. 2005; 164(3):789–97. https://doi.org/10.1016/j.amc.2004.06.011
7. Özdeş A, Aksan E. The method of lines solution of the Korteweg-de Vries equation for small times. International Journal of Contemporary Mathematical Sciences. 2006; 1(13):639–50. https://doi.org/10.12988/ijcms.2006.06067
8. Dehghan M, Shokri A. A numerical method for KdV equation using collocation and radial basis functions. Nonlinear Dynamics. 2007; 50(1–2):111–20. https://doi.org/10.1007/s11071-006-9146-5
9. Dağ İ, Dereli Y. Numerical solutions of KDV equation using radial basis functions. Applied Mathematical Modelling. 2008; 32(4):535–46. https://doi.org/10.1016/j.apm.2007.02.001
10. Khattak AJ, Tirmizi IA. A meshfree method for numerical solution of KdV equation. Engineering Analysis with Boundary Elements. 2008; 32(10):849–55. https://doi.org/10.1016/j.enganabound.2008.01.003
11. Lu XM, Hong WR, Gang ZC. Applying multiquadric quasi-interpolation to solve KdV equation. Journal of Mathematical Research and Exposition. 2011.
12. Sarboland M, Aminataei A. On the numerical solution of the nonlinear Korteweg–de Vries equation. Systems Science and Control Engineering. 2015; 3(1):69–80. https://doi.org/10.1080/21642583.2014.986340
13. Rashid A, Lu D, Md. Ismail AI, Abbas M. Numerical solution of the generalized Hirota-Satsuma coupled Korteweg-de Vries equation by Fourier Pseudospectral method. Journal of Computational Analysis and Applications. 2017; 23(8):1412–23.
14. Irk D, Dag I, Saka B. A small time solution for the Korteweg–de Vries equation using spline approximation. Applied Mathematics and Computation. 2006; 173(2):834–46. https://doi.org/10.1016/j.amc.2005.04.018
15. Aksan E, Özdeş A. Numerical solution of Korteweg–de Vries equation by Galerkin B-spline finite element method. Applied Mathematics and Computation. 2006; 175(2):1256–65. https://doi.org/10.1016/j.amc.2005.08.038
16. Saka B. Cosine expansion-based differential quadrature method for numerical solution of the KdV equation. Chaos, Solitons and Fractals. 2009; 40(5):2181–90. https://doi.org/10.1016/j.chaos.2007.10.004
17. Canıvar A, Sari M, Dag I. A Taylor–Galerkin finite element method for the KdV equation using cubic B-splines. Physica B: Condensed Matter. 2010; 405(16):3376–83. https://doi.org/10.1016/j.physb.2010.05.008
18. Yu R, Wang R, Zhu C. A numerical method for solving KdV equation with blended b-spline quasi-interpolation. Journal of Information and Computational Science. 2013; 10(16):5093–101.https://doi.org/10.12733/jics20102180
19. Micula G, Micula M. On the numerical approach of Korteweg-de Vries-Burger equations by spline finite element and collocation methods. Seminar on Fixed Point Theory Cluj-Napoca; 2002.
20. Ersoy O, Dag I. The exponential cubic B-spline algorithm for Korteweg-de Vries equation. Advances in Numerical Analysis; 2015.
21. Raslan K, El-Danaf TS, Ali KK. New numerical treatment for solving the KDV equation. Journal of Abstract and Computational Mathematics. 2017; 2(1):1–12.
22. Lakestani M. Numerical solutions of the KdV equation using B-spline functions. Iranian Journal of Science and Technology, Transactions A: Science. 2017; 41(2):409–17. https://doi.org/10.1007/s40995-017-0260-7
23. Dong B. Optimally convergent HDG method for third-order Korteweg–de Vries type equations. Journal of Scientific Computing. 2017; 73(2–3):712–35. https://doi.org/10.1007/s10915-017-0437-4
24. Zin SM, et al., A new trigonometric spline approach to numerical solution of generalized nonlinear Klien-Gordon equation. PloS One. 2014; 9(5):e95774. https://doi.org/10.1371/journal.pone.0095774. PMid:24796483. PMCid:PMC4010414
25. Lang F-G, Xu X-P. A new cubic B-spline method for approximating the solution of a class of nonlinear second-order boundary value problem with two dependent variables. Science Asia. 2014; 40: 444–50. https://doi.org/10.2306/scienceasia1513-1874.2014.40.444
26. Iqbal MK, Abbas M, Wasim I. New cubic B-spline approximation for solving third order Emden–Flower type equations. Applied Mathematics and Computation. 2018; 331:319–33. https://doi.org/10.1016/j.amc.2018.03.025
27. De Boor C. A practical guide to splines. Springer-Verlag New York. 1978; 27. https://doi.org/10.1007/978-1-4612-6333-3

28. Abbas M, Abd. Majid A, Md. Ismail AI, Rashid A. Numerical method using cubic B-spline for a strongly coupled reaction-diffusion system. PloS One. 2014; 9(1):e83265. https://doi.org/10.1371/journal.pone.0083265.PMid:24427270. PMCid:PMC3888394
29. Mittal R, Arora G. Quintic B-spline collocation method for numerical solution of the Kuramoto–Sivashinsky equation. Communications in Nonlinear Science and Numerical Simulation. 2010; 15(10):2798–808. https://doi.org/10.1016/j.cnsns.2009.11.012
30. Rashidinia J, Jokar M. Polynomial scaling functions for numerical solution of generalized Kuramoto–Sivashinsky equation. Applicable Analysis, 2017; 96(2):293–306. https://doi.org/10.1080/00036811.2015.1132310