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

Muhammad Abbas¹*, Muhammad Kashif Iqbal², Bushra Zafar³ and Shazalina Binti Mat Zin⁴

¹Department of Mathematics, University of Sargodha, Sargodha, Pakistan; muhammad.abbas@uos.edu.pk
²Department of Mathematics, Government College University, Faisalabad, Pakistan; kashifiqbal@gcuf.edu.pk
³Department of Computer Science, Government College University, Faisalabad, Pakistan; bushrazafar@gcuf.edu.pk
⁴Institute 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 straightforward 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, propagation of waves and propagation of bores in shallow water waves. The KdV equation is given by

\[ u_t + uu_x + \beta u_{xx} + \gamma u_{xxx} = 0, \quad x \in [a, b], \quad t \in [0, T], \]

with conditions

\[ u(x,0) = g(x), \]

\[ u(a,t) = \phi_1(t), \quad u(b,t) = \phi_2(t), \quad u_x(a,t) = \phi_3(t), \]

where, \( u = u(x,t), \) \( \alpha, \beta, \gamma \) 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. 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 using exponential finite difference scheme. Ozer and Kutluay proposed a numerical technique for solving KdV type equations. The authors in employed the method of lines for small times solution of KdV equation. Dehghan and Shokri proposed a numerical method based on multi-quadratic radial basis functions for solving KdV equation. Dag and Dereli 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 for approximate solution of KdV equation. Xiao et al. investigated the numerical solution to KdV equation using multi-quadratic quasi-interpolation operator. Sarboland and Aminataei proposed a numerical scheme based on integrated radial basis functions and multi-quadratic quasi-interpolation operator for solving of KdV equation. Rashid et al. 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. 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 for solving one dimensional KdV equation. Sakas employed differential quadrature method for solving KdV equation. Canivar et al. studied the numerical solution of KdV equation by means of third degree B-spline functions. Yu et al. proposed exponential cubic basis splines for numerical solution of KdV equation. The spline finite-element and collocation methods have been discussed by Micula and Micula for solving KdV-Burger equation. Ersoy and Dag 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. for numerical solution of one dimensional KdV equation. Lakestani presented a numerical scheme based on finite difference method and B-spline functions for solving third order non-linear KdV equation. Dong 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 and new Cubic B-Spline (CBS) approximations 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

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

(3)

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

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

(4)

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

$$b_i(x) = \frac{1}{6h^3}(x-x_i) + 3h(x-x_i)^2 - 3(x-x_i)^3, \quad x \in [x_i, x_{i+1}]$$

(5)

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

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

(6)

where, $c_p(t)$’s are, time dependent real constants, yet to be calculated. For simplicity, we express the CBS approximations $U(x)$, $U'(x)$, $U''(x)$ and $U'''(x)$ 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

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

(7)

$$m_i = \sum_{p=i+1}^{i+1} c_pB_p(x) = \frac{1}{2h}(-c_{i+1} + c_{i-1})$$

(8)

Moreover, for second and third order derivatives, we shall use the following new CBS approximations.
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 \( \theta \)-weighted scheme, the problem is discretized in time direction as

\[
\frac{u^{i+1} - u^i}{\Delta t} + \theta \left[ \alpha (uu_x)^{i+1} + \beta uu_{xx}^{i+1} + \gamma uu_{xxx}^{i+1} \right] + (1 - \theta) \left[ \alpha uu_x^i + \beta uu_{xx}^i + \gamma uu_{xxx}^i \right] = 0.
\]

(11)

where, \( \Delta t \) is the step size in time direction, \( 0 \leq \theta \leq 1 \) and \( u^{i+1} \) is used to denote \( u(x_i, t_{i+1} + \Delta t) \). The non-linear term \( (uu_x)^{i+1} \) is linearized as \(25,30\)

\[
(uu_x)^{i+1} = uu_x^i + uu_{xx}^i - uu_x^i.
\]

(12)

Substituting (12) into (11), we get

\[
\frac{u^{i+1} - u^i}{\Delta t} + \theta \left[ \alpha (uu_x^i + uu_{xx}^i - uu_x^i) + \beta uu_{xx}^{i+1} + \gamma uu_{xxx}^{i+1} \right] + (1 - \theta) \left[ \alpha uu_x^i + \beta uu_{xx}^i + \gamma uu_{xxx}^i \right] = 0.
\]

(13)

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

\[
\left[ \frac{2}{\Delta t} + \alpha uu_x^i \right] u^{i+1} + \alpha uu_x^i + \beta uu_{xx}^{i+1} + \gamma uu_{xxx}^{i+1} = \left[ \frac{2}{\Delta t} - \beta uu_x^i - \gamma uu_{xxx}^i \right].
\]

(14)

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

\[
w^i U^{i+1} + y^i M^{i+1} + \beta M^{i+1} + \gamma T^{i+1} = z_i,
\]

(15)

where, \( w^i = \frac{2}{\Delta t} + \alpha uu_x^i \), \( y^i = \alpha U^i \), and \( z_i = \frac{2}{\Delta t} U^i - \beta M^i - \gamma T^i \).

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

\[
w^i \left( c_{i-1} + 4c_i + c_{i+1} \right) + \frac{c_{i+1}}{2\Delta t} - c_{i-1} = c_{i+1} - c_{i-1}
\]

(16)

\[
+ \frac{\beta}{12\Delta t} \left( 14c_{i-1} + 33c_{i+1} + 28c_{i+2} - 14c_{i+1} \right) + \frac{c_{i+1}}{2\Delta t} - c_{i-1} = c_{i+1} - c_{i-1}
\]

(17)

\[
+ \frac{\gamma}{36\Delta t} \left( -12c_{i-1} + 24c_{i+1} - 35c_{i+2} + 238c_{i+3} - 78c_{i+4} + 9c_{i+5} + c_{i+6} \right) = z_{i+1}.
\]

(18)

\[
w^i \left( c_{i-1} + 4c_i + c_{i+1} \right) + \frac{c_{i+1}}{2\Delta t} - c_{i-1} = c_{i+1} - c_{i-1}
\]

(19)

\[
+ \frac{\beta}{12\Delta t} \left( c_{i-1} + 8c_i - 18c_{i+1} + 8c_{i+2} + c_{i+3} \right) + \frac{c_{i+1}}{2\Delta t} - c_{i-1} = c_{i+1} - c_{i-1}
\]

(20)

\[
+ \frac{\gamma}{144\Delta t} \left( -12c_{i-1} + 213c_{i+1} + 378c_{i+2} + 55c_{i+3} - 450c_{i+4} + 225c_{i+5} - 4c_{i+6} - 3c_{i+7} \right) = z_{i+1}.
\]

(21)

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

\[
U_0^{i+1} = \phi_1 \left( t_{j+1} \right)
\]

(22)

\[
U_n^{i+1} = \phi_2 \left( t_{j+1} \right)
\]

(23)

\[
m_n^{i+1} = \phi_3 \left( t_{j+1} \right)
\]

(24)

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

\[
AC^{i+1} = B,
\]

(25)

where \( A \) denotes the coefficient matrix of order \( n + 3 \), \( B \) is column matrix of order \( n + 3 \) and \( C^{i+1} = \left[ c_{i-1} \ c_i \ c_{i+1} \ \cdots \ c_{n+1} \right]^T \) 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)

\[ m_0^0 = g'(x_0), \]  
\[ U_i^0 = g(x_i), \quad i = 0(1)n, \]  
\[ m_n^0 = g'(x_n). \]  

Using (7)–(8), we get

\[ -c_{i-1}^0 + c_{i+1}^0 = 2h g'(x_i), \]  
\[ c_{i-1}^0 + 4c_i^0 + c_{i+1}^0 = 6g(x_i), \quad i = 0(1)n, \]  
\[ -c_n^0 + c_{n+1}^0 = 2h g'(x_n). \]  

The above system can be expressed in matrix form as

\[ AC^0 = B. \]  

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 \( L_\infty \), \( L_2 \) and Root Mean Square (RMS), which are calculated as

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

The exact solution is

\[ u(x,t) = \frac{\lambda}{2} \text{sech}^2 \left( \frac{\sqrt{\lambda}}{2} (x - \mu) \right). \]

The error norms \( L_\infty \), \( L_2 \) 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 \( 0 \leq x \leq 40 \), \( 0 \leq t \leq 5 \), \( \lambda = 0.5 \), \( \mu = 7 \)

<table>
<thead>
<tr>
<th>( t )</th>
<th>MQ(^0) ( \Delta t = 0.001 )</th>
<th>IMQ(^0) ( \Delta t = 0.001 )</th>
<th>MQQI(^11) ( \Delta t = 0.001 )</th>
<th>Proposed method ( \Delta t = 0.01 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 1.79 \times 10^{-5} )</td>
<td>( 6.96 \times 10^{-5} )</td>
<td>( 1.53 \times 10^{-3} )</td>
<td>( 8.63 \times 10^{-6} )</td>
</tr>
<tr>
<td>2</td>
<td>( 3.01 \times 10^{-5} )</td>
<td>( 1.96 \times 10^{-4} )</td>
<td>( 2.87 \times 10^{-3} )</td>
<td>( 1.11 \times 10^{-5} )</td>
</tr>
<tr>
<td>3</td>
<td>( 3.98 \times 10^{-5} )</td>
<td>( 3.83 \times 10^{-3} )</td>
<td>( 4.14 \times 10^{-3} )</td>
<td>( 1.26 \times 10^{-5} )</td>
</tr>
<tr>
<td>4</td>
<td>( 4.78 \times 10^{-5} )</td>
<td>( 5.91 \times 10^{-3} )</td>
<td>( 5.39 \times 10^{-3} )</td>
<td>( 1.36 \times 10^{-5} )</td>
</tr>
<tr>
<td>5</td>
<td>( 5.46 \times 10^{-5} )</td>
<td>( 8.37 \times 10^{-3} )</td>
<td>( 6.81 \times 10^{-3} )</td>
<td>( 1.45 \times 10^{-5} )</td>
</tr>
</tbody>
</table>
Table 2. Error norms for Example 1, when $0 \leq x \leq 40$, $0 \leq t \leq 5$, $\lambda = 0.5$, $\mu = 7$

<table>
<thead>
<tr>
<th></th>
<th>$t$</th>
<th>$L_\infty$</th>
<th>$L_2$</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Proposed method MQRBI $^8$</td>
<td>$1.67 \times 10^{-5}$</td>
<td>$6.00 \times 10^{-4}$</td>
<td>$4.23 \times 10^{-5}$</td>
</tr>
<tr>
<td>2</td>
<td>Proposed method IMQQI $^{12}$</td>
<td>$1.11 \times 10^{-5}$</td>
<td>$9.22 \times 10^{-4}$</td>
<td>$6.51 \times 10^{-5}$</td>
</tr>
<tr>
<td>3</td>
<td>Proposed method IMQQI $^{12}$</td>
<td>$1.26 \times 10^{-5}$</td>
<td>$1.13 \times 10^{-4}$</td>
<td>$8.00 \times 10^{-5}$</td>
</tr>
<tr>
<td>4</td>
<td>Proposed method IMQQI $^{12}$</td>
<td>$1.36 \times 10^{-5}$</td>
<td>$1.29 \times 10^{-3}$</td>
<td>$9.12 \times 10^{-5}$</td>
</tr>
<tr>
<td>5</td>
<td>Proposed method IMQQI $^{12}$</td>
<td>$1.45 \times 10^{-5}$</td>
<td>$1.42 \times 10^{-3}$</td>
<td>$1.00 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>$t$</th>
<th>$\Delta t$</th>
<th>$L_\infty$</th>
<th>$L_2$</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Proposed method MQRBI $^8$</td>
<td>0.01</td>
<td>$2.00 \times 10^{-7}$</td>
<td>$2.14 \times 10^{-5}$</td>
<td>$1.35 \times 10^{-6}$</td>
</tr>
<tr>
<td>2</td>
<td>Proposed method MQRBI $^8$</td>
<td>0.001</td>
<td>$4.43 \times 10^{-7}$</td>
<td>$3.50 \times 10^{-5}$</td>
<td>$2.21 \times 10^{-6}$</td>
</tr>
<tr>
<td>3</td>
<td>Proposed method MQRBI $^8$</td>
<td>0.01</td>
<td>$5.84 \times 10^{-7}$</td>
<td>$4.10 \times 10^{-5}$</td>
<td>$2.59 \times 10^{-6}$</td>
</tr>
<tr>
<td>4</td>
<td>Proposed method MQRBI $^8$</td>
<td>0.001</td>
<td>$6.84 \times 10^{-7}$</td>
<td>$4.28 \times 10^{-5}$</td>
<td>$2.70 \times 10^{-6}$</td>
</tr>
<tr>
<td>5</td>
<td>Proposed method MQRBI $^8$</td>
<td>0.01</td>
<td>$7.87 \times 10^{-7}$</td>
<td>$3.71 \times 10^{-6}$</td>
<td>$2.61 \times 10^{-7}$</td>
</tr>
</tbody>
</table>
New Cubic B-spline Approximations for Solving Non-linear Third-order Korteweg-de Vries Equation

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

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

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

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

Example 2:
Consider the following KdV equation\(^{23}\)

\[
\begin{align*}
    u_t + 6uu_x + u_{xxx} &= 0, \quad x \in [a, b], \ t \in [0, T], \\
    u(x, 0) &= 2 \operatorname{sech}^2 (x + 4).
\end{align*}
\]

The exact solution is \( u(x; t) = 2 \operatorname{sech}^2 (x - 4t + 4) \). The computational error norms \( L_\infty, L_2 \) and RMS are listed in Table 4 when \( n = 200 \) and \( \Delta t = 0.01 \). 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 \( \Delta t = 0.01 \).

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

<table>
<thead>
<tr>
<th>( t )</th>
<th>( L_\infty )</th>
<th>( L_2 )</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>( 3.39 \times 10^{-5} )</td>
<td>( 1.82 \times 10^{-4} )</td>
<td>( 1.29 \times 10^{-5} )</td>
</tr>
<tr>
<td>0.4</td>
<td>( 3.49 \times 10^{-5} )</td>
<td>( 2.40 \times 10^{-4} )</td>
<td>( 1.69 \times 10^{-5} )</td>
</tr>
<tr>
<td>0.6</td>
<td>( 5.12 \times 10^{-5} )</td>
<td>( 3.41 \times 10^{-4} )</td>
<td>( 2.40 \times 10^{-5} )</td>
</tr>
<tr>
<td>0.8</td>
<td>( 8.21 \times 10^{-5} )</td>
<td>( 4.76 \times 10^{-4} )</td>
<td>( 3.36 \times 10^{-5} )</td>
</tr>
<tr>
<td>1.0</td>
<td>( 8.09 \times 10^{-5} )</td>
<td>( 4.30 \times 10^{-4} )</td>
<td>( 3.03 \times 10^{-5} )</td>
</tr>
</tbody>
</table>
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 non-linear 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, MQ, IMQ, MQQI and IMQQI 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


