International Journal of Advanced and Applied Sciences, 4(5) 2017, Pages: 1-11

Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Numerical solution of generalized Burger's-Huxley equation using local radial basis functions

Mutia Bukhari [1,*], Muhammad Arshad [1], Safeera Batool [2], Syed Muhammad Saqlain [3]

[1]Department of Mathematics, Islamia College University, Peshawar, Pakistan
[2]Department of Basic Sciences, University of Engineering & Technology, Taxila, Pakistan
[3]Department of Computer Science & Software Engineering, International Islamic University, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 25 August 2016
Received in revised form
31 March 2017
Accepted 31 March 2017

Keywords:
Burger's-Huxley equation
Radial basis function
Differential collocation

ABSTRACT

Partial differential equations are well known with their use in different areas of applied mathematics and engineering. Numerical solutions to these equations are widely accepted but the development of robust, simple and efficient numerical scheme for the solution of partial differential equations is a challenging research issue. In this Paper a local radial basis function based differential quadrature collocation method is presented. It finds the numerical solution of Generalized Burger's Huxley equation and its various special cases such as Burger's Huxley equation, Generalized Burger's equation and Huxley equation. The proposed method is implemented through different examples using Gaussian radial basis functions (RBF) with central three point scheme and central five point scheme in the local support region. Accuracy of the method is investigated by computing L_2 and L_∞error norms and absolute errors. The results of the proposed technique in terms of error norms comparative to the existing techniques of adomian decomposition and global mesh free method are presented as well.

1. Introduction

Partial differential equations (PDEs) (linear and nonlinear) have lots of applications in the area of elasticity, fluid dynamics, electrostatics, electrodynamics, propagation of heat and sound, mechanics, computational physics and applied mathematics. These provide the source of mathematical models for chemical, physical and biological phenomena and are also used in the areas of economics, financial forecasting, image processing etc. As PDEs are used in almost all fields of engineering and science, so it is essential to find their accurate solutions. Thus utilization of numerical techniques is an important source to solve them. But it is a big issue to develop accurate numerical approximation techniques and thus development of robust as well as simple and efficient numerical schemes is a key interest in the research field.

A lot of different techniques are available for the numerical solutions of PDEs including finite difference method (FDM) (Özisik, 1994), boundary element method (Šarler and Kuhn, 1998a; 1998b), finite element method (FEM) (Zienkiewicz and Taylor, 2000) and finite volume method (FVM) (Hong, 2004) etc. FDM can solve strong type PDEs and its source is polynomial approximation or Taylor series expansion. FV method employs the physical conservation laws on the class of discretization schemes called the method of weighted residuals. Mesh generation is the first step in FE method. In this method mesh is constructed containing triangles or other polygons, called "elements". Mesh nodes, edges, and local information related to every element are stored in different arrays. The accuracy of the solution increases with more elements and number of elements increases by refined meshing. Several mesh less techniques have been introduced up to now including the diffuse element method (Nayroles et al., 1992) reproducing kernel particle method (Liu et al., 1995), smoothed particle hydrodynamics method (Lucy, 1977), the partition of unity method (Babuska and Melenk, 1997), the element-free Galerkin method (Belytschko et al., 1994), the hp-clouds method (Duarte and Oden, 1995), the finite point method (Onate et al., 1996), the method of finite spheres (De and Bathe, 2000) and radial basis functions method (Rippa, 1984) etc.

* Corresponding Author.
Email Address: mutiabukhari11@gmail.com (M. Bukhari)
https://doi.org/10.21833/ijaas.2017.05.001

In the past few decades, a new method for numerical solution of PDEs has got importance in the world of science which is known as the method of radial basis functions. Originally RBFs were introduced for multivariate data and function interpolation, mainly for higher dimension problems. As RBFs are truly mesh free in nature these are used in almost every field of science successfully including geology, mathematics, geophysics, geodesy, engineering, mapping, meteorology, photogrammetry, spacecraft designing, air pollution modeling, medical imaging and many more other fields.

LRBF was first initiated by Liu and Gu (2001). To investigate free vibration of 2D solids Liu and Gu (2001) used local radial point interpolation method (LRPIM). Wu and Liu (2003) employed LRPIM to the incompressible flow simulation. Further Shu et al. (2003) recommended the local RBF-based differential quadrature (LRBFDQ) collocation method for the solution of two-dimensional incompressible Navier–Stokes equations.

The DQ method is a numerical discretization technique to approximate derivatives. In early 1970's, Bellman and Casti (1971) introduced this technique, started from the concept of conventional integral quadrature. Early schemes of DQ in engineering utilized Bellman's approaches for determining the weighting coefficients. But there were some drawbacks using Bellman's approaches. To overcome these drawbacks some efforts have been done. In this regard a great work is done by Shu and Richard (1992). Further Shu and Chew (1997), Shu (2000), and Shu and Xue (1997) proposed easy algebraic formulations to determine weighting coefficients of first and second order derivatives when the function is approximated by a Fourier series expansion.

It is noted that all the work done in the past regarding the use of RBFs to solve PDEs numerically relays on function approximation rather than derivative approximation. This process is extremely complex, particularly dealing nonlinear problems. To overcome this problem, Wu and Shu (2002) developed RBF-based differential quadrature (RBF-DQ) method.

Whilst in the global RBF-DQ collocation method the function is approximated by using all the collocating points in the computational domain and it produces ill-conditioned matrix by the use of large number of nodes. Hence to overcome this drawback local RBF-DQ collocation method (Shu et al., 2003) is used. It can be employed to any complex problem.

The Generalized Burger's-Huxley equation was studied first by Satsuma et al. (1987) in 1987 and is given by (Eq. 1)

$$u_t + \alpha u^{\delta} u_x + u_{xx} = \beta u(1 - u^{\delta})(u^{\delta} - \gamma),$$
$$x \in \Omega = [a, b], t \geq 0 \qquad (1)$$

with initial condition (Eq. 2)

$$u^0(x) = u(x,0) = [\tfrac{\gamma}{2} + \tfrac{\gamma}{2}\tanh(\omega_1 x)]^{\frac{1}{\delta}} \qquad (2)$$

and boundary conditions (Eq. 3)

$$u\,(x,t) = [\tfrac{\gamma}{2} + \tfrac{\gamma}{2}\tanh(\omega_1(x - \omega_2 t)]^{\frac{1}{\delta}},$$
$$x \in \partial\Omega = \{a, b\}, t > 0 \qquad (3)$$

the exact solution of Eq. 1 is given by (Eq. 4)

$$u\,(x,t) = [\tfrac{\gamma}{2} + \tfrac{\gamma}{2}\tanh(\omega_1(x - \omega_2 t)]^{\frac{1}{\delta}},$$
$$x \in \Omega = [a, b], t \geq 0 \qquad (4)$$

where

$$\omega_1 = \frac{-\alpha\delta + \delta\sqrt{\alpha^2 + 4\beta(1+\delta)}}{4(1+\delta)}\gamma,$$

and

$$\omega_2 = \frac{\alpha\gamma}{1+\delta} - \frac{(1+\delta-\gamma)(-\alpha+\sqrt{\alpha^2+4\beta(1+\delta)})}{2(1+\delta)}$$

Here α, β, δ and γ are parameters such that $\alpha, \beta, \delta \geq 0$ and $\gamma \in (0,1)$ on exact solution the function of parameters was studied by Efimova and Kudryashov (2004). Eq. 1 takes the form of Huxley equation when $\alpha = 0$ and $\delta = 1$, which is used in nerve pulse propagation during nerve fibers and wall motion in liquid crystals. For $\beta = 0$, Eq. 1 becomes Burger's equation. The Burger's equation has an important role in shock wave formulation, sound waves in viscous medium, boundary layer properties and traffic flow. Hon and Mao (1998) have given its detail study (Hon and Mao, 1998).

Several techniques have been introduced in the literature for the numerical solution of the Generalized Burger's–Huxley equation. These include optimal homotopy asymptotic method by Ali et al. (2012), Adomian decomposition technique by Ismail et al. (2004), Haar wavelet method by Celik (2012), computational meshless method by Khattak (2009). Many other techniques can be found in references (Mittal and Tripathi, 2015; Jiwari el al., 2013; El-Kady et al., 2013; Macías-Díaz and Szafrańska, 2014; Ervin et al., 2015).

The global radial basis function collocation method approximates the space derivative which requires reasonably large set of collocation points in the neighborhood of any collocation node resulting in need of high computational cost. In this work the local radial basis function collocation method is that it approximates the space derivatives via RBF interpolation using a small set of collocation points in the neighborhood of any collocation node and therefore it saves much computational work than the globally supported RBF collocation method.

2. Proposed technique

In this section we will construct the LRBFDQ method for the generalized Burger's-Huxley equation. Moreover some examples regarding the numerical results of the generalized Burger's-Huxley equation and its different cases like Burger's-Huxley equation, Burger's equation, Huxley equation etc. will also be given. Also we will test the proposed (LRBFDQ) method by calculating error norms and comparing results with some other existing methods like ADM (Ismail et al., 2004) and global mesh free method (Khattak, 2009).

2.1. Construction of local RBF-based differential quadrature method

We consider the generalized Burger's-Huxley Eqs. 1-3. We choose N distinct nodes$x_1, x_2, \ldots, x_N$. In local RBF-based differential quadrature interpolation, the derivative of $u(x)$ is approximated by the differential quadrature method. i.e., the derivative of $u(x)$ at the center x_i is approximated by the linear weighted sum of all the functional values in the supporting region of

$$x_{i,}\, x_i, \{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in_i}\}$$

contained in (Eq. 5)

$$\{x_1, x_1, \ldots, x_1\}, n_i \ll N,$$
$$u^{(m)}(x_i) \approx \sum_{j=1}^{n_i} \lambda_j^{(m)} u(x_{i_j}), \quad \mathrm{i} = 1,2, \ldots, \mathrm{N} \tag{5}$$

The weighting coefficients $\lambda_j^{(m)}$ are computed by RBF $\phi(\|x - x_k\|)$ approximation of the function $u(x_{i_j})$ into Eq. 4 as under (Eq. 6):

$$\phi^{(m)}(\|x_i - x_k\|) = \sum_{j=1}^{n_i} \lambda_{i_j}^{(m)} \phi\left(\left\|x_{i_j} - x_k\right\|\right), \quad k = i_1, i_2, \ldots, i_n \tag{6}$$

Eq. 5 can be written in matrix from as (Eq. 7):

$$\begin{bmatrix} \phi_{i_1}^{(m)}(x_i) \\ \phi_{i_2}^{(m)}(x_i) \\ \vdots \\ \phi_{i_{n_i}}^{(m)}(x_i) \end{bmatrix} = \begin{bmatrix} \phi_{i_1}(x_{i_1}) & \phi_{i_2}(x_{i_1}) & \cdots & \phi_{i_{n_i}}(x_{i_1}) \\ \phi_{i_2}(x_{i_2}) & \phi_{i_2}(x_{i_2}) & & \phi_{i_{n_i}}(x_{i_2}) \\ \vdots & & \ddots & \vdots \\ \phi_{i_1}(x_{i_{n_i}}) & \phi_{i_2}(x_{i_{n_i}}) & \cdots & \phi_{i_{n_i}}(x_{i_{n_i}}) \end{bmatrix} \begin{bmatrix} \lambda_{i_1}^{(m)} \\ \lambda_{i_2}^{(m)} \\ \vdots \\ \lambda_{i_{n_i}}^{(m)} \end{bmatrix} \tag{7}$$

where, $\phi_k(x_j) = \phi(\|x_j - x_k\|)$, therefore

$$\phi_{n_i}^{(m)} = A_{n_i} \lambda_{n_i}^{(m)} \tag{8}$$

where,

$$\phi_{n_i}^{(m)} = [\phi_{i_1}^{(m)}(x_i), \phi_{i_2}^{(m)}(x_i), \ldots, \phi_{i_{n_i}}^{(m)}(x_i)]^T,$$

$$A_{n_i} = \begin{bmatrix} \phi_{i_1}(x_{i_1}) & \phi_{i_2}(x_{i_1}) & \cdots & \phi_{i_{n_i}}(x_{i_1}) \\ \phi_{i_2}(x_{i_2}) & \phi_{i_2}(x_{i_2}) & & \phi_{i_{n_i}}(x_{i_2}) \\ \vdots & & \ddots & \vdots \\ \phi_{i_1}(x_{i_{n_i}}) & \phi_{i_2}(x_{i_{n_i}}) & \cdots & \phi_{i_{n_i}}(x_{i_{n_i}}) \end{bmatrix}$$

and $\lambda_{n_i}^{(m)} = [\lambda_{i_1}^{(m)}, \lambda_{i_2}^{(m)}, \ldots, \lambda_{i_{n_i}}^{(m)}]^T$

thus the corresponding coefficients can be obtained as:

$$\lambda_{n_i}^{(m)} = A_{n_i}{}^{-1} \phi_{n_i}^{(m)} \tag{9}$$

by substituting Eq. 8 into Eq. 5, we have

$$u^{(m)}(x_i) = (\lambda_{n_i}^{(m)})^T \overline{u_{n_i}} \tag{10}$$

Where, $\overline{u_{n_i}} = [u(x_{i_1}), u(x_{i_2}), \ldots, u(x_{i_{ni}})]^T.z$

now applying LRBFDQ method to Eqs. 1-3 we get,

$$\frac{du_i}{dt} + \alpha u_i^\delta (\lambda_{\boldsymbol{n_i}}^{(\mathbf{1})})^T \overline{u_{n_i}} - \left(\lambda_{\boldsymbol{n_i}}^{(\mathbf{2})}\right)^T \overline{u_{n_i}} = \beta u_i (1 - u_i^\delta)(u_i^\delta - \gamma), i = 2,3, \ldots, N-1 \tag{11}$$
$$u(a,t) = f_1(t), u(b,t) = f_2(t) \tag{12}$$

Eqs. 10-11 can be expressed in matrix form as:

$$\frac{dU}{dt} = -\alpha U^\delta * (\lambda^{(1)} U) + (\lambda^{(2)} U) + \beta((1+\gamma) U^{1+\delta} - \gamma U - U^{2\delta+1}) \tag{13}$$

The symbol * shows component by component multiplication of two vectors and

$U = [u_1, u_2, \ldots, u_N]^T$, $U^\delta = [u_1^\delta, u_2^\delta, \ldots, u_{N-1}^\delta, u_N^\delta]^T$, $\lambda^{(1)} = [\lambda_k^{(1)}]_{NXN}$, $\lambda^{(2)} = [\lambda_k^{(2)}]_{NXN}$, $k = i_1, i_2, \ldots, i_n$, $i = 1,2,3, \ldots, N-1$

Eq. 12 can be written as:

$$\frac{dU}{dt} = M(U) \tag{14}$$

where,

$$M(U) = -\alpha U^\delta * (\lambda^{(1)} U) + (\lambda^{(2)} U) + \beta((1+\gamma) U^{1+\delta} - \gamma U - U^{2\delta+1})$$

the related initial condition is

$$U^0(x) = [u^0(x_1), u^0(x_2), \ldots, u^0(x_N)]^T \tag{15}$$

From the boundary conditions Eq. 3, we have

$$u(a,t) = f_1(t), u(b,t) = f_2(t) \tag{16}$$

to solve Eqs. 14-16 we use the following Classical fourth order Runge–Kutta scheme (RK4):

$$K_1 = M(U^n),\ K_2 = M\left(U^n + \tfrac{dt}{2} K_1\right),$$
$$K_3 = M\left(U^n + \tfrac{dt}{2} K_2\right),$$
$$K_4 = M(U^n + dt K_3),$$
$$U^{n+1} = U^n + \frac{\delta t(K_1 + 2K_2 + 2K_3 + K4)}{6}$$

In the next section we present numerical results obtained by the proposed method given in Eq. 14-16.

2.2. Numerical tests and discussion

In this section we provide numerical results through various examples of the generalized Burger's-Huxley equation to validate the method using Eqs. 14-16. Accuracy of the method is examined via absolute error (AE), L_2 and error norms defined as follows:

$$AE = |(u_{exact})_j - (u_{app})_j| ,$$
$$L_2 = \sqrt{h \sum_{j=1}^{N} |(u_{exact})_j - (u_{app})_j|^2} ,$$
$$L_\infty = \max_j |(u_{exact})_j - (u_{app})_j|$$

where, u_{exact} and u_{app} represent exact and approximate solutions respectively.

For computation purpose Gaussian RBF is chosen in LRBFDQ method using three and five point central scheme for the local support domain. Also the number of nodes selected in the local support region are ni=3 and ni=5. All the computations are performed with nodal distance h= 0.1. For the comparison purpose with ADM (Ismail et al., 2004) and GRBF (Khattak, 2009), spatial domain Ω=[0,1] will be used.

Example 1: In generalized Burger's-Huxely Eq. 1-4, consider parameter values as β=0 and δ=1, it becomes Burger's equation. Take α=1, Eq. 1 reduces to the following form:

$$u_t + uu_x - u_{xx} = 0, \quad x \in \Omega = [a, b], t \geq 0$$

with initial condition

$$u(x, 0) = \frac{\gamma}{2} + \frac{\gamma}{2} tanh(\omega_1 x), \qquad x \in [0,1]$$

and boundary conditions

$$u(0, t) = \frac{\gamma}{2} + \frac{\gamma}{2} tanh(-\omega_1 \omega_2 t)$$
$$u(1, t) = \frac{\gamma}{2} + \frac{\gamma}{2} tanh(\omega_1 (1 - \omega_2 t)), \qquad t \geq 0$$

Computations are performed by applying LRBFDQ and using shape parameter c=1.041×10-15 for n_i=5 and c=1.3881×10-15 for n_i=3 and γ=0.001 with time step dt=0.001 and h=0.1 for interval [0, 1]. The error norms of LRBFDQ method using n_i=5 and n_i=3 are reported in Table 1. Moreover for comparison purpose the absolute errors of LRBFDQ method, ADM (Ismail et al., 2004) and GRBF (Khattak, 2009) at specific nodes are also mentioned in Table 2 up to time t=50.

Example 2: Now consider Burger's equation with δ=2, 3. All the computations are performed by applying LRBFDQ with α=1 and using values c=1.041×10-15 for n_i=5 and c=1.388×10-15 for n_i=3 up to time t=5.

The error norms of LRBFDQ method are reported in Table 3 for δ=2 using time step dt=0.01. For δ=3 the error norms are reported in Table 5 and Table 7 with time steps dt=0.0001 and dt=0.01 respectively. Moreover for comparison purpose the absolute errors of LRBFDQ method, ADM (Ismail et al., 2004) and GRBF (Khattak, 2009) are also mentioned in Table 4 for δ=2 with dt=0.01 and in Table 6 and Table 8 for δ=3 with dt=0.0001, 0.01 respectively.

Example 3: Consider the value of parameters as α=0.001, β=0.001, δ=1 and γ=0.001 in the generalized Burger's-Huxely Eqs. 1-4 with time step dt=0.001.

Computations are performed by applying LRBFDQ by using shape parameter c=1.317×10-15 for n_i=5 and c=1.388×10-15 for n_i=3. The error norms of LRBFDQ method are reported in Table 9. Furthermore the absolute errors of LRBFDQ method at specific nodes are also mentioned in Table 10 up to time t=50.

Table 1: Error norms for α=1, β=0, δ=1, dt=0.001, γ=0.001

t	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.5	2.6053×10^{-7}	1.9542×10^{-7}	1.648×10^{-7}	1.5131×10^{-7}
1	5.211×10^{-7}	3.9099×10^{-7}	2.3296×10^{-7}	3.0263×10^{-7}
2	1.0429×10^{-6}	7.826×10^{-7}	4.6592×10^{-7}	6.0525×10^{-7}
5	2.611×10^{-6}	1.9611×10^{-6}	1.648×10^{-6}	1.5131×10^{-6}
50	2.6728×10^{-5}	2.0324×10^{-5}	1.656×10^{-5}	1.5147×10^{-5}

Table 2: Numerical and exact values and absolute errors for α=1, β=0, δ=1, dt=0.001, γ=0.001.

t	x	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)	GRBF (Khattak, 2009)	ADM (Ismail et al., 2004)
	0.1	0.000500260	0.000500195	0.000500000	2.6×10^{-7}	1.95×10^{-7}	1.0×10^{-6}	6.34×10^{-8}
0.5	0.5	0.000500000	0.000500000	0.000500000	0.0	6.11×10^{-11}	5.0×10^{-6}	5.66×10^{-8}
	0.9	0.000499740	0.000499805	0.000500000	2.6×10^{-7}	1.95×10^{-7}	3.0×10^{-6}	4.13×10^{-8}
	0.1	0.000500521	0.000500391	0.000500000	5.21×10^{-7}	3.9×10^{-7}	1.0×10^{-6}	2.0×10^{-6}
1	0.5	0.000500000	0.000500000	0.000500000	0.0	2.44×10^{-10}	3.0×10^{-6}	1.84×10^{-6}
	0.9	0.000499479	0.000499610	0.000500000	5.21×10^{-7}	3.9×10^{-7}	3.0×10^{-6}	1.37×10^{-6}
	0.1	0.000501043	0.000500783	0.000500000	1.0×10^{-6}	7.8×10^{-7}	1.0×10^{-6}	6.4×10^{-5}
2	0.5	0.000500000	0.000499999	0.000500000	0.0	9.77×10^{-10}	3.0×10^{-6}	6.1×10^{-5}
	0.9	0.000498959	0.000499220	0.000500000	1.0×10^{-6}	7.8×10^{-7}	3.0×10^{-6}	4.7×10^{-5}
	0.1	0.000502611	0.000501961	0.000500000	2.6×10^{-6}	1.96×10^{-6}	4.0×10^{-6}	
5	0.5	0.000500000	0.000499994	0.000500000	0.0	6.11×10^{-11}	9.0×10^{-6}	
	0.9	0.000497403	0.000498054	0.000500000	2.6×10^{-6}	1.95×10^{-6}	3.0×10^{-6}	
	0.1	0.000526728	0.000520324	0.000500000	2.67×10^{-5}	2.03×10^{-5}	6.0×10^{-6}	
50	0.5	0.000500000	0.000499390	0.000500000	1.12×10^{-14}	6.1×10^{-7}	2.0×10^{-5}	
	0.9	0.000474628	0.000481204	0.000500000	2.5×10^{-5}	1.88×10^{-5}	8.0×10^{-6}	

Table 3: Error norms for α=1, β=0, δ=2, dt=0.01, γ=0.001

t	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.5	1.1654×10^{-5}	8.7412×10^{-6}	5.2102×10^{-6}	6.7674×10^{-6}
1	2.332×10^{-5}	1.7493×10^{-5}	1.0419×10^{-5}	1.3534×10^{-5}
2	4.6689×10^{-5}	3.5026×10^{-5}	2.0837×10^{-5}	2.7068×10^{-5}
5	1.1709×10^{-4}	8.7875×10^{-5}	5.2094×10^{-5}	6.7671×10^{-5}

Table 4: Numerical and exact values and absolute errors for α=1, β=0, δ=2, dt=0.01, γ=0.001T

	x	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)	GRBF (Khattak,2009)
	0.1	0.022372334	0.022369421	0.022360680	1.17×10^{-5}	8.74×10^{-6}	1×10^{-6}
0.5	0.5	0.022360680	0.022360677	0.022360680	0.0	2.74×10^{-9}	2×10^{-6}
	0.9	0.022349038	0.022351949	0.022360680	1.16×10^{-5}	8.73×10^{-6}	0.0
	0.1	0.022384000	0.022378172	0.022360680	2.33×10^{-5}	1.75×10^{-5}	0.0
1	0.5	0.022360680	0.022360669	0.022360680	0.0	1.09×10^{-8}	1×10^{-6}
	0.9	0.022337408	0.022343228	0.022360680	2.33×10^{-5}	1.75×10^{-5}	2×10^{-6}
	0.1	0.022407369	0.022395706	0.022360680	4.67×10^{-5}	3.5×10^{-5}	2×10^{-6}
2	0.5	0.022360680	0.022360636	0.022360680	0.0	4.37×10^{-8}	5×10^{-6}
	0.9	0.022314184	0.022325817	0.022360680	4.65×10^{-5}	3.49×10^{-5}	1×10^{-6}
	0.1	0.022477770	0.022448555	0.022360680	1.17×10^{-4}	8.79×10^{-5}	4×10^{-6}
5	0.5	0.022360680	0.022360407	0.022360680	0.0	2.37×10^{-7}	1.1×10^{-5}
	0.9	0.022244803	0.022273828	0.022360680	1.16×10^{-4}	8.69×10^{-5}	3×10^{-6}

Table 5: Error norms for α=1, β=0, δ=3, dt=0.0001, γ=0.001.

t	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.0005	4.1345×10^{-8}	3.1009×10^{-8}	1.8856×10^{-8}	2.4179×10^{-8}
0.001	8.26×10^{-8}	6.2018×10^{-8}	3.71×10^{-8}	4.8119×10^{-8}

Table 6: Numerical and exact values and absolute errors for α=1, β=0, δ=3, dt=0.0001, γ=0.001

t	X	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)	GRBF (Khattak, 2009)	ADM (Ismail et al., 2004)
	0.1	0.079370094	0.079370084	0.079370053	4.1×10^{-8}	3.10×10^{-8}	6×10^{-6}	4.5×10^{-4}
0.0005	0.5	0.079370053	0.079370053	0.079370053	0.0	1.01×10^{-14}	5×10^{-6}	1.9×10^{-3}
	0.9	0.079370011	0.079370022	0.079370053	4.1×10^{-8}	3.10×10^{-8}	4×10^{-6}	9.2×10^{-3}
	0.1	0.079370135	0.079370115	0.079370053	8.27×10^{-8}	6.2×10^{-8}	1.9×10^{-5}	4.4×10^{-4}
0.001	0.5	0.079370053	0.079370053	0.079370053	0.0	3.95×10^{-14}	1.6×10^{-5}	1.9×10^{-3}
	0.9	0.079369970	0.079369991	0.079370053	8.27×10^{-8}	6.2×10^{-8}	1.5×10^{-5}	9.1×10^{-4}

Table 7: Error norms for α=1, β=0, δ=3, dt=0.01, γ=0.001

T	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.5	4.1377×10^{-5}	3.1033×10^{-5}	1.8494×10^{-5}	2.4021×10^{-5}
1	8.28×10^{-5}	6.211×10^{-5}	3.69×10^{-5}	4.804×10^{-5}
2	1.659×10^{-4}	1.2422×10^{-4}	7.39×10^{-5}	9.6079×10^{-5}
5	4.1671×10^{-4}	3.1253×10^{-5}	1.8492×10^{-4}	2.4021×10^{-5}

Table 8: Numerical and exact values and absolute errors for α=1, β=0, δ=3, dt=0.01, γ=0.001

T	X	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)	GRBF (Khattak, 2009)
	0.1	0.079411430	0.079401086	0.079370053	4.1×10^{-5}	3.1×10^{-5}	3×10^{-6}
0.5	0.5	0.079370053	0.079370043	0.079370053	0.0	9.37×10^{-9}	7×10^{-6}
	0.9	0.079328740	0.079339068	0.079370053	4.1×10^{-5}	3.1×10^{-5}	1×10^{-6}
	0.1	0.079452872	0.079432167	0.079370053	8.28×10^{-5}	6.2×10^{-5}	2×10^{-6}
1	0.5	0.079370053	0.079370014	0.079370053	0.0	3.88×10^{-8}	8×10^{-6}
	0.9	0.079287491	0.079308132	0.079370053	8.26×10^{-5}	6.19×10^{-5}	1×10^{-6}
	0.1	0.079535952	0.079494477	0.079370053	1.66×10^{-4}	1.24×10^{-4}	3×10^{-6}
2	0.5	0.079370053	0.079369897	0.079370053	0.0	1.55×10^{-7}	8×10^{-6}
	0.9	0.079205187	0.079246403	0.079370053	1.65×10^{-4}	1.24×10^{-4}	1×10^{-6}
	0.1	0.079786762	0.079682586	0.079370053	4.17×10^{-4}	3.13×10^{-4}	4×10^{-6}
5	0.5	0.079370053	0.079369083	0.079370053	0.0	9.7×10^{-7}	1.3×10^{-5}
	0.9	0.078959805	0.079062365	0.079370053	4.1×10^{-4}	3.08×10^{-4}	3×10^{-6}

Example 4: In generalized Burger's-Huxely Eq. 1-4, we take the parameters values α=β=δ=1, so the equation reduces to the following form, which is known as Burger's-Huxley equation:

$u_t + uu_x - uu_x - u(1-u)(u-\gamma), \ \ x \in \Omega = [0,1], \mathrm{t} \geq 0$

with initial condition of $u(x,0) = \frac{\gamma}{2} + \frac{\gamma}{2}\tanh(\omega_1 x)$, $x \in [0,1]$, and boundary conditions of $u(0,t) = \frac{\gamma}{2} + \frac{\gamma}{2}\tanh(-\omega_1\omega_2 t)$ and $u(1,t) = \frac{\gamma}{2} + \frac{\gamma}{2}\tanh(\omega_1(1-\omega_2 t))$, $t > 0$

Computations are performed by applying LRBFDQ by using shape parameter c=1.0409×10^{-15} for n_i=5 and c=1.388×10^{-15}for n_i=3 and γ=0.001.

Table 9: Error norms for α=0.001, β=0.001, δ=1, dt=0.001, γ=0.001

t	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.005	5.0757×10^{-12}	4.9425×10^{-12}	2.6305×10^{-12}	3.0367×10^{-12}
0.05	5.0757×10^{-11}	4.9425×10^{-11}	2.6179×10^{-11}	3.0267×10^{-11}
0.01	1.0151×10^{-11}	9.885×10^{-12}	5.2418×10^{-12}	6.0583×10^{-12}
0.1	1.0151×10^{-10}	9.885×10^{-11}	5.2355×10^{-11}	6.0533×10^{-11}
1	1.0151×10^{-9}	9.885×10^{-10}	5.2355×10^{-10}	6.053×10^{-10}
5	5.0757×10^{-9}	4.9425×10^{-9}	2.6177×10^{-9}	3.0266×10^{-9}
50	5.0756×10^{-8}	4.9424×10^{-8}	2.6177×10^{-8}	3.0266×10^{-8}

Table 10: Numerical and exact values and absolute errors α=0.001, β=0.001, δ=1, dt=0.001, γ=0.001

t	X	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)
0.005	0.1	0.000500001	0.000500001	0.000500001	1.33×10^{-13}	2.51×10^{-16}
	0.5	0.000500003	0.000500003	0.000500003	2.47×10^{-12}	2.47×10^{-12}
	0.9	0.000500005	0.000500005	0.000500005	5.08×10^{-12}	4.94×10^{-12}
0.05	0.1	0.000500001	0.000500001	0.000500001	1.33×10^{-12}	2.51×10^{-15}
	0.5	0.000500003	0.000500003	0.000500003	2.47×10^{-11}	2.47×10^{-11}
	0.9	0.000500005	0.000500005	0.000500005	5.08×10^{-11}	4.94×10^{-11}
0.01	0.1	0.000500001	0.000500001	0.000500001	2.67×10^{-13}	5.03×10^{-16}
	0.5	0.000500003	0.000500003	0.000500003	4.94×10^{-12}	4.94×10^{-12}
	0.9	0.000500005	0.000500005	0.000500005	1.02×10^{-11}	9.89×10^{-12}
0.1	0.1	0.000500001	0.000500001	0.000500001	2.67×10^{-12}	5.03×10^{-15}
	0.5	0.000500003	0.000500003	0.000500003	4.94×10^{-11}	4.94×10^{-11}
	0.9	0.000500005	0.000500005	0.000500005	1.02×10^{-10}	9.89×10^{-11}
1	0.1	0.000500001	0.000500001	0.000500001	2.67×10^{-11}	5.04×10^{-14}
	0.5	0.000500003	0.000500003	0.000500003	4.94×10^{-10}	4.94×10^{-10}
	0.9	0.000500004	0.000500004	0.000500005	1.02×10^{-9}	9.89×10^{-10}
5	0.1	0.000500002	0.000500002	0.000500002	1.33×10^{-10}	2.57×10^{-13}
	0.5	0.000500002	0.000500002	0.000500004	2.47×10^{-9}	2.47×10^{-9}
	0.9	0.000500001	0.000500001	0.000500006	5.08×10^{-9}	4.94×10^{-9}
50	0.1	0.000500014	0.000500013	0.000500013	1.33×10^{-9}	3.12×10^{-12}
	0.5	0.000499990	0.000499990	0.000500015	2.47×10^{-8}	2.47×10^{-8}
	0.9	0.000499966	0.000499968	0.000500017	5.08×10^{-8}	4.94×10^{-8}

The error norms of LRBFDQ method are reported in Table 11. Moreover for comparison purpose the absolute errors of LRBFDQ method, ADM (Ismail et al., 2004) and GRBF (Khattak, 2009) at specific nodes are also mentioned in Table 12 up to time t=5.

Example 5: In this example consider the values α=0.001, β=0.001, γ=0.001 in Eqs. 1-4. Computations are performed by applying LRBFDQ.

Table 11: Error norms for α=β= δ=1, dt=0.01, γ=0.001

t	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.05	4.4793×10^{-8}	3.8277×10^{-8}	2.1411×10^{-8}	2.3438×10^{-8}
0.1	8.9584×10^{-8}	7.6551×10^{-8}	4.2586×10^{-8}	4.6737×10^{-8}
1	8.954×10^{-7}	7.6523×10^{-7}	4.2503×10^{-7}	4.6677×10^{-7}
5	4.4671×10^{-6}	3.8198×10^{-6}	2.1238×10^{-6}	2.3307×10^{-6}

Table 12: Numerical and exact values and absolute errors for α=β= δ=1, γ=0.001 dt=0.01

t	x	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)	GRBF (Khattak, 2009)	ADM (Ismail et al., 2004)
0.05	0.1	0.000500026	0.000500020	0.000500019	7.31×10^{-9}	7.97×10^{-10}	1.0×10^{-9}	1.93×10^{-7}
	0.5	0.000500050	0.000500050	0.000500069	1.87×10^{-8}	1.87×10^{-8}	1.0×10^{-9}	1.93×10^{-7}
	0.9	0.000500074	0.000500080	0.000500119	4.48×10^{-8}	3.83×10^{-8}	1.0×10^{-9}	1.93×10^{-7}
0.1	0.1	0.000500040	0.000500027	0.000500025	1.46×10^{-8}	1.59×10^{-9}	1.0×10^{-9}	3.87×10^{-7}
	0.5	0.000500038	0.000500038	0.000500075	3.75×10^{-8}	3.75×10^{-8}	1.0×10^{-9}	3.87×10^{-7}
	0.9	0.000500035	0.000500048	0.000500125	8.96×10^{-8}	7.66×10^{-8}	1.0×10^{-9}	3.87×10^{-7}
1	0.1	0.000500284	0.000500153	0.000500137	1.46×10^{-7}	1.60×10^{-8}	0.0×10^{-9}	3.88×10^{-6}
	0.5	0.000499813	0.000499812	0.000500187	3.75×10^{-7}	3.75×10^{-7}	0.0×10^{-9}	3.88×10^{-6}
	0.9	0.000499342	0.000499472	0.000500237	8.95×10^{-7}	7.65×10^{-7}	0.0×10^{-9}	3.88×10^{-6}
5	0.1	0.000501369	0.000500719	0.00+0500637	7.32×10^{-7}	8.24×10^{-8}	1.0×10^{-9}	
	0.5	0.000498813	0.000498807	0.000500687	1.87×10^{-6}	1.88×10^{-6}	1.0×10^{-9}	-
	0.9	0.000496270	0.000496922	0.000500737	4.47×10^{-6}	3.82×10^{-6}	1.0×10^{-9}	-

The error norms of LRBFDQ method are reported in Table 13 and Table 15 for δ=2, 3 with dt=0.0001 and dt=0.001 respectively. Furthermore the absolute errors of LRBFDQ method are also listed in Table 14 and Table 16 for δ=2 with dt=0.0001 and δ=3 with dt=0.001 respectively up to time t=50. The value of c used for all these Tables are c= 1.388×10^{-15} for n_i=3 and c=1.32×10^{-15} for n_i=5.

Example 6: In this example take the values α=1, β=1, γ=0.001 in Eqs. 1-4. Computations are performed by applying LRBFDQ.

Table 13: Error norms for α=0.001, β=0.001, δ=2, dt=0.0001, γ=0.001

t	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.0005	2.2723×10^{-11}	2.2152×10^{-11}	1.1784×10^{-11}	1.361×10^{-11}
0.001	4.5446×10^{-11}	4.4301×10^{-11}	2.3482×10^{-11}	2.7153×10^{-11}
0.5	2.2723×10^{-8}	2.2152×10^{-8}	1.1727×10^{-8}	1.3565×10^{-8}
1	4.5446×10^{-8}	4.4304×10^{-8}	2.3453×10^{-8}	2.713×10^{-8}
5	2.2723×10^{-7}	2.2152×10^{-7}	1.1727×10^{-7}	1.3536×10^{-7}

Table 14: Numerical and exact values and absolute errors for α=0.001, β=0.001, δ=2, dt=0.0001, γ=0.001

t	x	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)
0.0005	0.1	0.022360700	0.022360700	0.022360700	5.7×10^{-13}	2.9×10^{-15}
	0.5	0.022360781	0.022360781	0.022360781	1.1×10^{-11}	1.1×10^{-11}
	0.9	0.022360862	0.022360862	0.022360862	2.3×10^{-11}	2.2×10^{-11}
0.001	0.1	0.022360700	0.022360700	0.022360700	1.1×10^{-12}	5.8×10^{-15}
	0.5	0.022360781	0.022360781	0.022360781	2.2×10^{-11}	2.2×10^{-11}
	0.9	0.022360862	0.022360862	0.022360862	4.5×10^{-11}	4.43×10^{-11}
0.5	0.1	0.022360706	0.022360705	0.022360705	5.7×10^{-10}	2.9×10^{-12}
	0.5	0.022360775	0.022360775	0.022360786	1.1×10^{-8}	1.1×10^{-8}
	0.9	0.022360845	0.022360845	0.022360867	2.27×10^{-8}	2.2×10^{-8}
1	0.1	0.022360712	0.022360711	0.022360711	1.1×10^{-9}	5.83×10^{-15}
	0.5	0.022360770	0.022360770	0.022360792	2.2×10^{-8}	2.2×10^{-8}
	0.9	0.022360827	0.022360828	0.022360873	4.5×10^{-8}	4.43×10^{-8}
5	0.1	0.022360761	0.022360755	0.022360755	5.72×10^{-9}	2.96×10^{-11}
	0.5	0.022360725	0.022360725	0.022360836	1.1×10^{-7}	1.1×10^{-7}
	0.9	0.022360689	0.022360695	0.022360917	2.27×10^{-7}	2.2×10^{-7}

Table 15: Error norms are computed at α=0.001, β=0.001, δ=3, dt=0.001, γ=0.001

t	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.005	8.0705×10^{-10}	7.8679×10^{-10}	4.1868×10^{-10}	4.8346×10^{-10}
0.01	1.6141×10^{-9}	1.5736×10^{-9}	8.3434×10^{-10}	9.6451×10^{-10}
0.5	8.0705×10^{-8}	7.8679×10^{-8}	4.1666×10^{-8}	4.8185×10^{-8}
1	1.6141×10^{-7}	1.5736×10^{-7}	8.333×10^{-8}	9.6371×10^{-8}

Table 16: Numerical and exact values and absolute errors for α=0.001, β=0.001, δ=3, dt=0.001, γ=0.001

t	x	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)
0.005	0.1	0.079370115	0.079370115	0.079370115	1.99×10^{-11}	3.9×10^{-13}
	0.5	0.079370364	0.079370364	0.079370364	3.94×10^{-10}	3.9×10^{-10}
	0.9	0.079370612	0.079370612	0.079370613	8.07×10^{-10}	7.87×10^{-10}
0.01	0.1	0.079370115	0.079370115	0.079370115	3.97×10^{-11}	7.81×10^{-13}
	0.5	0.079370363	0.079370363	0.079370364	7.87×10^{-10}	7.87×10^{-10}
	0.9	0.079370612	0.079370612	0.079370613	1.61×10^{-9}	1.57×10^{-9}
0.5	0.1	0.079370136	0.079370134	0.079370134	1.99×10^{-9}	3.9×10^{-11}
	0.5	0.079370344	0.079370344	0.079370383	3.94×10^{-8}	3.94×10^{-8}
	0.9	0.079370632	0.079370554	0.079370632	8.07×10^{-8}	7.87×10^{-8}
1	0.1	0.079370158	0.079370154	0.079370154	3.97×10^{-9}	7.79×10^{-11}
	0.5	0.079370324	0.079370324	0.079370403	7.86×10^{-8}	7.87×10^{-8}
	0.9	0.079370491	0.079370495	0.079370652	1.61×10^{-7}	1.57×10^{-7}
5	0.1	0.079370330	0.079370310	0.079370310	1.99×10^{-8}	3.87×10^{-10}
	0.5	0.079370166	0.079370166	0.079370559	3.94×10^{-7}	3.94×10^{-7}
	0.9	0.079370001	0.079370021	0.079370808	8.07×10^{-7}	7.87×10^{-7}
50	0.1	0.079372266	0.079372064	0.079372067	1.99×10^{-7}	3.61×10^{-9}
	0.5	0.079368381	0.079368380	0.079372316	3.94×10^{-6}	3.94×10^{-6}
	0.9	0.079364495	0.079364698	0.079372565	8.07×10^{-6}	7.87×10^{-6}

The error norms of LRBFDQ method are reported in Table 17 and Table 19 by the use of time step dt=0.01 and δ=2, 3 respectively. Moreover the absolute errors of LRBFDQ method are also mentioned in Table 18 and Table 20 with time step dt=0.01 and δ=2, 3 respectively up to time t=5. The value of c used for all these Tables are c= 1.388×10^{-15} for n_i=3 and c=1.0409×10^{-15} for n_i=5.

Example 7: In generalized Burger's-Huxely equation (1)-(4), put α=0 and δ=1, equation takes the form of Huxely equation. We take the parameter value as

β=1

$$u_t - u_{xx} = u(1-u)(u-\gamma), \qquad x \in \Omega = [a,b], t \geq 0$$

with initial condition:

$$u(x,0) = \frac{\gamma}{2} + \frac{\gamma}{2} tanh(\omega_1 x), \qquad x \in [0,1],$$

and boundary conditions

$$u(0,t) = \frac{\gamma}{2} + \frac{\gamma}{2} tanh(-\omega_1 \omega_2 t)$$
$$u(1,t) = \frac{\gamma}{2} + \frac{\gamma}{2} tanh(\omega_1 (1-\omega_2 t)), \qquad t > 0$$

Computations are performed by applying LRBFDQ by using shape parameter c=1.672×10-9 for ni=3 and c=1.0818×10-9 for ni=5.and γ=0.001.

Table 17: Error norms for α=1, β=1, δ=2, dt=0.01, γ=0.001

t	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.05	2.0398×10^{-6}	1.7486×10^{-6}	9.8662×10^{-7}	1.0749×10^{-6}
0.1	4.0795×10^{-6}	3.497×10^{-6}	1.963×10^{-6}	2.1437×10^{-6}
1	4.0766×10^{-5}	3.4953×10^{-5}	1.9594×10^{-5}	2.1409×10^{-5}
5	2.0319×10^{-4}	1.7438×10^{-4}	9.7899×10^{-5}	1.0687×10^{-4}

Table 18: Numerical and exact values and absolute errors for α=1, β=1, δ=2, dt=0.01, γ=0.001

t	X	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)
0.05	0.1	0.022361771	0.022361480	0.022361481	2.9×10^{-7}	1.15×10^{-9}
	0.5	0.022362549	0.022362549	0.022363423	8.75×10^{-7}	8.7×10^{-7}
	0.9	0.022363325	0.022363616	0.022365365	2.04×10^{-6}	1.75×10^{-6}
0.1	0.1	0.022362378	0.022361795	0.022361797	5.8×10^{-7}	2.26×10^{-9}
	0.5	0.022361990	0.022361990	0.022363739	1.75×10^{-6}	1.75×10^{-6}
	0.9	0.022361601	0.022362184	0.022365681	4.08×10^{-6}	3.5×10^{-6}
1	0.1	0.022373300	0.022367472	0.022367486	5.8×10^{-6}	1.37×10^{-8}
	0.5	0.022351932	0.022351921	0.022369427	1.75×10^{-5}	1.75×10^{-5}
	0.9	0.022330598	0.022336423	0.022371368	4.08×10^{-5}	3.49×10^{-5}
5	0.1	0.022422075	0.022392879	0.022392749	2.93×10^{-5}	1.3×10^{-7}
	0.5	0.022307174	0.022306904	0.022394689	8.75×10^{-5}	8.78×10^{-5}
	0.9	0.022193420	0.022222444	0.022396628	2.03×10^{-4}	1.74×10^{-4}

Table 19: Error norms for α=1, β=1, δ=3, dt=0.01, γ=0.001

t	L_∞ (n_i=3)	L_2 (n_i=3)
0.05	7.328×10^{-6}	3.572×10^{-6}
0.1	1.465×10^{-5}	7.108×10^{-6}
1	1.4642×10^{-4}	7.0952×10^{-5}

Table20: Numerical and exact values and absolute errors for α=1, β=1, δ=3, dt=0.01, γ=0.001

T	x	LRBFDQ (n_i=3)	Exact	LRBFDQ-AE (n_i=3)
0.05	0.1	0.079373754	0.079372811	9.4×10^{-7}
	0.5	0.079375815	0.079379007	3.19×10^{-6}
	0.9	0.079377874	0.079385202	7.3×10^{-6}
0.1	0.1	0.079375906	0.079374020	1.89×10^{-6}
	0.5	0.079373832	0.079380216	6.83×10^{-6}
	0.9	0.079371755	0.079386411	1.47×10^{-5}
1	0.1	0.079414710	0.079395776	1.89×10^{-5}
	0.5	0.079338123	0.079401969	6.38×10^{-5}
	0.9	0.079261739	0.079408160	1.46×10^{-4}
5	0.1	0.079588655	0.079492325	9.63×10^{-5}
	0.5	0.079179029	0.079498503	3.19×10^{-4}
	0.9	0.078775587	0.079504679	7.29×10^{-4}

The error norms of LRBFDQ method are reported in Table 21. Moreover for comparison purpose the absolute errors of LRBFDQ method, ADM (Ismail et al., 2004) and GRBF (Khattak, 2009) at specific nodes are also reported in Table 22 up to time t=50.

Example 8: Again consider the Huxley equation, i.e. take α=0 in the generalized Burger's-Huxely Eq. 1-4. Computations are performed by applying LRBFDQ and considering β=1 and γ=0.001.

The error norms of LRBFDQ method are reported in Table 23 and 25 for δ=2 and 3 respectively.

Similarly the absolute errors of LRBFDQ method, ADM (Ismail et al., 2004) and GRBF (Khattak, 2009) for δ=2, 3 are also mentioned in Table 24 and Table 26 up to time t=5. The value of c used in these tables are c= 1.672×10^{-9} for n_i=3 and c=1.0818×10^{-9} for n_i=5.

Table 21: Error norms for α=0, β=1, δ=1, dt=0.01, γ=0.001

t	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.05	2.4988×10^{-8}	5.1347×10^{-8}	2.1195×10^{-8}	3.0711×10^{-8}
0.1	4.9977×10^{-8}	1.0269×10^{-7}	4.2072×10^{-8}	6.1418×10^{-8}
1	4.9996×10^{-7}	1.0262×10^{-6}	4.197×10^{-7}	6.1359×10^{-7}
5	2.5041×10^{-6}	5.1155×10^{-6}	2.0993×10^{-6}	3.0649×10^{-6}
50	2.5541×10^{-5}	4.9458×10^{-5}	2.1085×10^{-5}	3.0125×10^{-5}

Table 22: Numerical and exact values and absolute errors for α=0, β=δ=1,dt=0.01, γ=0.001.

t	X	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)	GRBF (Khattak, 2009)	ADM (Ismail et al., 2004)
0.05	0.1	0.000500026	0.000500032	0.000500030	4.0×10^{-9}	1.37×10^{-9}	0.0×10^{-9}	1.88×10^{-7}
	0.5	0.000500076	0.000500076	0.000500101	2.50×10^{-8}	2.5×10^{-8}	1.0×10^{-9}	1.87×10^{-7}
	0.9	0.000500168	0.000500173	0.000500172	4.0×10^{-9}	1.37×10^{-9}	1.0×10^{-9}	1.87×10^{-7}
0.1	0.1	0.000500035	0.000500045	0.000500043	8.01×10^{-9}	2.7×10^{-9}	1.0×10^{-9}	3.75×10^{-7}
	0.5	0.000500063	0.000500063	0.000500113	5×10^{-8}	5×10^{-8}	0.0×10^{-9}	3.75×10^{-7}
	0.9	0.000500176	0.000500187	0.000500184	7.99×10^{-9}	2.7×10^{-9}	0.0×10^{-9}	3.75×10^{-7}
1	0.1	0.000500188	0.000500295	0.000500268	7.98×10^{-8}	2.7×10^{-8}	1.0×10^{-9}	37.5×10^{-7}
	0.5	0.000499839	0.000499838	0.000500338	5×10^{-7}	5×10^{-7}	0.0×10^{-9}	37.5×10^{-7}
	0.9	0.000500329	0.000500436	0.000500409	7.97×10^{-8}	2.7×10^{-8}	1.0×10^{-9}	37.5×10^{-7}
5	0.1	0.000500874	0.000501395	0.000501267	3.93×10^{-7}	1.28×10^{-7}	0.0×10^{-9}	
	0.5	0.000498839	0.000498837	0.000501338	2.5×10^{-6}	2.5×10^{-6}	1.0×10^{-9}	-
	0.9	0.000501016	0.000501537	0.000501408	3.93×10^{-7}	1.29×10^{-7}	0.0×10^{-9}	-
50	0.1	0.000509232	0.000513036	0.000512509	3.27×10^{-6}	5.27×10^{-7}	1.0×10^{-9}	
	0.5	0.000487597	0.000487410	0.000512579	2.5×10^{-5}	2.52×10^{-5}	0.0×10^{-9}	-
	0.9	0.000509381	0.000513182	0.000512650	3.27×10^{-6}	5.32×10^{-7}	0.0×10^{-9}	-

Table 23: Error norms for α=0, β=1, δ=2, dt=0.01, γ=0.001

T	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.05	1.1175×10^{-6}	2.2961×10^{-6}	9.4782×10^{-7}	1.3734×10^{-6}
0.1	2.2351×10^{-6}	4.592×10^{-6}	1.8814×10^{-6}	2.7466×10^{-6}
1	2.236×10^{-5}	4.5894×10^{-5}	1.8768×10^{-5}	2.7458×10^{-5}
5	1.1199×10^{-4}	2.2889×10^{-4}	9.3871×10^{-5}	1.3712×10^{-4}

Table 24: Numerical and exact values and absolute errors for α=0, β=1, δ=2, dt=0.01, γ=0.001

T	X	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)	GRBF (Khattak, 2009)	ADM (Ismail et al., 2004)
0.05	0.1	0.022361705	0.022361945	0.022361884	1.79×10^{-7}	6.1×10^{-8}	2×10^{-8}	5.8×10^{-7}
	0.5	0.022363348	0.022363348	0.022364466	1.12×10^{-6}	1.1×10^{-6}	3×10^{-8}	5.7×10^{-7}
	0.9	0.022366869	0.022367109	0.022367047	1.79×10^{-7}	6.1×10^{-8}	3×10^{-8}	5.50×10^{-7}
0.1	0.1	0.022362085	0.022362565	0.022362443	3.58×10^{-7}	1.2×10^{-7}	2×10^{-8}	1.14×10^{-6}
	0.5	0.022362790	0.022362790	0.022365025	2.23×10^{-6}	2.2×10^{-6}	2×10^{-8}	1.12×10^{-6}
	0.9	0.022367249	0.022367729	0.022367606	3.57×10^{-7}	1.2×10^{-7}	3×10^{-8}	1.11×10^{-6}
1	0.1	0.022368933	0.022373710	0.022372499	3.57×10^{-6}	1.1×10^{-6}	2×10^{-8}	1.12×10^{-5}
	0.5	0.022352731	0.022352728	0.022375079	2.33×10^{-5}	2.2×10^{-5}	3×10^{-8}	1.19×10^{-5}
	0.9	0.022374105	0.022378880	0.022377659	3.55×10^{-6}	1.2×10^{-6}	2×10^{-8}	1.12×10^{-5}
5	0.1	0.022399653	0.022423003	0.022417137	1.75×10^{-5}	5.87×10^{-6}	3×10^{-8}	
	0.5	0.022307971	0.022307888	0.022419712	1.12×10^{-4}	1.12×10^{-4}	2×10^{-8}	-
	0.9	0.022404856	0.022428197	0.022422287	1.74×10^{-5}	5.91×10^{-6}	3×10^{-8}	

Table 25: Error norms for α=0, β=1, δ=3, dt=0.01, γ=0.001

T	L_∞ (n_i=3)	L_∞ (n_i=5)	L_2 (n_i=3)	L_2 (n_i=5)
0.05	3.9667×10^{-6}	8.1497×10^{-6}	3.3641×10^{-6}	4.87468×10^{-6}
0.1	7.9336×10^{-6}	1.6299×10^{-5}	6.6777×10^{-6}	9.7486×10^{-6}
1	7.9336×10^{-5}	1.6292×10^{-4}	6.6614×10^{-5}	9.7466×10^{-5}

3. Conclusion

In this paper the local radial basis functions based differential quadrature collocation method has been extended to generalized Burger's-Huxley equation. Numerical results using the proposed method are obtained for generalized Burger's-Huxley equation and its variants (Burger's-Huxley equation, Huxley equation, Burger's equation and modified Burger's equation etc.). Gaussian radial basis function with three and five point central scheme in local support of each node is used. Infinity error norm, L_2 error norm and absolute error are used to measure the accuracy of the proposed method. Considerable accuracy through the proposed method is achieved. The results obtained in the form of accuracy are comparable with reference to global radial basis function collocation method while better than Adomian decomposition. It is also observed that in most cases the proposed method produces highly accurate results for small values of the parameters $\alpha, \beta, \gamma, dt, \delta$.

Table 26: Numerical and exact values and absolute errors for α=0, β=1, δ=3, dt=0.01, γ=0.001

T	x	LRBFDQ (n_i=3)	LRBFDQ (n_i=5)	Exact	LRBFDQ-AE (n_i=3)	LRBFDQ-AE (n_i=5)	GRBF[61] (Khattak, 2009)	ADM[59] (Ismail et al., 2004)
0.05	0.1	0.079373385	0.079374237	0.079374020	6.36×10^{-7}	2.17×10^{-7}	8×10^{-8}	1.02×10^{-6}
	0.5	0.079377990	0.079377990	0.079381956	3.97×10^{-6}	3.97×10^{-6}	9×10^{-8}	2×10^{-6}
	0.9	0.079389256	0.079390108	0.079389890	6.3×10^{-7}	2.19×10^{-7}	8×10^{-8}	2×10^{-6}
0.1	0.1	0.079374733	0.079376437	0.079376004	1.27×10^{-6}	4.33×10^{-7}	8×10^{-8}	4×10^{-6}
	0.5	0.079376007	0.079376007	0.079383939	7.9×10^{-6}	7.93×10^{-6}	9×10^{-8}	3.94×10^{-6}
	0.9	0.079390606	0.079392310	0.079391872	1.27×10^{-6}	4.37×10^{-7}	8×10^{-8}	3.97×10^{-6}
1	0.1	0.079399050	0.079416009	0.079411690	1.26×10^{-5}	4.32×10^{-6}	8×10^{-8}	3.97×10^{-5}
	0.5	0.079340297	0.079340286	0.079419618	7.93×10^{-5}	7.93×10^{-5}	9×10^{-8}	3.96×10^{-5}
	0.9	0.079414952	0.079431907	0.079427544	1.26×10^{-5}	4.36×10^{-6}	8×10^{-8}	3.96×10^{-5}

References

Ali A, Ali S, Arif M, and Hussain I (2012). Optimal Homotopy Asymptotic Method for the approximate solution of generalized Burger's–Huxley equation. Life Science Journal, 9(4): 3823-3828.

Babuska I and Melenk JM (1997). The partition of unity method. International Journal for Numerical Methods in Engineering, 40(4): 727–758.

Bellman RE and Casti J (1971). Differential quadrature and long-term integration. Journal of Mathematical Analysis and Applications, 34(2): 235-238.

Belytschko T, Lu YY, and Gu L (1994). Element-free Galerkin methods. International Journal for Numerical Methods in Engineering, 37(2): 229–256.

Celik I (2012). Haar wavelet method for solving generalized Burgers–Huxley equation. Arab Journal of Mathematical Sciences, 18(1): 25-37.

De S and Bathe KJ (2000). The method of finite spheres. Computational Mechanics, 25(4): 329-345.

Duarte C A and Oden JT (1995). Hp clouds–a meshless method to solve boundary-value problems. Taxes Institute Computational and Applied Mathematics Report, University of Texas at Austin, Austin, USA.

Efimova OYu and Kudryashov NA (2004). Exact solutions of the Burgers-Huxley equation. Journal of Applied Mathematics and Mechanics, 68(3): 413-420.

El-Kady M, El-Sayed SM, and Fathy HE (2013). Development of galerkin method for solving the generalized Burger's-Huxley equation. Mathematical Problems in Engineering, 2013: Article ID 165492, 9 pages. http://dx.doi.org/10.1155/2013/165492

Ervin VJ, Macías-Díaz JE, and Ruiz-Ramírez J (2015). A positive and bounded finite element approximation of the generalized Burgers–Huxley equation. Journal of Mathematical Analysis and Applications, 424(2): 1143-1160.

Hon YC and Mao XZ (1998). An efficient numerical scheme for Burgers' equation. Applied Mathematics and Computation, 95(1): 37–50.

Hong CP (2004). Computer modelling of heat and fluid flow in materials processing. CRC Press, Boca Raton, USA.

Ismail HNA, Raslan K, and Rabboh AAA (2004). Adomian decomposition method for Burgers'–Huxley and Burger's–Fisher equations. Applied Mathematics and Computation, 159(1): 291–301.

Jiwari R, Mittal RC, and Sharma KK (2013). A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers' equation. Applied Mathematics and Computation, 219(12): 6680-6691.

Khattak AJ (2009). A computational meshless method for the generalized Burger's–Huxley equation. Applied Mathematical Modelling, 33(9): 3718-3729.

Liu GR and Gu YT (2001). A local radial point interpolation method (LRPIM) for free vibration analyses of 2-D solids. Journal of Sound and Vibration, 246(1): 29–46.

Liu WK, Jun S, and Zhang YF (1995). Reproducing kernel particle methods. International Journal for Numerical Methods in Fluids, 20(8-9): 1081–1106.

Lucy LB (1977). A numerical approach to the testing of the fission hypothesis. Astrophysics Journal, 82: 1013–1024.

Macías-Díaz JE and Szafrańska A (2014). Existence and uniqueness of monotone and bounded solutions for a finite-difference discretization à la Mickens of the generalized Burgers–Huxley equation. Journal of Difference Equations and Applications, 20(7): 989-1004.

Mittal RC and Tripathi A (2015). Numerical solutions of generalized burgers–fisher and generalized Burgers–Huxley equations using collocation of cubic b-splines. International Journal of Computer Mathematics, 92(5): 1053-1077.

Nayroles B, Touzot G, and Villon P (1992). Generalizing the finite element method: Diffuse approximation and diffuse element. Computational Mechanics, 10(5): 307–318.

Onate E, Idelsohn S, Zienkiewicz OC, and Taylor RL (1996). A finite point method in computational mechanics, Application to convective transport and fluid flow. International Journal for Numerical Methods in Engineering, 39(22): 3839–3866.

Özisik MN (1994). Finite difference methods in heat transfer. CRC Press, Boca Raton, USA.

Rippa S (1984). Interpolation and smoothing of scattered data by radial basis functions. M.Sc. Thesis, Tel Aviv University, Tel Aviv, Israel.

Šarler B and Kuhn G (1998a). Dual reciprocity boundary element method for convective-diffusive solid-liquid phase change problems, Part 1 (Formulation). Engineering Analysis with Boundary Elements, 21(1): 53-63.

Šarler B and Kuhn G (1998b). Dual reciprocity boundary element method for convective-diffusive solid-liquid phase change problems, Part 2 (Numerical examples). Engineering Analysis with Boundary Elements, 21(1): 65-79.

Satsuma J, Ablowitz M, Fuchssteiner B, and Kruskal M (1987). Topics in soliton theory and exactly solvable nonlinear equations. World Scientific, Singapore, Singapore.

Shu C (2000). Differential quadrature and its application in engineering. Springer-Verlag, London, UK.

Shu C and Chew YT (1997). Fourier expansion-based differential quadrature and its application to Helmholtz eigenvalue problems. Communications in Numerical Methods in Engineering, 13: 643–653.

Shu C and Richards BE (1992). Application of generalized differential quadrature to solve two-dimens1ional incompressible Navier–Stokes equations. International Journal for Numerical Methods in Fluids, 15(7): 791–798.

Shu C and Xue H (1997). Explicit computation of weighting coefficients in the harmonic differential quadrature. Journal of Sound and Vibration, 204(3): 549-555.

Shu C, Ding H, and Yeo KS (2003). Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier– Stokes equations. Computer Methods in Applied Mechanics and Engineering, 192(7-8): 941–954.

Wu YL and Liu GR (2003). A meshfree formulation of local radial point interpolation method (LRPIM) for incompressible flow simulation. Computational Mechanics, 30(5): 355–365.

Wu YL and Shu C (2002). Development of RBF-DQ method for derivative approximation and its application to simulate natural convection in concentric annuli. Computational Mechanics, 29(6): 477-485.

Zienkiewicz OC and Taylor RL (2000). The finite element method: The basics. Butterworth-Heinemann, Oxford, UK.