

A Computational Simulation of Fractional Advection-Diffusion Model Using Differential Quadrature and Local Radial Basis Functions

Arshed Ali¹, Raza Ali Khan¹, Imtiaz Ahmad²

¹Department of mathematics, Islamia College Peshawar, Pakistan.

²Institute of Informatics and Computing in Energy (IICE), University Tenaga Nasional (UNITEN), Kajang, Selangor, Malaysia.

* **Correspondence:** Arshed Ali Email ID: arshad.ali@icp.edu.pk.

Citation | Ali. A, Khan. R. A, Ahmad. I, “A Computational Simulation of Fractional Advection-Diffusion Model Using Differential Quadrature and Local Radial Basis Functions”, IJIST, Vol. 07 Issue. 02 pp 926-938, May 2025.

Received | April 18, 2025 **Revised |** May 20, 2025 **Accepted |** May 22, 2025 **Published |** May 24, 2025.

This article presents a local radial basis function-based differential quadrature method for solving the time-fractional advection-diffusion equation. Backward difference formula is utilized to approximate Caputo fractional derivative. Differential quadrature approach is employed to compute the space derivatives by 3-point central scheme in the neighborhood of a node. Two types of radial basis functions are utilized in numerical simulations. Accuracy and computational efficiency of proposed technique is assessed via L_∞ , L_2 error norms, fractional order, time and spatial step sizes, rate of convergence and execution time. Three nonhomogeneous test problems are solved to validate the method, and the results are compared with finite volume method to show its superiority.

Keywords: Caputo fractional derivative, Radial basis functions, Fractional Advection-Diffusion equation, Differential quadrature, Backward difference formula.



Introduction:

Fractional partial differential equations (FPDEs) extend classical partial differential equations by incorporating derivatives of non-integer (fractional) order. This advancement provides a flexible mathematical framework for modeling complex phenomena across various disciplines including physics, biology, imaging science, and engineering, particularly in systems characterized by unusual behaviors such as memory effects, spatial heterogeneity, and long-range dependencies [1], [2], [3]. Unlike conventional PDEs, FPDEs are capable of capturing long-range dependencies in both time and space, making them particularly effective for describing processes such as viscoelastic behavior, fluid flow in porous media, and anomalous diffusion in irregular environments. Although the use of fractional derivatives adds complexity to both analytical and computational methods, it enables more accurate and realistic modeling of intricate systems that traditional equations may not adequately represent. Given the memory-dependent characteristics and intrinsic complexity of fractional differential equations, obtaining analytical solutions is often highly challenging and impractical.

Consequently, the design of efficient and accurate computational algorithms has become crucial for their numerical solution. In contrast to traditional mesh-based methods such as finite element, finite difference, and spectral techniques [4], [5], Radial Basis Function (RBF) based meshless methods utilize scattered nodes, removing the requirement for structured grid generation and providing enhanced flexibility for modeling complex and irregular domains [6], [7]. Moreover, mesh-based methods are time consuming, expensive and hard to implement while RBF provides highly accurate results, easily extendable to higher dimensional problems, and having simple implementation in complicated geometries [8], [9], [10], [11], [12]. Radial basis function (RBF) techniques that incorporate all interpolation nodes across the entire problem domain are referred to as global RBF interpolation methods. These methods construct a full interpolation matrix incorporating contributions from every node, which generally enhances accuracy as the number of nodes increases. However, this improvement comes at the cost of increased computational expense and ill-conditioning of the interpolation matrix, often leading to instability. To address the challenges of numerical differentiation, authors [13] introduced the differential quadrature (DQ) method in 1971 for approximating derivatives of sufficiently smooth functions. Recognized as an efficient alternative to finite difference and finite element methods, DQ offers high accuracy with fewer computational resources by using a relatively small set of nodal points [14], [15].

To mitigate the instability and ill-conditioning issues of global RBFs, author [16] proposed the local RBF method combined with differential quadrature for solving the incompressible 2D Navier–Stokes equations. In recent years, RBF-based methods have been successfully applied to a wide range of problems, including space-fractional diffusion equations on 3D irregular domains [17], multi-dimensional hyperbolic telegraph equations in nuclear materials science [18], dam break problems [19], 2D time-fractional Sobolev equations [20], the time-fractional cable equation [21], and oil water two-phase Darcy flow [22]. The fractional advection-diffusion equation is a significant type of FPDE used to model transport phenomena in complex systems, including air pollution, groundwater contamination, chemical solute dispersion, discharge of contaminated fluids, heat transfer processes, thermal pollution in river systems, and seawater intrusion, among others [23], [24]. In this study, we consider the following time-fractional advection-diffusion equation (TFADE), as presented in [24]:

$$\frac{\partial^\alpha}{\partial t^\alpha} u(x, t) + K_1 \frac{\partial}{\partial x} u(x, t) - K_2 \frac{\partial^2}{\partial x^2} u(x, t) = f(x, t), \quad 0 < \alpha < 1, a < x < b, 0 < t \leq T, \quad (1)$$

with

$$u(x, 0) = \varphi(x), \quad a \leq x \leq b, \quad (2)$$

$$u(a, t) = \chi(t), \quad u(b, t) = \zeta(t), \quad 0 < t \leq T, \quad (3)$$

where $u(x, t)$ denotes the field variable which can describe solute concentration besides other phenomenon, K_1 represents the constant velocity of fluid and K_2 represents the coefficient of dispersion, x is space variable and t is the time variable, and $\frac{\partial^\alpha}{\partial t^\alpha}$ denotes the α -order Caputo fractional derivative. $\varphi(x), \chi(t)$ and $\zeta(t)$ are given smooth functions while $f(x, t)$ represents sinks/sources.

Several researchers dedicated significant effort and proposed various methods for the solution of Eq. (1). author[23] solved Eq. (1) using finite volume element method. Author[24] utilized an upwind implicit finite difference method to get solution of Eq. (1). Author[25] used Lagrange square interpolation and weighted and shifted Legendre polynomials to get the solution of Eq. (1). Author[26] provided a Haar wavelet based technique to solve Eq. (1). Author[27] applied Legendre collocation method to find the solution of Eq. (1). Recently, author[28] obtained the approximate analytical solution of Eq. (1) using the homotopy analysis method.

Novelty Statement:

The novelty of this work lies in the integration of local RBFs with the differential quadrature method and the backward difference formula to overcome the limitations of global RBF methods, such as ill-conditioning and high computational cost. This hybrid meshfree technique offers an effective approach for approximating space-time fractional derivatives in complex domains, providing improved accuracy and convergence compared to existing finite volume and finite difference methods. The results of the proposed method are compared with those obtained using the finite volume element method (FVEM) [23] to demonstrate its superiority.

Objectives of the Study:

The primary objective of this study is to develop an accurate, stable, and computationally efficient numerical method for solving time-fractional advection-diffusion equations, which are widely used to model transport processes exhibiting memory and non-local effects.

Proposed Methodology:

To develop the local radial basis functions-based differential quadrature method, we discretized the space interval $[a, b]$ and time interval $[0, T]$ as $a = x_1 < x_2 < x_3 < \dots < x_M = b$ and $t_j = j\Delta t, j = 0, 1, 2, \dots, L$, respectively.

The following mathematical notions were used to achieve the suggested method:

Definition-1:

A function $\Phi: \mathbb{R}^n \rightarrow \mathbb{R}$ is called radial if there exists a univariate function $\varphi: [0, \infty) \rightarrow \mathbb{R}$ such that

$$\Phi(x) = \varphi(\|x\|), \quad \text{where } x \in \mathbb{R}^n \text{ and } \|\cdot\| \text{ represents the Euclidean norm [6].}$$

Definition-2:

A radial basis function $\varphi(r)$ is a one variable continuous real valued function. Furthermore, it relies on the distance from the origin (or any other fixed center point d) [6].

The following types of infinitely smooth RBFs are used in the proposed method [6]:

Multiquadric (MQ):

$$\varphi(r_j) = \sqrt{r_j^2 + c^2},$$

Gaussian (GA): $\varphi(r_j) = e^{-cr_j^2}$,

where $r_j = \|x - d_j\|$ and c is a shape parameter that impacts both the solution's accuracy and the conditioning of the system matrix (for details we refer to paper [6]).

Definition-3

The Caputo derivative $\frac{\partial^\alpha u(x,t)}{\partial t^\alpha}$ of order $\alpha, 0 \leq \alpha \leq 1$, has the following form [29]:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} \frac{\partial u(x,s)}{\partial s} ds. \quad (4)$$

Temporal Fractional Derivative Approximation:

The following theorem was used to approximate the time derivative in Eq. (1).

Theorem

The Caputo derivative given in (4) has the following approximation using backward difference formula [29]:

$$\frac{\partial^\alpha u(x,t_{N+1})}{\partial t^\alpha} = \frac{1}{\Gamma(2-\alpha)} \sum_{j=0}^N b_j \frac{u(x,t_{N+1-j}) - u(x,t_{N-j})}{(\Delta t)^\alpha} + r_{\Delta t}^{N+1} \quad (5)$$

where $b_j = (j+1)^{1-\alpha} - j^{1-\alpha}, j = 0, 1, 2, \dots, N$, and $r_{\Delta t}^{N+1}$ is the truncation error such that $r_{\Delta t}^{N+1} \leq C_u \Delta t^{2-\alpha}$.

Spatial Discretization and Approximation:

We used the DQ method [13] to approximate the spatial derivatives in Eq.(1) at $x = x_i$ as follows:

$$\frac{\partial^n u(x_i,t)}{\partial x^n} = \sum_{l=1}^M a_{il}^{(n)} u(x_l, t) \quad (6)$$

where $a_{il}^{(n)}, i = 1, 2, \dots, M$, are called weighting coefficients, and $n = 1, 2$.

Now using RBF to find the weighting coefficients $a_{il}^{(n)}, i = 1, 2, \dots, M$, we choose $x_{i_1}, x_{i_2}, \dots, x_{i_{M_i}}, M_i \ll M$, in the neighborhood of x_i and substituting in (6),

$$\varphi_k^{(n)}(x_i) \approx \sum_{l=i_1}^{i_{M_i}} a_{il}^{(n)} \varphi_k(x_l), \quad (7)$$

where $\varphi_k^{(n)}(x_i) = \frac{\partial^n \varphi(\|x_i - x_k\|)}{\partial x^n}, k = i_1, i_2, \dots, i_{M_i}$.

Matrix form of Eq. (7) is as follows:

$$\text{where } \boldsymbol{\varphi}_i = \begin{bmatrix} \varphi_{i_1}^{(n)}(x_i) \\ \vdots \\ \varphi_{i_{M_i}}^{(n)}(x_i) \end{bmatrix}, \quad \mathbf{A}_i = \begin{bmatrix} \varphi_{i_1}(x_{i_1}) & \dots & \varphi_{i_{M_i}}(x_{i_1}) \\ \vdots & \ddots & \vdots \\ \varphi_{i_1}(x_{i_{M_i}}) & \dots & \varphi_{i_{M_i}}(x_{i_{M_i}}) \end{bmatrix} \text{ and } \mathbf{a}_i^{(n)} = \begin{bmatrix} a_{ii_1}^{(n)} \\ \vdots \\ a_{ii_{M_i}}^{(n)} \end{bmatrix}.$$

Thus,

$$\mathbf{a}_i^{(n)} = \mathbf{A}_i^{-1} \boldsymbol{\varphi}_i. \quad (8)$$

Thus, using Eqs. (4) and (8) in (1), we get

$$\frac{1}{\Gamma(2-\alpha)} \sum_{j=0}^N \frac{u(x_i,t_{N+1-j}) - u(x_i,t_{N-j})}{(\Delta t)^\alpha} b_j + \left(K_1 \mathbf{a}_i^{(1)} - K_2 \mathbf{a}_i^{(2)} \right)^t \mathbf{u}_i^{N+1} = f(x_i, t_{N+1}), \quad (9)$$

where $\mathbf{u}_i^{N+1} = [u_{i_1}^{N+1}, u_{i_2}^{N+1}, \dots, u_{i_{M_i}}^{N+1}]^t, i = 1, 2, \dots, M$.

Algorithm:

The proposed method (9) can be implemented in the following steps:

1. M nodes were selected from the space domain $[a, b]$ such that $a = x_1 < x_2 < \dots < x_M = b$.
2. The time step size Δt and fractional order α were chosen such that $0 < \alpha \leq 1$.
3. For each node x_i a set of neighboring nodes $x_{i_1}, x_{i_2}, \dots, x_{i_{M_i}}$ was selected.
4. The weighting coefficients were computed using the radial basis function (RBF) approach.

5. The initial solution u_i^0 was calculated from Eq. (2), and Eq. (9) was solved to obtain u_i^{N+1}

6. Steps 3 to 5 were repeated to determine the solution at each node x_i .

The flow diagram of the proposed algorithm is shown below:

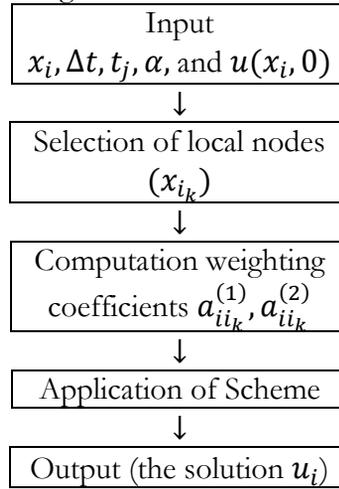


Figure 1. Flow diagram showing implementation of the local RBF-DQ Method

This section presents numerical simulations of the proposed method. To validate its effectiveness, three examples were implemented over the spatial domain $[0, 1]$. To assess reliability, performance and accuracy of the present method, we utilized the following error norms L_∞ , L_2 , and rate of convergence (RoC) defined as [10]:

$$L_\infty = \max_{1 \leq i \leq M} |u_i - \hat{u}_i|, L_2 = \sqrt{h \sum_{i=1}^M (u_i - \hat{u}_i)^2}, \text{ and } \text{RoC} = \frac{\log_{10}(\|u - \hat{u}_{\delta t_i}\| / \|u - \hat{u}_{\delta t_{i+1}}\|)}{\log_{10}(\delta t_i / \delta t_{i+1})}$$

where u_i and \hat{u}_i represent exact and approximate solutions respectively. Moreover, to compare our results with the finite volume element method [23], the values of $K_1 = 1$ and $K_2 = 1$ were taken. In each example, the functions $\varphi(x), \chi(t), \zeta(t)$ were extracted from the exact solution. The values of shape parameter c used to get accurate results are mentioned in tables. Two-GB RAM, Corei3, 2.4GHz processor, and uniformly distributed spatial nodes were used for implementation of the proposed method. Computer execution time (RT) was calculated in seconds.

Results and Discussion:

Test Example-1

We utilized analytical solution as $u(x, t) = e^{xt^\beta}$, in the first example, and source function as:

$$f(x, t) = \frac{\Gamma(\beta + 1)}{\Gamma(\beta + 1 - \alpha)} e^{xt^{\beta-\alpha}} \quad (10)$$

Simulations were performed with various values of the parameters $\alpha, \beta, \delta t, h$ and MQ and GA types of RBFs, and the computed results are recorded in Tables 1-3. In Table 1, L_∞, L_2 and computer run time are provided at different times $t, 0 \leq t \leq 1$ using $\alpha = 0.5, \beta = 5, \delta t = 0.001, h = 0.01$, MQ and GA. Table 2 shows the computed values of L_∞, L_2 and temporal RoC for $\alpha = 0.1, 0.5, 0.9, \beta = 5, N = 100$ and $\delta t = 0.1, 0.05, 0.025, 0.0125$, MQ and GA. Table 3 provides L_∞, L_2 and spatial RoC for $\alpha = 0.1, 0.5, 0.9, \beta = 5, h = 0.1, 0.05, 0.025, 0.0125, \delta t = 0.001$, MQ and GA. It can be observed from Tables 1-3, that MQ provides better accuracy than GA. Moreover, the accuracy of the proposed method improves as the time step is reduced. Graphical illustration of approximate solutions by the present scheme using MQ and error in approximation for $\delta t = 0.001, h = 0.05, \alpha = 0.5, t = 1$ is given in Figures 2 and 3. Figure 2(a) presents both

the exact and approximate solutions, while Figure 2(b) illustrates the error associated with the approximate solutions. Figure 3 displays approximate solutions at different times in the interval $0 \leq t \leq 1$.

Table 1. Error norms and RT using $\delta t = 0.001, N = 100, \alpha = 0.5$

t	MQ ($c = 5.2$)			GA ($c = 0.1$)		
	L_∞	L_2	RT(sec)	L_∞	L_2	RT(sec)
0.1	$1.80e^{-8}$	$1.31e^{-8}$	0.2931	$1.80e^{-8}$	$1.31e^{-8}$	0.2345
0.2	$1.65e^{-7}$	$1.20e^{-7}$	0.6018	$1.65e^{-7}$	$1.20e^{-7}$	0.4858
0.3	$5.90e^{-7}$	$4.29e^{-7}$	0.9439	$5.95e^{-7}$	$4.32e^{-7}$	0.7659
0.4	$1.44e^{-6}$	$1.05e^{-6}$	1.3110	$1.46e^{-6}$	$1.06e^{-6}$	1.0848
0.5	$2.85e^{-6}$	$2.07e^{-6}$	1.7386	$2.92e^{-6}$	$2.12e^{-6}$	1.4345
0.6	$4.95e^{-6}$	$3.60e^{-6}$	2.1723	$5.13e^{-6}$	$3.72e^{-6}$	1.8676
0.7	$7.84e^{-6}$	$5.69e^{-6}$	2.6665	$8.23e^{-6}$	$5.97e^{-6}$	2.2579
0.8	$1.16e^{-5}$	$8.40e^{-6}$	3.2106	$1.24e^{-5}$	$8.96e^{-6}$	2.7502
0.9	$1.62e^{-5}$	$1.18e^{-5}$	3.7848	$1.76e^{-5}$	$1.28e^{-5}$	3.2801
1	$2.17e^{-5}$	$1.58e^{-5}$	4.3995	$2.42e^{-5}$	$1.75e^{-6}$	3.8526

Table 2. Error Norms and RoC at $t = 1$ using $N = 100$

α	δt	c	MQ			GA			
			L_∞	L_2	RoC (L_∞)	c	L_∞	L_2	RoC (L_∞)
0.1	0.1	0.66	$1.32e^{-3}$	$9.57e^{-4}$	---	0.1	$1.37e^{-3}$	$9.95e^{-4}$	---
	0.05		$3.84e^{-4}$	$2.78e^{-4}$	1.7846		$4.36e^{-4}$	$3.16e^{-4}$	1.6567
	0.025		$8.02e^{-5}$	$5.81e^{-5}$	2.2582		$1.32e^{-4}$	$9.58e^{-5}$	1.7199
	0.0125		$1.36e^{-5}$	$9.88e^{-6}$	2.5552		$3.84e^{-5}$	$2.78e^{-5}$	1.7831
0.5	0.1	0.3	$1.84e^{-2}$	$1.34e^{-2}$	---	0.05	$1.99e^{-2}$	$1.44e^{-2}$	---
	0.05		$6.29e^{-3}$	$4.56e^{-3}$	1.5503		$7.71e^{-3}$	$5.59e^{-3}$	1.3660
	0.025		$1.48e^{-3}$	$1.07e^{-3}$	2.0861		$2.89e^{-3}$	$2.10e^{-3}$	1.4161
	0.0125		$3.43e^{-4}$	$2.50e^{-4}$	2.1099		$1.06e^{-3}$	$7.69e^{-4}$	1.4467
0.9	0.1	0.2	$9.38e^{-2}$	$6.81e^{-2}$	---	0.11	$1.01e^{-1}$	$7.35e^{-2}$	---
	0.05		$4.26e^{-2}$	$3.10e^{-2}$	1.1378		$4.95e^{-2}$	$3.60e^{-2}$	1.0307
	0.025		$1.70e^{-2}$	$1.24e^{-2}$	1.3228		$2.37e^{-2}$	$1.72e^{-2}$	1.0635
	0.0125		$4.66e^{-3}$	$3.38e^{-3}$	1.8696		$1.12e^{-2}$	$8.14e^{-3}$	1.0809

Table 3. Error Norms and RoC at $t = 1$ using $\delta t = 0.001$

α	h	c	MQ			GA			
			L_∞	L_2	RoC (L_∞)	c	L_∞	L_2	RoC (L_∞)
0.1	0.1	1.32	$7.73e^{-5}$	$5.60e^{-5}$	---	0.19	$9.40e^{-5}$	$6.82e^{-5}$	---
	0.05		$1.95e^{-5}$	$1.41e^{-5}$	1.9865		$2.32e^{-5}$	$1.68e^{-5}$	2.0206
	0.025		$4.59e^{-6}$	$3.33e^{-6}$	2.0879		$5.47e^{-6}$	$3.96e^{-6}$	2.0823
	0.0125		$8.26e^{-7}$	$5.98e^{-7}$	2.4749		$1.04e^{-6}$	$7.51e^{-7}$	2.3992
0.5	0.1	0.5	$1.46e^{-2}$	$1.06e^{-2}$	---	3.0	$1.47e^{-2}$	$1.06e^{-2}$	---
	0.05		$4.06e^{-3}$	$2.95e^{-3}$	1.8460		$3.76e^{-3}$	$2.73e^{-3}$	1.9656
	0.025		$1.03e^{-3}$	$7.45e^{-4}$	1.9844		$9.62e^{-4}$	$6.99e^{-4}$	1.9644
	0.0125		$2.39e^{-4}$	$1.74e^{-4}$	2.0992		$2.60e^{-4}$	$1.89e^{-4}$	1.8875
0.9	0.1	0.32	$6.72e^{-2}$	$4.90e^{-2}$	---	14.0	$2.75e^{-1}$	$2.00e^{-2}$	---
	0.05		$2.10e^{-2}$	$1.53e^{-2}$	1.6759		$7.05e^{-2}$	$5.13e^{-2}$	1.9644
	0.025		$5.12e^{-3}$	$3.73e^{-3}$	2.0391		$1.82e^{-2}$	$1.32e^{-2}$	1.9546
	0.0125		$7.76e^{-4}$	$5.67e^{-4}$	2.7208		$5.08e^{-3}$	$3.70e^{-3}$	1.8407

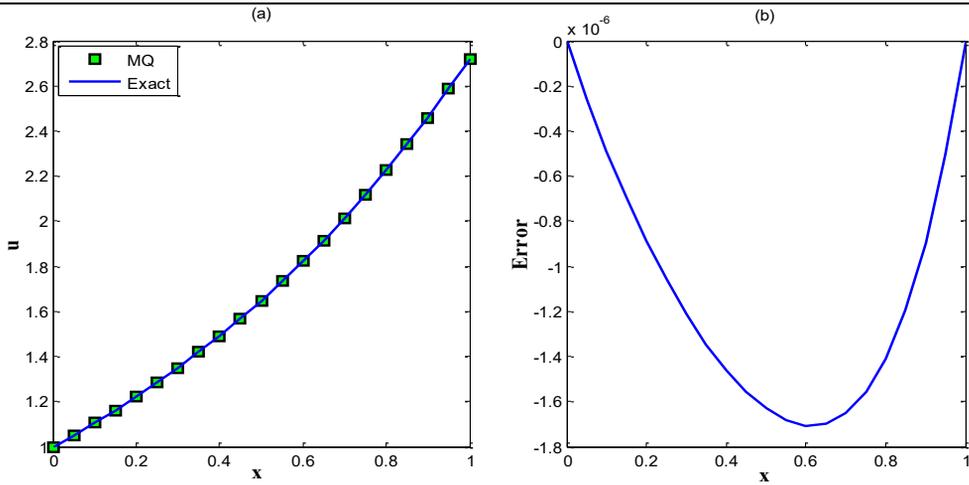


Figure 2. Plots of (a) Exact and approximate solutions (b) Error in approximate solution, at $t = 1$ for $\delta t = 0.001, h = 0.05, \alpha = 0.5$ corresponding to Test Example-1

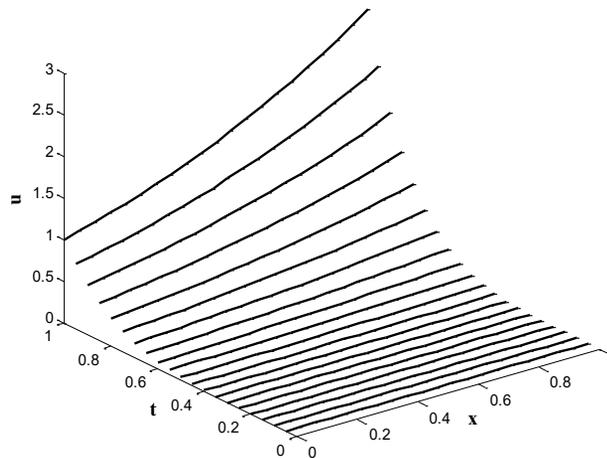


Figure 3. Profile of approximate solutions at different times over the space interval $[0, 1]$ for $\delta t = 0.001, h = 0.05, \alpha = 0.5$ corresponding to Test Example-1

Test Example 2:

In this example, we considered the exact analytical solution $u(x, t) = t^2x(1 - x^2)$, so that the source term is given by:

$$f(x, t) = \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} x(1 - x) + t^2(3 - 2x) \quad (11)$$

Computations are carried out and the results of the proposed method are noted in Tables 4-5 along with the results of FVEM for comparison. In Table 4, L_∞, L_2 are provided at time $t = 0.5$ using $\alpha = 0.5, h = 0.001$, different time step sizes δt and MQ. Table 5 provides the computed values of L_∞, L_2 for $\alpha = 0.5, \delta t = 0.001$, various space step sizes h , and MQ. It can be seen from Tables 3 and 4, that the present method provided more accurate solution than the FVEM. Furthermore, accuracy improves with mesh refinement, i.e., smaller δt and h , reflecting the convergence characteristics of the proposed numerical scheme. Plot of approximate solutions obtained by the current method with MQ and error in approximation for $\delta t = 0.001, h = 0.03125, \alpha = 0.5$ is given in Figures 4 and 5. Figure 4(a) displays exact and approximate solutions while Figure 4(b) shows error in the approximate solutions, at $t = 1$. Surface graph of approximate solutions at various times over the interval $0 \leq t \leq 1$, is shown in Figure 5.

Table 4. Comparison of errors in approximate solutions at $t = 0.5$ using the present method with MQ, $h = 0.001, \alpha = 0.5$ and FVEM

δt	c	MQ		FVEM	
		L_∞	L_2	L_∞	L_2
0.5	0.036	$4.2042e^{-4}$	$2.9716e^{-4}$	$2.9458e^{-3}$	$2.0888e^{-3}$
0.25	0.043	$4.5684e^{-5}$	$3.3322e^{-5}$	$1.1591e^{-3}$	$8.2174e^{-3}$
0.125	0.056	$1.4337e^{-5}$	$9.5842e^{-6}$	$4.3374e^{-4}$	$3.0747e^{-4}$
0.0625	0.071	$2.6269e^{-6}$	$1.59188e^{-6}$	$1.6155e^{-4}$	$1.1451e^{-4}$
0.03125	0.094	$1.5269e^{-6}$	$9.3414e^{-7}$	$5.8876e^{-5}$	$4.1734e^{-5}$
0.015625	0.123	$2.8622e^{-7}$	$1.7318e^{-7}$	$2.4184e^{-5}$	$1.7144e^{-5}$
0.007812	0.164	$1.52136e^{-7}$	$9.2554e^{-8}$	$4.0481e^{-6}$	$2.8682e^{-6}$

Table 5. Comparison of errors in approximate solutions at $t = 0.5$ using the present method with MQ, $\delta t = 0.001, \alpha = 0.5, c = 4$ and FVEM

h	MQ		FVEM	
	L_∞	L_2	h	L_∞
0.25	$1.9477e^{-4}$	$1.4294e^{-4}$	0.25	$1.9477e^{-4}$
0.125	$4.9091e^{-5}$	$3.6120e^{-5}$	0.125	$4.9091e^{-5}$
0.0625	$1.2056e^{-5}$	$8.8780e^{-6}$	0.0625	$1.2056e^{-5}$
0.03125	$2.7571e^{-6}$	$2.0367e^{-6}$	0.03125	$2.7571e^{-6}$
0.015625	$4.3102e^{-7}$	$3.2544e^{-7}$	0.015625	$4.3102e^{-7}$
0.0078125	$2.4053e^{-8}$	$1.2094e^{-8}$	0.0078125	$2.4053e^{-8}$

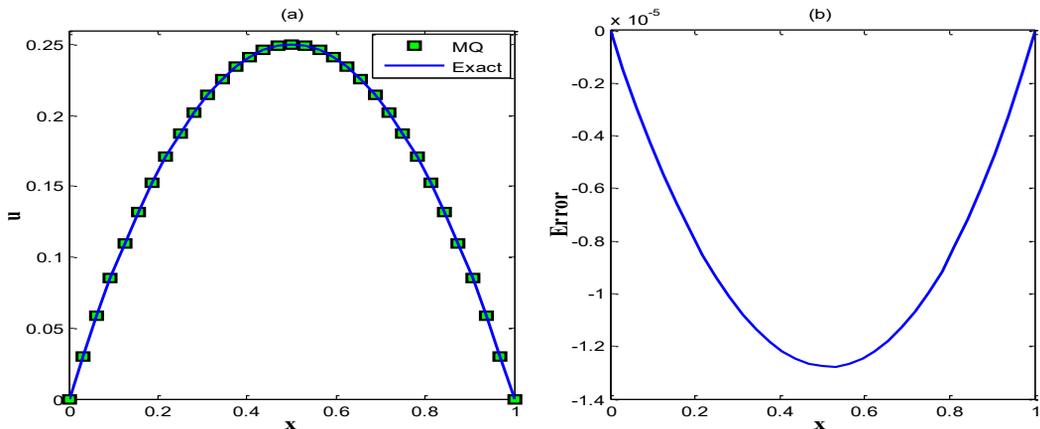


Figure 4. Plots of (a) Exact and approximate solutions (b) Error in approximate solution, at $t = 1$ for $\delta t = 0.001, h = 0.03125, \alpha = 0.5$ corresponding to Test Example-2

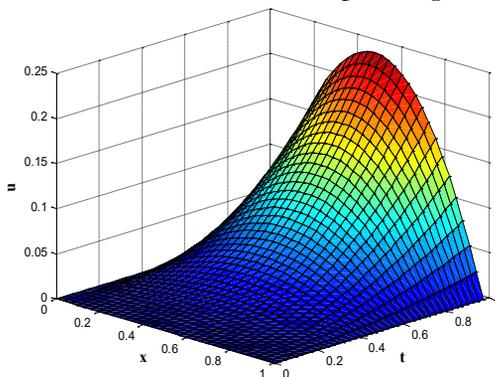


Figure 5. 3D Surface plot of approximate solutions over the domain $[0, 1] \times [0, 1]$ for $\delta t = 0.001, h = 0.03125, \alpha = 0.5$ corresponding to Test Example-2.

Test Example 3

In the third example, we take $u(x, t) = t^2 \sin(\pi x)$ as the analytical solution (1), then the source term is given by

$$f(x, t) = \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} \sin(\pi x) + \pi t^2 (\cos(\pi x) + \pi \sin(\pi x)) \quad (12)$$

Numerical results are given in Tables 6-7 for $\alpha = 0.05$ and $\alpha = 0.95$ alongwith the results of FVEM for comparison. In Table 6, L_∞, L_2 are provided at time $t = 0.5$ using $h = 0.01$, different time step sizes δt and MQ. Table 7 provides the computed values of L_∞, L_2 for $\delta t = 0.01, t = 0.5$, different space step sizes h , and MQ. It can be noted from Tables 5 and 6, that the proposed method provided better accuracy than the FVEM. Additionally, the accuracy of the proposed method improves as both the time step δt and the space step size h are reduced. Plot of approximate solutions obtained by the current method with MQ and error in approximation for $\delta t = 0.01, h = 0.03125, \alpha = 0.05$ is given in Figures. 6 and 7. Figure 6(a) displays exact and approximate solutions and Figure 6(b) shows error plot at $t = 1$. Profile of approximate solutions at different times over the interval $0 \leq t \leq 1$, is given in Figure 7.

Table 6. Comparison of errors in approximate solutions at $t = 0.5$ using present method with MQ, $h = 0.01, \alpha = 0.05, 0.95$ and FVEM

	δt	MQ			FVEM	
		c	L_∞	c	L_∞	c
$\alpha = 0.05$	0.25	0.41	$1.7980e^{-5}$	0.41	$1.7980e^{-5}$	0.41
	0.125	0.61	$4.3982e^{-6}$	0.61	$4.3982e^{-6}$	0.61
	0.0625	0.81	$3.0722e^{-6}$	0.81	$3.0722e^{-6}$	0.81
	0.03125	1.05	$2.6443e^{-6}$	1.05	$2.6443e^{-6}$	1.05
	0.015625	1.10	$2.3538e^{-6}$	1.10	$2.3538e^{-6}$	1.10
$\alpha = 0.95$	0.25	0.095	$4.4368e^{-4}$	0.095	$4.4368e^{-4}$	0.095
	0.125	0.112	$1.3345e^{-4}$	0.112	$1.3345e^{-4}$	0.112
	0.0625	0.135	$8.6710e^{-5}$	0.135	$8.6710e^{-5}$	0.135
	0.03125	0.165	$4.2574e^{-5}$	0.165	$4.2574e^{-5}$	0.165
	0.015625	0.204	$2.4899e^{-5}$	0.204	$2.4899e^{-5}$	0.204

Table 7. Comparison of errors in approximate solutions at $t = 0.5$ using present method with MQ, $\delta t = 0.01, \alpha = 0.05, 0.95$ and FVEM

	h	c	MQ		FVEM	
			L_∞	L_2	L_∞	L_2
$\alpha = 0.05$	0.25	1.1	$1.7817e^{-3}$	$1.0465e^{-3}$	$1.7817e^{-3}$	$1.0465e^{-3}$
	0.125	1.1	$3.3501e^{-4}$	$2.2696e^{-4}$	$3.3501e^{-4}$	$2.2696e^{-4}$
	0.0625	1.1	$9.9547e^{-5}$	$5.9224e^{-5}$	$9.9547e^{-5}$	$5.9224e^{-5}$
	0.03125	1.1	$2.4476e^{-5}$	$1.4934e^{-5}$	$2.4476e^{-5}$	$1.4934e^{-5}$
	0.015625	1.1	$5.8366e^{-6}$	$3.6418e^{-6}$	$5.8366e^{-6}$	$3.6418e^{-6}$
$\alpha = 0.95$	0.25	1.0	$1.3573e^{-3}$	$8.1712e^{-4}$	$1.3573e^{-3}$	$8.1712e^{-4}$
	0.125	1.0	$3.8755e^{-4}$	$2.3198e^{-4}$	$3.8755e^{-4}$	$2.3198e^{-4}$
	0.0625	0.78	$8.5797e^{-5}$	$5.2524e^{-5}$	$8.5797e^{-5}$	$5.2524e^{-5}$
	0.03125	0.51	$3.7031e^{-5}$	$2.4495e^{-5}$	$3.7031e^{-5}$	$2.4495e^{-5}$
	0.015625	0.313	$1.7548e^{-5}$	$1.2310e^{-5}$	$1.7548e^{-5}$	$1.2310e^{-5}$

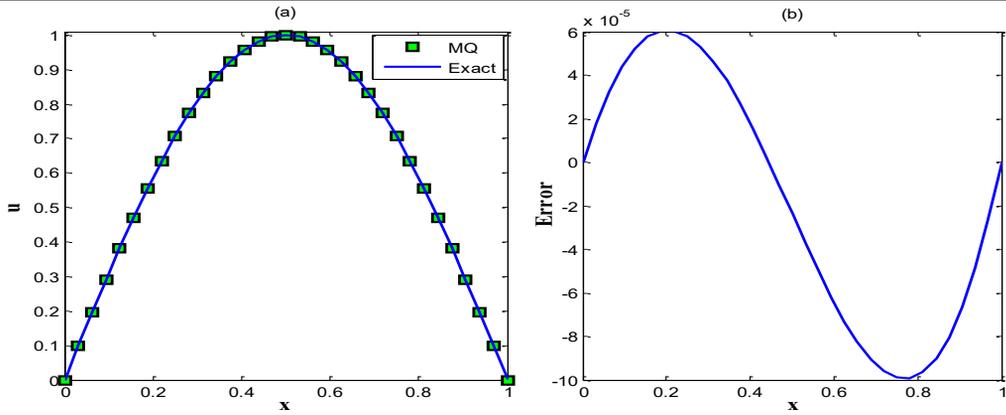


Figure 6. Plots of (a) Exact and approximate solutions (b) Error in approximate solution, at $t = 1$ for $\delta t = 0.01, h = 0.03125, \alpha = 0.05$ corresponding to Test Example-3

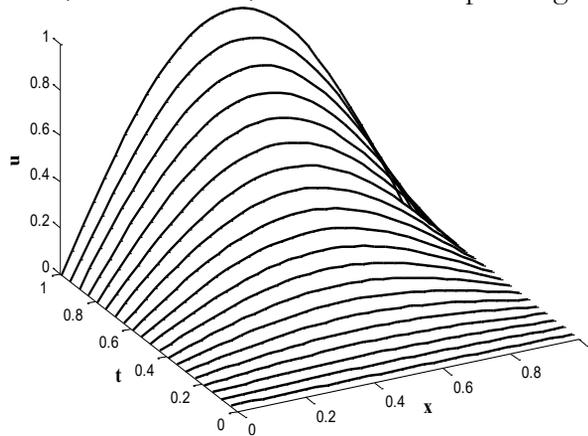


Figure 7. Profile of approximate solutions at different times over the space interval $[0, 1]$ for $\delta t = 0.01, h = 0.03125, \alpha = 0.05$ corresponding to Test Example-3

Conclusion:

A local radial basis functions based differential quadrature method is presented for approximate solution of time-fractional Advection-Diffusion equations having Caputo derivative. Three benchmark nonhomogeneous problems are provided for its validation, and the results are analyzed through error norms, fractional order, numerical convergence, and factional order. The method is computationally efficient and the reported results demonstrate that the present method produced better accuracy for relatively small time and space step sizes. The outlined technique achieved better accuracy compared to finite volume element method. Based on remarkably agreement with the exact solution, this approach is efficient, accurate, simple, and economical for obtaining approximate solutions of wide class of fractional Advection-Diffusion equations.

Author’s Contribution:

Conceptualization: Arshed Ali and Raza Ali Khan; Data curation: Raza Ali Khan and Imtiaz Ahmad; Formal analysis: Imtiaz Ahmad; Investigation: Arshed Ali and Raza Ali Khan; Methodology: Arshed Ali and Raza Ali Khan; Resources: Arshed Ali and Imtiaz Ahmad; Software: Arshed Ali and Raza Ali Khan; Supervision: Arshed Ali; Validation, Arshed Ali and Raza Ali Khan; Writing original draft: Raza Ali Khan; Writing review & editing, Arshed Ali and Imtiaz Ahmad.

Conflict of interest: The authors declare that they no competing interest for publishing this manuscript in IJIST.

References:

- [1] “Theory and applications Of Fractal Differential Equations | Request PDF.” Accessed: May 19, 2025. [Online]. Available: https://www.researchgate.net/publication/216225153_Theory_and_applications_Of_Fractal_Differential_Equations
- [2] Z. Z. Yunying Zheng, Changpin Li, “A note on the finite element method for the space-fractional advection diffusion equation,” *Comput. Math. with Appl.*, vol. 59, no. 5, pp. 1718–1726, 2010, doi: <https://doi.org/10.1016/j.camwa.2009.08.071>.
- [3] Ercília Sousa, “A second order explicit finite difference method for the fractional advection diffusion equation,” *Comput. Math. with Appl.*, vol. 64, no. 10, pp. 3141–3152, 2012, doi: <https://doi.org/10.1016/j.camwa.2012.03.002>.
- [4] S. A. T. Algazaa and J. Saeidian, “Spectral methods utilizing generalized Bernstein-like basis functions for time-fractional advection–diffusion equations,” *Math. Methods Appl. Sci.*, vol. 48, no. 2, pp. 1411–1429, Jan. 2024, doi: [10.1002/MMA.10390;PAGE:STRING:ARTICLE/CHAPTER](https://doi.org/10.1002/MMA.10390;PAGE:STRING:ARTICLE/CHAPTER).
- [5] M. W. Harbir Antil, Thomas Brown, Ratna Khatri, Akwum Onwunta, Deepanshu Verma, “Chapter 3 - Optimal control, numerics, and applications of fractional PDEs,” *Handb. Numer. Anal.*, vol. 23, pp. 87–114, 2022, doi: <https://doi.org/10.1016/bs.hna.2021.12.003>.
- [6] E.J. Kansa, “Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—I surface approximations and partial derivative estimates,” *Comput. Math. with Appl.*, vol. 19, no. 8–9, pp. 127–145, 1990, doi: [https://doi.org/10.1016/0898-1221\(90\)90270-T](https://doi.org/10.1016/0898-1221(90)90270-T).
- [7] S. H. Siraj-ul-Islam, Arshed Ali, “A computational modeling of the behavior of the two-dimensional reaction–diffusion Brusselator system,” *Appl. Math. Model.*, vol. 34, no. 12, pp. 3896–3909, 2010, doi: <https://doi.org/10.1016/j.apm.2010.03.028>.
- [8] M. Dehghan and A. Shokri, “A meshless method for numerical solution of the one-dimensional wave equation with an integral condition using radial basis functions,” *Numer. Algorithms*, vol. 52, no. 3, pp. 461–477, Oct. 2009, doi: [10.1007/S11075-009-9293-0/METRICS](https://doi.org/10.1007/S11075-009-9293-0/METRICS).
- [9] A. A. Iltaf Hussain, Safyan Mukhtar and, “A Numerical Meshless Technique for the Solution of the two Dimensional Burger’s Equation Using Collocation Method,” *World Appl. Sci. J.*, vol. 23, no. 12, pp. 29–40, 2013, doi: [10.5829/idosi.wasj.2013.23.12.2840](https://doi.org/10.5829/idosi.wasj.2013.23.12.2840).
- [10] A. Ali, F. Haq, I. Hussain, and U. Shah, “A meshless method of lines for numerical solution of some coupled nonlinear evolution equations,” *Int. J. Nonlinear Sci. Numer. Simul.*, vol. 15, no. 2, pp. 121–128, Mar. 2014, doi: [10.1515/IJNSNS-2012-0035/MACHINEREADEABLECITATION/RIS](https://doi.org/10.1515/IJNSNS-2012-0035/MACHINEREADEABLECITATION/RIS).
- [11] J. Lin, J. Bai, S. Reutskiy, and J. Lu, “A novel RBF-based meshless method for solving time-fractional transport equations in 2D and 3D arbitrary domains,” *Eng. Comput.*, vol. 39, no. 3, pp. 1905–1922, Jun. 2023, doi: [10.1007/S00366-022-01601-0/METRICS](https://doi.org/10.1007/S00366-022-01601-0/METRICS).
- [12] Q. X. Luchuan Shi, “Application of Generalized Finite Difference Method and Radial Basis Function Neural Networks in Solving Inverse Problems of Surface Anomalous Diffusion,” *Math. Comput. Appl.*, vol. 30, no. 1, p. 7, 2025, doi: <https://doi.org/10.3390/mca30010007>.
- [13] R. Bellman and J. Casti, “Differential quadrature and long-term integration,” *J. Math. Anal. Appl.*, vol. 34, no. 2, pp. 235–238, 1971, doi: [https://doi.org/10.1016/0022-247X\(71\)90110-7](https://doi.org/10.1016/0022-247X(71)90110-7).
- [14] Siraj-Ul-Islam, A. Ali, A. Zafar, and I. Hussain, “A Differential Quadrature Based

- Approach for Volterra Partial Integro-Differential Equation with a Weakly Singular Kernel,” *Comput. Model. Eng. Sci.*, vol. 124, no. 3, pp. 915–935, Aug. 2020, doi: 10.32604/CMES.2020.011218.
- [15] T. A. Imtiaz Ahmad, Mehwish Saleem, Arshed Ali, Aziz Khan, “A Computational Analysis of Convection-Diffusion Model with Memory using Caputo-Fabrizio Derivative and Cubic Trigonometric B-Spline Functions,” *Eur. J. Pure Appl. Math.*, vol. 18, no. 2, 2025, [Online]. Available: <https://www.ejpam.com/index.php/ejpam/article/view/6186>
- [16] C. Shu, H. Ding, and K. . Yeo, “Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations,” *Comput. Methods Appl. Mech. Eng.*, vol. 192, no. 7–8, pp. 941–954, 2003, doi: [https://doi.org/10.1016/S0045-7825\(02\)00618-7](https://doi.org/10.1016/S0045-7825(02)00618-7).
- [17] X. G. Zhu, Y. F. Nie, Z. H. Ge, Z. B. Yuan, and J. G. Wang, “A class of RBFs-based DQ methods for the space-fractional diffusion equations on 3D irregular domains,” *Comput. Mech.*, vol. 66, no. 1, pp. 221–238, Jul. 2020, doi: 10.1007/S00466-020-01848-8/METRICS.
- [18] I. Ahmad, A. R. Seadawy, H. Ahmad, P. Thounthong, and F. Wang, “Numerical study of multi-dimensional hyperbolic telegraph equations arising in nuclear material science via an efficient local meshless method,” *Int. J. Nonlinear Sci. Numer. Simul.*, vol. 23, no. 1, pp. 115–122, Feb. 2022, doi: 10.1515/IJNSNS-2020-0166/MACHINEREADABLECITATION/RIS.
- [19] M. V. Abdol Mahdi Behrooz, Claudio I. Meier, “Radial basis function-based differential quadrature for dam break problems,” *Eng. Anal. Bound. Elem.*, vol. 155, pp. 307–322, 2023, doi: <https://doi.org/10.1016/j.enganabound.2023.06.020>.
- [20] B. Almutairi, I. Ahmad, B. Almohsen, H. Ahmad, and D. U. Ozsahin, “NUMERICAL SIMULATIONS OF TIME-FRACTIONAL PDES ARISING IN MATHEMATICS AND PHYSICS USING THE LOCAL MESHLESS DIFFERENTIAL QUADRATURE METHOD,” *Therm. Sci.*, vol. 27, no. Special Issue 1, pp. 263–272, 2023, doi: 10.2298/TSCI23S1263A.
- [21] N. Biranvand and A. Ebrahimijahan, “Utilizing differential quadrature-based RBF partition of unity collocation method to simulate distributed-order time fractional Cable equation,” *Comput. Appl. Math.*, vol. 43, no. 1, pp. 1–26, Feb. 2024, doi: 10.1007/S40314-023-02507-3/METRICS.
- [22] S. Lv, D. Li, W. Zha, and Y. Xing, “Physics-informed radial basis function neural network for efficiently modeling oil-water two-phase Darcy flow,” *Phys. Fluids*, vol. 37, no. 1, Jan. 2025, doi: 10.1063/5.0249560/3329262.
- [23] M. Badr, A. Yazdani, and H. Jafari, “Stability of a finite volume element method for the time-fractional advection-diffusion equation,” *Numer. Methods Partial Differ. Equ.*, vol. 34, no. 5, pp. 1459–1471, Sep. 2018, doi: 10.1002/NUM.22243.
- [24] A. V. K. and L. J. SWAPNALI DOLEY, “UPWIND SCHEME OF CAPUTO TIME FRACTIONAL ADVECTION DIFFUSION EQUATION,” *Adv. Appl. Math. Sci.*, vol. 21, no. 3, pp. 1239–1247, 2022, [Online]. Available: https://www.mililink.com/upload/article/1223602015aams_vol_213_january_2022_a14_p1239-1247_swapnali_doley_a_vanav_kumar_and_l_jino.pdf
- [25] Y. E. Aghdam, H. Mesgarani, and Z. Asadi, “Estimate of the fractional advection-diffusion equation with a time-fractional term based on the shifted Legendre polynomials,” *J. Math. Model.*, vol. 11, no. 4, pp. 731–744, Dec. 2023, doi: 10.22124/JMM.2023.24479.2191.
- [26] K. S. N. Sh. Ahmed, Sh. Jahan, “Haar wavelet based numerical technique for the solutions of fractional advection diffusion equations,” *J. Math. Comput. Sci.*, vol. 34, no.

- 3, pp. 217--233, 2024, doi: <https://dx.doi.org/10.22436/jmcs.034.03.02>.
- [27] S. Kumar, K. Kumar, R. K. Pandey, and Y. Xu, "Legendre collocation method for new generalized fractional advection-diffusion equation," *Int. J. Comput. Math.*, Oct. 2024, doi: 10.1080/00207160.2024.2305640;REQUESTEDJOURNAL:JOURNAL:GCOM20;SUBPAGE:STRING:ACCESS.
- [28] Z. Odibat, "Approximate analytical solutions of time-fractional advection–diffusion equations using singular kernel derivatives: a comparative study," *Nonlinear Dyn.*, vol. 113, no. 13, pp. 16427–16442, Jul. 2025, doi: 10.1007/S11071-025-10991-X/METRICS.
- [29] S. Arshed, "B-spline solution of fractional integro partial differential equation with a weakly singular kernel," *Numer. Methods Partial Differ. Equ.*, vol. 33, no. 5, pp. 1565–1581, Sep. 2017, doi: 10.1002/NUM.22153.



Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.