A local stabilized approach for approximating the modified time-fractional diffusion problem arising in heat and mass transfer

O. Nikan, Z. Avazzadeh, J.A. Tenreiro Machado

School of Mathematics, Iran University of Science and Technology, Narmak, Tehran, Iran
Department of Applied Mathematics, Xi’an Jiaotong-Liverpool University, Suzhou 215123, China
Institute of Engineering, Polytechnic of Porto, Department of Electrical Engineering, Rua Dr. António Bernardino de Almeida, 431, 4249-015 Porto, Portugal

Highlights
- The modified time-fractional diffusion model plays an important role in heat transfer and fluid flow problems.
- A meshless method based on hybridization of Gaussian and cubic kernels is developed to solve the model.
- The proposed method has good accuracy and applicable over complex domains with various node distributions.
- The time-discrete algorithm is examined in terms of the convergence and unconditional stability properties using the energy method.

Graphical Abstract

Abstract

Introduction: During the last years the modeling of dynamical phenomena has been advanced by including concepts borrowed from fractional order differential equations. The diffusion process plays an important role not only in heat transfer and fluid flow problems, but also in the modeling of pattern formation that arises in porous media. The modified time-fractional diffusion equation provides a deeper understanding of several dynamic phenomena.

Methods: The temporal discretization is performed by integrating both sides of the modified time-fractional diffusion model. The unconditional stability of the time discretization scheme and the optimal convergence rate are obtained. Then, the spatial derivatives are discretized through a local hybridization of the cubic and Gaussian radial basis function. This hybrid kernel improves the condition of the system matrix. Therefore, the solution of the linear system can be obtained using direct solvers that reduce significantly computational cost. The main idea of the method is to consider the distribution of data points over the local support domain where the number of points is almost constant.

Results: Three examples show that the numerical procedure has good accuracy and applicable over complex domains with various node distributions. Numerical results on regular and irregular domains illustrate the accuracy, efficiency and validity of the technique.
Introduction

During the last years the modeling of dynamical phenomena has been advanced by including concepts borrowed from fractional order differential equations (FDEs). This generalisation of derivatives and integrals led to the field of mathematics called fractional calculus [1,2]. The history of fractional calculus started in the 17th century, but only recently became popular in applied sciences. The application of FDEs has expanded rapidly and we can mention for example the analysis of problems such as creep or relaxation in visco-elastoplastic materials, diffusion process models, control, and plasma physics problems [3-6]. Nonetheless, in many applications is necessary to solve FDEs using efficient numerical techniques [7-14].

The diffusion process plays an important role not only in heat transfer and fluid flow problems, but also in the modeling of pattern formation that arise in porous media. The time-fractional diffusion equation describes the mass transfer processes involving memory effects [15]. The mean square displacement (MSD) of the diffusing species in standard diffusion is linear with respect to time. One can define the MSD as a product of the diffusion coefficient and time [16].

If the memory effects are present to some mass transfer, then the process no longer follows a Brownian motion and the diffusion species cannot be described by Fick's second law [17]. In these cases, the time-fractional diffusion equation can explain better the time-fractional diffusion problem. The main results of the research is the numerical technique with non-uniform distribution in irregular grids.

Conclusion: This paper adopts a local hybrid kernel meshless approach to solve the modified time-fractional diffusion problem. The main results of the research is the numerical technique with non-uniform distribution in irregular grids.

© 2021 The Authors. Published by Elsevier B.V. on behalf of Cairo University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
rather than following the connections among them. Spectral meshless techniques based on the radial basis functions (RBFs) are gaining popularity in the geosciences due to their competitive accuracy, functionality on unstructured meshes. The RBF technique is an efficient method for interpolation of multidimensional scattered data [36–38], since it does not requires meshes and is able to solve easily complex geometrical and high-dimension problems.

Therefore, the RBFs became popular for approximating partial differential equations on complex domains and have been applied in the solution of physical problems with theoretical proofs on solvability and convergence [36–38].

Hardy [39] initially used the RBF to interpolate scattered points for the approximation of irregular surfaces that have an important role in meshfree techniques. Kansa [40] employed the RBF interpolation technique to obtain the approximate solution of a PDE using all the domain points (thereafter named the Kansa’s method). The results obtained from applying the Kansa’s method lead to a large, dense, and ill-conditioned linear system [41]. For overcoming the above problems with the Kansa’s approach, a “local” version of RBF was proposed giving spectral accuracy for a sparse, better-conditioned linear system [42]. For overcoming the time step limitations of each one and keeping the formulation as a standard advantage of the merits of the two kernels while compensating for the RBF combining Gaussian kernels with cubic splines. The basic idea produces an approach to deal with these two issues through hybrid functions in meshless techniques for specific node arrangements conditioned. On the other hand, the cubic RBF is infinitely smooth, provide an invertible system matrix in meshless methods. The infinitely smooth RBF, such as Gaussian, Fisher-Kolmogorov equations in one-, two-, and three-dimensions. Dehghan and Abbaszadeh [43] developed a fast and efficient numerical algorithm by combining the local RBF-FD approach with the proper orthogonal decomposition (POD) and the discrete empirical interpolation method (DEIM) to simulate the time dependent incompressible Navier–Stokes model. Dehghan et al. [44] proposed a local meshless RBF-FD method for solving the regularized long-wave and extended Fisher-Kolmogorov equations in one-, two-, and three-dimensions. Dehghan and Abbaszadeh [45] considered a linear combination of the shape functions of local RBF collocation method with the moving Kriging interpolation technique. Avazzadeh et al. [46] applied the local RBF-FD to obtain the solitary wave solutions of the generalized Rosenau–Korteweg-de Vries–regularized-long wave equation. Nikan et al. [47–49] adopted the local RBF-FD collocation approach for solving the time fractional equations.

This work proposes a local stabilized approach, generated through the hybrid Gaussian-cubic RBF and FD for obtaining the solution of the MTFDE. The infinitely smooth RBF, such as Gaussian, provide an invertible system matrix in meshless methods. Nevertheless, for a large number of domain nodes and small shape parameters, a Gaussian RBF yields a system of equations that is ill-conditioned. On the other hand, the cubic RBF is infinitely smooth and without any shape parameter, unlike what occurs with the Gaussian RBF. However, using a cubic RBF to interpolate shape functions in meshless techniques for specific node arrangement has the risk of obtaining a singular system. Indeed, this paper introduces an approach to deal with these two issues through hybrid RBF combining Gaussian kernels with cubic splines. The basic idea behind such hybridization is to obtain a kernel that takes advantage of the merits of the two kernels while compensating for the limitations of each one and keeping the formulation as a standard RBF method. The outline of this paper is as follows. The Section entitled “Local hybridization of the Gaussian and cubic kernels” describes the spatial desensitization with the of the local hybrid kernel meshless techniques. The Section entitled “Numerical examples” illustrates the numerical algorithm by means of three problems that highlight the accuracy and efficiency of the proposed approach. Finally, the Section entitled “Conclusion” synthesizes the main concluding remarks.

**Time-discrete formulation**

In this section, a time discrete algorithm to approximate the (1) in the time variable is formulated. For this purpose, we split the time interval [0, T] into small cells with time step length δt and nodes \( t_k = k\delta t \), for \( k = 0, 1, 2, \ldots, M \), where \( M \) is a positive integer.

**Definition 1.** Suppose that \( y(t) \in L^1(a, b) \), and that the expression

\[
\ell^t y(t) = \frac{1}{(\gamma)^{1/2}} \int_0^t y(\eta) (t - \eta)^{\gamma - 1/2} d\eta, \quad t > a, \gamma > 0,
\]

denotes the R-L fractional integral operator of order \( \gamma \).

**Lemma 1.** ([50]) If \( y(t) \in C^1[0, 1] \), then

\[
\ell^t y(t_{k+1}) - \ell^t y(t_k) = \frac{\delta t^2}{(\gamma + 1)^{1/2}} \left[ y(t_{k+1}) + \sum_{j=0}^{k-1} (b_j - b_{j+1}) y(t_{k-j}) \right] + R_k,
\]

where \( |R_k| \leq C_0 \delta t^{1+\gamma} \), \( 1 = b_0 > b_1 > \ldots > b_k > 0 \) and \( b_j = (j+1)^\gamma - j^\gamma \).

Integrating both sides of Eq. (1) on \( [t_k, t_{k+1}] \), gives

\[
\frac{1}{\delta t} f(x, \eta) d\eta = \delta t f(x, t_{k+1}) + O(\delta t^2),
\]

By means of Lemma 1 and the following equation

\[
\int_{t_k}^{t_{k+1}} f(x, \eta) d\eta = \delta t f(x, t_{k+1}) + O(\delta t^2),
\]

we obtain

\[
u^{k+1} - (\mu_1 + \mu_2)\Delta u^{k+1} = u^k + \mu_1 \sum_{j=0}^{k-1} (\lambda_j - \lambda_j) \Delta u^{k-j} + \mu_2 \sum_{j=0}^{k-1} (\omega_j - \omega_j) \Delta u^{k-j} + \delta t f^{k+1} + \delta t R_k^{k+1},
\]

where \( \mu_1 = \frac{\nu_1 \delta t^2}{(\gamma + 1)^{1/2}}, \mu_2 = \frac{\nu_2 \delta t^2}{(\gamma + 1)^{1/2}}, \lambda_j = (j + 1)^\gamma - j^\gamma, \omega_j = (j + 1)^\gamma - j^\gamma \).

By eliminating the small error term \( R_k^{k+1} \) and introducing \( U^{k+1} \) as the numerical approximation, results:

\[
U^{k+1} - (\mu_1 + \mu_2)\Delta U^{k+1} = U^k + \mu_1 \sum_{j=0}^{k-1} (\lambda_j - \lambda_j) \Delta U^{k-j} + \mu_2 \sum_{j=0}^{k-1} (\omega_j - \omega_j) \Delta U^{k-j} + \delta t f^{k+1}.
\]

**Theoretical analysis of the difference scheme**

We define the functional spaces endowed with the standard norms and inner products for the proposed difference algorithm as
A finite difference algorithm is called to be stable for the norm \( \| \cdot \| \), if we have two constants \( C_1 > 0 \) and \( C_2 > 0 \), independent of \( \delta t \), such that when \( \delta t \) approaches zero, we can write:

\[
\left\| U^k \right\|^2 \leq C_1 \left\| V^0 \right\|^2 + C_2 \| f \|,
\]

where \( f \) and \( V^0 \) represent the source term and the initial data, respectively.

**Lemma 2.** ([52]) If we consider

\[
\eta_{k+1} = \frac{\delta t}{\Gamma(\alpha)(k+1)^{\frac{\alpha}{2}}}, \quad \xi_{k+1} = \frac{\delta t}{\Gamma(\beta)(k+1)^{1+\beta}},
\]

then we have the following inequalities:

\[
\mu_1 \eta_{k+1} \leq \eta_{k+2}, \quad \mu_2 \eta_{k+1} \leq \xi_{k+1} \leq \mu_2 \xi_{k+1}.
\]

**Theorem 1.** If \( U^k \in H_2^1(\Omega) \), then the implicit numerical method (6) is unconditionally stable in the \( H^1 \)-norm.

**Proof.** We multiply Eq. (6) by \( U^{k+1} \) and integrate on \( \Omega \),

\[
\left\| U^{k+1} \right\|^2 + \mu_1 \left\| \nabla U^{k+1} \right\|^2 + \mu_2 \left\| \Delta U^{k+1} \right\|^2 = \left\langle \left( \delta_t^2 U^k, U^{k+1} \right) + \mu_1 \sum_{j=0}^{k-1} \left( \xi_{j+1} - \xi_j \right) \left\langle \nabla U^{-j}, \nabla U^{k+1} \right\rangle \right.
\]

\[
+ \mu_2 \sum_{j=0}^{k-1} \left( \eta_{j+1} - \eta_j \right) \left\langle \nabla U^{-j}, \nabla U^{k+1} \right\rangle + \delta t \left\langle j^{k+1}, U^{k+1} \right\rangle.
\]

Considering the Cauchy–Schwarz inequality, one obtains

\[
\left\| U^{k+1} \right\|^2 + \mu_1 \left\| \nabla U^{k+1} \right\|^2 + \mu_2 \left\| \Delta U^{k+1} \right\|^2
\]

\[
\leq \left\langle \left( \delta_t^2 U^k, U^{k+1} \right) + \mu_1 \sum_{j=0}^{k-1} \left( \xi_{j+1} - \xi_j \right) \left\langle \nabla U^{-j}, \nabla U^{k+1} \right\rangle + \delta t \left\langle j^{k+1}, U^{k+1} \right\rangle \right.
\]

\[
+ \mu_2 \sum_{j=0}^{k-1} \left( \eta_{j+1} - \eta_j \right) \left\langle \nabla U^{-j}, \nabla U^{k+1} \right\rangle + \delta t \left\langle j^{k+1}, U^{k+1} \right\rangle.
\]

According to the Poincaré and Young’s inequalities, we get

\[
\left\| U^{k+1} \right\|^2 \leq C \left\| \nabla U^{k+1} \right\|^2,
\]

\[
\forall \theta = 0 \rightarrow |ab| \leq \frac{1}{2 \theta^2} a^2 + \frac{\theta^2}{2} b^2,
\]

respecively, and we obtain

\[
\left\| U^{k+1} \right\|^2 + \mu_1 \left\| \nabla U^{k+1} \right\|^2 + \mu_2 \left\| \Delta U^{k+1} \right\|^2
\]

\[
\leq \left\langle \left( \delta_t^2 U^k, U^{k+1} \right) + \mu_1 \sum_{j=0}^{k-1} \left( \xi_{j+1} - \xi_j \right) \left\langle \nabla U^{-j}, \nabla U^{k+1} \right\rangle + \delta t \left\langle j^{k+1}, U^{k+1} \right\rangle \right.
\]

\[
+ \mu_2 \sum_{j=0}^{k-1} \left( \eta_{j+1} - \eta_j \right) \left\langle \nabla U^{-j}, \nabla U^{k+1} \right\rangle + \delta t \left\langle j^{k+1}, U^{k+1} \right\rangle \right.
\]

\[
+ \left( \frac{\mu_1}{2} + \frac{\mu_2}{2} \right) \left\| \nabla U^{k+1} \right\|^2 + \left( \frac{\mu_1}{2} + \frac{\mu_2}{2} \right) \left\| \nabla U^{k+1} \right\|^2 + \frac{\delta t^2}{2} \left\langle j^{k+1}, U^{k+1} \right\rangle \right.
\]

\[
+ \left( \frac{\mu_1}{2} + \frac{\mu_2}{2} \right) \left\| \nabla U^{k+1} \right\|^2 + \left( \frac{\mu_1}{2} + \frac{\mu_2}{2} \right) \left\| \nabla U^{k+1} \right\|^2 + \frac{\delta t^2}{2} \left\langle j^{k+1}, U^{k+1} \right\rangle \right.
\]

After simplification, we arrive at

\[
\frac{1}{2} \left\| U^{k+1} \right\|^2 + \frac{\mu_1}{2} \sum_{j=0}^{k-1} \left\| \nabla U^{-j} \right\|^2 + \frac{\mu_2}{2} \sum_{j=0}^{k-1} \left\| \nabla U^{k+1} \right\|^2
\]

\[
\leq \frac{1}{2} \left\| U^k \right\|^2 + \frac{\mu_1}{2} \sum_{j=0}^{k-1} \left\| \nabla U^{-j} \right\|^2 + \frac{\mu_2}{2} \sum_{j=0}^{k-1} \left\| \nabla U^{k+1} \right\|^2
\]

\[
+ \frac{\delta t^2}{2} \left\langle j^{k+1}, U^{k+1} \right\rangle \right.
\]

\[
+ \frac{\delta t^2}{2} \left\langle j^{k+1}, U^{k+1} \right\rangle \right.
\]

\[
+ \frac{\delta t^2}{2} \left\langle j^{k+1}, U^{k+1} \right\rangle \right.
\]

\[
+ \frac{\delta t^2}{2} \left\langle j^{k+1}, U^{k+1} \right\rangle \right.
\]
Therefore, we obtain
\[ \varsigma_{k+1} \leq \varsigma^0 + \left( \frac{c_2}{n} \right)^{k}\|f^{k+1}\|^2 \]
\[ \leq \varsigma^0 + \sum_{j=0}^{k} \frac{c_2^j}{n^j} \|f^{k+1}\|^2 \]
\[ \leq \varsigma^0 + \frac{c_2^k}{n^k} \max_{0 \leq j \leq k} \|f^{j}\|^2, \]
where
\[ \varsigma^0 = \|U^0\|^2 + \mu_1 \sum_{j=0}^{k} \gamma_j \|\nabla U^{k-j}\|^2 + \mu_2 \sum_{j=0}^{k} \alpha_j \|\nabla U^{k-j}\|^2. \]

If we consider
\[ \varsigma^0 = \|U^0\|^2 + \mu_1 \|\nabla U^0\|^2 + \mu_2 \|\nabla U^0\|^2 = \|U^0\|^2, \]
then expression (8) can be rewritten as
\[ \|U^{k+1}\|_{H^1(\Omega)} \leq \|U^0\|_{H^1(\Omega)} + \left( \frac{C \delta T}{\varsigma_{k+1} + \eta_{k+1}} \right)^{1/2} \max_{0 \leq j \leq k} \|f^{j}\|^2, \quad k \geq 0. \]

From [53], for \( k \geq 0 \), it holds that
\[ \frac{T \delta t}{\varsigma_{k+1} + \eta_{k+1}} = \frac{T \delta t}{\varsigma_{k} + \eta_{k}} \leq T^2 \delta t \Gamma(\varsigma), \]
\[ \frac{T \delta t}{\varsigma_{k+1} + \eta_{k+1}} = \frac{T \delta t}{\varsigma_{k} + \eta_{k}} \leq T^2 \delta t \Gamma(\beta). \]

In view of the aforesaid inequalities and the relation (9), we get
\[ \|U^{k+1}\|_{H^1(\Omega)} \leq \|U^0\|_{H^1(\Omega)} + C T \delta t \left( \frac{\min(\varsigma, \beta)}{\delta t} \right)^{1/2} \max_{0 \leq j \leq k} \|f^{j}\|^2, \]
and the proof is concluded. \( \square \)

**Theorem 2.** Let \( u^{k+1} = u(x, t_{k+1}) \in H^1(\Omega) \) represent the exact solution of Eqs. (1)–(3) and \( U^{k+1} \) be the approximate solution obtained by (6). Then, the semi-explicit difference algorithm (6) has convergence rate \( c(\delta t^{1-\min(\varsigma, \beta)}) \).

**Proof.** We define \( \rho^{k+1} = u^k - U^{k+1} \) at \( t = t_{k+1}, \quad k = 0, 1, \ldots, M \). By subtracting Eq. (5) from Eq. (6), we obtain
\[ \rho^{k+1} - \left( \mu_1 + \mu_2 \right) \Delta \rho^{k+1} = \rho_0 + \mu_2 \sum_{j=1}^{k} (\gamma_j - 1) \Delta \rho^{k-j} + \mu_2 \sum_{j=1}^{k} (\alpha_j - 1) \Delta \rho^{k-j} + \delta t \rho^{k+1}, \]
where \( \rho^{k+1} \leq \delta t^{1-\min(\varsigma, \beta)} \) and \( \rho_0 = 0 \). Regarding Theorem 1 and Eq. (10), the following inequality is obtained:
\[ \|\rho^{k+1}\|_{L^2(\Omega)} \leq C T \delta t^{1-\min(\varsigma, \beta)}, \]
which finishes the proof of Theorem 2. \( \square \)

**Local hybridization of the Gaussian and cubic kernels**

The RBF is a value dependent function on the distance from the center point. The RBF can be easily implemented by using the distance functions to solve multi-dimensional PDE. In the RBF method, the approximation of the function \( u(x) \) to the data values
\[ u_j = u(x_j), \quad j = 1, \ldots, N, \]
at the centers \( X_C = \{x_1, \ldots, x_N\} \subseteq \mathbb{R}^d \), is represented as
\[ u(x) \simeq S(x) = \sum_{j=1}^{N} \beta_j \phi_j(x, \varepsilon). \]
where \( \beta_j \) are unknown coefficients, \( \phi_j(x, \varepsilon) = \phi_j(||x - x_j||, \varepsilon) \), and \( \varepsilon \) denotes the shape parameter.

Mishra et al. [56] first introduced the hybrid Gaussian-cubic RBF as a stable method for interpolation problems involving scattered data. Later, Mishra et al. [57,58] developed a stabilized RBF-FD approach with a hybrid kernel, generated through the cubic and Gaussian RBF hybridization based on FD. It was shown that this hybrid kernel was able to calculate stable interpolants for scattered nodes when there were a large number of degrees of freedom. In addition, the ill-conditioning in the numerical solution of PDEs was reduced. The hybrid Gaussian-cubic kernel is expressed by the relationship:
\[ \phi(r, \varepsilon) = c_1 \exp(-c_2 r^2) + c_2 r^3, \]
with \( c_1 \) and \( c_2 \) representing the weights that control the Gaussian and cubic kernels, respectively, and \( \varepsilon \) standing for the shape parameter of the Gaussian kernel
\[ \phi(r, \varepsilon) = \exp(-c_1 r^2) + \varepsilon \theta r^3, \]
where the parameters \( \varepsilon \) and \( \theta = c_2/c_1 \) play a significant role in the stability and accuracy of the proposed technique. The parameters that control the cubic and Gaussian kernels in the hybrid RBF are chosen via global particle swarm optimization. The proposed strategy was observed to considerably decrease the ill-conditioning in the global RBF (GRBF) collocation technique. Moreover, it maintains the accuracy and stability for considerably small shape parameters. The term "stabilized" can be interpreted in two ways here [57,58]. On one hand, it can refer to the significant reduction in the interpolation matrix condition number. On the other hand, it can also point to the stabilization of the eigenvalue spectra of the system matrix belonging to the linear system irrespective of the stencil size or any stencil irregularity. This hybrid RBF overcomes the ill-conditioning issue in RBF-FD and results in a stabilized evaluation for a low computational cost, equivalent to the evaluation obtained from the GRBF. The proposed strategy decreases considerably the ill-conditioning in the GRBF. Moreover, it maintains the accuracy and stability for considerably small shape parameters [57,58]. Orcu [59] applied the local hybrid kernel meshless method to approximate the fractional cable equation.

Kansa [60,61] adopted a linear differential operator \( \mathcal{L} \) for interpolating (11) in order approximate \( \mathcal{L} u \) at the N scatter nodes,
\[ \mathcal{L} u(x_j) \simeq \sum_{j=1}^{N} \beta_j \phi_j(x_j, \varepsilon). \]

Eq. (14) represents a GRBF approximation since one needs all points in the domain to approximate \( \mathcal{L} \) at one of the points. That is to say, in order to determine the interpolation coefficient, one needs to consider all the spatial domain points. However, a large and ill-conditioned linear system is generated in the GRBF, which may leads to uncertain results. On the other hand, one can use the local RBF method only for the local support domain (restricted nodes) on each center rather than for the whole set of points. The local RBF results in a sparse and better-conditioned linear system and, at the same time, is reliable for ill-conditioned problems.
Suppose that \( \Xi = \{ x_1, \ldots, x_k \} \subseteq \mathbb{R}^d \) is a set of \( N \) distinct points computational region \( \Omega \) that splits into two sets \( I \) and \( B \), which are the indices of the internal and boundary nodes, respectively, and that \( N \) denotes the total number of nodes so that \( N = N_I + N_B \). Moreover, consider the subset \( \Xi \) that is a vector with center \( x \) that including the center number and the indices of the \( N_I - 1 \) nearest neighboring centers. Each center and its \( N_I - 1 \) neighbors are named a stencil. Fig. 1 represents a typical stencil. Our target of this part is to approximate the Laplacian operator \( \Delta u(x) = \sum_{i=1}^{N_I} w_i \left( u(x_i) - u(x) \right) \) through the proposed method, where \( x = (x, y)^T \). The linear differential operator \( \Delta \) at a reference node \( x \), can be approximated using the weighted linear sum of function values at all nodes in its support domain as:

\[
\Delta u(x) \approx \sum_{j=1}^{N_I} w_j u(x_j).
\]  

(15)

The LRBF-FD stencil weights, \( \{ w_j \}_{j=1}^{N_I} \), will be computed by enforcing the linear constraint (15) and the RBF \( \{ \phi_j(x, \varepsilon) \}_{j=1}^{N_I} \), centered at the stencil point locations [42,62,63], that can be formulated as

\[
\Delta \phi_j(x, \varepsilon) = \sum_{j=1}^{N_I} w_j \phi_j(x_k, \varepsilon), \quad k = 1, \ldots, N_I.
\]  

(16)

Therefore, we have a system of equations with size \( N_I \times N_I \) at each stencil as follows

\[
\Phi w^{(0)} = \Psi^{(0)},
\]  

(17)

where \( \phi_{ij} = \phi_j(x_i, \varepsilon), \quad k, j = 1, \ldots, N_I \), are the elements of the coefficient matrix \( \Phi \), \( \{ w_{j=1}^{(0)} \}_{j=1}^{N_I} \), is differentiation weights for the point \( x \), and \( \Psi^{(0)} \) has the elements \( \Delta \phi_j(x, \varepsilon) \). The weight coefficients \( w^{(0)} \) on the corresponding local domain can be calculated as

\[
w^{(0)} = \Phi^{-1}\Psi^{(0)}.
\]  

(18)

Solving Eq. (18) and substituting the obtained result into Eq. (15) gives the approximate solution of the linear differential operator, that is, gives \( \Delta u \) at node \( x \). The following system of equations can be integrated by substituting (15) into Eq. (6) at an internal point \( x_i = (x_i, y_i) \) and Eq. (3) at all Dirichlet boundary points as

\[
A_{il} = \int_{\Omega} B_{il} + \mu_1 \sum_{k=0}^{n-1} (\Delta x_{i,k} - \Delta x_{k}) |D| + \mu_2 \sum_{k=0}^{n-1} (\Delta x_{i,k} - \Delta x_{k}) |D| + b,
\]  

(19)

where the elements of the matrices \( A, B, C, D \) and \( b \) can be written as follows:

\[
A_{il} = \begin{cases} 
\delta_{ij} - \mu w_{j}^{(0)}, & i = 1, \ldots, N_I, j = 1, \ldots, N, \\
\delta_{ij}, & i = N_I + 1, \ldots, N, j = 1, \ldots, N, 
\end{cases}
\]

\[
B_{il} = \begin{cases} 
\delta_{ij}, & i = 1, \ldots, N_I, j = 1, \ldots, N, \\
0, & i = N_I + 1, \ldots, N, j = 1, \ldots, N, 
\end{cases}
\]

\[
C_{il} = \begin{cases} 
\mu w_{j}^{(0)}, & i = 1, \ldots, N_I, j = 1, \ldots, N, \\
0, & i = N_I + 1, \ldots, N, j = 1, \ldots, N, 
\end{cases}
\]

\[
D_{il} = \begin{cases} 
\mu w_{j}^{(0)}, & i = 1, \ldots, N_I, j = 1, \ldots, N, \\
0, & i = N_I + 1, \ldots, N, j = 1, \ldots, N, 
\end{cases}
\]

\[
b = \begin{cases} 
\tau_{i}^{m+1}, & i = 1, \ldots, N_I, \\
\varepsilon^{m+1}(x_i), & i = N_I + 1, \ldots, N, 
\end{cases}
\]

\[
U^{(n+1)} = (U^{(n)}(x_1), U^{(n)}(x_2), \ldots, U^{(n)}(x_N))^T.
\]

Moreover, we have \( \mu = \mu_1 + \mu_2 \) and \( \delta_{ij} \) for the Kronecker delta function. The implementation of the proposed method, underlying computational aspects, can be summarized as follows:

Step 1 Generate nodes distributions
The uniform or Halton node distributions are generated in the solution domain as the set of evaluation points. The uniformly random Halton nodes are generated by using the MATLAB program `haltonseq.m` [64]. The set \( \Omega = \{ x_1, \ldots, x_N \} \) of collocation points is split into two sets \( I \) and \( B \), which are the indices of the internal and boundary nodes, respectively, and where \( N \) denotes the total number of nodes so that \( N = N_I + N_B \).

Step 2 Construction of the support domain
For each \( x_i \), we first create a local support domain \( \Xi = \{ x_1, \ldots, x_N \} \) that includes the nearest \( N_I - 1 \) evaluation points to \( x_i \). In this approach, each local support domain includes only evaluation nodes. The neighboring points can be selected via several methods. One of these methods consists of defining a finite radius around a node and assuming that the points within that radius are the corresponding neighboring points. One can achieve the \( \mu_1 \)-tree with the help of the `knnsearch` function available in MATLAB [65,64].

Step 2 Determination of the differentiation weights for the point \( x \)
The weight coefficients \( w^{(0)} \) are obtained to approximate the Laplacian operator \( \Delta \) on the corresponding local domain in the center \( x \) by using the relation (18).

Step 3 Create the coefficient matrix
We create the matrices \( A, B, C, D \) and \( b \) by using Step 2.

Step 4 Calculate the approximate solution in the main time stepping loop
We set Eq. (19) in a loop for the number of time steps, \( n = 1, \ldots, M \), and we calculate the approximate solution \( U^{(n)} \) at each time step.

Numerical examples

This section includes three problems and four domains for illustrating the effectiveness and accuracy of the proposed method. In order to assess the performance, we calculate the time convergence order by means of the formula:

\[
C_{\text{it}} = \log_{2} \left( \frac{\| L_{\infty} (2 \delta t, h) \|}{\| L_{\infty} (\delta t, h) \|} \right).
\]
The condition number (C-N) of the coefficient matrix is evaluated by means of the Matlab command `condest`. All computations are obtained in MATLAB R2016a on a Pentium IV, 2800 MHz CPU with 8 Gbyte memory. Fig. 2 considers four distinct types of computational regions \((\Omega, i = 1, 2, 3, 4)\). The domain \(\Omega_1 = [0, 1]^2\) is discretized including the uniform points. The contour of \(\Omega_2\) is determined by the polar form \(r(\theta) = \frac{1}{2}(17 - 8 \cos(3\theta)), 0 \leq \theta \leq 2\pi\). The domains \(\Omega_3\) and \(\Omega_4\) exhibit irregular geometries given by \(r(\theta) = \sqrt{\sin^2(5\theta) + \sin^2(2\theta) + \cos^2(\theta)}\) and \(r(\theta) = \sqrt{\cos(3\theta) + \sqrt{2 - \sin^2(3\theta)}}, 0 \leq \theta \leq 2\pi\), respectively, discretized with irregular distribution of nodes.

Example 1. Consider the following MTFDE:

\[
\begin{align*}
\frac{\partial u(x, y, t)}{\partial t} &= -D_1^{1-\alpha} \Delta u(x, y, t) - D_1^{1-\beta} \Delta u(x, y, t) \\
&= \exp(x + y) \left[ (1 + x + \beta) t^{2x+\beta} - 2 \Gamma(2x+\beta) t^{2x+\beta} \right] \\
&- 2 \Gamma(2/\alpha+2\beta) t^{2x+2\beta}, \quad (x, y) \in \Omega, \quad 0 < t < T.
\end{align*}
\]

The boundary and initial conditions are obtained from the analytic solution \(u(x, y, t) = \exp(x + y) t^{1-\alpha+\beta}\).

The new method is employed for determining the solution with different values of \(\alpha, \beta, \partial t, \partial x, \partial \beta\) and \(N\) on the four domains. The numerical results are summarised in Tables 1–4 and Figs. 3–6. Tables 1 and 2 list the rates of convergence in the temporal direction at \(T = 1\) on the domains \((\Omega_1, \Omega_2, \Omega_3, \Omega_4)\) with

![Fig. 2. The computational regions \((\Omega_1, \Omega_2, \Omega_3, \Omega_4)\) for the test problems.](image-url)

**Table 1**

Numerical orders of convergence in the time variable with \((x, \beta) = \{0.6, 0.9\}\) on domains \(\Omega_1 \) and \(\Omega_2\).

| \(\partial t\) | \(L_\infty\) | \(\Omega_1\) | \(C_m\) | \(L_\infty\) | \(\Omega_2\) | \(C_m\) |
|---|---|---|---|---|---|---|
| 1/10 | 2.194e – 03 | 1.4622 | 6.3279e – 03 | 1.4105 |
| 1/20 | 7.964e – 04 | 1.4921 | 2.3805e – 03 | 1.4829 |
| 1/40 | 2.831e – 04 | 1.5681 | 8.0166e – 03 | 1.5518 |
| 1/80 | 9.549e – 05 | 1.5736 | 2.7344e – 03 | 1.6099 |
| 1/160 | 3.208e – 05 | 1.6146 | 8.9587e – 04 | 1.6698 |
| 1/320 | 1.037e – 05 | 1.6284 | 2.9213e – 04 | 1.6167 |
| 1/640 | 3.388e – 06 | 1.6146 | 9.1815e – 05 | 1.6698 |
| 1/1280 | 1.104e – 06 | 1.6176 | 2.9825e – 05 | 1.6222 |
The numerical orders of convergence in the time variable with \( \{x, \beta\} = \{0.6, 0.9\} \) on domains \( \Omega_1 \) and \( \Omega_4 \) of Example 1.

| \( \Delta t \) | \( L^0 \) | \( \Omega_1 \) | \( C_H \) | \( L^0 \) | \( \Omega_4 \) | \( C_H \) |
|---|---|---|---|---|---|---|
| 1/10 | 3.6239e - 03 | 1.5286 | 4.1765e - 03 | | | |
| 1/20 | 1.2596e - 03 | | 1.5332e - 03 | | | |
| 1/40 | 4.2525e - 04 | 1.5666 | 5.2825e - 04 | | | |
| 1/80 | 1.4188e - 04 | 1.5836 | 1.7552e - 04 | | | |
| 1/160 | 4.6799e - 05 | 1.6001 | 5.7206e - 05 | | | |
| 1/320 | 1.4833e - 05 | 1.6577 | 1.8761e - 05 | | | |
| 1/640 | 4.8304e - 06 | 1.6186 | 6.0341e - 06 | | | |
| 1/1280 | 1.3887e - 06 | 1.6146 | 9.1815e - 05 | | | |

Table 3

| \( \Delta t \) | \( \text{Ref. [31]} \) | \( \text{Ref. [34]} \) | \( \text{Proposed method} \) |
|---|---|---|---|
| \( L^0 \) | \( C_H \) | \( L^0 \) | \( C_H \) | \( L^0 \) | \( C_H \) |
| 1/4 | 1.1481e - 02 | - | 1.1093e - 02 | - | 1.0635e - 02 | - |
| 1/8 | 3.2554e - 03 | 1.8183 | 2.9280e - 03 | 1.9214 | 2.9948e - 03 | 1.8283 |
| 1/16 | 5.0574e - 04 | 2.6864 | 4.2727e - 04 | 2.7769 | 3.8661e - 04 | 2.9641 |
| 1/32 | 2.3163e - 04 | 1.1264 | 1.6510e - 04 | 1.3718 | 1.5721e - 04 | 1.3056 |
| 1/64 | - | - | - | - | 6.5273e - 05 | 1.2681 |
| 1/128 | - | - | - | - | 2.7591e - 05 | 1.2423 |
| 1/256 | - | - | - | - | 1.1874e - 05 | 1.2164 |

Table 4

| \( N_t \) | \( \text{Ref. [31]} \) | \( \text{Ref. [34]} \) | \( \text{Proposed method} \) |
|---|---|---|---|
| \( L^0 \) | \( C_H \) | CPU | \( L^0 \) | \( C_H \) | CPU | \( L^0 \) | \( C_H \) | CPU |
| 43 | 7.1436e - 03 | 3.4725e + 04 | 6.451 | 2.4877e - 04 | 5.1020e - 05 | 6.616 |
| 63 | 3.6604e - 03 | 3.7118e + 04 | 11.172 | 1.0377e - 04 | 5.0928e - 05 | 11.983 |
| 83 | 9.0852e - 04 | 3.8164e + 04 | 15.302 | 9.2497e - 05 | 5.2278e + 05 | 15.443 |
| 103 | 4.0743e - 04 | 3.9388e + 04 | 22.081 | 4.6659e - 05 | 5.2502e + 05 | 22.465 |

\( \varepsilon = 2.1, \vartheta = 0.00001, N_t = 75, \) and \( N = 325. \) From these tables, it can be observed that the rate of convergence in the temporal variable is \( e^{(x^1 \min(x, \beta))} \), which is compatible with the theoretical formulation. Table 3 compares the computational errors of the proposed technique with those obtained by other methods [31,34] for \( \{x, \beta\} = \{0.5, 0.2\} \). These results are in agreement with those reported in [31,34]. Table 4 lists the computational errors, the condition number and computational time (in seconds) for different stencil sizes \( N_t \) with \( \varepsilon = 2.1, \vartheta = 0.00055, \{x, \beta\} = \{0.7, 0.4\}, \Delta t = 1/500. \) It can be concluded that the \( L^0 \) error dimin-

![Fig. 3](image-url)
ishes by increasing the stencil sizes $N_i$. Furthermore, the condition numbers are quite acceptable. Table 5 compares the $L_1$ error for the schemes described in [30] with those obtained by the proposed technique when $T = 1$ on $\Omega_i$. Table 6 demonstrates the $L_1$ error for the proposed technique at different final time $T$ on $\Omega_i$. From this table, it can be seen that the numerical results are acceptable in larger $T$. Fig. 3 represents the numerical solutions and the computational errors by letting $\epsilon = 2.4, \theta = 0.0001, (\alpha, \beta) = (0.5, 0.5), \delta t = 1/100, N_i = 33$ and $N = 289$, when $T = 1$ on $\Omega_i$. From Figs. 4–6, it can be seen that the proposed method performs well when solving problems involving complicated and irregular domains.

Example 2. Consider the following two-dimensional MTFDE:

$$\begin{align*}
\frac{\partial u(x, y, t)}{\partial t} - D_1^{\alpha, \beta} \Delta u(x, y, t) - D_1^{\alpha, \beta} a & = \sin(x + y) \left( 1 + \alpha + \beta \right) t^{2+\beta} + 2 \frac{\Gamma(2+\alpha+\beta)}{\Gamma(1+2+\alpha+\beta)} t^{2+\beta} \\
& + 2 \frac{\Gamma(2+\alpha+\beta)}{\Gamma(1+2+\alpha+\beta)} t^{2+\beta}, (x, y) \in \Omega, 0 < t \leq T.
\end{align*}$$

The boundary and initial conditions are calculated using the analytic solution $u(x, y, t) = \sin(x + y) t^{1+\alpha+\beta}$.

The proposed technique is adopted for computing the solution with various values of $\epsilon, \theta, \delta t, \alpha, \beta$ and $N$ on $\Omega_i, i = 1, 2, 3, 4$. The
Table 5
The $L_\infty$ error for the schemes described in [30] and the proposed method at $T = 1$ on $\Omega_1$ of Example 1.

| $(\alpha, \beta)$ | $N$  | $\Delta t$ | Ref. [30]   | Proposed method |
|-------------------|------|------------|-------------|-----------------|
| (0.1, 0.2)        | 81   | 1/64       | 8.1150e−03  | 1.0893e−03      |
|                   | 289  | 1/1024     | 2.9754e−04  | 7.0079e−05      |
| (0.5, 0.8)        | 81   | 1/64       | 1.2422e−03  | 1.5288e−03      |
|                   | 289  | 1/1024     | 1.1255e−03  | 8.8765e−05      |

Fig. 6. The numerical solutions and the resulting errors at final time $T = 1$ on region $\Omega_1$ of Example 1.

Table 6
The $L_\infty$ error at the various final times on $\Omega_1$ with $N = 441$ and $\Delta t = 1/80$ of Example 1.

| $T$      | $L_\infty$ | C-N               | CPU(s) |
|----------|------------|-------------------|--------|
| 0.25     | 8.3437e−05 | 9.2621e−04        | 0.751  |
| 0.50     | 1.0555e−04 | 9.2621e−04        | 0.979  |
| 0.75     | 2.7365e−04 | 9.2621e−04        | 1.299  |
| 1.00     | 3.4154e−04 | 9.2621e−04        | 1.668  |
| 1.25     | 4.7164e−04 | 9.2621e−04        | 2.087  |
| 1.50     | 5.4350e−04 | 9.2621e−04        | 2.611  |
| 1.75     | 7.4821e−04 | 9.2621e−04        | 3.131  |
| 2.00     | 8.7651e−04 | 9.2621e−04        | 3.763  |

Table 7
Numerical orders of convergence in the time variable with $(\alpha, \beta) = (0.3, 0.8)$ on domains $\Omega_1$ and $\Omega_2$ of Example 2.

| $\Delta t$ | $L_\infty$ | $\Omega_1$ | $C_N$ | $L_\infty$ | $\Omega_2$ | $C_N$ |
|------------|------------|------------|-------|------------|------------|-------|
| 1/10       | 3.6116e−03 | –          | 5.5384e−03 | –          | 1.2600     |
| 1/20       | 1.5493e−03 | 1.2210     | 2.3126e−03 | 1.2764     |
| 1/40       | 6.5849e−04 | 1.2344     | 9.5471e−04 | 1.2989     |
| 1/80       | 2.7379e−04 | 1.2661     | 3.8803e−04 | 1.3105     |
| 1/160      | 1.1047e−04 | 1.3094     | 1.5645e−04 | 1.3081     |
| 1/320      | 4.4231e−05 | 1.3166     | 6.1382e−05 | 1.3242     |
| 1/640      | 1.7520e−05 | 1.3361     | 2.5233e−05 | 1.3103     |
| 1/1280     | 7.0384e−06 | 1.3157     | 1.0175e−05 | 1.3103     |
Table 8
Numerical orders of convergence in the time variable with \( \alpha \) and \( \beta \) on domains \( \Omega_3 \) and \( \Omega_4 \) of Example 2.

| \( \Delta t \) | \( \Omega_3 \) | \( \Omega_4 \) |
|-------|-------|-------|
| 1/10  | 3.5765e – 03 | 4.3275e – 03 |
| 1/20  | 1.3478e – 03 | 1.6282e – 03 |
| 1/40  | 5.0307e – 04 | 6.0520e – 04 |
| 1/80  | 1.8205e – 04 | 2.1945e – 04 |
| 1/160 | 6.5172e – 05 | 7.8313e – 05 |
| 1/320 | 2.2605e – 05 | 2.7658e – 05 |
| 1/640 | 7.7686e – 06 | 9.5447e – 06 |

Table 9
The error \( L_1 \), C-N and CPU time (in seconds) with \( \Delta t = 1/500 \) and \( N = 857 \) of Example 2.

| \( N_i \) | \( \Omega_3 \) | \( \Omega_4 \) |
|-------|-------|-------|
|       | \( L_1 \) | C-N | CPU(s) | \( L_1 \) | C-N | CPU(s) |
| 37    | 7.6294e – 04 | 1.8611e + 04 | 6.352 | 4.2745e – 04 | 5.1576e + 05 | 5.422 |
| 57    | 5.5539e – 04 | 2.3097e + 04 | 12.820 | 2.5233e – 04 | 5.0639e + 05 | 11.219 |
| 77    | 3.0384e – 04 | 3.1927e + 04 | 17.452 | 9.5042e – 05 | 5.1611e + 05 | 16.015 |
| 97    | 8.3628e – 05 | 3.3489e + 04 | 26.523 | 5.3608e – 05 | 5.1793e + 05 | 24.836 |

Table 10
The \( L_1 \) error at the various final times on \( \Omega_i \) with \( N = 676 \) and \( \Delta t = 1/80 \) of Example 2.

| \( T \) | \( \Omega_3 \) | \( \Omega_4 \) |
|-------|-------|-------|
| 0.25  | 7.7525e – 05 | 1.9095e + 05 | 1.435 |
| 0.50  | 1.4909e – 04 | 1.9095e + 05 | 1.835 |
| 0.75  | 2.5460e – 04 | 1.9095e + 05 | 2.263 |
| 1.00  | 5.9131e – 04 | 1.9095e + 05 | 2.920 |
| 1.25  | 6.5221e – 04 | 1.9095e + 05 | 3.630 |
| 1.50  | 7.8056e – 04 | 1.9095e + 05 | 4.449 |
| 1.75  | 9.0255e – 04 | 1.9095e + 05 | 5.337 |
| 2.00  | 1.8387e – 03 | 1.9095e + 05 | 6.270 |

Fig. 7. The numerical solutions and the resulting errors for at final time \( T = 1 \) on region \( \Omega_i \) of Example 2.
numerical results are given in Tables 7–10 and Figs. 7–10. Tables 7 and 8 illustrate the rate of convergence in the temporal direction at $T = 1$ on domains $\{\Omega_1, \Omega_2, \Omega_3, \Omega_4\}$ with $\varepsilon = 2.3, \vartheta = 0.00001$, $N_\xi = 43$ and $N = 275$. We can observe that the rate of convergence in temporal direction is $O(\delta t^{1+\min\{a,b\}})$, which is compatible with the theoretical result. Table 9 reports the computational errors, the C-N and computational time (in seconds) for different stencil sizes $N_{\eta i}$ with $\varepsilon = 2.3, \vartheta = 0.00015, \{a, b\} = \{0.3, 0.6\}$ and $\delta t = 1/500$. We observe that the $L_\infty$ error diminishes by increasing the stencil sizes $N_{\eta i}$, and that the condition numbers are quite acceptable. Table 10 lists the $L_\infty$ error for the proposed technique at various final time $T$ on $\Omega_i$. It can be observed that the method yields accurate results in large $T$ and exhibits adequate stability. Fig. 7 represents the numerical solutions and the computational errors by letting $\varepsilon = 2.0, \vartheta = 0.00005, \{a, b\} = \{0.5, 0.5\}, \delta t = 1/100, N_\xi = 65$ and $N = 255$, when $T = 1$ on $\Omega_1$. Fig. 8 portrays the numerical solutions and the computational errors by taking $\varepsilon = 2.1, \vartheta = 0.000055, \{a, b\} = \{0.25, 0.55\}, \delta t = 1/150, N_\xi = 71$ and $N = 359$, when $T = 1$ on $\Omega_2$. Fig. 9 plots the numerical solutions and the computational errors by letting $\varepsilon = 2.4, \vartheta = 0.000001, \{a, b\} = \{0.45, 0.65\}, \delta t = 1/150, N_\xi = 59$ and $N = 431$, when $T = 1$ on $\Omega_3$. Finally, Fig. 10 illustrates the numerical solutions and the computational errors by taking $\varepsilon = 2.5, \vartheta = 0.000055, \{a, b\} = \{0.85, 0.35\}, \delta t = 1/150, N_\xi = 47$ and $N = 501$, when $T = 1$ on $\Omega_4$. We verify that the accuracy of the proposed method is not affected by the complex shaped domains with the uniform and Halton nodes.
Example 3. Consider the following Gaussian pulse example

\[
\frac{\partial u(x,y,t)}{\partial t} - D_t^{1-\beta} \Delta u(x,y,t) - D_t^{1-\beta} \Delta u(x,t) = f(x,y,t), \quad (x,y) \in \Omega, \quad 0 < t \leq T.
\]

All conditions and source term are deduced from the analytic solution

\[ u(x,y,t) = t^{1+\alpha} \exp \left( \frac{-(x-0.5)^2-(y-0.5)^2}{\gamma} \right). \]

The new method is adopted for determining the numerical solution with various values of \(\alpha, \beta\) and \(\gamma\) at final times \(T\). Table 11 lists the \(L_\infty\) errors and the time convergence orders \(C_{dt}\) of the proposed method for various values of \(\alpha, \beta\) and \(\gamma\) at final time \(T = 1\) on the square domain \([-2,2]^2\) including the uniform nodes. It can be seen that the obtained convergence rates tend to a limit close to \(O(\delta t^{1+\min\{\alpha,\beta\}})\), which is in accordance with the theoretical convergence order. Fig. 11 depicts the approximate solutions and associated errors by choosing various values of \(\gamma \in \{1/5,1/10,1/20\}\) on the square domain \([-2,2]^2\) having the uniform nodes with \(\alpha = 0.55, \beta = 0.85\) \(\delta t = 1/100, N = 441\) and \(N_t = 87\) at \(T = 1\).

**Table 11**

| \(\delta t\) | \(\alpha = 0.65, \beta = 0.45, \gamma = 1/5\) | \(\alpha = 0.35, \beta = 0.95, \gamma = 1/10\) |
| --- | --- | --- |
| \(L_\infty\) | \(C_{dt}\) | \(L_\infty\) | \(C_{dt}\) |
| 1/4 | 6.4325e-02 | - | 8.5265e-02 | - |
| 1/8 | 2.5107e-02 | 1.3573 | 3.5791e-02 | 1.2711 |
| 1/16 | 9.7363e-03 | 1.3966 | 1.4630e-02 | 1.3019 |
| 1/32 | 3.6518e-03 | 1.4148 | 6.0150e-03 | 1.3276 |
| 1/64 | 1.3574e-03 | 1.4278 | 2.3965e-03 | 1.3636 |
| 1/128 | 4.9716e-04 | 1.4491 | 9.3132e-04 | 1.3636 |

Fig. 10. The numerical solutions and the resulting errors at final time \(T = 1\) on region \(\Omega\) of Example 2.
Fig. 11. Approximate solutions and errors on the square domain for $\gamma \in \{1/5, 1/10, 1/20\}$ with $\alpha t = 1/100, N = 225$ on $[-2, 2]^2$ at $T = 1$ for Example 3.
Conclusion

Fractional diffusion models phenomena of anomalous diffusion in transport processes through complex and disordered systems with fractal media. This paper adopted a local hybrid kernel meshless strategy to solve the TFMD. We observe in the published literature that some numerical algorithms have a high-order numerical accuracy on domains with simple shapes, but they are not applicable in the case of complex and irregular domains. On the other hand, some other techniques can be used on complex domains, but they exhibit in sufficient accuracy. Given these limitations, this paper presented a new local hybrid kernel meshless method. This approach not only provides adequate precision, but is also usable over complex domains with various node distributions. First, a finite difference scheme was employed to approximate the time direction. The unconditionally stability and convergence analysis of the semi-discrete approach were examined. Second, the local hybridization of Gaussian and cubic kernels method were used to discrete the spatial direction. Numerical simulations verified the good performance of the meshless algorithm on regular and irregular domains. The numerical results are compatible with the theoretical conclusions.

Declaration of Competing Interest

The authors declare no competing interests.

Compliance with ethics requirement

This paper does not include any studies with human or animal subjects.

Acknowledgements

The authors would like to thank the editors and three anonymous reviewers for their insightful comments and suggestions that greatly improved the quality of this paper.

References

[1] Oldham KB, Spanier J. The Fractional Calculus, vol. 111 of Mathematics in science and engineering; 1974.
[2] Podlubny I. Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and applications, vol. 198. Elsevier; 1998.
[3] Uchaikin VV. Fractional derivatives for physicists and engineers, vol. 2. Springer; 2013.
[4] Milici C, Drăgănescu G, Machado JT. Introduction to fractional differential equations, vol. 25. Springer; 2019.
[5] Pinto C, Carvalho AR. Fractional dynamics of an infection model with time-varying drug exposure. J Comput Nonlinear Dyn 2018;13(9). Article 090904.
[6] Carvalho AR, Pinto CM, Tavares JN. Maintenance of the latent reservoir by pyropothisis and superinfection in a fractional order HIV transmission model. Int J Optim Control: Theories Appl (IJCOTA) 2019;9(3):69–75.
[7] Sweilam N, Al-Mekhlafi S, Albalawi A, Machado JT. Optimal control of variable-order fractional model for delay cancer treatments. Appl Math Model 2021;89:1557–74.
[8] Sweilam N, Al-Mekhlafi S, Baleanu D. Nonstandard finite difference method for solving complex-order fractional Burgers’ equations. J Adv Res 2020;25:19–29.
[9] Tian NH, Baleanu D, Thach TN, O’Regan D, Can NH. Final value problem for nonlinear time fractional reaction–diffusion equation with discrete data. J Comput Appl Math 2020;376:112883.
[10] Dubey VP, Kumar R, Kumar D, Khan I, Singh J. An efficient computational scheme for nonlinear time fractional systems of partial differential equations arising in physical sciences. Adv Differ Equ 2020;2020(1):46.
[11] Luk NH, Huynh VN, Baleanu D, Can NH. Identifying the space source term problem for a generalization of the fractional diffusion equation with hyper-Besel operator. Adv Differ Equ 2020;2020(1):1–23.
[12] Diwedi KD, Das S, Baleanu D. Numerical solution of nonlinear space-time fractional-order advection–reaction–diffusion equation. J Comput Nonlinear Dyn 2020;15(6):061005.
[46] Avazzadeh Z, Nikan O, Machado JAT. Solitary wave solutions of the generalized Rosenau-KdV-RLW equation. Mathematics 2020;8(9):1601.
[47] Nikan O, Avazzadeh Z, Machado JT. Numerical investigation of fractional nonlinear sine-Gordon and Klein-gordon models arising in relativistic quantum mechanics. Eng Anal Bound Elem 2020;120:223–37.
[48] Nikan O, Machado J, Avazzadeh Z, Jafari H. Numerical evaluation of fractional Tricomi-type model arising from physical problems of gas dynamics. J Adv Res 2020;25:205–16.
[49] Nikan O, Machado JT, Golbabai A. Numerical solution of time-fractional fourth-order reaction-diffusion model arising in composite environments. Appl Math Model 2020;85:819–36.
[50] Zhuang P-H, Liu Q-X. Numerical method of Rayleigh-Stokes problem for heated generalized second grade fluid with fractional derivative. Appl Math Mech 2009;30(12):1533.
[51] Liu F, Zhuang P, Burrage K. Numerical methods and analysis for a class of fractional advection-dispersion models. Comput Math Appl 2012;64 (10):2990–3007.
[52] Lin Y, Li X, Xu C. Finite difference/spectral approximations for the fractional cable equation. Math Comput 2011;80(275):1369–96.
[53] Zhang N, Deng W, Wu Y. Finite difference/element method for a two-dimensional modified fractional diffusion equation. Adv Appl Math Mech 2012;4(4):496–518.
[54] Wendland H. Scattered data approximation, Vol. 17. Cambridge University Press; 2004.
[55] Buhmann MD. Radial basis functions: theory and implementations, Vol. 12. Cambridge University Press; 2003.
[56] Mishra PK, Nath SK, Sen MK, Fasshauer GE. Hybrid Gaussian-cubic radial basis functions for scattered data interpolation. Comput Geosci 2018;22 (5):1203–18.
[57] Mishra PK, Nath SK, Kosec G, Sen MK. An improved radial basis-pseudospectral method with hybrid Gaussian-cubic kernels. Eng Anal Bound Elem 2017;80:162–71.
[58] Mishra PK, Fasshauer GE, Sen MK, Ling L. A stabilized radial basis-finite difference (RBF-FD) method with hybrid kernels. Comput Math Appl 2019;77 (9):2354–68.
[59] Oruç O. A local hybrid kernel meshless method for numerical solutions of two-dimensional fractional cable equation in neuronal dynamics. Numer Methods Partial Diff Eqs 2020;36(6):1699–717.
[60] Kansa EJ. Multiquadratics-A scattered data approximation scheme with applications to computational fluid-dynamics-i surface approximations and partial derivative estimates. Comput Math Appl 1990;19(8–9):127–45.
[61] Kansa EJ. Multiquadratics-a scattered data approximation scheme with applications to computational fluid-dynamics-ii solutions to parabolic, hyperbolic and elliptic partial differential equations. Comput Math Appl 1990;19(8–9):147–61.
[62] Shu C, Ding H, Yeo K. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations. Comput Methods Appl Mech Eng 2003;192(7–8):941–54.
[63] Wright GB, Fornberg B. Scattered node compact finite difference-type formulas generated from radial basis functions. J Comput Phys 2006;212(1):99–123.
[64] Fasshauer GE. Meshfree Approximation Methods with Matlab; (With CD-ROM), Vol. 6, World Scientific Publishing Co Inc; 2007.
[65] Shechter G. Matlab package kd-tree; 2004.

Zakieh Avazzadeh is an Associate Professor at Xi’an Jiaotong-Liverpool University Suzhou, Jiangsu, China. She received her Ph.D. degree in Applied Mathematics from Yazd University in 2011. From 2012 to 2014, she was a postdoctoral fellowship at Hohai University. Her research areas are Numerical Approximation, Meshless Methods and Orthogonal Basis Functions. She is also interested in Fractional Calculus and Fractional Dynamical Systems.