



ARTICLE

A Novel Accurate Method for Multi-Term Time-Fractional Nonlinear Diffusion Equations in Arbitrary Domains

Tao Hu¹, Cheng Huang², Sergiy Reutskiy^{3,*}, Jun Lu⁴ and Ji Lin^{5,*}

¹Ningbo Yongxin Engineering Consulting Co., Ltd., Ningbo, 315042, China

²Taizhou Huanfeng Water Conservancy Engineering Co., Ltd., Taizhou, 318050, China

³A. Pidhorny Institute of Mechanical Engineering Problems, National Academy of Sciences of Ukraine, Kharkiv, 61046, Ukraine

⁴Material and Structure Department, Nanjing Hydraulic Research Institute, Nanjing, 210029, China

⁵College of Mechanics and Materials, Hohai University, Nanjing, 211100, China

*Corresponding Authors: Sergiy Reutskiy. Email: sergiyreutskiy@gmail.com; Ji Lin. Email: linji@hhu.edu.cn

Received: 06 April 2023 Accepted: 20 June 2023 Published: 17 November 2023

ABSTRACT

A novel accurate method is proposed to solve a broad variety of linear and nonlinear (1+1)-dimensional and (2+1)-dimensional multi-term time-fractional partial differential equations with spatial operators of anisotropic diffusivity. For (1+1)-dimensional problems, analytical solutions that satisfy the boundary requirements are derived. Such solutions are numerically calculated using the trigonometric basis approximation for (2+1)-dimensional problems. With the aid of these analytical or numerical approximations, the original problems can be converted into the fractional ordinary differential equations, and solutions to the fractional ordinary differential equations are approximated by modified radial basis functions with time-dependent coefficients. An efficient backward substitution strategy that was previously provided for a single fractional ordinary differential equation is then used to solve the corresponding systems. The straightforward quasilinearization technique is applied to handle nonlinear issues. Numerical experiments demonstrate the suggested algorithm's superior accuracy and efficiency.

KEYWORDS

Müntz polynomial basis; backward substitution method; collocation method; meshless method; fractional equation

1 Introduction

In this study, we consider an efficient and accurate numerical technique for the problem of the following form:

$$\mathcal{L}_t[v] = \mathcal{M}_t[L(\mathbf{x})[v]] + N(v, v_x, v_y) + h(\mathbf{x}, t), \quad t \geq 0, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^2, \quad (1)$$



where

$$L(\mathbf{x})[v] = \text{div}(\widehat{\mathbf{D}}(\mathbf{x}) \nabla v) = d_{1,1}(\mathbf{x}) \frac{\partial^2 v}{\partial x_1^2} + 2d_{1,2}(\mathbf{x}) \frac{\partial^2 v}{\partial x_1 \partial x_2} + d_{2,2}(\mathbf{x}) \frac{\partial^2 v}{\partial x_2^2} + \left(\frac{\partial d_{1,1}(\mathbf{x})}{\partial x_1} + \frac{\partial d_{1,2}(\mathbf{x})}{\partial x_2} \right) \frac{\partial v}{\partial x_1} + \left(\frac{\partial d_{1,2}(\mathbf{x})}{\partial x_1} + \frac{\partial d_{2,2}(\mathbf{x})}{\partial x_2} \right) \frac{\partial v}{\partial x_2}, \quad (2)$$

is the operator of the anisotropic diffusivity. $d_{1,1}(\mathbf{x})$, $d_{1,2}(\mathbf{x})$, and $d_{2,2}(\mathbf{x})$ are the elements of the anisotropic matrix $\widehat{\mathbf{D}}$. \mathcal{L}_t and \mathcal{M}_t are given as follows:

$$\mathcal{L}_t = D_t^{(\mu)} + \sum_{k=1}^I a_k(t) D_t^{(\mu_k)}, \quad \mathcal{M}_t = \sum_{k=I+1}^K a_k(t) D_t^{(\mu_k)}, \quad (3)$$

which are time differential operators of integer or fractional orders. And $N(u, u_x, u_y)$ is a nonlinear function in the unknown solution and its derivatives. The values $0 \leq \mu_k < \mu \leq 1$ are fractional constant values and $a_k(t)$, $k = 1, \dots, K$ are real smooth time-varying functions available in advance. The operator $D_t^{(\nu)}$ is the Caputo-type fractional derivative

$$D_t^{(\nu)} [f(x, t)] = \begin{cases} \frac{1}{\Gamma(n-\nu)} \int_0^t \frac{\partial_t^{(n)} f(x, \tau) d\tau}{(t-\tau)^{\nu-n+1}}, & n-1 < \nu < n, \\ \partial_t^{(n)} f(x, t), & \nu = n. \end{cases} \quad (4)$$

Eq. (1) can be used to describe a wide variety of mathematical models in engineering and science as particular cases. First of all, these are the models of the heat equations obeying the classical Fourier law. At the same time, there are many models of heat exchange based on the non-Fourier laws which also can be represented by this equation. Besides, it can also be used for simulating the unsteady flow of the non-Newtonian fractional Maxwell fluid [1] and the Oldroyd-B fluid [2]. The general equation given in Eq. (1) also includes the Benjamin-Bona-Mahony and the Benjamin-Bona-Mahony-Burgers equations [3,4], which have been used for modeling long waves of small amplitudes. Besides, the Sobolev equation also falls into this group [5,6].

So, the multi-term systems of equations are introduced for modeling many real-life applications which involved some complicated processes that cannot be accurately described by systems of the single-term. It is evident that only a small number of problems can be analyzed analytically under some idealized conditions which are useful for parametric research. However, the derivation of analytical solutions for the nonlinear fractional equations with general spatial differential operators is not a trivial task. Applying numerical methods to deal with these equations would be more desirable. The finite difference method (FDM) [7–9] and the finite element method (FEM) [10–12] have been proposed widely as available techniques. Jin et al. [13] investigated the multi-term time-fractional diffusion equations using the Galerkin FEM. Recently an alternating direction implicit (ADI) scheme has been presented by Huang et al. in [14]. The weighted meshless spectral method was proposed by Hussain and Haq for problems arising in mass and heat transfer [15]. A method based on the Haar wavelets and the finite difference scheme was proposed by Oruç, Esen, and Bulut for two-dimensional time fractional reaction-sub-diffusion equation in [16]. An accurate computational method based on the linear barycentric interpolation method has been proposed by Oruç [17] recently for solving fractional Rayleigh-Stokes problems. The numerical method which combines the Laplace transform with the local radial basis functions method was presented by Li et al. in [18] for fractional diffusion models. A Galerkin method based on the second kind Chebyshev wavelets was presented by Soltani Sarvestani et al. in [19] for solving the multi-term time fractional diffusion-wave equation. Very recently an alternating direction implicit Legendre spectral method has been developed by Liu et al. in

[20]. A mixed meshless method was proposed by Nikan et al. in [21]. The weak Galerkin finite element method for the space discretization combined with the discretization of the Caputo time derivative by L1 method was proposed by Zhou et al. in [22] for time-fractional quasi-linear diffusion equation. A method that combines the L1 interpolation of the fractional derivative with a high-order compact finite volume scheme was applied by Su et al. in [23] for the 2D multi-term time fractional sub-diffusion equation. The technique which combines approximation of the time-fractional derivative via the L1 formula and approximation of the integer order space derivatives by truncated one and two-dimensional wavelet series was presented by Ghafoor et al. for one and two dimensional higher order multi-term time-fractional partial differential equations in [24].

Inspired by the advantages of meshless methods which do not need the requirement of domain mesh, a variety of meshless methods have been proposed in the literature for solving fractional equations in both theoretical and numerical aspects. There exist several types of meshless methods, from which the radial basis function (RBF)-based meshless methods are the most popular type. With the merits of Euclidian distances as variables, RBF-based methods are flexible and usual for high-dimension problems under irregular complicated domains. Piret et al. [25] made the first attempt to use the RBF discretization for the fractional diffusion. Hosseini et al. [26] solved the time-fractional telegraph equations using the RBF discretization with the finite difference method for temporal discretization. Liu et al. [27] provided the implicit RBF approach with error analysis. The radial basis function-finite difference method (RBF-FD) [28] was also used to solve the fractional equations. In this paper, an accurate and efficient method based on the key idea of the backward substitution method has been proposed [29–34]. In this method, the general approximation based on the pure radial basis functions and their correcting terms are formed. Applying the collocation method, we can reduce the considered problems into the system of fractional ordinary differential equations. Then the corresponding problems are solved efficiently with the help of the Müntz polynomial basis since the fractional derivative of a Müntz polynomial is again a Müntz polynomial.

The remainder is organized as follows. These preliminaries including the definition of fractional derivative, the backward substitution method for the linear system of fractional ordinary differential equations, and the quasilinearization techniques are described in Section 2. The main method of application to (1+1)-dimensional and (2+1)-dimensional problems are described in Section 3 where the description of the method in application to the problems with the spatial operator of the fourth order has also been discussed. The numerical examples which illustrate the method presented are placed with the comparisons by some well-known numerical methods in Section 4. Finally, in Section 5, a short conclusion and discussion is provided.

2 Preliminaries

2.1 Solutions for Linear System of Fractional Ordinary Differential Equations

The method presented in this study is based on the effective algorithm for the solution of linear systems of the fractional ordinary differential equations (FODEs). This algorithm is described detailed in this subsection. Let us consider the linear system of fractional ordinary differential equation

$$\widehat{\mathbf{A}} D_t^{(\mu)} [\mathbf{P}(t)] = \sum_{k=1}^K \widehat{\mathbf{A}}_k(t) D_t^{(\mu_k)} [\mathbf{P}(t)] + \widehat{\mathbf{B}}(t) \mathbf{P}(t) + \mathbf{F}(t), \quad 0 \leq t \leq T, \quad (5)$$

with zero initial conditions

$$\mathbf{P}(0) = \mathbf{0}, \quad (6)$$

where $\mathbf{P}(t) = [P_1(t), P_2(t), \dots, P_N(t)]^T$, $\widehat{\mathbf{A}}$ is a constant non-singular matrix of $N \times N$ dimension, $\widehat{\mathbf{A}}_k(t)$ and $\widehat{\mathbf{B}}(t)$ are time dependent matrices of the same size, $\mathbf{F}(t) = [f_1(t), f_2(t), \dots, f_N(t)]^T$. In order to solve the fractional equation easily, we introduce the Müntz polynomial basis (MPB)

$$\varphi_m(t) = t^{\delta_m}, \delta_m = \sigma(m-1), 0 < \sigma \leq 1, m = 1, 2, 3, \dots, \quad (7)$$

as the basis function where the right hand side of Eq. (5) can be approximated by the linear combination of these functions

$$\sum_{k=1}^K \widehat{\mathbf{A}}_k(t) D_t^{(\mu_k)} [\mathbf{P}(t)] + \widehat{\mathbf{B}}(t) \mathbf{P}(t) + \mathbf{F}(t) = \widehat{\mathbf{A}} \sum_{m=1}^{\infty} \mathbf{q}_m \varphi_m(t), \quad 0 \leq t \leq T, \quad (8)$$

where \mathbf{q}_m are unknown coefficients to be determined which will be discussed in the following section.

Some properties of Müntz polynomials should be discussed and noted here. Let $\{\delta_0, \delta_1, \delta_2, \dots\}$ be a sequence of distinct real numbers such that $0 \leq \delta_0 < \delta_1 < \delta_2 < \dots \rightarrow \infty$. The Müntz polynomials of the form $\sum_{k=0}^n c_k t^{\delta_k}$, with real coefficients c_0, c_1, \dots, c_n are dense in $L^2[0, 1]$ if and only if

$$\sum_{k=1}^{\infty} \frac{1}{\delta_k} = +\infty. \quad (9)$$

Also if $\delta_0 = 0$, then the Müntz polynomials are dense in $C[0, 1]$, with the uniform norm, if and only if Eq. (9) is established [35,36]. With the help of Eqs. (8) and (5), we have

$$\widehat{\mathbf{A}} D_t^{(\mu)} [\mathbf{P}(t)] = \widehat{\mathbf{A}} \sum_{m=1}^{\infty} \mathbf{q}_m \varphi_m(t). \quad (10)$$

Followed from Eq. (10), we have the following equation:

$$D_t^{(\mu)} [\mathbf{P}(t)] = \sum_{m=1}^{\infty} \mathbf{q}_m \varphi_m(t), \quad (11)$$

if the matrix $\widehat{\mathbf{A}}$ in Eq. (10) is a non-singular matrix. As it is easily to prove that the following analytical expression:

$$\Phi_m(t) = \frac{\Gamma(\delta_m + 1)}{\Gamma(\delta_m + \mu + 1)} t^{\delta_m + \mu}, \quad (12)$$

satisfies

$$\begin{aligned} D_t^{(\mu)} [\Phi_m(t)] &= \varphi_m(t), \\ \Phi_m(0) &= 0. \end{aligned} \quad (13)$$

Taking into account of Eq. (6), the following series:

$$\mathbf{P}(t) = \sum_{m=1}^{\infty} \mathbf{q}_m \Phi_m(t), \quad (14)$$

is the analytical solution to Eq. (11) and the unknown vectors \mathbf{q}_m can be solved by

$$\sum_{k=1}^K \widehat{\mathbf{A}}_k(t) D_t^{(\mu_k)} \left[\sum_{m=1}^{\infty} \mathbf{q}_m \Phi_m(t) \right] + \widehat{\mathbf{B}}(t) \sum_{m=1}^{\infty} \mathbf{q}_m \Phi_m(t) + \mathbf{F}(t) = \widehat{\mathbf{A}} \sum_{m=1}^{\infty} \mathbf{q}_m \varphi_m(t), \quad (15)$$

or

$$\sum_{m=1}^{\infty} \left[\widehat{\mathbf{A}} \varphi_m(t) - \widehat{\mathbf{B}}(t) \Phi_m(t) - \sum_{k=1}^K \widehat{\mathbf{A}}_k(t) \Phi_m^{(\mu_k)}(t) \right] \mathbf{q}_m = \mathbf{F}(t), \quad t \in [0, T], \quad (16)$$

here

$$\Phi_m^{(\mu_k)}(t) = D_t^{(\mu_k)} [\Phi_m(t)] = D_t^{(\mu_k)} \left[\frac{\Gamma(\delta_m + 1)}{\Gamma(\delta_m + \mu + 1)} t^{\delta_m + \mu} \right] = \frac{\Gamma(\delta_m + 1) t^{\delta_m + \mu - \mu_k}}{\Gamma(\delta_m + \mu + 1 - \mu_k)}. \quad (17)$$

It is worth noting that if Eq. (16) is fulfilled for any t in the time sequence, the infinite sequence given in Eq. (14) is the desired solution. For the sake of calculation, we consider the truncated series

$$\mathbf{P}_M(t) = \sum_{m=1}^M \mathbf{q}_m \Phi_m(t), \quad (18)$$

which satisfies the corresponding equation of truncated series as follows:

$$D_t^{(\alpha)} [\mathbf{P}_M(t)] = \sum_{m=1}^M \mathbf{q}_m \varphi_m(t), \quad (19)$$

where the unknown parameters can be obtained by enforcing the Eq. (16) at several selected time steps t_j . To ensure the solvability of the corresponding system, the number of equations (i.e., the number of time steps) should be taken larger than the number of unknown variables.

2.2 Quasilinearization Procedure for Nonlinear Problem

The nonlinear term $N(v, v_x, v_y)$ in the Eq. (1) can be transformed into linear terms using the quasilinearization technique [37] which is briefly illustrated as follows.

Let us denote $\xi_i = \frac{\partial v}{\partial x_i}$ and assume that $v_0(\mathbf{x})$ and $\xi_{i,0}(\mathbf{x})$ are known functions which are used as initial approximations of the exact values. So, we can write

$$v = v_0 + (v - v_0) = v_0 + \Delta v, \quad \xi_i = \xi_{i,0} + (\xi_i - \xi_{i,0}) = \xi_{i,0} + \Delta \xi_i. \quad (20)$$

Assuming that Δv and $\Delta \xi_i$ are small enough, then we can neglect the squares of their values, in this way

$$\begin{aligned} N(v, v_x, v_y, \mathbf{x}, t) &\approx N(v_0, \xi_{1,0}, \xi_{2,0}, \mathbf{x}, t) + \frac{\partial N(v_0, \xi_{1,0}, \xi_{2,0}, \mathbf{x}, t)}{\partial v} (v - v_0) \\ &\quad + \sum_{i=1}^2 \frac{\partial N(v_0, \xi_{1,0}, \xi_{2,0}, \mathbf{x}, t)}{\partial \xi_i} (\xi_i - \xi_{i,0}). \end{aligned} \quad (21)$$

As a result the original equation is transformed to a sequence of linear equations with the initial approximations of the form

$$\mathcal{L}_t[v] = \mathcal{M}_t[L(\mathbf{x})[v]] + b_1(\mathbf{x}, t) v_x + b_2(\mathbf{x}, t) v_y + c(\mathbf{x}, t) v + f(\mathbf{x}, t), \quad (22)$$

and the system of iteration will stop under the control of the error given by two successive evaluations $\max_{\mathbf{x}, t} |v(\mathbf{x}, t) - v_0(\mathbf{x}, t)|$ which is smaller than the selected tolerance. In practical calculations, we can simply fix the number of iterations and from the numerical examples, the iteration will converge in only several steps.

3 The Solution Scheme for the Considered Problem

In this section, the solution scheme for the Eq. (22) with the known boundary conditions (BC) and initial conditions (IC) will be discussed in details, as follows:

$$\mathcal{B}[v(\mathbf{x}, t)] = g(\mathbf{x}, t), \mathbf{x} \in \partial\Omega, \quad (23)$$

$$v(\mathbf{x}, 0) = v_0(\mathbf{x}), \quad (24)$$

in which \mathcal{B} is the boundary condition operator. In this case, we consider the first kind and the third kind boundary conditions where the second kind boundary condition can be easily obtained, as follows:

$$u = g_1(\mathbf{x}, t), \mathbf{x} \in \Gamma_1, \quad (25)$$

$$r(\mathbf{x})u + \frac{\partial u}{\partial \mathbf{n}} = r(\mathbf{x})u + n_1 \frac{\partial u}{\partial x_1} + n_2 \frac{\partial u}{\partial x_2} = g_2(\mathbf{x}, t), \mathbf{x} \in \Gamma_2, \quad (26)$$

in which $\mathbf{n} = (n_1, n_2)$ represents the unit outward normal vector to the physical boundary and $r(\mathbf{x})$, $g_1(\mathbf{x}, t)$, and $g_2(\mathbf{x}, t)$ given in Eqs. (25) and (26) are known in advances with sufficient smoothness. Note that the boundary condition of the second kind (Neumann BC) is given if $r(\mathbf{x}) \equiv 0$ in Eq. (26).

3.1 Algorithm for (1+1)-Dimensional Problem

In this case the governing equation takes the form

$$\mathcal{L}_t[v] = \mathcal{M}_t[L(x)[v]] + b(x, t)v_x + c(x, t)v + f(x, t), \quad (27)$$

or in the explicit form

$$\begin{aligned} \mathcal{L}_t[v] &= D_t^{(\mu)}[v] + \sum_{k=1}^I a_k(t) D_t^{(\mu_k)}[v] = \sum_{k=t+1}^K a_k(t) \left[\partial_x \left(d(x) \partial_x D_t^{(\mu_k)}[v] \right) \right] \\ &+ b(x, t) \partial_x v + c(x, t)v + f(x, t) = \mathcal{M}_t[L(x)[v]] + b(x, t) \partial_x v + c(x, t)v + f(x, t), \end{aligned} \quad (28)$$

with the IC

$$v(x, 0) = v_0(x), \quad (29)$$

and the BCs

$$L_W[v(x, t)] \equiv \alpha_W \partial_x v(0, t) + \beta_W v(0, t) = g_W(t), \quad (30)$$

$$L_E[v(x, t)] \equiv \alpha_E \partial_x v(1, t) + \beta_E v(1, t) = g_E(t), \quad (31)$$

at the endpoints of the interval $\Omega = [0, 1]$. Let us define the function

$$u(x, t) = v(x, t) - v_0(x), \quad (32)$$

which satisfies the equation

$$\mathcal{L}_t[u] = \mathcal{M}_t[L(x)[u]] + b(x, t) \partial_x u + c(x, t)u + f_1(x, t), \quad (33)$$

the BCs

$$L_W[u(x, t)] = g_W(t) - L_W[v_0(x)] = h_W(t), \quad (34)$$

$$L_E[u(x, t)] = g_E(t) - L_E[v_0(x)] = h_E(t), \quad (35)$$

and zero IC

$$u(x, 0) = 0. \quad (36)$$

Here

$$f_1(x, t) = f(x, t) + b(x, t) \partial_x v_0(x) + c(x, t) v_0(x). \quad (37)$$

Let us define the following two functions where the parameters are determined from the initial condition and boundary condition:

$$\theta_E(x) = \frac{\alpha_W - \beta_W x}{\alpha_W \beta_E - \beta_W (\alpha_E + \beta_E)}, \quad (38)$$

$$\theta_W(x) = \frac{\beta_E x - (\alpha_E + \beta_E)}{\alpha_W \beta_E - \beta_W (\alpha_E + \beta_E)}, \quad (39)$$

satisfying the conditions

$$L_W[\theta_W(x)] = 1, \quad L_E[\theta_W(x)] = 0, \quad (40)$$

$$L_W[\theta_E(x)] = 0, \quad L_E[\theta_E(x)] = 1. \quad (41)$$

Then, it is easy to prove that the following combination:

$$u_g(x, t) = \theta_E(x) h_E(t) + \theta_W(x) h_W(t), \quad (42)$$

satisfies the boundary conditions

$$L_W[u_g(x, t)] = h_W(t), \quad L_E[u_g(x, t)] = h_E(t). \quad (43)$$

Suppose that the $u_g(x, t)$ provided in Eq. (42) is the initial approximation to the solution, then by using the decomposition method, we have the equation for the correction item $w(x, t)$, as follows:

$$u(x, t) = u_g(x, t) + w(x, t), \quad (44)$$

where $w(x, t)$ should satisfy the following governing equation:

$$\mathcal{L}_i[w] = \mathcal{M}_i[L(x)[w]] + b(x, t) \partial_x w + c(x, t) w + f_2(x, t), \quad (45)$$

where

$$f_2(x, t) = f_1(x, t) - \mathcal{L}_i[u_g] + \mathcal{M}_i[L(x)[u_g]] + b(x, t) \partial_x u_g + c(x, t) u_g, \quad (46)$$

and the following initial condition and boundary condition:

$$w(x, 0) = 0, \quad (47)$$

$$L_W[w(x, t)] = 0, \quad L_E[w(x, t)] = 0. \quad (48)$$

In order to approximate $w(x, t)$, we consider the radial basis function $\psi_i(x)$ which has been widely used in meshless methods

$$\psi_i(x) = \sqrt{(x - \zeta_i)^2 + c^2}, \quad (49)$$

where c is the artificial selection shape parameter and ζ_i are the centers of the considered function distributed in the solution domain. We define the modified basis functions as follows:

$$\phi_i(x) = \psi_i(x) + c_{i,0} + c_{i,1}x, \quad (50)$$

where coefficients $c_{i,0}$ and $c_{i,1}$ are chosen in the way that the following equations are fulfilled:

$$L_W[\phi_i(x)] = L_E[\phi_i(x)] = 0. \quad (51)$$

As a result we get the linear system

$$\alpha_W c_{i,1} + \beta_W c_{i,0} = -\alpha_W \partial_x \psi_i(x)_{x=0} - \beta_W \psi_i(0), \quad (52)$$

$$\alpha_E c_{i,1} + \beta_E (c_{i,0} + c_{i,1}) = -\alpha_E \partial_x \psi_i(x)_{x=1} - \beta_E \psi_i(1), \quad (53)$$

for each pair of the coefficients $c_{i,0}$ and $c_{i,1}$. The system can be solved easily in the analytical form. By considering the properties of the modified functions $\phi_i(x)$ which satisfying the homogeneous boundary conditions, the linear combination of $\phi_i(x)$ is used as the approximation to $w_N(x, t)$, as follows:

$$w_N(x, t) = \sum_{i=1}^N \phi_i(x) P_i(t), \quad (54)$$

in which the unknown $P_i(t)$ have to be determined. Substituting Eq. (54) into Eq. (45), we can yield the following system

$$\begin{aligned} \sum_{i=1}^N \phi_i(x) \mathcal{L}_i[P_i(t)] &= \sum_{i=1}^N L(x) [\phi_i(x)] \mathcal{M}_i[P_i(t)] \\ &+ \sum_{i=1}^N [b(x, t) \partial_x \phi_i(x) + c(x, t) \phi_i(x)] P_i(t) + f_2(x, t). \end{aligned} \quad (55)$$

Let $x_j \in [0, 1]$, $j = 1, \dots, N$ be selected nodes in the solution domain, the following system of FODEs will be obtained after considering collocation procedure:

$$\begin{aligned} \sum_{i=1}^N \phi_i(x_j) \mathcal{L}_i[P_i(t)] &= \sum_{i=1}^N L(x) [\phi_i(x_j)] \mathcal{M}_i[P_i(t)] \\ &+ \sum_{i=1}^N [b(x_j, t) \partial_x \phi_i(x_j) + c(x_j, t) \phi_i(x_j)] P_i(t) + f_2(x_j, t), \end{aligned} \quad (56)$$

or simplified in the matrix form as shown below:

$$\widehat{\mathbf{A}} D_t^{(\mu)} [\mathbf{P}(t)] + \sum_{k=1}^I a_k(t) \widehat{\mathbf{A}} D_t^{(\mu_k)} [\mathbf{P}(t)] = \sum_{k=I+1}^K a_k(t) \mathbf{D}_k D_t^{(\mu_k)} [\mathbf{P}(t)] + \widehat{\mathbf{B}}(t) \mathbf{P}(t) + \mathbf{F}(t), \quad 0 \leq t \leq T. \quad (57)$$

Here the derivatives can be obtained in the analytical form

$$\partial_x \phi_i(x) = \frac{x - \zeta_i}{\psi_i(x)} + c_{i,1}, \quad \partial_{xx} \phi_i(x) = \frac{1}{\psi_i(x)} - \frac{(x - \zeta_i)^2}{(\psi_i(x))^3}. \quad (58)$$

$\widehat{\mathbf{A}}$, $\widehat{\mathbf{D}}_k$, and $\widehat{\mathbf{B}}(t)$ are $N \times N$ matrices with the components

$$\widehat{\mathbf{A}} = (a_{j,i})_{j,i=1}^N = (\phi_i(x_j))_{j,i=1}^N, \quad (59)$$

$$\widehat{\mathbf{B}}(t) = (b(x_j, t) \partial_x \phi_i(x_j) + c(x_j, t) \phi_i(x_j))_{j,i=1}^N, \quad (60)$$

$$\widehat{\mathbf{D}}_k = \left(\frac{\partial}{\partial x} (d(x_j) \partial_x \phi_i(x_j)) \right)_{j=i=1}^N \tag{61}$$

and $\mathbf{P}(t) = [P_1(t), P_2(t), \dots, P_N(t)]^T$, $\mathbf{F}(t) = [f_2(x_1, t), f_2(x_2, t), \dots, f_2(x_N, t)]^T$ are N -vectors. If we denote

$$a_k(t) \widehat{\mathbf{A}} = -\widehat{\mathbf{A}}_k(t), k = 1, \dots, I; a_k(t) \widehat{\mathbf{D}}_k = \widehat{\mathbf{A}}_k(t), k = I + 1, \dots, K, \tag{62}$$

then the system Eq. (57) coincides with the linear system of FODEs shown in Eq. (5) and can be solved using the algorithm described there. Using M Müntz polynomials in solving of the system of the FODEs shown in Eq. (5), we get the approximate solution

$$w_{N,M}(x, t) = \sum_{i=1}^N \phi_i(x) P_{M,i}(t), \tag{63}$$

with the vector $\mathbf{P}_M(t)$ given in Eq. (18). Then, the approximate solution of the original problem Eqs. (27), (29)–(31) can be given in the following way:

$$v_{N,M}(x, t) = u_{N,M}(x, t) + v_0(x) = w_{N,M}(x, t) + u_g(x, t) + v_0(x) = \sum_{i=1}^N \phi_i(x) P_{M,i}(t) + u_g(x, t) + v_0(x). \tag{64}$$

The same algorithm can be applied for solving problems of Eq. (27) with the spatial operator of the fourth order

$$L(x)[v] = -\partial_{xx}(r(x) \partial_{xx}v) + \partial_x(q(x) \partial_x v) - p(x)v. \tag{65}$$

In this case, the modified RBF takes the form

$$\phi_i(x) = \psi_i(x) + c_{i,0} + c_{i,1}x + c_{i,2}x^2 + c_{i,3}x^3, \tag{66}$$

where the coefficients $c_{i,0}, c_{i,1}, c_{i,2}, c_{i,3}$ are chosen to satisfy the homogeneous BCs. For example, let us consider the BCs:

$$\phi_i(0) = \phi_i(1) = \partial_{xx}\phi_i(0) = \partial_{xx}\phi_i(1) = 0, \tag{67}$$

which leads to the four conditions

- 1) $c_{i,0} = -\psi_i(0),$
- 2) $c_{i,0} + c_{i,1} + c_{i,2} + c_{i,3} = -\psi_i(1),$
- 3) $2c_{i,2} = -\partial_{xx}\phi_i(0),$
- 4) $2c_{i,2} + 6c_{i,3} = -\partial_{xx}\phi_i(1).$

$$\tag{68}$$

So, from 3) $\Rightarrow c_{i,2} = -1/2\partial_{xx}\phi_i(0)$, then from 4) $\Rightarrow c_{i,3} = (\partial_{xx}\phi_i(0) - \partial_{xx}\phi_i(1))/6$ and finally, from 2) $c_{i,1} = -\psi_i(1) - c_{i,0} - c_{i,2} - c_{i,3}$. The derivatives of the modified RBF can be calculated in explicit analytical form:

$$\begin{aligned} \partial_x \phi_i(x) &= \frac{x - \zeta_i}{\psi_i(x)} + c_{i,1} + 2c_{i,2}x + 3c_{i,3}x^2, \partial_{xx} \phi_i(x) = \frac{1}{\psi_i(x)} - \frac{(x - \zeta_i)^2}{\psi_i^3(x)} + 2c_{i,2} + 6c_{i,3}x, \\ \partial_{xxx} \phi_i(x) &= -\frac{3(x - \zeta_i)}{\psi_i^3(x)} + \frac{3(x - \zeta_i)^3}{\psi_i^5(x)} + 6c_{i,3}, \partial_{xxxx} \phi_i(x) = -\frac{3}{\psi_i^3(x)} + \frac{18(x - \zeta_i)^2}{\psi_i^3(x)} - \frac{15(x - \zeta_i)^4}{\psi_i^7(x)}. \end{aligned} \tag{69}$$

3.2 Algorithm for (2+1)-Dimensional Problem

Following the decomposition for the (1+1)-dimensional problem, we have

$$u(\mathbf{x}, t) = v(\mathbf{x}, t) - v(\mathbf{x}, 0) \equiv v(\mathbf{x}, t) - v_0(\mathbf{x}), \quad (70)$$

which can transform problems of Eq. (22) into the form

$$\mathcal{L}_t[u] = \mathcal{M}_t[L(\mathbf{x})[u]] + b_1(\mathbf{x}, t) \partial_x u(\mathbf{x}, t) + b_2(\mathbf{x}, t) \partial_y u(\mathbf{x}, t) + c(\mathbf{x}, t) u(\mathbf{x}, t) + f_1(\mathbf{x}, t), \quad (71)$$

where

$$f_1(\mathbf{x}, t) = f(\mathbf{x}, t) + b_1(\mathbf{x}, t) \partial_x v_0(\mathbf{x}) + b_2(\mathbf{x}, t) \partial_y v_0(\mathbf{x}) + c(\mathbf{x}, t) v_0(\mathbf{x}). \quad (72)$$

The solution $u(\mathbf{x}, t)$ satisfies homogeneous initial condition:

$$u(\mathbf{x}, 0) = 0, \quad (73)$$

and the updated boundary conditions, as follows:

$$\mathcal{B}[u(\mathbf{x}, t)] = g(\mathbf{x}, t) - \mathcal{B}[v(\mathbf{x}, 0)], \quad \mathbf{x} \in \partial\Omega, \quad (74)$$

where the operator in the last equation is described by the formulae of Eqs. (25) and (26). To transform the time-dependent fractional equations into a system of fractional ordinary equations and apply the algorithm described in the previous subsection for (1+1)-dimensional problems, we should perform the following steps for this goal:

1) We transform Eq. (71) to the homogeneous BCs by introducing the primary approximation u_g :

$$u(\mathbf{x}, t) = u_g(\mathbf{x}, t) + w(\mathbf{x}, t), \quad (75)$$

where $u_g(\mathbf{x}, t)$ is a man-made function which has to satisfy

$$\mathcal{B}[u_g(\mathbf{x}, t)] = g(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega. \quad (76)$$

As discussed in the (1+1)-dimensional problems, by considering the Eq. (75) and the governing equation, we have the equation to determine the correcting solution w , as follows:

$$\mathcal{L}_t[w] = \mathcal{M}_t[L(\mathbf{x})[w]] + b_1(\mathbf{x}, t) \partial_x w(\mathbf{x}, t) + b_2(\mathbf{x}, t) \partial_y w(\mathbf{x}, t) + c(\mathbf{x}, t) w(\mathbf{x}, t) + f_2(\mathbf{x}, t), \quad (77)$$

where the new source term is

$$f_2(\mathbf{x}, t) = f_1(\mathbf{x}, t) - \mathcal{L}_t[u_g(\mathbf{x}, t)] + \mathcal{M}_t[L(\mathbf{x})[u_g(\mathbf{x}, t)]] + b_1(\mathbf{x}, t) \partial_x u_g(\mathbf{x}, t) + b_2(\mathbf{x}, t) \partial_y u_g(\mathbf{x}, t) + c(\mathbf{x}, t) u_g(\mathbf{x}, t), \quad (78)$$

which can now be solved using the method as described in the (1+1)-dimensional problems. The main difference for the solution process of the (1+1)-dimensional problems and the (2+1)-dimensional problems is that the initial approximation of the solution u_g and the correcting function $\omega_i(\mathbf{x})$ for the (1+1)-dimensional problem can be obtained analytically. However, for the (2+1)-dimensional problems, these functions can only be obtained numerically. In the following subsection, the goal for obtaining such functions are detailed.

3.3 The Procedure for Obtaining $u_g(\mathbf{x}, t)$ and $\omega_i(\mathbf{x})$

Since the artificially designed $u_g(\mathbf{x}, t)$ and $\omega_i(\mathbf{x})$ do not have the requirements to satisfy the governing equations, we apply the trigonometric basis, as follows:

$$\theta_k(\beta, x) = \sin\left(k\pi \frac{x + \beta}{2\beta}\right), \quad (79)$$

$$\theta_{k_1, k_2}(\beta, \mathbf{x}) = \sin\left(k_1\pi \frac{x_1 + \beta}{2\beta}\right) \sin\left(k_2\pi \frac{x_2 + \beta}{2\beta}\right), \quad (80)$$

for the (1+1)-dimensional problems and the (2+1)-dimensional problems, respectively, where the parameter β is selected in the way that $\Omega \subset \Omega_\beta$. Then we use the following approximations of the linear combination of θ with weighted parameters

$$\omega_i(\mathbf{x}) = \sum_{k=1}^{\mathcal{K}} p_{i,k} \theta_k(\beta, \mathbf{x}), \quad i = 1, \dots, N, \quad (81)$$

$$u_g(\mathbf{x}, t) = \sum_{k=1}^{\mathcal{K}} s_k(t) \theta_k(\beta, \mathbf{x}), \quad (82)$$

where the weighted parameters are determined using the collocation approach

$$\sum_{k=1}^{\mathcal{K}} p_{i,k} \mathcal{B}[\theta_k(\beta, \xi_l)] = \mathbf{H}_i, \quad \xi_l \in \partial\Omega, \quad i = 1, \dots, N, \quad l = 1, \dots, \mathcal{K}_c, \quad (83)$$

$$\sum_{k=1}^{\mathcal{K}} s_k(t) \mathcal{B}[\theta_k(\beta, \xi_l)] = \mathbf{G}(t), \quad \xi_l \in \partial\Omega, \quad l = 1, \dots, \mathcal{K}_c, \quad (84)$$

and the \mathbf{H}_i and $\mathbf{G}(t)$ are given as follows:

$$\mathbf{H}_i = -[B[\psi_i(\xi_1)], B[\psi_i(\xi_2)], \dots, B[\psi_i(\xi_{\mathcal{K}_c})]]^T, \quad \mathbf{G}(t) = [g(\xi_1, t), g(\xi_2, t), \dots, g(\xi_{\mathcal{K}_c}, t)]^T. \quad (85)$$

It is important to note that we need approximation of the function u_g in the way which admits calculation the fractional derivatives (see Eq. (78)). When the sought solution has the particular form $u(\mathbf{x}, t) = u_0(\mathbf{x}) \zeta(t)$, then a simplified procedure can be applied: we find $u_{g,0}(\mathbf{x})$ as a boundary approximation of $u_0(\mathbf{x})$ and define $u_g(\mathbf{x}, t) = u_{g,0}(\mathbf{x}) \zeta(t)$. Then the derivatives can be found as $D_t^{(\alpha)}[u_g(\mathbf{x}, t)] = u_{g,0}(\mathbf{x}) D_t^{(\alpha)}[\zeta(t)]$ at each time moment t . In general case, this approach is impossible.

To make this possible, we use the point interpolation method (PIM) [38,39]. Let $0 \leq t_1 < t_2 < \dots < t_I \leq T$ be the collocation points in the time interval. Let $\mathbf{S}_k = [s_k(t_1), s_k(t_2), \dots, s_k(t_I)]$ be the vector of the values of the smooth function $s_k(t)$ calculated at these points. We seek approximation of $s_k(t)$ at a point of interest t in the form of

$$\tilde{s}_k(t, I) = \sum_{i=1}^I t^{i-1} a_{k,i}, = \underbrace{[1, t, \dots, t^{I-1}]}_{\mathbf{p}^T} \left\{ \begin{array}{c} a_{k,1} \\ \dots \\ a_{k,I} \end{array} \right\} = \mathbf{p}^T \mathbf{a}_k, \quad (86)$$

and the coefficients $a_{k,i}$ in Eq. (86) can be determined by enforcing $\tilde{s}_k(t, I)$ to pass through the values at the collocation points t_j . This yields I equations for each collocation point, i.e.,

$$\left. \begin{array}{l} s_k(t_1) = a_{k,1} + a_{k,2}t_1 + \dots + a_{k,I}t_1^{I-1} \\ s_k(t_2) = a_{k,1} + a_{k,2}t_2 + \dots + a_{k,I}t_2^{I-1} \\ \dots \\ s_k(t_I) = a_{k,1} + a_{k,2}t_{M_p} + \dots + a_{k,I}t_I^{I-1} \end{array} \right\} \Rightarrow \mathbf{S}_k = \mathcal{P} \mathbf{a}_k, \quad (87)$$

where

$$\widehat{\mathcal{P}} = \begin{bmatrix} 1, t_1, \dots, t_1^{I-1} \\ 1, t_2, \dots, t_2^{I-1} \\ \dots \\ 1, t_{M_p}, \dots, t_{M_p}^{I-1} \end{bmatrix}, \quad (88)$$

is the so-called square moment matrix. Assuming that \mathcal{P} is a non-singular matrix, we get

$$\mathbf{a}_k = \widehat{\mathcal{P}}^{-1} \mathbf{S}_k. \quad (89)$$

Using vectors \mathbf{a}_k , $k = 1, \dots, \mathcal{K}$, we obtain $u_g(\mathbf{x}, t)$ in the form of the smooth function differentiable in time.

$$\begin{aligned} u_g(\mathbf{x}, t) &= \sum_{k=1}^{\mathcal{K}} \sum_{i=1}^I t^{i-1} a_{k,i} \theta_k(\beta, \mathbf{x}) = \sum_{i=1}^I \left(\sum_{k=1}^{\mathcal{K}} a_{k,i} \theta_k(\beta, \mathbf{x}) \right) t^{i-1} \\ &= \sum_{i=1}^I \Theta_i(\beta, \mathbf{x}) t^{i-1}. \end{aligned} \quad (90)$$

As a result, we get the approximate solution $v_{N,M}(\mathbf{x}, t)$ in the form similar to the one obtained in the case of the (1+1) dimensional problem

$$v_{N,M}(\mathbf{x}, t) = \sum_{i=1}^N \phi_i(\mathbf{x}) P_{M,i}(t) + u_g(\mathbf{x}, t) + v_0(\mathbf{x}), \quad (91)$$

where the approximate solution also depends on the parameters \mathcal{K} and I of the function $u_g(\mathbf{x}, t)$.

4 Numerical Examples

In this section, the accuracy and efficiency of the proposed method is verified. The maximum absolute error (MAE , $E_{\max}(t)$), the relative root mean square error ($E_{rel}(t)$), and also the error in discrete $H^1(\Omega)$ norm ($E_{H^1}(t)$) are used as criteria. The detailed computational strategies are given:

$$E_{\max}(t) = \max_{1 \leq i \leq N_t} |u_{N,M}(\mathbf{x}_i, t) - u_{ex}(\mathbf{x}_i, t)|, \quad (92)$$

$$E_{rel}(t) = \sqrt{\sum_{i=1}^{N_t} [u_{ex}(\mathbf{x}_i, t) - u_{N,M}(\mathbf{x}_i, t)]^2 / \sum_{i=1}^{N_t} u_{ex}^2(\mathbf{x}_i, t)}, \quad (93)$$

$$E_{H^1}(t) = \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} \left[(u_{ex}(\mathbf{x}_i, t) - u_{N,M}(\mathbf{x}_i, t))^2 + \sum_{k=1}^2 \left(\frac{\partial u_{ex}}{\partial x_k}(\mathbf{x}_i, t) - \frac{\partial u_{N,M}}{\partial x_k}(\mathbf{x}_i, t) \right)^2 \right]}, \quad (94)$$

where $u_{ex}(\mathbf{x}_i, t)$ and $u_{N,M}(\mathbf{x}_i, t)$ are analytical and numerical solutions, respectively, N_t is the number of the test points. To investigate the influence of N (the number of RBFs) and M (the number of Müntz polynomials) on the accuracy of the proposed scheme, we introduce the convergence orders (COs)

$$\begin{aligned} CO_N &= \frac{\log(E_{\max}(N_1)/E_{\max}(N_2))}{\log(N_1/N_2)}, \quad M \text{ is fixed,} \\ CO_M &= \frac{\log(E_{\max}(M_1)/E_{\max}(M_2))}{\log(M_1/M_2)}, \quad N \text{ is fixed.} \end{aligned} \quad (95)$$

Furthermore, for the 1D problems, we let $c = 0.5$ and for the 2D problems, we take $c = 2$. The data provided in Examples 1 to 4 demonstrate that in the whole range $\sigma \in [0.1, 0.3]$, the Müntz basis provides a quite precise approximation in time. The calculation results are very close for all values of the parameter σ from this range. As the parameter σ increases, the error grows rapidly. It reaches a maximum at $\sigma = 1$ when the usual polynomials of the integer order t^m , $m = 0, 1, 2, \dots$ are used in approximation of time. This demonstrates the advantage of using the Müntz basis. Therefore, for the parameters in the Müntz polynomials, we have $\sigma = 0.3$ for the rest examples and illustrations.

4.1 (1+1)-Dimensional Problems

4.1.1 Example 1

Let us consider the multi-term problem discussed in [13]

$$D_t^{(0.95)}[u] + D_t^{(0.2)}[u] = \partial_{xx}u + f(x, t), \quad 0 \leq x, t \leq 1, \tag{96}$$

with the first kind boundary condition and initial condition conform to $u(x, t) = (1 + t^2)(x^2 - x)$.

Fig. 1 and Table 1 show the behaviour of the errors with respect to the number of the modified RBFs $\phi_i(x)$, N in the approximation of the solution (see Eq. (54)) with fixed M . For $M = 10$, the predicted errors can be shapely reduced by increasing of the N for all the computed $2 \leq N \leq 26$. This means that the error occurring in spatial approximation is the main error source in this case. While for $M = 5$ the computed errors cannot be decreased for the selected $N > 10$. Therefore, in this case, the dominant error is the error in the solution of the system. With the increasing of N , the minimum error of this example is $5.80E-10$ and the convergence order vs. N is larger than 2. It is noticeable that for the $M = 10$ and $N = 26$, we can obtain the solution in 2.9 s which is sufficiently efficient. Fig. 1 shows that all the curves $E_{max}(N)$, $E_{H^1}(N)$ originally lie on the same curve. The predicted errors move away from this curve depending on the value of M , when the error occurring in the approximation in time plays an essential role.

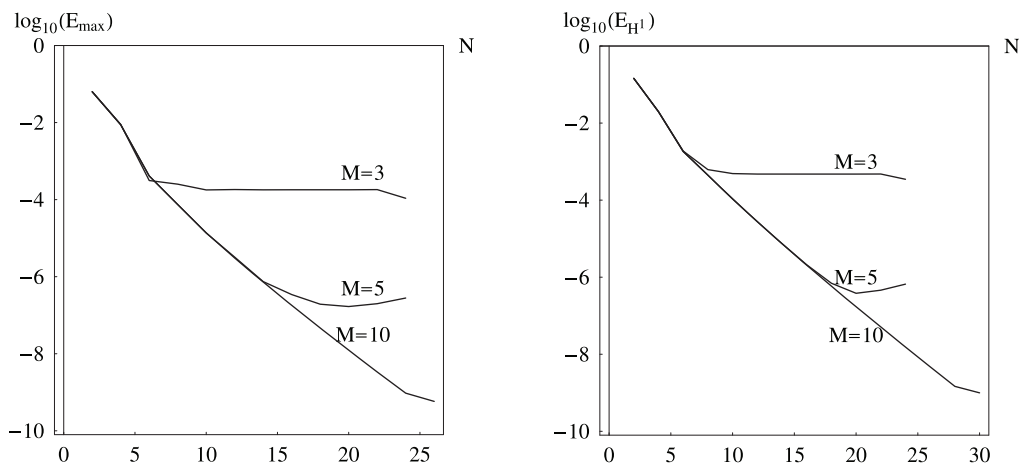


Figure 1: The E_{max} (left) and E_{H^1} (right) errors with the respect to the number of the RBFs used in the approximate solution

Table 1: The predicted errors vs. N at $t = 1$ with $M = 10$

N	E_{\max}	E_{rel}	E_{H^1}	CO_N	CPU, sec.
4	8.77E-3	2.96E-2	1.96E-2	2.8	0.5
8	7.45E-5	1.94E-4	4.45E-4	5.9	0.6
12	3.05E-6	7.21E-6	2.76E-5	8.3	0.9
16	1.81E-7	3.88E-7	2.07E-6	10.4	1.3
20	1.24E-8	2.52E-8	1.71E-7	12.6	1.9
24	9.49E-10	1.87E-9	1.52E-8	14.5	2.7
26	5.80E-10	1.22E-9	4.72E-9	6.1	2.9

Fig. 2 and Table 2 show the computed E_{\max} and E_{H^1} errors vs. the number of Müntz polynomials M with fixed N . The same problem has already been considered by Jin et al. in [13] using the Galerkin finite element method and finite difference discretization of the time-fractional derivatives. In their practical computations, the size of the mesh is $h = 2^{-10}$ and the time step for the temporal approximation is $\tau = 1/160$. They obtained $E_{\max} = 4.20E-4$ which was placed in the Table 1 of the reference [13]. It is evident that the proposed method is far more accurate than the reference. The domain errors with $N = 26$ and $M = 10$ is given in Fig. 3 from which we can see that in the whole domain, the proposed method maintains the same error level for the test nodes which are close to the boundary.

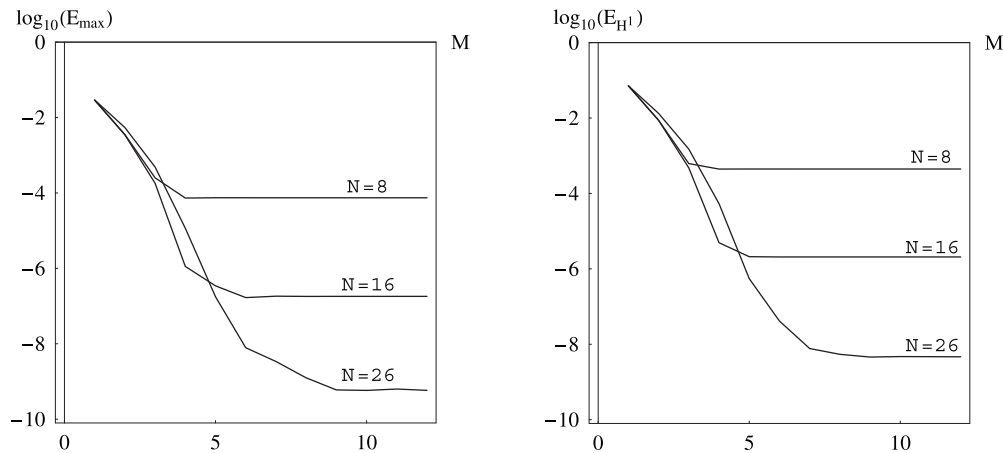


Figure 2: The E_{\max} (left) and E_{H^1} (right) errors vs. the number of Müntz’s polynomials M

Table 2: The errors with M at $t = 1$ with $N = 26$

M	E_{\max}	E_{rel}	E_{H^1}	CO	CPU, sec.
2	5.43E-3	2.18E-2	1.33E-2	2.4	0.5
4	1.16E-5	3.10E-5	5.34E-5	13.0	0.8
6	7.85E-9	2.17E-8	4.13E-8	17.1	1.2

(Continued)

Table 2 (continued)

M	E_{\max}	E_{rel}	E_{H^1}	CO	CPU, sec.
8	1.25E-9	3.12E-9	5.41E-9	7.5	1.8
10	5.80E-10	1.22E-9	4.72E-9	0.3	2.3

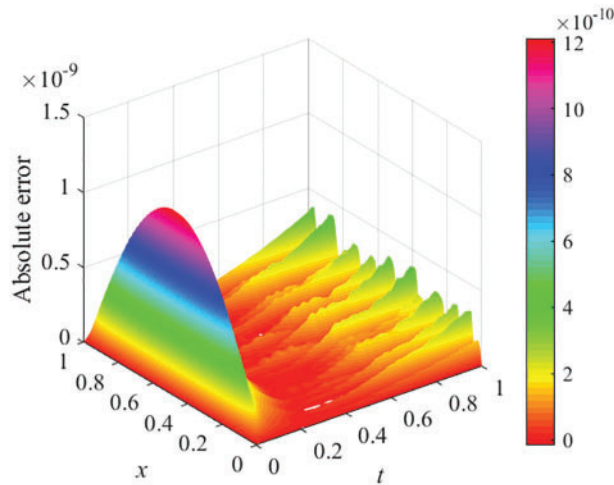


Figure 3: The domain absolute errors with $N = 26$ and $M = 10$

Finally, we show the behaviour of the errors E_{rel} with the growth of N for different σ in Table 3. This table demonstrates that in the whole range $\sigma \in [0.1, 0.3]$, the Müntz basis provides a quite precise approximation in time.

Table 3: The behaviour of the errors with the growth of N different σ

N	5	10	15	20	25
$\sigma = 0.1$	1.79E-3	1.37E-5	1.41E-7	1.24E-8	3.55E-10
$\sigma = 0.2$	1.79E-3	1.37E-5	1.41E-7	1.24E-8	3.91E-10
$\sigma = 0.3$	1.79E-3	1.37E-5	1.41E-7	1.24E-8	4.20E-10
$\sigma = 0.6$	1.79E-3	1.38E-5	1.40E-7	1.88E-8	6.97E-9
$\sigma = 1.0$	1.79E-3	1.38E-5	1.83E-7	1.10E-7	5.42E-7

4.1.2 Example 2

Let us consider the multi-term problem of the following form:

$$\begin{aligned}
 & D_t^{(\pi/8)} [u] + (1 + t) D_t^{(0.25)} [u] + (1 + t^2) D_t^{(0.125)} [u] \\
 & = \sin(t) D_t^{(0.2)} [L(x) [u]] + \cos(t) D_t^{(0.1)} [L(x) [u]] + e^t L(x) [u] + f(x, t), \quad 0 \leq x, t \leq 1,
 \end{aligned} \tag{97}$$

with the boundary condition of third kind (Robin BCs)

$$\partial_x u(0, t) - \pi u(0, t) = g_0(t), \quad \partial_x u(1, t) - eu(1, t) = g_1(t). \quad (98)$$

The BCs and IC conform the exact solution $u(x, t) = \exp(x + t)$. Here the spatial operator is $L(x)[u(x, t)] = \partial_x(\cosh(x) \partial_x u(x, t))$.

Some results of the calculations are presented in Tables 4 and 5 and displayed in the Figs. 4 and 5. From Tables 4 and 5, we can see that for the problem of the Robin boundary conditions, the proposed method can yield accurate results with small number of basis functions and collocation nodes. And the overall convergence order of the propose scheme is large than two. The same conclusions can be made from Figs. 4 and 5, where the computed maximum absolute errors vs. the M and N are displayed. From these figures, we can see that with the increasing of N for the case $N \geq 24$, the errors adjust within a small range which may due to properties of the coefficient matrix such as the ill-conditioning. The following three ways can be used to solve this problem. In the first strategy, we can increase the number of M , for $N \geq 24$, then we can obtain more accurate and stable results from the over-determined system. The second attempt can be done to use the high-precision matrix solver to overcome the ill-conditioning of the matrix. Furthermore, the recently proposed multiple-scale method by Oruç [40] can also be used to reduce the condition number of the coefficient matrix. For the fact that with the increasing of M and the fixed N , the approximate solutions become more accurate and the errors only adjust within a small range, we will not discuss in details of these techniques for improving the solution stability.

Table 4: The errors vs. N at $t = 1$ with $M = 12$

N	E_{\max}	E_{rel}	E_{H^1}	CO_N	CPU, sec.
4	1.48E-2	2.98E-3	3.38E-2	2.0	0.7
8	2.03E-4	2.91E-5	1.01E-3	5.1	1.3
12	8.30E-6	1.02E-6	6.35E-5	8.6	1.5
16	4.55E-7	5.49E-8	4.86E-6	10.6	1.9
20	2.96E-8	3.64E-9	4.11E-7	13.4	4.2
24	3.82E-9	5.15E-10	3.74E-8	13.9	5.4
26	5.16E-9	8.34E-10	1.57E-8	–	6.1

Table 5: The errors vs. M at $t = 1$ with $N = 26$

M	E_{\max}	E_{rel}	E_{H^1}	CO_M	CPU, sec.
2	1.92E-2	4.43E-3	4.62E-2	1.9	0.8
4	2.41E-3	5.46E-4	5.91E-3	3.8	1.5
6	4.69E-5	9.59E-6	1.18E-4	10.1	2.4
8	2.56E-6	4.38E-7	8.78E-6	10.5	3.4
10	4.36E-8	6.28E-9	2.30E-7	31.5	4.6
12	5.16E-9	8.34E-10	1.57E-8	11.7	6.6

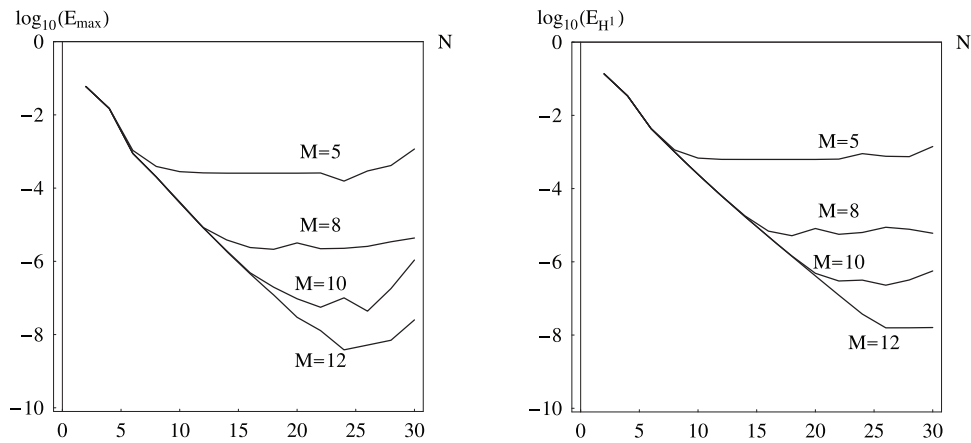


Figure 4: The E_{max} (left) and E_{H^1} (right) errors vs. N

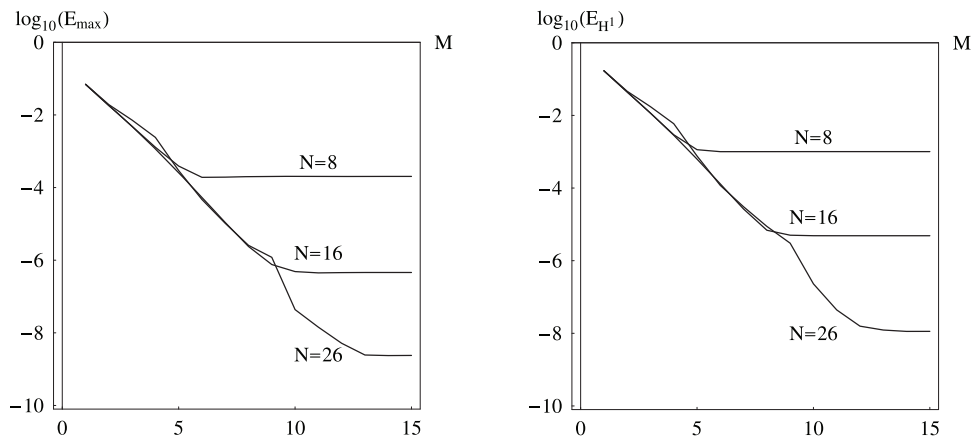


Figure 5: The E_{max} (left) and E_{H^1} (right) errors vs. M

In [Table 6](#), we show the effect of the σ on the accuracy. The same conclusion can be made that in the whole range $\sigma \in [0.1, 0.3]$, the Müntz basis provides a quite precise approximation in time.

Table 6: The behaviour of the errors with the growth of N for different σ

N	5	10	15	20	25
$\sigma = 0.1$	8.86E-4	5.29E-6	1.39E-7	8.20E-9	6.33E-9
$\sigma = 0.2$	8.86E-4	5.28E-6	1.48E-7	2.58E-9	2.36E-10
$\sigma = 0.3$	8.86E-4	2.58E-6	1.48E-7	2.61E-9	2.44E-10
$\sigma = 0.6$	8.86E-4	4.01E-6	3.24E-6	4.29E-6	3.67E-6
$\sigma = 1.0$	6.11E-4	7.32E-4	7.40E-4	9.04E-4	1.10E-3

4.1.3 Example 3

Consider the nonlinear time-fractional Huxley equation of the following form:

$$D_t^{(0.5)} [u(x, t)] = \partial_{xx} u(x, t) + u(x, t) (1 - u(x, t))(u(x, t) - \gamma) + f(x, t), \quad 0 \leq x, t \leq 1, \quad (99)$$

with $u(x, t) = x^3(t + 1)$.

Table 7 and Fig. 6 show the trends of the computed errors with the growth of N with fixed M . The predicted solutions are obtained after 3 iteration of the quasilinearization procedure. The initial approximations of the solution and its derivatives are provided randomly. From the results, we can verify that for the randomly provided initial approximations, we can obtained the accurate solutions in three iterations. And with the increasing of the N , the proposed algorithm converges fast. The same problem was considered by Hadhoud et al. in [41] using a numerical technique based on the cubic B-spline collocation method and the mean value theorem for integrals. The maximal absolute errors obtained there are near $\sim 10^{-6}$ for the mesh size $\Delta x = 0.01$ and the time step $\Delta t = 0.01$ (see Tables 4–6 of the original paper).

Table 7: The errors vs. N at $t = 1$ with $M = 10$

N	E_{\max}	E_{rel}	E_{H^1}	CO_N	CPU, sec.
4	1.48E-2	2.46E-2	3.40E-2	3.7	0.6
8	2.43E-4	2.98E-4	1.04E-3	5.8	1.6
12	9.09E-6	8.43E-6	5.83E-5	9.2	3.3
16	4.50E-7	3.99E-7	4.15E-6	11.1	5.6
20	2.80E-8	2.44E-8	3.22E-7	9.7	7.9
24	2.24E-9	1.79E-9	2.86E-8	0.03	9.7

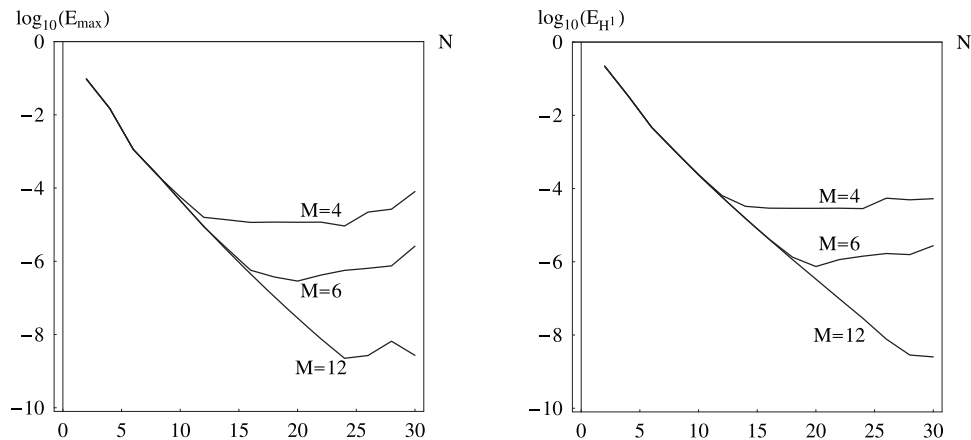


Figure 6: The maximal absolute E_{\max} (left) and E_{H^1} (right) errors as functions of the number of the RBFs used in the approximate solution

Now let us move onto the multi-term nonlinear time-fractional equation with the Huxley nonlinear term,

$$\begin{aligned}
 & D_t^{(0.5)} [u(x, t)] + \sin(t)D_t^{(0.3)} [u(x, t)] + \cos(t)D_t^{(0.2)} [u(x, t)] \\
 &= \sin h(t)D_t^{(0.29)} [L(x) [u(x, t)]] + \cosh(t)D_t^{(0.19)} [L(x) [u(x, t)]] + e^t L(x) [u(x, t)] \\
 &+ u(x, t) (1 - u(x, t))(u(x, t) - \gamma) + f(x, t), \quad 0 \leq x, t \leq 1,
 \end{aligned}
 \tag{100}$$

with the same analytical solution. Here $L(x) [u(x, t)] = \partial_x ((1 + x^2) \partial_x u(x, t))$.

Fig. 7 demonstrates the behaviour of the errors with the grows of the total amount of basis function $\phi_i(x)$ for the approximation of the solution (see Eq. (54)). The data correspond to the 3th iterations of the quasilinearization procedure with randomly provided initial approximations of the solution and its derivatives. If we apply much more steps of iterations, the solution accuracy do not change which means that three steps of iterations are sufficient for the proposed method.

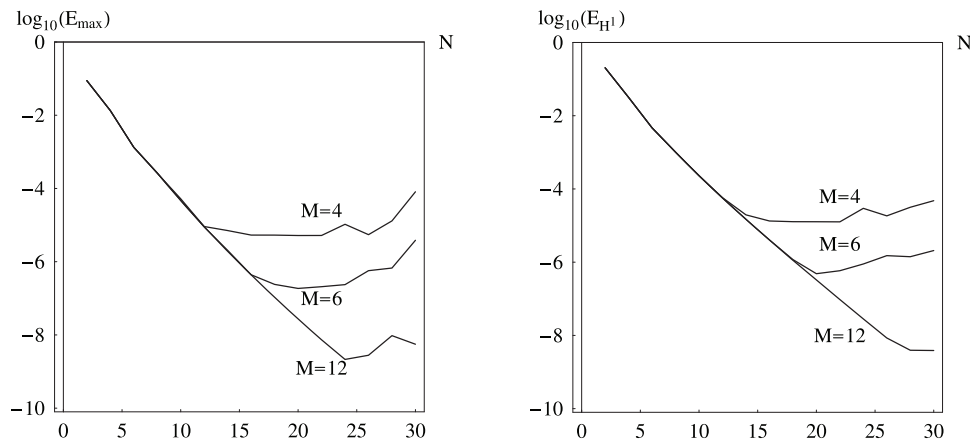


Figure 7: The E_{max} (left) and E_{H^1} (right) errors vs. the number of N

Then, we discuss the solution accuracy on the σ in Table 8. It is evident that the $\sigma = 0.3$ is sufficient to do the approximation.

Table 8: The behaviour of the errors with the growth of N different σ

N	5	10	15	20	25
$\sigma = 0.1$	6.69E-3	4.72E-5	1.28E-6	2.74E-8	2.25E-9
$\sigma = 0.2$	6.69E-3	4.72E-5	1.28E-6	2.74E-8	2.18E-9
$\sigma = 0.3$	6.69E-3	4.72E-5	1.28E-6	2.73E-8	2.15E-9
$\sigma = 0.6$	6.69E-3	4.71E-5	1.40E-6	2.43E-7	1.05E-6
$\sigma = 1.0$	6.68E-3	3.93E-5	6.69E-6	5.02E-5	6.78E-5

4.1.4 Example 4

Let us consider the nonlinear problem with the spatial differential operator of the fourth order

$$\begin{aligned}
 & D_t^{(0.9)} [u(x, t)] + t D_t^{(0.1)} [u(x, t)] + t^2 D_t^{(0.2)} [u(x, t)] \\
 &= \frac{1}{1+t} D_t^{(0.29)} [L(x) [u(x, t)]] + \frac{1}{1+t^2} D_t^{(0.19)} [L(x) [u(x, t)]] \\
 &+ L(x) [u(x, t)] + N(u(x, t), \partial_x u(x, t)) + f(x, t), \quad 0 \leq x, t \leq 1,
 \end{aligned} \tag{101}$$

where the spatial operator

$$L(x) [u(x, t)] = -\partial_{xx} [\cos(x) \partial_{xx} u(x, t)] + \partial_x [\cosh(x) \partial_x u(x, t)] - \sinh(x) u(x, t), \tag{102}$$

and the nonlinear term is

$$N(u(x, t), \partial_x u(x, t)) = \sin(u(x, t)) + (\partial_x u(x, t))^2. \tag{103}$$

The boundary condition and initial condition conform to the exact $u(x, t) = \sin(t) \sin(\pi x)$. The predicted results shown in Table 9 and Fig. 8 are obtained after 3th iterations of the quasilinearization procedure with randomly provided initial guess of solution and its derivatives. From Table 9, we can conclude that the proposed approach can produce accurate solution for the problems of high-order with a small number of basis functions both in terms of the spatial and temporal approximations. For the $M = 6$, the method converges with $N = 14$. And if we increase the $M = 12$, the proposed method converges with $N = 20$. And even with the small number of $N = 10$ and $M = 6$, the computational accuracy of the problem with the spatial differential operator of the fourth order is high around 10^{-5} .

Table 9: The errors vs. N at $t = 1$ with $M = 15$

N	$M = 6$				$M = 12$			
	E_{\max}	E_{rel}	CO_N	CPU, sec.	E_{\max}	E_{rel}	CO_N	CPU, sec.
10	8.19E-5	5.66E-5	14.1	1.7	7.36E-5	5.04E-5	14.5	6.4
12	1.09E-5	7.76E-6	11	2.3	2.68E-6	1.88E-6	18.2	9.3
14	8.38E-6	5.88E-6	1.70	3.5	6.20E-8	2.84E-8	24.2	13
16	8.31E-6	5.89E-6	–	4.2	2.90E-8	1.15E-8	5.7	17
18	8.32E-6	5.88E-6	–	5.1	6.08E-9	2.44E-9	13.89	22
20	8.32E-6	5.88E-6	–	6.2	1.77E-9	6.65E-10	11.7	27
22	8.29E-6	5.86E-6	–	7.6	5.08E-10	1.88E-10	13	33
24	8.26E-6	5.84E-6	–	9.1	1.73E-10	5.29E-11	12	39
26	8.10E-6	5.73E-6	–	10.1	2.23E-10	1.08E-10	–	46

In Table 10, we display the effect of the parameter σ in the calculations. From this table, we can see that $\sigma = 0.3$ provides a quite precise approximation in time. Therefore, in the following examples, we take $\sigma = 0.3$ without further discussion.

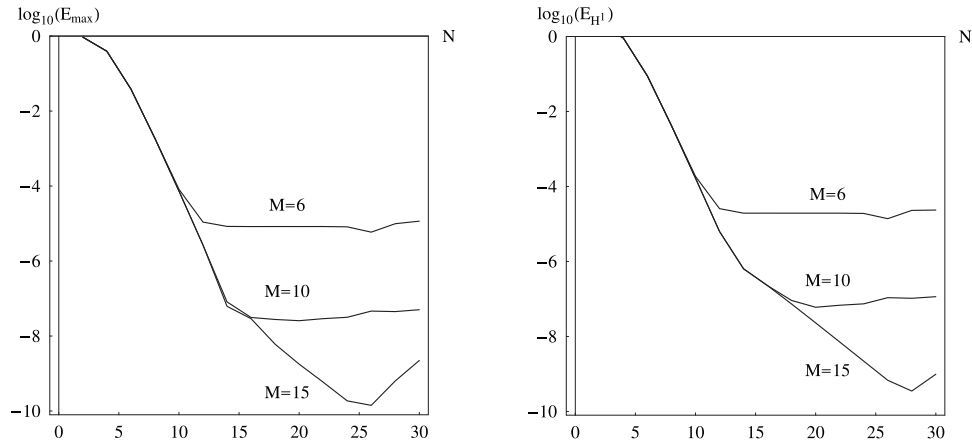


Figure 8: The maximal absolute E_{max} (left) and E_{H^1} (right) errors as functions of the number of the RBFs

Table 10: The behaviour of the errors with the growth of N for different σ

N	5	10	15	20	25
$\sigma = 0.1$	1.14E-1	7.36E-5	4.11E-8	4.65E-9	3.41E-9
$\sigma = 0.2$	1.14E-1	7.36E-5	3.82E-8	1.78E-9	6.02E-11
$\sigma = 0.3$	1.14E-1	7.36E-5	3.80E-8	1.96E-9	2.23E-10
$\sigma = 0.6$	1.14E-1	6.51E-5	8.74E-6	4.33E-6	2.04E-6
$\sigma = 1.0$	1.15E-1	7.08E-5	4.81E-4	2.49E-5	2.34E-4

4.2 (2+1)-Dimensional Problems

4.2.1 Example 5

Let us consider the linear fractional equation of the following form:

$$\begin{aligned}
 & D^{(\pi/4)} [u] + tD^{(0.3)} [u] + t^2D^{(0.2)} [u] + t^3D^{(0.1)} [u] \\
 & = (1 + t^2) D_t^{(0.33)} [L(x, y) [u]] + (1 + t^3) D_t^{(0.22)} [L(x, y) [u]] \\
 & \quad + (1 + t^4) D_t^{(0.11)} [L(x, y) [u]] + (1 + t) L(x, y) [u] + f(x, y, t),
 \end{aligned} \tag{104}$$

in the domain governed by

$$x = \rho(\theta) \cos \theta, y = \rho(\theta) \sin \theta, \rho(\theta) = \left(\cos 4\theta + \sqrt{3.6 + \sin^2 4\theta} \right)^{1/3}, 0 \leq \theta \leq 2\pi, \tag{105}$$

as shown in Fig. 9.

Here

$$L(x, y) [u] = \text{div} (\widehat{\mathbf{D}}(x, y) \nabla u), \tag{106}$$

where the matrix of diffusivity is

$$\widehat{\mathbf{D}} = \begin{pmatrix} \cosh(x + y), & 0.1(x^2 + y^2) \\ 0.1(x^2 + y^2), & \cosh(x - y) \end{pmatrix}. \tag{107}$$

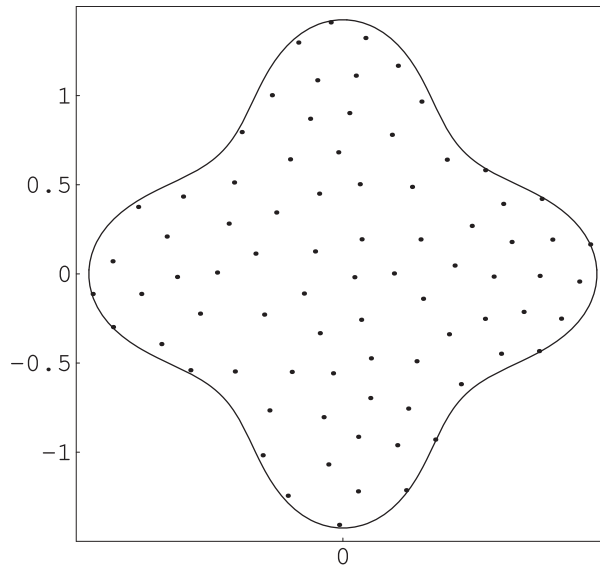


Figure 9: Profile of the irregular domain

The boundary condition of the Robin type:

$$\frac{\partial u}{\partial \mathbf{n}} + (x^2 + y^2) u = g(x, y, t), \quad (x, y) \in \partial\Omega, \quad (108)$$

and IC conform the exact solution $u_{ex}(x, t) = \exp(x + y + t)$.

Table 11 shows the errors as the functions of N where M is fixed at 25, and the maximum error is about 10^{-7} for $M = 25$ and $N = 10$. From this table, it is clear that with the increasing of N , the errors decrease slowly indicating that $N = 10$ is sufficient for the approximation. Additionally, for the small number of basis function and collocation nodes, the computational cost is low. For larger parameters, more computational resources should be deployed with less efficiency.

Table 11: The errors vs. N at $t = 1$ with fixed $M = 25$

N	E_{\max}	E_{rel}	E_{H^1}	CPU, sec.
10	6.99E-7	4.41e-8	9.32E-7	27
20	5.82E-7	4.86E-8	8.23E-7	66
40	1.41E-7	1.01E-8	1.94E-7	249
60	1.66E-8	9.61E-10	1.87E-8	360
80	7.44E-9	3.78E-10	1.45E-8	756

4.2.2 Example 6

In the last example, we consider a nonlinear problem

$$D^{(0.75)}[u] + \cosh(t) D^{(0.3)}[u] + t^2 D^{(0.2)}[u] = (1 + t^2) D_t^{(0.33)}[L(x, y)[u]] + L(x, y)[u] + u \partial_x u + u \partial_y u - u^2 + f(x, y, t), \quad (109)$$

in the star-like domain shown in Fig. 10. Here

$$L(x, y)[u] = \text{div}(\widehat{\mathbf{D}}(x, y) \nabla u), \tag{110}$$

where the matrix of diffusivity is

$$\widehat{\mathbf{D}} = \begin{pmatrix} \cosh(x+y), & 0.1(x+y)^2 \\ 0.1(x+y)^2, & \cosh(x-y) \end{pmatrix}. \tag{111}$$

The Dirichlet boundary conditions and IC can be obtained from $u_{ex}(x, t) = \exp(t) \sinh(x) \sinh(y)$. The results displayed in Table 12 are obtained after 3th iterations of the quasilinearization procedure where $M = 10$ and $t = 1$ and the initial guesses are randomly provided. From the table, we can see that the parameters $M = 10$ and $N = 10$ are sufficient to obtain approximate solutions and its derivatives where the three computed errors E_{\max} , E_{rel} , and E_{H^1} are about 10^{-9} . From this table, we notice that for the $N = 50$ and $M = 10$, the computational time is about 505 s for the fact that the full dense matrix system should be solved at each time step of the iterations. The computational cost can be reduced if the localized scheme is absorbed where the sparse matrix system can be solved efficiently. We refer the readers to [42]. The accuracy of the proposed method can also be verified from Figs. 11 to 13 where the domain absolute errors of u , $\partial u/\partial x$, and $\partial u/\partial y$ are plotted. The maximum errors are occurred at the boundary.

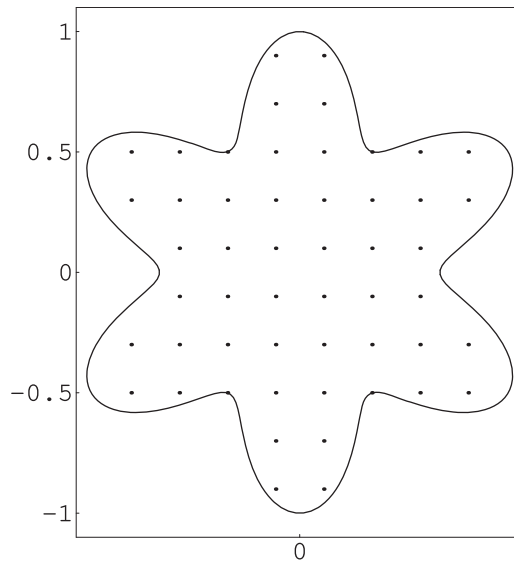


Figure 10: Profile of irregular star-like solution domain

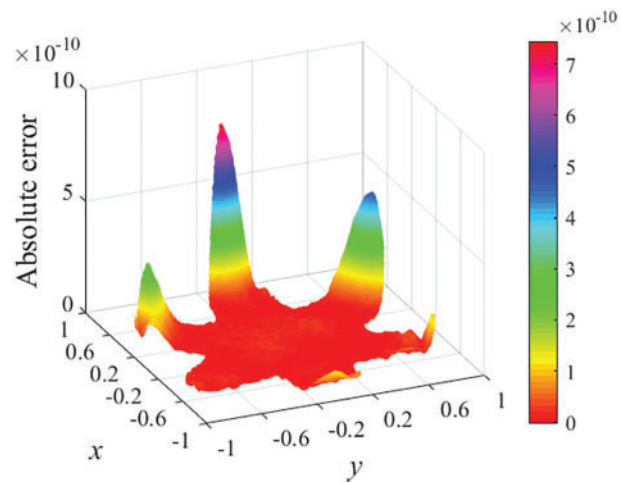
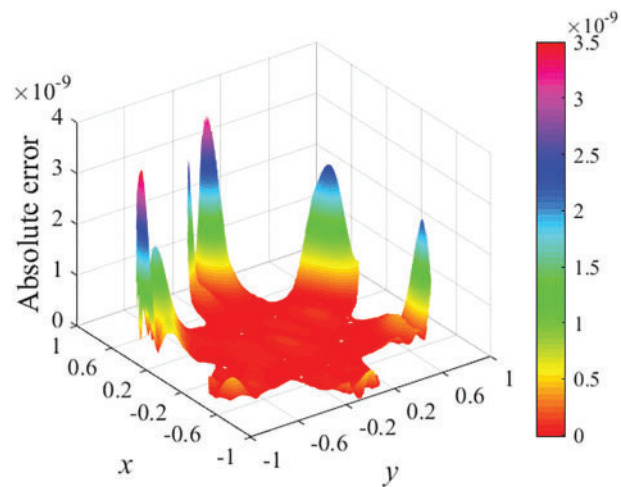
Table 12: The errors vs. N at $t = 1$ with fixed $M = 10$

N	E_{\max}	E_{rel}	E_{H^1}	CPU, sec.
10	5.88E-9	2.53E-9	6.59E-9	30
15	5.30E-9	2.34E-9	7.07E-9	58
20	3.17E-8	1.40E-8	2.82E-8	94
25	2.57E-9	1.03E-9	3.49E-9	138

(Continued)

Table 12 (continued)

N	E_{\max}	E_{rel}	E_{H^1}	CPU, sec.
30	3.97E-9	9.81E-10	2.92E-9	281
35	7.08E-10	3.86E-10	1.06E-9	334
40	4.83E-10	1.73E-10	5.42E-10	380
50	2.37E-10	8.95E-11	2.45E-10	505

**Figure 11:** Absolute errors of u **Figure 12:** Absolute errors of $\partial u/\partial x$

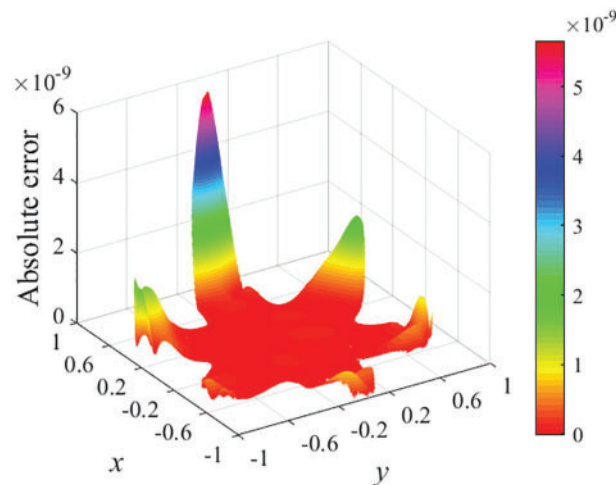


Figure 13: Absolute errors of $\partial u/\partial y$

5 Conclusion

In order to solve (1+1)-dimensional and (2+1)-dimensional time-fractional nonlinear diffusion equations of multi-term accurately, we propose a simple technique in this paper. First, we convert the original problem into a new one under homogeneous initial conditions by subtracting the initial conditions from the unknown desired solutions. Then the key issue is to solve the problem with zero initial conditions. After that, we try to seek the function $u_g(\mathbf{x}, t)$ analytical or numerically which satisfies the specify boundary conditions of the transformed problem. By the decomposition with the computed $u_g(\mathbf{x}, t)$ which makes it possible that the original problem of homogeneous conditions can be solved efficiently. By this conversion, we can seek the function in the form of the truncated series over the modified radial basis functions. The modified radial basis functions and the linear combinations pose the properties satisfying the homogeneous boundary conditions. The collocation at the centers of the RBF transforms the problems into a system of ordinary equations which are solved by the recently proposed backward substitution technique. The numerical examples including linear and nonlinear problems prove the accuracy and efficiency of the proposed method. From the numerical results, we can see that with a small number of collocations nodes, the problems can be solved efficiently with high accuracy and converging order compared with the results in the literature. It should be noted here that this paper provides a general algorithm for solving time-dependent fractional equations which can be considered as an alternative tool for engineering.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Funding Statement: This research was funded by the National Key Research and Development Program of China (No. 2021YFB2600704), the National Natural Science Foundation of China (No. 52171272), and the Significant Science and Technology Project of the Ministry of Water Resources of China (No. SKS-2022112).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: T. Hu, C. Huang, and S. Reutskiy; data collection: J. Lu; analysis and interpretation of results:

J. Lin and S. Reutskiy; draft manuscript preparation: J. Lu and J. Lin. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: None.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Liu, L., Feng, L., Xu, Q., Zheng, L., Liu, F. (2020). Flow and heat transfer of generalized Maxwell fluid over a moving plate with distributed order time fractional constitutive models. *International Communications in Heat and Mass Transfer*, 116(6), 104679.
2. Zheng, L., Liu, Y., Zhang, X. (2012). Slip effects on MHD flow of a generalized Oldroyd-B fluid with fractional derivative. *Nonlinear Analysis: Real World Applications*, 13(2), 513–523.
3. Lyu, P., Vong, S. (2019). A high-order method with a temporal nonuniform mesh for a time-fractional benjamin-bona-mahony equation. *Journal of Scientific Computing*, 80(3), 1607–1628.
4. Bayarassou, K. (2021). Fourth-order accurate difference schemes for solving benjamin-bona-mahony-burgers (BBMB) equation. *Engineering with Computers*, 37(1), 123–138.
5. Dehghan, M., Shafieeabyaneh, N., Abbaszadeh, M. (2020). Application of spectral element method for solving sobolev equations with error estimation. *Applied Numerical Mathematics*, 158, 439–462.
6. Zhao, J., Fang, Z., Li, H., Liu, Y. (2020). A crank-nicolson finite volume element method for time fractional sobolev equations on triangular grids. *Mathematics*, 8(9), 1591.
7. Ren, J., Sun, Z. (2015). Efficient numerical solution of the multi-term time fractional diffusion-wave equation. *East Asian Journal on Applied Mathematics*, 5(1), 1–28.
8. Salehi, R. (2017). A meshless point collocation method for 2-D multi-term time fractional diffusion-wave equation. *Numerical Algorithms*, 74, 1145–1168.
9. Yang, X., Qiu, W., Zhang, H., Tang, L. (2021). An efficient alternating direction implicit finite difference scheme for the three-dimensional time-fractional telegraph equation. *Computers & Mathematics with Applications*, 102, 233–247.
10. Shen, S., Liu, F., Anh, V. (2019). The analytical solution and numerical solutions for a two-dimensional multi-term time fractional diffusion and diffusion-wave equation. *Journal of Computational and Applied Mathematics*, 345, 515–534.
11. Stynes, M., O’Riordan, E., Gracia, J. (2017). Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation. *SIAM Journal on Numerical Analysis*, 55(2), 1057–1079.
12. Soori, Z., Aminataei, A. (2018). Sixth-order non-uniform combined compact difference scheme for multi-term time fractional diffusion-wave equation. *Applied Numerical Mathematics*, 131, 72–94.
13. Jin, B., Lazarov, R., Liu, Y., Zhou, Z. (2015). The Galerkin finite element method for a multi-term time-fractional diffusion equation. *Journal of Computational Physics*, 281, 825–843.
14. Huang, J., Zhang, J., Arshad, S., Tang, Y. (2021). A numerical method for two-dimensional multi-term time-space fractional nonlinear diffusion-wave equations. *Applied Numerical Mathematics*, 159, 159–173.
15. Hussain, M., Haq, S. (2019). Weighted meshless spectral method for the solutions of multi-term time fractional advection-diffusion problems arising in heat and mass transfer. *International Journal of Heat and Mass Transfer*, 129, 1305–1316.
16. Oruç, O., Esen, A., Bulut, F. (2019). A haar wavelet approximation for two-dimensional time fractional reaction-subdiffusion equation. *Engineering with Computers*, 35(1), 75–86.

17. Oruç, O. (2022). An accurate computational method for two-dimensional (2D) fractional Rayleigh-Stokes problem for a heated generalized second grade fluid via linear barycentric interpolation method. *Computers & Mathematics with Applications*, 118, 130–131.
18. Li, J., Dai, L., Nazeer, W. (2020). Numerical solution of multi-term time fractional wave diffusion equation using transform based local meshless method and quadrature. *AIMS Mathematics*, 5(6), 5813–5839.
19. Soltani Sarvestani, F., Heydari, M., Niknam, A., Avazzadeh, Z. (2019). A wavelet approach for the multi-term time fractional diffusion-wave equation. *International Journal of Computer Mathematics*, 96(3), 640–661.
20. Liu, Y., Yin, X., Liu, F., Xin, X., Shen, Y. et al. (2022). An alternating direction implicit legendre spectral method for simulating a 2D multi-term time-fractional Oldroyd-B fluid type diffusion equation. *Computers & Mathematics with Applications*, 113, 160–173.
21. Nikan, O., Tenreiro Machado, J., Golbabai, A. (2021). Numerical solution of time-fractional fourth-order reaction-diffusion model arising in composite environments. *Applied Mathematical Modelling*, 89, 819–836.
22. Zhou, J., Xu, D., Qiu, W., Qiao, L. (2022). An accurate, robust, and efficient weak Galerkin finite element scheme with graded meshes for the time-fractional quasi-linear diffusion equation. *Computers & Mathematics with Applications*, 124, 188–195.
23. Su, B., Jiang, Z. (2020). High-order compact finite volume scheme for the 2D multi-term time fractional sub-diffusion equation. *Advances in Difference Equations*, 2020, 1–22.
24. Ghafoor, A., Khan, N., Hussain, M., Ullah, R. (2022). A hybrid collocation method for the computational study of multi-term time fractional partial differential equations. *Computers & Mathematics with Applications*, 128, 130–144.
25. Piret, C., Hanert, E. (2013). A radial basis functions method for fractional diffusion equations. *Journal of Computational Physics*, 238, 71–81.
26. Hosseini, V., Chen, W., Avazzadeh, Z. (2014). Numerical solution of fractional telegraph equation by using radial basis functions. *Engineering Analysis with Boundary Elements*, 38, 31–39.
27. Liu, Q., Gu, Y., Zhuang, P., Nie, Y. (2011). An implicit RBF meshless approach for time fractional diffusion equations. *Computational Mechanics*, 48, 1–12.
28. Qiao, Y., Zhai, S., Feng, X. (2017). RBF-FD method for the high dimensional time fractional convection-diffusion equation. *International Communications in Heat and Mass Transfer*, 89, 230–240.
29. Reutskiy, S. (2017). A new semi-analytical collocation method for solving multi-term fractional partial differential equations with time variable coefficients. *Applied Mathematical Modelling*, 45, 238–254.
30. Reutskiy, S., Fu, Z. (2018). A semi-analytic method for fractional-order ordinary differential equations: Testing results. *Fractional Calculus and Applied Analysis*, 21(6), 1598–1618.
31. Reutskiy, S., Lin, J. (2018). A semi-analytic collocation method for space fractional parabolic PDE. *International Journal of Computer Mathematics*, 95(6–7), 1326–1339.
32. Lin, J., Zhang, Y., Reutskiy, S. (2021). A semi-analytical method for 1D, 2D and 3D time fractional second order dual-phase-lag model of the heat transfer. *Alexandria Engineering Journal*, 60(6), 5879–5896.
33. Tian, X., Reutskiy, S., Fu, Z. (2022). A novel meshless collocation solver for solving multi-term variable-order time fractional PDEs. *Engineering with Computers*, 38, 1527–1538.
34. Lin, J., Bai, J., Reutskiy, S., Lu, J. (2023). A novel RBF-based meshless method for solving time-fractional transport equations in 2D and 3D arbitrary domains. *Engineering with Computers*, 39, 1905–1922.
35. Pourbabaee, M., Saadatmandi, A. (2022). A new operational matrix based on Müntz-Legendre polynomials for solving distributed order fractional differential equations. *Mathematics and Computers in Simulation*, 194, 210–235.
36. Borwein, P., Erdelyi, T., Zhang, J. (1994). Müntz systems and orthogonal Müntz-Legendre polynomials. *Transactions of the American Mathematical Society*, 342(2), 523–542.

37. Bellman, R., Kalaba, R. (1965). *Quasilinearization and nonlinear boundary-value problems*. New York, USA: American Elsevier Publishing Company.
38. Liu, G., Gu, Y. (2001). A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering*, 50(4), 937–951.
39. Liu, G., Gu, Y. (2005). *An introduction to meshfree methods and their programming*. Berlin, Germany: Springer Science & Business Media.
40. Oruç, O. (2023). A strong-form meshfree computational method for plane elastostatic equations of anisotropic functionally graded materials via multiple-scale Pascal polynomials. *Engineering Analysis with Boundary Elements*, 146, 132–145.
41. Hadhoud, A., Abd Alaal, F., Abdelaziz, A., Radwan, T. (2021). Numerical treatment of the generalized time-fractional Huxley-Burgers' equation and its stability examination. *Demonstratio Mathematica*, 54(1), 436–451.
42. Lin, J., Xu, Y., Zhang, Y. (2020). Simulation of linear and nonlinear advection-diffusion-reaction problems by a novel localized scheme. *Applied Mathematics Letters*, 99, 106005.