The Journal of Supercomputing (2023) 79:14457–14488
https://doi.org/10.1007/s11227-023-05198-z

New numerical methods for solving the partial fractional differential equations with uniform and non-uniform meshes

Mohammad Javidi[1] · Mahdi Saedshoar Heris[1]

Accepted: 15 March 2023 / Published online: 7 April 2023

Abstract
In this paper, we design and develop some algorithms by using the piecewise linear interpolation polynomial for solving the partial fractional differential equations involving Caputo derivative, with uniform and non-uniform meshes. For designing new methods, we select the mesh points based on the two equal-height and equal-area distribution. Furthermore, the error bounds of proposed methods with uniform and equidistributing meshes are obtained. We also show that our numerical method is stable and convergent with the accuracy of $O(\kappa^2 + h)$. Also, some numerical examples are constructed to demonstrate the efficacy and usefulness of the numerical methods. Finally, a comparative study for different values of parameters is also presented.

Keywords Partial fractional differential equations · Uniform and non-uniform meshes · Stability and convergence

Mathematics Subject Classification 35R11 · 65L20 · 65N06 · 65N22

1 Introduction

The study of fractional calculus dates back to times when Leibnitz and Newton invented differential calculus. Fractional calculus deals with derivatives and integrals of arbitrary real order. It is a powerful tool for modeling phenomena arising in diverse fields such as mechanics, physics, engineering, economics, finance, medicine, biology, and chemistry [1–6]. In the past few decades, fractional differential equations (FDEs)

✉ Mohammad Javidi
mo_javidi@tabrizu.ac.ir

Mahdi Saedshoar Heris
m.saed@tabrizu.ac.ir

1 Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

have been used in increasingly more applications. Recently, there has been a tremendous increase in the use of fractional differential equations to simulate dynamics in many fields, e.g., physics, chemistry, biology, engineering and so on. For example, ultrasonic wave propagation in human cancellous bone [7], modeling of speech signals [8], modeling the cardiac tissue electrode interface [9], the sound waves propagation in rigid porous materials [10], lateral and longitudinal control of autonomous vehicles [11], the theory of viscoelasticity [12], fractional differentiation for edge detection [13], fluid mechanics [14], Electrical spectroscopy impedance [15], Frequency-dependent acoustic wave propagation in porous media [16], etc.

In general, there does not exist method that yields an exact solution for fractional differential equations. Several analytical methods have been suggested to solve fractional differential equations, such as, the homotopy perturbation method [17], Adomian's decomposition method [18–20], homotopy analysis method [21], the Laplace transform method, fractional Green's function, Power series method, and method of orthogonal polynomials [22–25].

There have been several numerical methods published for producing approximate solutions for fractional differential equations. These methods include the Implicit Quadrature method, introduced by Diethelm [26], the Predictor-Corrector method, discussed by Diethelm, Ford and Freed [27], the Approximate Mittag-Leffler method, considered by Diethelm and Luchko [28], a Collocation method, described by Blank [29], the Finite Differences method, discussed by Gorenflo [6], etc. [30–36].

The modeling of real-world problems and physical systems leads to partial FDEs (PFDEs). Analytical solutions as in the case of PFDEs are available only for a few simple PFDEs. Though researchers have developed efficient numerical solution methods for partial FDEs, in general, the literature on the numerical approximation of partial fractional derivative and present a simple general efficient numerical methods for the solution of PFDEs, are limited. Some analytical techniques are presented in the literature for solving PFDEs, such as, method of separating variables [37], decomposition method [38], variational iteration method [39], and homotopy-perturbation method [40]. To study numerical methods for solving partial fractional differential equations, see [36, 41–50, 52].

One of the disadvantages of finite difference methods by uniform meshes for solving fractional differential equations is its high computational cost. We show that the computational cost of the non-uniform meshes scheme is lower compared to the method of uniform meshes scheme and does not lose the numerical accuracy of this method.

This paper focuses on designing a new numerical method by uniform and non-uniform meshes for the partial fractional differential equation as:

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = \lambda_\alpha {}_0^C D_x^\alpha u(x,t) + f(x,t), \quad t > 0, \ x \in [0, L], \\ u(x,0) = g(x), \quad 0 < \alpha < 1, \\ u(0,t) = \mu_1(t), \quad u(L,t) = \mu_2(t), \end{cases} \tag{1}$$

where, $\lambda_\alpha < 0$ and $L > 0$ are constants. Also, the fractional derivative operator ${}^C{}_0D_x{}^\alpha$ is Caputo's derivative as [22]

$$ {}_0^C D_x^\alpha Z(x) = \frac{1}{\Gamma(n-\alpha)} \int_0^x \frac{Z^{(n)}(s)}{(x-s)^{\alpha-n+1}} \text{ds}, \quad n-1<\alpha<n. \tag{2} $$

In this paper, an initial value problem for the partial fractional differential equation is considered. We design new methods with uniform meshes and non-uniform meshes. The error bounds are obtained for solving our problem. Finally, some examples are presented, and also, we compared results obtained by the new methods with uniform and non-uniform meshes.

The rest of this paper is organized as follows. In Sect. 2, a new numerical method with uniform meshes is presented. In Sect. 3, a new numerical method with non-uniform meshes is developed. We perform the error analysis for those methods in Sect. 4. In Sect. 5, examples illustrating the performance of the new numerical schemes are presented. In the last section, conclusions are given.

2 Numerical method with uniform meshes

The purpose of this section is to present a new numerical method by using the piecewise linear interpolation polynomial with uniform meshes for solving the partial fractional differential Eq. (1). We partition $[0, L]$ into a uniform mesh with the space step size $h = L/M$ and the time step size $t = T/N$, where M, N are two positive integers. Also we have, $x_n = nh$ for $n = 1, ..., M$ and $t_j = j\kappa$ for $j = 1, ..., N$.

By using Eq. (2), we can write

$$ \begin{aligned} \frac{\partial u(x,t)}{\partial t} &= \lambda_\alpha {}_0^C D_x^\alpha u(x,t) + f(x,t) \\ &= \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_0^x (x-\tau)^{-\alpha} \frac{\partial u(\tau,t)}{\partial \tau} d\tau + f(x,t), \end{aligned} \tag{3} $$

if we take, $x = x_{n+1}, \quad t = t_j$, we have

$$ \begin{aligned} \frac{\partial u(x_{n+1},t_j)}{\partial t} &= \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_0^{x_{n+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial u(\tau,t_j)}{\partial \tau} d\tau + f(x_{n+1},t_j) \\ &= \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_0^{x_n} (x_{n+1}-\tau)^{-\alpha} \frac{\partial u(\tau,t_j)}{\partial \tau} d\tau \\ &\quad + \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_{x_n}^{x_{n+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial u(\tau,t_j)}{\partial \tau} d\tau + f(x_{n+1},t_j) \\ &= I_1 + I_2 + f(x_{n+1},t_j). \end{aligned} \tag{4} $$

The integral I_2 approximate by the piecewise linear interpolation at the nodes x_n and x_{n+1} for u, by the following approach

$$I_2 = \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_{x_n}^{x_{n+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial u(\tau,t_j)}{\partial \tau} d\tau \approx \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_{x_n}^{x_{n+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial \hat{u}(\tau,t_j)}{\partial \tau} d\tau$$
$$= \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_{x_n}^{x_{n+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial}{\partial \tau}\left[\frac{\tau - x_{n+1}}{x_n - x_{n+1}} u_n^j + \frac{\tau - x_n}{x_{n+1}-x_n} u_{n+1}^j\right] d\tau$$
$$= \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)}\left[u_{n+1}^j - u_n^j\right], \tag{5}$$

where $\hat{u}$ is the piecewise linear interpolation for u and $u_n^j = u(x_n, t_j)$. Also, the integral I_1 approximate by the piecewise linear interpolation at the nodes x_k and x_{k+1} with $k = 0, 1, ..., n-1$ for u, by the following approach

$$I_1 = \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_0^{x_n} (x_{n+1}-\tau)^{-\alpha} \frac{\partial u(\tau,t_j)}{\partial \tau} d\tau$$
$$\approx \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_0^{x_n} (x_{n+1}-\tau)^{-\alpha} \frac{\partial \widehat{u}(\tau,t_j)}{\partial \tau} d\tau = \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial \hat{u}(\tau,t_j)}{\partial \tau} d\tau$$
$$= \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial}{\partial \tau}\left[\frac{\tau - x_{k+1}}{x_k - x_{k+1}} u(x_k,t_j) + \frac{\tau - x_k}{x_{k+1}-x_k} u(x_{k+1},t_j)\right] d\tau$$
$$= \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n-1} \left[u_k^j - u_{k+1}^j\right] \frac{(n+1-[k+1])^{1-\alpha} - (n+1-k)^{1-\alpha}}{([k+1]-k)}$$
$$= \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n} \left[\rho_{k,n+1}^R + \rho_{k,n+1}^L\right] u_k^j = \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n-1} \left[\rho_{k,n+1}^R u_k^j + \rho_{k+1,n+1}^L u_{k+1}^j\right]$$
$$= \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n-1} \rho_{k,n+1}^R \left[u_k^j - u_{k+1}^j\right] = \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n} \rho_{k,n+1} u_k^j, \tag{6}$$

where $\hat{u}$ is the piecewise linear interpolation for u and

$$\rho_{k,n+1}^R = \begin{cases} \frac{(n+1-[k+1])^{1-\alpha} - (n+1-k)^{1-\alpha}}{([k+1]-k)}, & 0 \le k \le n-1, \\ 0, \quad k = n, \end{cases}$$
$$\rho_{k,n+1}^L = \begin{cases} 0, \quad k = 0, \\ \frac{(n+1-[k-1])^{1-\alpha} - (n+1-k)^{1-\alpha}}{(k-[k-1])}, & 1 \le k \le n. \end{cases} \tag{7}$$

By using Eqs. (6) and (7), we can write

$$\rho_{k,n+1} = \rho_{k,n+1}^R + \rho_{k,n+1}^L, \quad \rho_{k+1,n+1}^L = -\rho_{k,n+1}^R. \tag{8}$$

Suppose, we take

$$\begin{aligned}\delta_{n+1}^{j} &= \lambda_{\alpha 0}^{C}D_{x}^{\alpha}u(x_{n+1},t_{j})+f(x_{n+1},t_{j})\\ &= \frac{\lambda_{\alpha}}{\Gamma(1-\alpha)}\int_{0}^{x_{n+1}}(x_{n+1}-\tau)^{-\alpha}\frac{\partial u(\tau,t_{j})}{\partial\tau}d\tau+f(x_{n+1},t_{j})\\ &= \frac{\lambda_{\alpha}h^{-\alpha}}{\Gamma(2-\alpha)}\left[u_{n+1}^{j}-u_{n}^{j}\right]+\frac{\lambda_{\alpha}h^{-\alpha}}{\Gamma(2-\alpha)}\sum_{k=0}^{n}\rho_{k,n+1}u_{k}^{j}+f_{n+1}^{j}.\end{aligned} \tag{9}$$

Thus, we approximate solution by using the Crank–Nicolson scheme for Eq. (1). So we apply numerical method to Eq. (1) as follows.

Let $u(x_n,t_j)=u_n^j,\quad f(x_n,t_j)=f_n^j$. Then,

$$\frac{u_{n+1}^{j}-u_{n+1}^{j-1}}{\kappa}=\frac{1}{2}\left[\delta_{n+1}^{j}+\delta_{n+1}^{j-1}\right]. \tag{10}$$

Therefore, after some calculations for Eq. (10) by using (9), we have

$$\begin{aligned}&u_{n+1}^{j}-\frac{\lambda_{\alpha}\kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\left[u_{n+1}^{j}-u_{n}^{j}\right]-\frac{\lambda_{\alpha}\kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\sum_{k=0}^{n}\rho_{k,n+1}u_{k}^{j}\\ &=u_{n+1}^{j-1}+\frac{\lambda_{\alpha}\kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\left[u_{n+1}^{j-1}-u_{n}^{j-1}\right]+\frac{\lambda_{\alpha}\kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\sum_{k=0}^{n}\rho_{k,n+1}u_{k}^{j-1}+\frac{\kappa(f_{n+1}^{j}+f_{n+1}^{j-1})}{2},\end{aligned} \tag{11}$$

finally, we can write

$$u_{n+1}^{j}+\sum_{k=0}^{n+1}\Psi_{k,n+1}^{\alpha}u_{k}^{j}=u_{n+1}^{j-1}-\sum_{k=0}^{n+1}\Psi_{k,n+1}^{\alpha}u_{k}^{j-1}+\frac{\kappa(f_{n+1}^{j}+f_{n+1}^{j-1})}{2}, \tag{12}$$

where

$$\psi_{k,n+1}^{\alpha}=\begin{cases}\dfrac{-\lambda_{\alpha}\kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\left[\rho_{k,n+1}^{R}-\rho_{k-1,n+1}^{R}\right], & k=1,2,...,n,\\ \dfrac{-\lambda_{\alpha}\kappa h^{-\alpha}}{2\Gamma(2-\alpha)}, & k=n+1.\end{cases} \tag{13}$$

By using Eq. (12) and (13), introducing

$$D=\begin{pmatrix}\psi_{1,1}^{\alpha} & 0 & 0 & \cdots & 0\\ \psi_{1,2}^{\alpha} & \psi_{2,2}^{\alpha} & 0 & \cdots & 0\\ \psi_{1,3}^{\alpha} & \psi_{2,3}^{\alpha} & \psi_{3,3}^{\alpha} & & 0\\ \vdots & \vdots & \vdots & & \vdots\\ \psi_{1,M-1}^{\alpha} & \psi_{2,M-1}^{\alpha} & \psi_{3,M-1}^{\alpha} & & 0\\ \psi_{1,M}^{\alpha} & \psi_{2,M}^{\alpha} & \psi_{3,M}^{\alpha} & \cdots & \psi_{M,M}^{\alpha}\end{pmatrix}, \tag{14}$$

and

$$U^j = [u_1^j, u_2^j, ..., u_M^j]^T, \tag{15}$$

Eq. (12) takes the matrix-form as:

$$(I + D)U^j = (I - D)U^{j-1} + F^j, \tag{16}$$

where

$$F^j = \begin{bmatrix} \frac{\kappa}{2}\left[f_1^j + f_1^{j-1}\right] & -\Psi_{0,1}^{\alpha} & \left[u_0^j + u_0^{j-1}\right] \\ \frac{\kappa}{2}\left[f_2^j + f_2^{j-1}\right] & -\Psi_{0,2}^{\alpha} & \left[u_0^j + u_0^{j-1}\right] \\ \frac{\kappa}{2}\left[f_3^j + f_3^{j-1}\right] & -\Psi_{0,3}^{\alpha} & \left[u_0^j + u_0^{j-1}\right] \\ \vdots & & \\ \frac{\kappa}{2}\left[f_{M-1}^j + f_{M-1}^{j-1}\right] & -\Psi_{0,M-1}^{\alpha} & \left[u_0^j + u_0^{j-1}\right] \\ \frac{\kappa}{2}\left[f_M^j + f_M^{j-1}\right] & -\Psi_{0,M}^{\alpha} & \left[u_0^j + u_0^{j-1}\right] \end{bmatrix}.$$

3 Numerical method with non-uniform meshes

In Sect. 2, we designed the proposed scheme with uniform meshes (12–13), to approximate the integral $\int_0^{x_n} d\tau$ by

$$\begin{aligned} \frac{\partial u(x_{n+1}, t_j)}{\partial t} =& \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_0^{x_n} (x_{n+1} - \tau)^{-\alpha} \frac{\partial u(\tau, t_j)}{\partial \tau} d\tau \\ &+ \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_{x_n}^{x_{n+1}} (x_{n+1} - \tau)^{-\alpha} \frac{\partial u(\tau, t_j)}{\partial \tau} d\tau + f(x_{n+1}, t_j) \end{aligned} \tag{17}$$

Since $(x_{n+1} - \tau)^{-\alpha}$ decays with power α, we can actually select lesser number of mesh points of $[0, L]$, as $0 = \sigma_{0,n} < \sigma_{1,n} < \sigma_{2,n} < ... < \sigma_{m_n,n} = x_n$ to approximate the integral $\int_0^{x_n} d\tau$.

3.1 Algorithms for selecting the equidistributing meshes

For selecting the equidistributing meshes, we introduce two algorithms in this subsection [51].

Algorithm 1: *Equal − height distribution algorithm* [51]

Assume that we have already got the points $\sigma_{i,n}$, we have two principles for selecting the next point $\sigma_{i+1,n}$. By this two principles, the numerical method does not lose the accuracy but reduce the computation cost.

Principle 1: The next point $\sigma_{i+1,n}$ is at least one step away from $\sigma_{i,n}$. The function values $u(\tau) = (x_{n+1} - \tau)^{-\alpha}$ are as equally distributed as possible, i.e.,

$$\bar{\sigma}_{i+1,n} = \max \left\{ \begin{array}{l} \text{solve}(\bar{\sigma}_{i+1,n} - \sigma_{i,n} = h,\ \bar{\sigma}_{i+1,n}), \\ \text{solve}(u(\bar{\sigma}_{i+1,n}) - u(\sigma_{i,n}) = \Delta u,\ \bar{\sigma}_{i+1,n}) \end{array} \right\}, \tag{18}$$

where Δu is a given small positive real number and solve(equ, var) *means the solution of equ with unknown variable var, e.g.*, solve$(u(\bar{\sigma}_{i+1,n}) - u(\sigma_{i,n}) = \Delta u, \bar{\sigma}_{i+1,n})$ *means solving*

$$(x_{n+1} - \bar{\sigma}_{i+1,n})^{-\alpha} - (x_{n+1} - \sigma_{i+1,n})^{-\alpha} = \Delta u. \tag{19}$$

Therefore, we have

$$\bar{\sigma}_{i+1,n} = x_{n+1} - [(x_{n+1} - \sigma_{i,n})^{-\alpha} + \Delta u]^{\frac{-1}{\alpha}}; \tag{20}$$

Principle 2: *To avoid involving non-equally divided nodes, we take*

$$\sigma_{i+1,n} = \left\lfloor \frac{\bar{\sigma}_{i+1,n}}{h} \right\rfloor * h. \tag{21}$$

therefore, we have $\sigma_{i+1,n} = \sigma_{i,n} + h$ *or*

$$(x_{n+1} - \sigma_{i+1,n})^{-\alpha} - (x_{n+1} - \sigma_{i,n})^{-\alpha} \leq \Delta u. \tag{22}$$

This algorithm is called equal – height distribution algorithm [51] (*see Algorithm* 1).

Algorithm 2: *Equal – area distribution algorithm* [51]

Principle 1: For design second algorithm to choosing the mesh points $\sigma_{i,n}$, we integrate of $u(\tau) = (x_{n+1} - \tau)^{-\alpha}$ as

$$\int_{\sigma_{i,n}}^{\bar{\sigma}_{i+1,n}} (x_{n+1} - \tau)^{-\alpha} d\tau = \Delta S,$$

where ΔS is a given small positive real number. For $\bar{\sigma}_{i+1,n}$, we approximate it by

$$\bar{\sigma}_{i+1,n} = x_{n+1} - [(x_{n+1} - \sigma_{i,n})^{1-\alpha} - (1-\alpha)\Delta S]^{\frac{1}{1-\alpha}}; \tag{23}$$

Principle 2: To avoid involving non-equally divided nodes, we take

$$\sigma_{i+1,n} = \left\lfloor \frac{\bar{\sigma}_{i+1,n}}{h} \right\rfloor * h, \tag{24}$$

therefore, $\sigma_{i+1,n}$ belongs to the uniform nodes $\{x_i\}_{i=0}^{n}$. It can be checked that

$$(x_{n+1} - \sigma_{i,k})^{1-\alpha} - (x_{n+1} - \sigma_{i+1,n})^{1-\alpha} \leq (1-\alpha)\Delta S, \quad or \quad \sigma_{i+1,n} = \sigma_{i,n} + h \tag{25}$$

This algorithm is called *equal – area distribution algorithm* [51] (see Algorithm 2).

Algorithm 1 The equal-height distribution algorithm for (18-22)

Function GENXI $(n, h, \alpha, \Delta u)$

$i=0;\ \sigma_{i,n}=0;$ ⁕ In this stage we have $\sigma_{0,n}=0$
$\sigma_c=0;$ ⁕ In this stage, current node is $\sigma_{0,n}$
while $x_c < x_n$ **do**
  $\sigma_{i+1,n}=x_{n+1}-[(x_{n+1}-\sigma_c)^{-\alpha}+\Delta u]^{\frac{-1}{\alpha}};$
  if $\sigma_{i+1,n}>x_n$ **then**
    $\sigma_{i+1,n}=x_n;$
    break;
  end if
  ⁕ if $u=(x_{n+1}-\tau)^{-\alpha}$ changes too fast, go to the following stage
  if $\sigma_{i+1,n}-\sigma_c<h$ **then**
    $\sigma_{i+1,n}=\sigma_c+h;$ ⁕ make $\sigma_{i+1,n}$ be one step away from $\sigma_{i,n}$
  else
    $\sigma_{i+1,k}=\left\lfloor\frac{\sigma_{i+1,n}}{h}\right\rfloor*h;$ ⁕ let $\sigma_{i+1,n}$ belong to the uniform mesh $\{t_j\}_{j=0}^{n}$
  end if
  $\sigma_c=\sigma_{i+1,n}; i=i+1;$
end while
End Function

Algorithm 2 The equal-area distribution algorithm for (23-25)

Function GENXII $(n, h, \alpha, \Delta S)$

$i=0;\sigma_{i,n}=0;$ ⁕ In this stage we have $\sigma_{0,n}=0$
$\sigma_c=0;$ ⁕ In this stage, current node is $\sigma_{0,n}$
while $x_c < x_n$ **do**
  $\sigma_{i+1,n}=x_{n+1}-[(x_{n+1}-\sigma_c)^{1-\alpha}-(1-\alpha)\Delta S]^{\frac{1}{1-\alpha}};$
  if $\sigma_{i+1,n}>x_n$ **then**
    $\sigma_{i+1,n}=x_n;$
    break;
  end if
  ⁕ if $u=(x_{n+1}-\tau)^{-\alpha}$ changes too fast, go to the following stage
  if $\sigma_{i+1,n}-\sigma_c<h$ **then**
    $\sigma_{i+1,n}=\sigma_c+h;$ ⁕ make $\sigma_{i+1,n}$ be one step away from $\sigma_{i,n}$
  else
    $\sigma_{i+1,k}=\left\lfloor\frac{\sigma_{i+1,n}}{h}\right\rfloor*h;$ ⁕ let $\sigma_{i+1,n}$ belong to the uniform mesh $\{t_j\}_{j=0}^{n}$
  end if
  $\sigma_c=\sigma_{i+1,n}; i=i+1;$
end while
End Function

3.2 Formulation of numerical method with equidistributing meshes

In the second section, we partition the interval [0, L] into a uniform mesh. The non-uniform mesh points $\sigma_{i,n}$ chosen from Algorithm 1 or 2 still belong to the set of the uniform meshes. Also, we take $x_{n_0} = 0$ and $x_{n_{m_n}} = x_n$. Thus, $\sigma_{i,n} = x_{n_i}, \quad i = 0, 1, ..., m_n$. Now, we assume that

$$\begin{aligned} x &= \left\{x_0, x_1, x_2, ..., x_n\right\}, \\ \sigma(i) &= \left\{\sigma_{0,n}, \sigma_{1,n}, \sigma_{2,n}, ..., \sigma_{m_i,n}\right\}. \end{aligned} \tag{26}$$

To design a new numerical method with the non-uniform mesh points, we have

$$\begin{aligned} \frac{\partial u(x_{n+1}, t_j)}{\partial t} =& \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_0^{x_{n+1}} (x_{n+1} - \tau)^{-\alpha} \frac{\partial u(\tau, t_j)}{\partial \tau} d\tau + f(x_{n+1}, t_j) \\ =& \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_0^{x_n} (x_{n+1} - \tau)^{-\alpha} \frac{\partial u(\tau, t_j)}{\partial \tau} d\tau \\ &+ \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_{x_n}^{x_{n+1}} (x_{n+1} - \tau)^{-\alpha} \frac{\partial u(\tau, t_j)}{\partial \tau} d\tau + f(x_{n+1}, t_j) \\ =& \hat{I}_1 + I_2 + f(x_{n+1}, t_j). \end{aligned} \tag{27}$$

We approximate $\hat{I}_1$ as

$$\begin{aligned} \hat{I}_1 &= \frac{\lambda_\alpha}{\varGamma(1-\alpha)} \int_0^{x_n} (x_{n+1} - \tau)^{-\alpha} \frac{\partial u(\tau, t_j)}{\partial \tau} d\tau \approx \frac{\lambda_\alpha}{\varGamma(1-\alpha)} \int_0^{x_n} (x_{n+1} - \tau)^{-\alpha} \frac{\partial \bar{u}(\tau, t_j)}{\partial \tau} d\tau \\ &= \frac{\lambda_\alpha}{\varGamma(1-\alpha)} \sum_{i=0}^{m_n-1} \int_{x_{n_i}}^{x_{n_{i+1}}} (x_{n+1} - \tau)^{-\alpha} \frac{\partial \bar{u}(\tau, t_j)}{\partial \tau} d\tau \\ &= \frac{\lambda_\alpha}{\varGamma(1-\alpha)} \sum_{i=0}^{m_n-1} \int_{x_{n_i}}^{x_{n_{i+1}}} (x_{n+1} - \tau)^{-\alpha} \frac{\partial}{\partial \tau} \left[\frac{\tau - x_{n_{i+1}}}{x_{n_i} - x_{n_{i+1}}} u^j_{n_i} + \frac{\tau - x_{n_i}}{x_{n_{i+1}} - x_{n_i}} u^j_{n_{i+1}} \right] d\tau \\ &= \frac{\lambda_\alpha h^{-\alpha}}{\varGamma(2-\alpha)} \sum_{i=0}^{m_n-1} \left[u^j_{n_{i+1}} - u^j_{n_i} \right] \frac{(n+1-n_i)^{1-\alpha} - (n+1-n_{i+1})^{1-\alpha}}{(n_{i+1} - n_i)} \\ &= \frac{\lambda_\alpha h^{-\alpha}}{\varGamma(2-\alpha)} \sum_{i=0}^{m_n-1} \left[\theta^R_{i,n+1} + \theta^L_{i,n+1} \right] u^j_{n_i} = \frac{\lambda_\alpha h^{-\alpha}}{\varGamma(2-\alpha)} \sum_{i=0}^{m_n-1} \left[\theta^R_{i,n+1} u^j_{n_i} + \theta^L_{i+1,n+1} u^j_{n_{i+1}} \right] \\ &= \frac{\lambda_\alpha h^{-\alpha}}{\varGamma(2-\alpha)} \sum_{i=0}^{m_n-1} \theta^R_{i,n+1} \left[u^j_{n_i} - u^j_{n_{i+1}} \right] = \frac{\lambda_\alpha h^{-\alpha}}{\varGamma(2-\alpha)} \sum_{i=0}^{m_n} \theta_{i,n+1} u^j_{n_i}, \end{aligned} \tag{28}$$

where $\bar{u}$ is the piecewise linear interpolation for u at the nodes x_{n_i} and $x_{n_{i+1}}$ with $i = 0, 1, ..., m_n - 1$, and

$$\theta^R_{i,n+1} = \begin{cases} \dfrac{(n+1-n_{i+1})^{1-\alpha} - (n+1-n_i)^{1-\alpha}}{(n_{i+1}-n_i)}, 0 \le i \le m_n - 1, \\ 0, \qquad i = m_n, \end{cases}$$
$$\theta^L_{i,n+1} = \begin{cases} 0, \qquad i = 0, \\ -\dfrac{(n+1-n_i)^{1-\alpha} - (n+1-n_{i-1})^{1-\alpha}}{(n_i - n_{i-1})}, 1 \le i \le m_n. \end{cases} \tag{29}$$

Also, the integral I_2 is approximated by Eq. (5). By using Eqs. (28) and (29), we have

$$\theta_{i,n+1} = \theta^R_{i,n+1} + \theta^L_{i,n+1}, \theta^L_{i+1,n+1} = -\theta^R_{i,n+1}. \tag{30}$$

Remark 1 If we take, $n_k = k, \ \ 0 \le k \le n$ (for uniform meshes), we can write

$$\begin{aligned} \sum_{k=0}^{n} \rho_{k,n+1} u^j_k &= \sum_{k=0}^{n-1} \left[\rho^R_{k,n+1} u^j_k + \rho^L_{k+1,n+1} u^j_{k+1} \right] \\ &= \sum_{i=0}^{m_n-1} \sum_{k=n_i}^{n_{i+1}-1} \rho^R_{k,n+1} \left[u^j_k - u^j_{k+1} \right]. \end{aligned} \tag{31}$$

For non-uniform meshes case, we take

$$\begin{aligned} \gamma^j_{n+1} =& \lambda_{\alpha 0}^{C} D^\alpha_x u(x_{n+1}, t_j) + f(x_{n+1}, t_j) \\ =& \frac{\lambda_\alpha}{\Gamma(1-\alpha)} \int_0^{xn+1} (x_{n+1} - \tau)^{-\alpha} \frac{\partial u(\tau, t_j)}{\partial \tau} d\tau + f(x_{n+1}, t_j) \\ =& \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)} \left[u^j_{n+1} - u^j_n \right] + \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{i=0}^{mn-1} \theta_{i,n+1} u^j_{n_i} + f^j_{n+1}. \end{aligned} \tag{32}$$

Let we take, $u(x_n, t_j) = u^j_n$, $u(x_{n_i}, t_j) = u^j_{n_i}$ and $f(x_n, t_j) = f^j_n$. Then, by using the Crank–Nicolson scheme for Eq. (1), numerical method for Eq. (1) is as the following form.

$$\frac{u_{n+1}^{j}-u_{n+1}^{j-1}}{\kappa}=\frac{1}{2}[\gamma_{n+1}^{j}+\gamma_{n+1}^{j-1}]. \tag{33}$$

Therefore, Eq. (33) by using (32) will be as the following form

$$\begin{aligned} & u_{n+1}^{j}-\frac{\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\left[u_{n+1}^{j}-u_{n}^{j}\right]-\frac{\lambda_\alpha h^{-\alpha}}{2\Gamma(2-\alpha)}\sum_{i=0}^{m_n}\theta_{i,n+1}u_{n_i}^{j} \\ & =u_{n+1}^{j-1}+\frac{\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\left[u_{n+1}^{j-1}-u_{n}^{j-1}\right]+\frac{\lambda_\alpha h^{-\alpha}}{2\Gamma(2-\alpha)}\sum_{i=0}^{m_n}\theta_{i,n+1}u_{n_i}^{j-1} \\ & +\frac{\kappa(f_{n+1}^{j}+f_{n+1}^{j-1})}{2}, \end{aligned} \tag{34}$$

after some calculations, we have

$$u_{n+1}^{j}+\sum_{i=0}^{m_n+1}\Phi_{n_i,n+1}^{\alpha}u_{n_i}^{j}=u_{n+1}^{j-1}-\sum_{i=0}^{m_n+1}\Phi_{n_i,n+1}^{\alpha}u_{n_i}^{j-1}+\frac{\kappa(f_{n+1}^{j}+f_{n+1}^{j-1})}{2}, \tag{35}$$

where

$$\Phi_{n_i,n+1}^{\alpha}=\begin{cases}\frac{-\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\left[\theta_{i,n+1}^{R}-\theta_{i-1,n+1}^{R}\right], & i=1,2,...,m_n,\\ \frac{-\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)}, & i=m_n+1.\end{cases}$$

If we take $U^j=[u_1^j,u_2^j,...,u_M^j]^T$ therefore, Eq. (35) takes the matrix-form as:

$$(I+\hat{D})U^j=(I-\hat{D})U^{j-1}+G^j, \tag{36}$$

where

$$G^j=\begin{bmatrix}\frac{\kappa}{2}\left[f_1^j+f_1^{j-1}\right]-\Phi_{n_0,1}^{\alpha}\left[u_{n_0}^{j}+u_{n_0}^{j-1}\right]\\ \frac{\kappa}{2}\left[f_2^j+f_2^{j-1}\right]-\Phi_{n_0,2}^{\alpha}\left[u_{n_0}^{j}+u_{n_0}^{j-1}\right]\\ \frac{\kappa}{2}\left[f_3^j+f_3^{j-1}\right]-\Phi_{n_0,3}^{\alpha}\left[u_{n_0}^{j}+u_{n_0}^{j-1}\right]\\ \vdots\\ \frac{\kappa}{2}\left[f_{M-1}^j+f_{M-1}^{j-1}\right]-\Phi_{n_0,M-1}^{\alpha}\left[u_{n_0}^{j}+u_{n_0}^{j-1}\right]\\ \frac{\kappa}{2}\left[f_M^j+f_M^{j-1}\right]-\Phi_{n_0,M}^{\alpha}\left[u_{n_0}^{j}+u_{n_0}^{j-1}\right]\end{bmatrix}$$

and matrix $\hat{D}$ will be introduced in the next subsection.

3.3 An algorithm for generating the matrix $\hat{D}$

In this subsection, we design an algorithm for generate the matrix $\hat{D}$ by using the Algorithm 1 or 2.

Algorithm 3: *Matrix Generation′s Algorithm* 3

We use the function GENXI or GENXII to generate the matrix $\hat{D}$ by using the non-uniform mesh points on [0, L] chosen from Algorithm 1 (equal-height distribution algorithm) or 2 (area-height distribution algorithm). We design an algorithm for generating the matrix $\hat{D}$, as the following process:

Step 1 We partition [0, L] into a uniform mesh with the space step size $h = L/M$ and the time step size $t = T/M$, where M is a positive integer. Also we have, $x_n = nh$ for $n = 1, ..., M$ and $t_j = j\kappa$ for $j = 1, ..., N$.

Step 2 In this stage, we use the function GENXI or GENXII to selecting non-uniform mesh points on [0, L] by Algorithm 1 (equal-height distribution algorithm) or 2 (area-height distribution algorithm). We consider these non-uniform meshes as a vector and call it X as:

$$X = [n_0, n_1, n_2, ..., n_{m_n}].$$

In the partition [0, L] into a uniform mesh, we replace zero instead of unused points. We consider these meshes as a vector and call it $\bar{X}$ as:

$$\bar{X} = [0, .., 0, n_1, 0, .., 0, n_2, 0, ..., 0, ..., 0, ..., 0, n_{m_n}].$$

Step 3 In this stage, we look for the coefficients of u_i^j, which are the matrix elements. If the i-th element of the vector X is zero, this coefficient will be zero. And if the i-th element is nonzero, the coefficient is obtained from the following relation:

$$\hat{D}_{n,i-1} = \frac{-\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)} \left[\frac{(n - X(i))^{1-\alpha} - (n - X(i+1))^{1-\alpha}}{(X(i+1) - X(i))} - \frac{(n - X(i+1))^{1-\alpha} - (n - X(i+2))^{1-\alpha}}{(X(i+2) - X(i+1))} \right]; \quad 2 \le i \le \text{length}(\bar{X})$$

also, $\hat{D}_{2,1}=\frac{-\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\left[2^{1-\alpha} - 2\right]$ and $\hat{D}_{i,i}=\frac{-\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)}$, $i = 1, ..., n$.

With these three steps, all the matrix elements will be obtained (see Algorithm 3).

Algorithm 3 Matrix Generation Algorithm for non-uniform meshes' method

Input: α, λ_α, h, m_n, M and Δu or ΔS.

Output: Matrix $\hat{D}$ for proposed method by using non-uniform meshes.

for $n = m_n$ to M **do**
  X= Function GENXI (n,h,α,Δu); ⁎ GENXI is equal-height algorithm
  OR
  X= Function GENXII (n,h,α,ΔS); ⁎ GENXII is area-height algorithm
  $\bar{X}$=[0,0,0,...,0]; ⁎ $\bar{X}$ is a zero's vector with n-dimension
  for $k = 2$ to $length(X)$ **do**
    $\bar{X}(X(k)+1)$=$X(k)$;⁎$\bar{X}$=[0,...,0,n_1,0,...,0,n_2,0,...,0,...,0,...,0,$n_{m_{n-1}}$,0,...,0,n_{m_n}]
  end for
  $\hat{D}_{2,1}$=$\omega_{2,1} = \dfrac{-\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\left[2^{1-\alpha}-2\right]$; ⁎ $\omega_{n+1,k} = \Phi^{\alpha}_{k,n+1}$
  for $k = 1$ to n **do**
    $\hat{D}_{k,k}$=$\omega_{k,k} = \dfrac{-\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)}$; ⁎ The diagonal elements of Matrix
  end for
  $j = 1$;
  for $l = 2$ to length($\bar{X}$) **do**
    if $\bar{X}(l) = 0$ **then**
      $\hat{D}_{n,l-1}$=$\omega_{n,l-1} = 0$; ,⁎ The other elements of Matrix
    else

$$\hat{D}_{n,l-1} = \omega_{n,l-1} = \frac{-\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)}\left[\frac{(n-X(l))^{1-\alpha}-(n-X(l+1))^{1-\alpha}}{(X(l+1)-X(l))} - \frac{(n-X(l+1))^{1-\alpha}-(n-X(l+2))^{1-\alpha}}{(X(l+2)-X(l+1))}\right];$$

    end if
    l=l+1;
  end for
end for

Remark 2 For Computing the total times of the nodes (N) being used in the our methods, we design Algorithm 4. For example, the total times of the nodes (N) which used in proposed method with uniform meshes for solving PFDEs compute form $N = n(n + \frac{n(n+1)}{2})$. So N is 650, 4600, 34400 and 265600, respectively, when $h = \kappa = 1/10, 1/20, 1/40, 1/80$ and $T = 1$, $L = 1$. So, the computation cost of numerical method with uniform meshes for solving the PFDEs is increasing.

Algorithm 4 The Algorithm for computing the total times of nodes(N)

$k=3$; ⋇ Three starting points are known.
$p=3$;
while $k<n+1$ **do**
  $N=n$(p+length($\sigma(k)$)); ⋇ Sums up all the points used before $\sigma(k)$.
  p=N;k=k+1;
end while

4 Error analysis of methods

In this section, we study error analysis of methods with uniform meshes and non-uniform meshes. So, let A be a matrix $d\times d$ and $\|.\|$ be a norm in C^d. Let $\lambda_1,\ \lambda_2,...,\lambda_{\rm d}$ be the eigenvalues of a matrix A. Then, its spectral radius will be as:

$$\rho(A)=\max\left\{|\lambda_1|,\ |\lambda_2|,...,|\lambda_{\rm d}|\right\}.$$

Lemma 1 [53] (**Gelfand's Formula**) *Given any matrix norm* $\|.\|$ *on* C^d

$$\rho(A)=\lim_{n\to\infty}\|A^n\|^{\frac{1}{n}}. \tag{37}$$

if $A_1,A_2,...,A_n$ *are matrices that all commute, by using Gelfand's formula, we can write*

$$\rho(A_1A_2...A_n)\le\rho(A_1)\rho(A_2)...\rho(A_n), \tag{38}$$

because

$$\begin{aligned}\rho(A1A2\cdots As)=&\lim_{n\to\infty}\|(A1A2\cdots As)^n\|^{\frac{1}{n}}=\lim_{n\to\infty}\left\|(A1^nA2^n\cdots As^n)^n\right\|^{\frac{1}{n}}\\ &\le\lim_{n\to\infty}\|A1^n\|^{\frac{1}{n}}\lim_{n\to\infty}\|A2^n\|^{\frac{1}{n}}\cdots\lim_{n\to\infty}\|As^n\|^{\frac{1}{n}}\\ =&\rho(A1)\rho(A2)\cdots\rho(As)\end{aligned} \tag{39}$$

Theorem 1 *The proposed method with uniform meshes is obtained as the following form,*

$$(I+D)U^j=(I-D)U^{j-1}+F^j, \tag{40}$$

for every initial vector U^0*, is stable.*

Proof Since all eigenvalues of matrix D are nonzero, thus the matrix $(I+D)^{-1}$ is invertible. We can write

$$U^j=(I+D)^{-1}(I-D)U^{j-1}+(I+D)^{-1}F^j.$$

If we take

$$A = (I + D)^{-1}(I - D), \quad B = (I + D)^{-1},$$

therefore, we have

$$U^j = AU^{j-1} + B.$$

suppose v_i, $i = 1, 2, ..., M$, be eigenvalues of matrix D. Since we have for matrix D, $v_i = \dfrac{-\lambda_\alpha \kappa h^{-\alpha}}{2\Gamma(2-\alpha)} > 0$, $i = 1, 2, ..., M$. We can write

$$\rho(I - D) < 1, \quad \rho((I + D)^{-1}) < 1. \tag{41}$$

Also, we can write

$$\begin{aligned} I &= (I + D)(I + D)^{-1} \\ &= (D^{-1}D + D^{-1}DD)(I + D)^{-1} \\ &= (D^{-1} + D^{-1}D)D(I + D)^{-1} \\ &= D^{-1}(I + D)D(I + D)^{-1} \end{aligned}$$

$$\begin{aligned} &D^{-1}(I + D)(I + D)^{-1}D = D^{-1}(I + D)D(I + D)^{-1} \\ &-(I + D)^{-1}D = -D(I + D)^{-1} \\ &(I + D)^{-1} - (I + D)^{-1}D = (I + D)^{-1} - D(I + D)^{-1} \\ &(I + D)^{-1}(I - D) = (I - D)(I + D)^{-1}, \end{aligned}$$

thus, $(I + D)^{-1}$ and $(I - D)$ are commutative matrices. Therefore, by using Lemma 1 and (41), we have

$$\begin{aligned} \rho(A) &= \rho((I + D)^{-1}(I - D)) \\ &\leq \rho((I + D)^{-1})\rho((I - D)) \\ &< 1. \end{aligned}$$

Thus, the proposed method with uniform meshes (12) is stable. □

Lemma 2 *Let* $u \in C^2[0, L]$ *and* $0 < \alpha < 1$, *then*

$$\left| \int_0^{x_{n+1}} (x_{n+1} - \tau)^{-\alpha} \frac{\partial}{\partial \tau} \left[u(\tau, t_j) - \hat{u}(\tau, t_j) \right] d\tau \right| < Ch.$$

Proof By using the Taylor theorem, for $\tau \in [x_i, x_{i+1}]$, there exist $\xi_i \in [x_i, x_{i+1}]$. Therefore,

$$
\begin{aligned}
&\left|\int_0^{x_{n+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial}{\partial \tau}\left[u(\tau,t_j)-\hat{u}(\tau,t_j)\right] d\tau\right| \\
&\le \sum_{i=0}^{n} \int_{x_i}^{x_{i+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial}{\partial \tau}\left|u(\tau,t_j)-\hat{u}(\tau,t_j)\right| d\tau \\
&= \sum_{i=0}^{n} \int_{x_i}^{x_{i+1}} (x_{n+1}-\tau)^{-\alpha} \frac{\partial}{\partial \tau}\left|(\tau-x_i)(\tau-x_{i+1}) \frac{\partial^2 u}{2\partial \tau^2}\bigg|_{\tau=\xi_i}\right| d\tau \\
&\le \frac{M_2}{2} \sum_{i=0}^{n} \int_{x_i}^{x_{i+1}} (x_{n+1}-\tau)^{-\alpha} \left|(2\tau - x_i - x_{i+1})\right| d\tau \\
&\le \frac{M_2}{2} \sum_{i=0}^{n} \int_{x_i}^{x_{i+1}} (x_{n+1}-\tau)^{-\alpha} \left|x_{i+1}-x_i\right| d\tau \\
&= M_2 h \sum_{i=0}^{n} \frac{(x_{n+1}-x_i)^{1-\alpha}-(x_{n+1}-x_{i+1})^{1-\alpha}}{(1-\alpha)} = M_2 h\left[\frac{(x_{n+1})^{1-\alpha}}{(1-\alpha)}\right] \le Ch,
\end{aligned}
$$

where $M_2 = \sup_{z\in[0,L]} \left|\frac{\partial^2 u(\tau,t)}{\partial \tau^2}\right|_{\tau=z}\Bigg|$.

Lemma 3 [54] *Let S be a positive definite matrix of order $m-1$. Then, for any parameter $\eta \ge 0$, the following inequalities hold*:

$$
\left\|(I+\eta S)^{-1}(I-\eta S)\right\|_\infty \le 1.
$$

By using Lemma 2, *we study convergence of the method. So, for the method* (16), *we can write*

$$
\begin{aligned}
\frac{u_i^j-u_i^{j-1}}{\kappa} &= \frac{\lambda_\alpha}{2}\left({}_0^C D_x^\alpha u(x_{n+1},t_j) +{}_0^C D_x^\alpha u(x_{n+1},t_{j-1})\right) + \mathrm{O}(\kappa^2),\\
\lambda_{\alpha 0}^{C} D_x^\alpha u(x_{n+1},t_j) &= \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)}\left[u_{n+1}^j - u_n^j\right] + \frac{\lambda_\alpha h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n-1} \rho_{k,n+1} u_k^j + \mathrm{O}(h).
\end{aligned}
\tag{42}
$$

Thus, the local truncation error of (12) *can be written as*:

$$
T_{i,j} = \mathrm{O}(\kappa^3 + \kappa h).
$$

Theorem 2 *Let* U^j *and* u^j *be the numerical solution and exact solution of* (12), *respectively. Then, we have*

$$\left\| U^j - u^j \right\|_\infty \leq CO(\kappa^2 + h), \tag{43}$$

where C is a positive constant.

Proof We can write

$$U^j_{n+1} + \sum_{k=0}^{n+1} \Psi^\alpha_{k,n+1} U^j_k = U^{j-1}_{n+1} - \sum_{k=0}^{n+1} \Psi^\alpha_{k,n+1} U^{j-1}_k + \frac{\kappa(f^j_{n+1} + f^{j-1}_{n+1})}{2} + \mathrm{O}(\kappa^3 + \kappa h) \tag{44}$$

and

$$u^j_{n+1} + \sum_{k=0}^{n} \Psi^\alpha_{k,n+1} u^j_k = u^{j-1}_{n+1} - \sum_{k=0}^{n} \Psi^\alpha_{k,n+1} u^{j-1}_k + \frac{\kappa(f^j_{n+1} + f^{j-1}_{n+1})}{2}, \tag{45}$$

Let us set $e^j_i = U^j_i - u^j_i$ and by using (44) and (45), we have

$$e^j_{n+1} + \sum_{k=0}^{n} \Psi^\alpha_{k,n+1} e^j_k = e^{j-1}_{n+1} - \sum_{k=0}^{n} \Psi^\alpha_{k,n+1} e^{j-1}_k + \mathrm{O}(\kappa^3 + \kappa h), \tag{46}$$

thus, matrix–vector form of (46) can be expressed as

$$(I + D)\mathrm{E}^j = (I - D)\mathrm{E}^{j-1} + \mathrm{O}(\kappa^3 + \kappa h)\chi,$$

where $\mathrm{E}^j = [e^j_1, e^j_2, ..., e^j_n]^T$ and $\chi = [1, 1, ..., 1]^T$. Let us take

$$\Theta = (I + D)^{-1}(I - D), \qquad \Xi = \mathrm{O}(\kappa^3 + \kappa h)(I + D)^{-1},$$

therefore, we can write

$$\mathrm{E}^j = \Theta \mathrm{E}^{j-1} + \Xi.$$

By iterating, we have

$$\mathrm{E}^j = (\Theta^{\mathrm{j}-1} + \Theta^{\mathrm{j}-2} + ... + I)\Xi.$$

Since the eigenvalues of matrix D are positive, then matrix D is a positive definite matrix. By Lemma 1 and Lemma 3, we can write

$$\begin{aligned}\left\|\mathrm{E}^j\right\|_\infty &\leq (\left\|\Theta^{j-1}\right\|_\infty + \left\|\Theta^{j-2}\right\|_\infty + \ldots + \|I\|_\infty)\|\Xi\|_\infty \\ &\leq (1+1+\ldots+1)\|\Xi\|_\infty \\ &\leq j\mathrm{O}(\kappa^3 + \kappa h) = T\mathrm{O}(\kappa^2 + h).\end{aligned}$$

Finally,

$$\left\|\mathrm{E}^j\right\|_\infty \leq C\mathrm{O}(\kappa^2 + h).$$

□

Theorem 3 *Let* $u \in C[0, L]$ *and* $\alpha \in (0, 1)$, *then for the equal-area distribution method, we have*

$$\frac{1}{\Gamma(1-\alpha)}\left|\int_0^{x_{n+1}}(x_{n+1}-\tau)^{-\alpha}\left[\frac{\partial\hat{u}(\tau,t_j)}{\partial\tau}-\frac{\partial\bar{u}(\tau,t_j)}{\partial\tau}\right]d\tau\right| \leq C\frac{\Delta S}{h} \tag{47}$$

and, for the equal-height distribution method

$$\frac{1}{\Gamma(1-\alpha)}\left|\int_0^{x_{n+1}}(x_{n+1}-\tau)^{-\alpha}\left[\frac{\partial\hat{u}(\tau,t_j)}{\partial\tau}-\frac{\partial\bar{u}(\tau,t_j)}{\partial\tau}\right]d\tau\right| \leq C\frac{\Delta u}{h^2}, \tag{48}$$

specifically, when $\Delta S = \mathrm{O}(h^2)$ *or* $\Delta u = \mathrm{O}(h^3)$, *then*

$$\frac{1}{\Gamma(1-\alpha)}\left|\int_0^{x_{n+1}}(x_{n+1}-\tau)^{-\alpha}\left[\frac{\partial\hat{u}(\tau,t_j)}{\partial\tau}-\frac{\partial\bar{u}(\tau,t_j)}{\partial\tau}\right]d\tau\right| \leq Ch, \tag{49}$$

where $\hat{u}$ *is the piecewise linear interpolation for* u *at the method with uniform meshes and* $\bar{u}$ *is the piecewise linear interpolation for* u *at the method with non-uniform meshes.*

Proof Let $\hat{u}$ and $\bar{u}$ are the piecewise linear interpolations for u at the method with uniform meshes and the method with non-uniform meshes, respectively. Thus, for the equal-area distribution method, by using (25), we can write

$$(x_{n+1}-x_{n_i})^{1-\alpha} - (x_{n+1}-x_{n_{i+1}})^{1-\alpha} \leq (1-\alpha)\Delta S,$$

thus, we have

$$\left[(n+1-n_i)^{1-\alpha} - (n+1-n_{i+1})^{1-\alpha}\right] \leq \frac{(1-\alpha)\Delta S}{h^{1-\alpha}}. \tag{50}$$

By using (50), (6) and (28), we can write

$$\begin{aligned}
&\frac{1}{\Gamma(1-\alpha)}\left|\int_0^{x_{n+1}}(x_{n+1}-\tau)^{-\alpha}\left[\frac{\partial\hat{u}(\tau,t_j)}{\partial\tau}-\frac{\partial\bar{u}(\tau,t_j)}{\partial\tau}\right]d\tau\right|\\
&=\frac{h^{-\alpha}}{\Gamma(2-\alpha)}\left|\sum_{k=0}^{n}\rho_{k,n+1}u_k^j-\sum_{i=0}^{m_n}\theta_{i,n+1}u_{n_i}^j\right|\\
&=\frac{h^{-\alpha}}{\Gamma(2-\alpha)}\left|\sum_{i=0}^{m_n-1}\sum_{k=n_i}^{n_{i+1}-1}\rho_{k,n+1}^R(u_k^j-u_{k+1}^j)-\sum_{i=0}^{m_n-1}\theta_{i,n+1}^R(u_{n_i}^j-u_{n_{i+1}}^j)\right|\\
&=\frac{h^{-\alpha}}{\Gamma(2-\alpha)}\left|\sum_{i=0}^{m_n-1}\sum_{k=n_i}^{n_{i+1}-1}\left[\rho_{k,n+1}^R\,(u_{n_i}^j+\frac{\partial u(\xi_k,t_j)}{\partial x}(x_k-x_{n_i}))\right.\right.\\
&\left.-\rho_{k,n+1}^R(u_{n_i}^j+\frac{\partial u(\xi_{k+1},t_j)}{\partial x}(x_{k+1}-x_{n_i}))\right]-\sum_{i=0}^{m_n-1}\left[\theta_{i,n+1}^R\,u_{n_i}^j-\theta_{i,n+1}^R(u_{n_i}^j\right.\\
&\left.\left.+\frac{\partial u(\xi_{n_{i+1}},t_j)}{\partial x}(x_{n_{i+1}}-x_{n_i}))\right|\le\frac{h^{-\alpha}}{\Gamma(2-\alpha)}\left|\sum_{i=0}^{m_n-1}\left[\sum_{k=n_i}^{n_{i+1}-1}(\rho_{k,n+1}^R-\rho_{k,n+1}^R)\right.\right.\\
&\left.\left.-(\theta_{i,n+1}^R-\theta_{i,n+1}^R)\right]u_{n_i}^j\right|+\frac{h^{-\alpha}}{\Gamma(2-\alpha)}\left\|\frac{\partial u}{\partial x}\right\|_\infty\sum_{i=0}^{m_n-1}(x_{n_{i+1}}-x_{n_i})\left|-\theta_{i,n+1}^R\right|\\
&\le\frac{h^{-\alpha}}{\Gamma(2-\alpha)}\left\|\frac{\partial u}{\partial x}\right\|_\infty\sum_{i=0}^{m_n-1}(x_{n_{i+1}}-x_{n_i})\left[(n+1-n_i)^{1-\alpha}-(n+1-n_{i+1})^{1-\alpha}\right]\\
&\le\frac{h^{-\alpha}}{\Gamma(2-\alpha)}\left\|\frac{\partial u}{\partial x}\right\|_\infty x_n\frac{(1-\alpha)}{h^{1-\alpha}}\Delta S=\frac{x_n}{\Gamma(1-\alpha)}\left\|\frac{\partial u}{\partial x}\right\|_\infty\frac{\Delta S}{h}\\
&\le C\frac{\Delta S}{h}.
\end{aligned}\tag{51}$$

We assume

$$\int_{x_{n_{i^*}}}^{x_{n_{i^*+1}}}(x_{n+1}-\tau)^{-\alpha}d\tau=\underset{0\le i\le m_n-1}{Max}\int_{x_{n_i}}^{x_{n_{i+1}}}(x_{n+1}-\tau)^{-\alpha}d\tau,\tag{52}$$

by using (22), we have

$$(x_{n+1}-x_{n_{i^*+1}})^{-\alpha}-(x_{n+1}-x_{n_{i^*}})^{-\alpha}\le\Delta u,$$

thus, we can write

$$(x_{n+1}-x_{n_{i^*+1}})^{-\alpha}\le(x_{n+1}-x_{n_{i^*}})^{-\alpha}+\Delta u.\tag{53}$$

Therefore, we can write

$$\begin{aligned}
&\int_{x_{n_{i^*}}}^{x_{n_{i^*+1}}}(x_{n+1}-\tau)^{-\alpha}d\tau\le(x_{n+1}-x_{n_{i^*+1}})^{-\alpha}(x_{n_{i^*+1}}-x_{n_{i^*}})\\
&\le\left[(x_{n+1}-x_{n_{i^*}})^{-\alpha}+\Delta u\right](x_{n_{i^*+1}}-x_{n_{i^*}}),
\end{aligned}\tag{54}$$

by means of the mean value theorem for $u(\tau)=(x_{n+1}-\tau)^{-\alpha}$, there is a $x_{n_{i^*}}$ that we can write

$$
\begin{aligned}
(x_{n+1}-x_{n_{i+1}})^{-\alpha}-(x_{n+1}-x_{n_i})^{-\alpha} \approx & (x_{n_{i+1}}-x_{n_i})(-\alpha)(x_{n+1}-x_{n_{i^*}})^{-\alpha-1} \\
= & h(n_{i+1}-n_i)(-\alpha)(x_{n+1}-x_{n_{i^*}})^{-\alpha-1} \\
\leq & \Delta u,
\end{aligned}
$$

therefore, we have

$$
(x_{n+1}-x_{n_{i^*}})^{-\alpha} \leq \frac{\Delta u}{h(-\alpha)}(x_{n+1}-x_{n_{i^*}}). \tag{55}
$$

By using (54) and (55), we can write

$$
\int_{x_{n_{i^*}}}^{x_{n_{i^*}+1}} (x_{n+1}-\tau)^{-\alpha} d\tau \leq \left[\frac{\Delta u}{h(-\alpha)}(x_{n+1}-x_{n_{i^*}})+\Delta u\right](x_{n_{i^*+1}}-x_{n_{i^*}}). \tag{56}
$$

Finally, by using (51), and the following relation

$$
\left[(n+1-n_i)^{1-\alpha}-(n+1-n_{i+1})^{1-\alpha}\right] = \frac{(1-\alpha)}{h^{1-\alpha}} \int_{x_{n_i}}^{x_{n_{i+1}}} (x_{n+1}-\tau)^{-\alpha} d\tau,
$$

we can write

$$
\begin{aligned}
& \frac{1}{\Gamma(1-\alpha)} \left| \int_0^{x_{n+1}} (x_{n+1}-\tau)^{-\alpha} \left[\frac{\partial \hat{u}(\tau,t_j)}{\partial \tau} - \frac{\partial \bar{u}(\tau,t_j)}{\partial \tau}\right] d\tau \right| \\
& \leq \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \left\| \frac{\partial u}{\partial x} \right\|_\infty \sum_{i=0}^{m_n-1} (x_{n_{i+1}}-x_{n_i}) \left[(n+1-n_i)^{1-\alpha}-(n+1-n_{i+1})^{1-\alpha}\right] \\
& = \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \left\| \frac{\partial u}{\partial x} \right\|_\infty \sum_{i=0}^{m_n-1} (x_{n_{i+1}}-x_{n_i}) \frac{(1-\alpha)}{h^{1-\alpha}} \int_{x_{n_i}}^{x_{n_{i+1}}} (x_{n+1}-\tau)^{-\alpha} d\tau \\
& \leq \frac{1}{\Gamma(1-\alpha)h} \left\| \frac{\partial u}{\partial x} \right\|_\infty \sum_{i=0}^{m_n-1} (x_{n_{i+1}}-x_{n_i}) \int_{x_{n_{i^*}}}^{x_{n_{i^*}+1}} (x_{n+1}-\tau)^{-\alpha} d\tau \\
& \leq \frac{x_n}{\Gamma(1-\alpha)h} \left\| \frac{\partial u}{\partial x} \right\|_\infty \left[\frac{\Delta u}{h(-\alpha)}(x_{n+1}-x_{n_{i^*}})+\Delta u\right](x_{n_{i^*+1}}-x_{n_{i^*}}) \\
& \leq C\frac{\Delta u}{h^2}.
\end{aligned}
$$

5 Numerical experiments

In this section, some examples to illustrate the error bounds of the two methods with uniform and non-uniform meshes are presented.

Example 1 Consider the following partial fractional differential equation:

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = -{}_0^C D_x^\alpha u(x,t) + f(x,t), \quad x \in [0,1], \\ u(x,0) = x^2(1-x)^2, \quad 0 < \alpha < 1, \\ u(0,t) = 0, \qquad u(1,t) = 0, \end{cases} \tag{57}$$

where

$$f(x,t) = -x^2(1-x)^2 e^{-t} + e^{-t}\left[\frac{\Gamma(5)x^{4-\alpha}}{\Gamma(5-\alpha)} - \frac{2\Gamma(4)x^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{\Gamma(3)x^{2-\alpha}}{\Gamma(3-\alpha)}\right].$$

The exact solution of (57) is $u(x,t) = x^2(1-x)^2 e^{-t}$.

For solving this example by uniform and equidistributing meshes, different values of α, $h = \kappa$, Δu and ΔS with $T = 1, L = 1$ are utilized. In Tables 1 and 2, we have reported the results of this problem. This process has more benefits since the proposed method by equidistributing meshes does not lose computational accuracy and the computation cost of the methods (36) is decreased compared to the computation cost of the proposed method by uniform meshes(16) (see column N at Tables). Other numerical results are shown in Fig. 1.

Collections of Δu and ΔS in Algorithm 1 or Algorithm 2 for collecting the point meshes are very important. Because this process depends on $h = \kappa, \alpha$ and Δu or ΔS. Therefore, if we choose suitable Δu and ΔS, then the computation cost of the

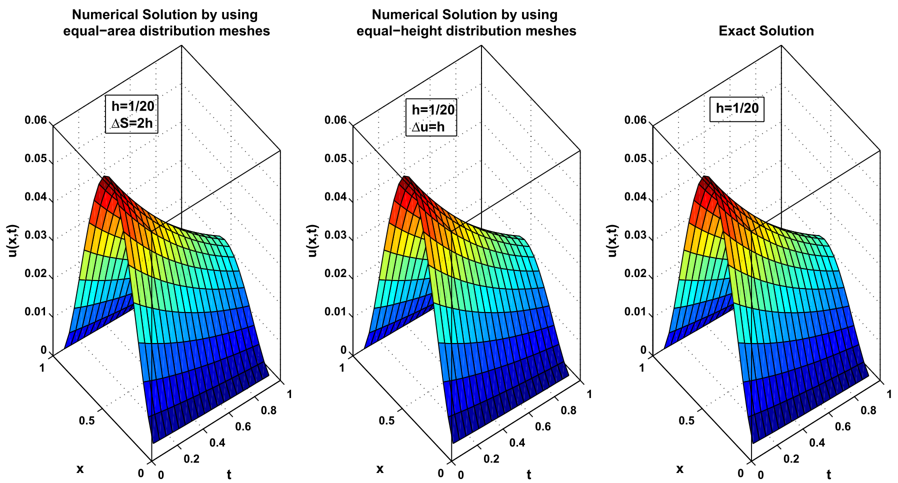

Fig. 1 The exact and numerical solutions by (16) and (36) (by using Algorithm 1 and 2), for example 1 (57), at $T = 1$, $L = 1$ and $\alpha = 0.2$, $\Delta u = h$, $\Delta S = 2h$, $h = 1/20$

Table 1 Absolute errors, convergence orders of Example 1 by the following methods at $L = 1$, $T = 1$, for different Δu, Δs, α and h, respectively

α	h	α=0.2 (Δu=h,ΔS=2.5 h)			α=0.5(Δu=2.5 h,ΔS=6 h)			α=0.8(Δu=7 h,ΔS=20 h)		
Method		Error	Order	N	Error	Order	N	Error	Order	N
Proposed method with uniform meshes (16)	$\frac{1}{10}$	3.70e−04	–	650	0.0016	–	650	0.0053	–	650
	$\frac{1}{20}$	1.39e−04	1.41	4600	6.70e−04	1.25	4600	0.0025	1.08	4600
	$\frac{1}{40}$	4.81e−05	1.54	34400	2.68e−04	1.32	34400	0.0011	1.18	34400
	$\frac{1}{80}$	1.57e−05	1.61	265600	1.02e−04	1.39	265600	5.07e−04	1.11	265600
Equal-height (36) by Algorithm 1	$\frac{1}{10}$	3.94e−04	–	550	0.0016	–	580	0.0053	–	540
	$\frac{1}{20}$	1.57e−04	1.32	3900	6.94e−04	1.20	4040	0.0025	1.08	3660
	$\frac{1}{40}$	5.24e−05	1.59	28800	2.75e−04	1.33	30160	0.0011	1.18	27280
	$\frac{1}{80}$	1.69e−05	1.63	222400	1.04e−04	1.39	234320	5.06e−04	1.12	212240
Equal-area (36) by Algorithm 2	$\frac{1}{10}$	3.52e−04	–	530	0.0018	–	390	0.0078	–	290
	$\frac{1}{20}$	1.19e−04	1.56	3620	6.37e−04	1.50	2460	0.0028	1.48	1640
	$\frac{1}{40}$	4.38e−05	1.44	27120	2.36e−04	1.43	17360	0.0011	1.35	10560
	$\frac{1}{80}$	1.45e−05	1.59	207280	1.01e−04	1.22	136160	5.07e−04	1.11	82320

Table 2 Absolute errors, convergence orders of Example 1 by the following methods at $L = 1$, $T = 1$, for different Δu, Δs, α and h, respectively

α	h	α=0.2 (Δu=2 h,ΔS=4 h)			α=0.5(Δu=5 h,ΔS=10 h)			α=0.8(Δu=10 h,ΔS=30 h)		
Method		Error	Order	N	Error	Order	N	Error	Order	N
Proposed method with uniform meshes (16)	$\frac{1}{10}$	3.70e−04	–	650	0.0016	–	650	0.0053	–	650
	$\frac{1}{20}$	1.39e−04	1.41	4600	6.70e−04	1.25	4600	0.0025	1.08	4600
	$\frac{1}{40}$	4.81e−05	1.54	34400	2.68e−04	1.32	34400	0.0011	1.18	34400
	$\frac{1}{80}$	1.57e−05	1.61	265600	1.02e−04	1.39	265600	5.07e−04	1.11	265600
Equal-height (36) by Algorithm 1	$\frac{1}{10}$	3.52e−04	–	440	0.0015	–	490	0.0052	–	490
	$\frac{1}{20}$	1.26e−04	1.49	3060	7.15e−04	1.07	3280	0.0025	1.06	3400
	$\frac{1}{40}$	5.12e−05	1.30	21720	2.90e−04	1.30	24320	0.0011	1.18	24960
	$\frac{1}{80}$	1.90e−05	1.43	165360	1.09e−04	1.41	188560	5.04e−04	1.12	189600
Equal-area (36) by Algorithm 2	$\frac{1}{10}$	4.22e−04	–	410	0.0027	–	330	0.0078	–	290
	$\frac{1}{20}$	1.74e−04	1.28	2420	8.97e−04	1.59	1800	0.0042	0.89	1420
	$\frac{1}{40}$	6.05e−05	1.52	16000	3.20e−04	1.49	11200	0.0013	1.69	8200
	$\frac{1}{80}$	1.79e−05	1.76	118320	9.77e−05	1.71	83440	5.26e−04	1.30	54480

non-uniform method (36) is decreased compared to the computation cost of the uniform method (16). Also, the numerical accuracy of non-uniform method does not decrease.

Example 2 We consider the following PFDEs as:

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = -{}_0^C D_x^{\alpha} u(x,t) + g(x,t), \quad x \in [0,1], \\ u(x,0) = x^2(1-x)^2, \quad 0 < \alpha < 1, \\ u(0,t) = 0, \qquad u(1,t) = 0, \end{cases} \tag{58}$$

where $g(x, t)$, define as:

$$g(x,t) = -x^2(1-x)^2 \sin(t) + \cos(t) \left[\frac{\Gamma(5) x^{4-\alpha}}{\Gamma(5-\alpha)} - \frac{2\Gamma(4) x^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{\Gamma(3) x^{2-\alpha}}{\Gamma(3-\alpha)} \right].$$

For this example(58), the exact solution is $u(x,t) = x^2(1-x)^2 \cos(t)$.

In Tables 3 and 4, we show the absolute errors of proposed methods with uniform (16) and non-uniform meshes (36). In those Tables, the results of proposed methods for different values of $h = \kappa$, Δu, ΔS and α, with $T = 1, L = 1$ are compared. Tables 3

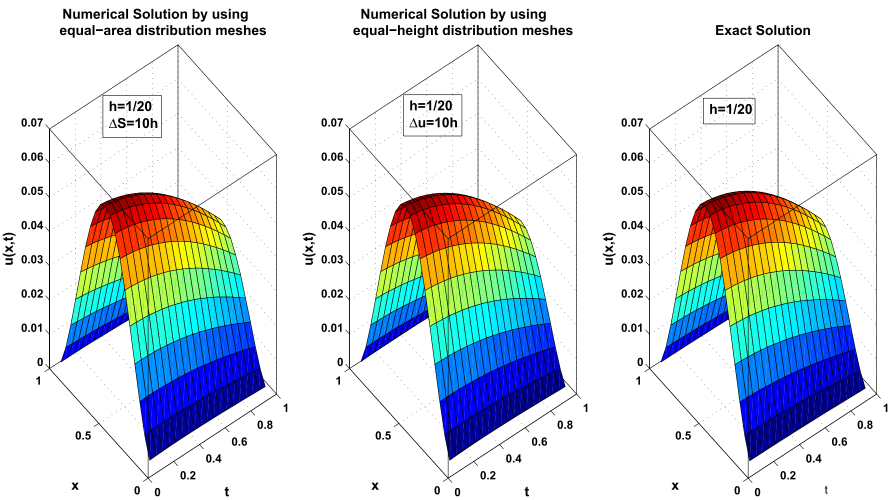

Fig. 2 The exact and numerical solutions by (16) and (36) (by using Algorithm 1 and 2), for example 2 (58), at $T = 1$, $L = 1$ and $\alpha = 0.5$, $\Delta u = 5h$, $\Delta S = 10h$, $h = 1/20$

Table 3 Absolute errors, convergence orders of Example 2 by the following methods at $L = 1$, $T = 1$, for different Δu, Δs, α and h, respectively

α	h	α=0.2 (Δu=h,ΔS=4 h)			α=0.5(Δu=2.5 h,ΔS=10 h)			α=0.8(Δu=7 h,ΔS=30 h)		
Method		Error	Order	N	Error	Order	N	Error	Order	N
Proposed method with uniform meshes (16)	$\frac{1}{10}$	5.09e−04	–	650	0.0022	–	650	0.0069	–	650
	$\frac{1}{20}$	1.92e−04	1.40	4600	8.63e−04	1.35	4600	0.0032	1.11	4600
	$\frac{1}{40}$	6.63e−05	1.54	34400	3.47e−04	1.31	34400	0.0015	1.09	34400
	$\frac{1}{80}$	2.16e−05	1.61	265600	1.33e−04	1.38	265600	6.5824e−04	1.19	265600
Equal-height (36) by Algorithm 1	$\frac{1}{10}$	5.41e−04	–	550	0.0021	–	580	0.0068	–	540
	$\frac{1}{20}$	2.16e−04	1.32	3900	8.96e−04	1.23	4040	0.0032	1.09	3660
	$\frac{1}{40}$	7.22e−05	1.59	28800	3.57e−04	1.33	30160	0.0015	1.09	27280
	$\frac{1}{80}$	2.32e−05	1.63	222400	1.36e−04	1.39	234320	6.56e−04	1.19	212240
Equal-area (36) by Algorithm 2	$\frac{1}{10}$	5.59e−04	–	410	0.0036	–	330	0.0104	–	290
	$\frac{1}{20}$	2.32e−04	1.27	2420	0.0012	1.58	1800	0.0056	0.89	1420
	$\frac{1}{40}$	8.14e−05	1.51	16000	4.32e−04	1.47	11200	0.0017	1.72	8200
	$\frac{1}{80}$	2.44e−05	1.73	118320	1.32e−04	1.71	83440	6.82e−04	1.32	54480

Table 4 Absolute errors, convergence orders of Example 2 by the following methods at $L = 1$, $T = 1$, for different Δu, Δs, α and h, respectively

α	h	α=0.2 (Δu=h,Δs=4 h)			α=0.5(Δu=2.5 h,Δs=10 h)			α=0.8(Δu=7 h,ΔS=30 h)		
Method		Error	Order	N	Error	Order	N	Error	Order	N
Proposed method with uniform meshes (16)	$\frac{1}{10}$	5.09e−04	–	650	0.0022	–	650	0.0069	–	650
	$\frac{1}{20}$	1.92e−04	1.40	4600	8.63e−04	1.35	4600	0.0032	1.11	4600
	$\frac{1}{40}$	6.63e−05	1.54	34400	3.47e−04	1.31	34400	0.0015	1.09	34400
	$\frac{1}{80}$	2.16e−05	1.61	265600	1.33e−04	1.38	265600	6.5824e−04	1.19	265600
Equal-height (36) by Algorithm 1	$\frac{1}{10}$	4.88e−04	–	440	0.0020	–	490	0.0068	–	490
	$\frac{1}{20}$	1.74e−04	1.48	3060	9.25e−04	1.11	3280	0.0032	1.09	3400
	$\frac{1}{40}$	7.08e−05	1.30	21720	3.77e−04	1.29	24320	0.0015	1.09	24960
	$\frac{1}{80}$	2.62e−05	1.43	165360	1.41e−04	1.41	188560	6.5341e−04	1.20	189600
Equal-area (36) by Algorithm 2	$\frac{1}{10}$	4.88e−04	–	530	0.0024	–	390	0.0104	–	290
	$\frac{1}{20}$	1.64e−04	1.57	3620	8.59e−04	1.48	2460	0.0037	1.49	1640
	$\frac{1}{40}$	6.03e−05	1.44	27120	3.05e−04	1.49	17360	0.0015	1.30	10560
	$\frac{1}{80}$	2.00e−05	1.59	207280	1.04e−04	1.55	136160	6.58e−04	1.19	82320

and 4 show that the proposed method with non-uniform meshes works well and convergence order of our proposed method with uniform meshes is $O(\kappa^2 + h)$. Other results are shown at Figs. 2 and 3.

Example 3 Consider the following partial fractional differential equation:

$$\begin{aligned} &\frac{\partial u(x,t)}{\partial t} + {}_0^C D_x^\alpha u(x,t) = 0, \quad x \in [0,1], \\ &u(x,0) = x^2(1-x)^2, \quad 0 < \alpha < 1, \\ &u(0,t) = 0, u(1,t) = 0, \end{aligned} \tag{59}$$

the exact solution for 59 is unavailable.

In Table 5, by using proposed methods with uniform and equidistributing meshes, we have reported numerical solutions of this problem at $x = 1$ and $t = 1$ ($u_{1,1}^{(h)}$) and $\left| u_{1,1}^{(h)} - u_{1,1}^{(\frac{h}{2})} \right|$ with different values of α, $h = \kappa$, Δu and ΔS. Where $u_{1,1}^{(h)}$ is numerical solution of example 3 at $x = 1$ and $t = 1$ with step size h. Other results are shown at Fig. 4.

6 Conclusion

In this paper, we design and develop some algorithms by using the piecewise linear interpolation polynomial for solving the PFDEs, with uniform and non-uniform meshes. The equal-height and equal-area distribution meshes are product by means of these algorithms. Also, we have used these algorithms (the equal-height or equal-area distribution algorithm) to generate the matrix at the proposed method with non-uniform meshes. Next, the error bounds of the proposed methods are obtained. The computation cost of numerical method with uniform meshes for the PFDEs is non-linearly increasing with time. This work shows that the computation cost of numerical method with non-uniform meshes for solving PFDEs increases linearly and the numerical accuracy of these methods dose not lose. Finally, we proved that the presented numerical method has a convergence order of $O(\kappa^2 + h)$.

Table 5 Numerical solutions at $x = 1$ and $t = 1$ ($u_{1,1}^{(h)}$) and $\left|u_{1,1}^{(h)} - u_{1,1}^{(\frac{h}{2})}\right|$ for Example 3 by the following methods for different Δu, Δs, α and h, respectively.

α	h	α=0.2 (Δu=h,ΔS=2.5 h)			α=0.5(Δu=2.5 h,ΔS=6 h)			α=0.8(Δu=7 h,ΔS=20 h)		
Method		$u_{1,1}^{(h)}$	$\left\|u_{1,1}^{(h)} - u_{1,1}^{(\frac{h}{2})}\right\|$	N	$u_{1,1}^{(h)}$	$\left\|u_{1,1}^{(h)} - u_{1,1}^{(\frac{h}{2})}\right\|$	N	$u_{1,1}^{(h)}$	$\left\|u_{1,1}^{(h)} - u_{1,1}^{(\frac{h}{2})}\right\|$	N
Proposed method with uniform meshes (16)	$\frac{1}{10}$	0.0065	–	650	0.0167	–	650	0.0217	–	650
	$\frac{1}{20}$	0.0064	1.00e−04	4600	0.0171	4.00e−04	4600	0.0228	0.0011	4600
	$\frac{1}{40}$	0.0063	1.00e−04	34400	0.0173	2.00e−04	34400	0.0235	7.00e−04	34400
	$\frac{1}{80}$	0.0063	0	265600	0.0173	0	265600	0.0239	1.00e−04	265600
Equal-height (36) by Algorithm 1	$\frac{1}{10}$	0.0065	–	550	0.0168	–	580	0.0217	–	540
	$\frac{1}{20}$	0.0064	1.00e−04	3900	0.0172	4.00e−04	4040	0.0228	0.0011	3660
	$\frac{1}{40}$	0.0063	1.00e−04	28800	0.0173	1.00e−04	30160	0.0235	7.00e−04	27280
	$\frac{1}{80}$	0.0063	0	222400	0.0173	0	234320	0.0239	1.00e−04	212240
Equal-area (36) by Algorithm 2	$\frac{1}{10}$	0.0065	–	530	0.0167	–	390	0.0235	–	290
	$\frac{1}{20}$	0.0063	1.00e−04	3620	0.0171	4.00e−04	2460	0.0237	2.00e−04	1640
	$\frac{1}{40}$	0.0063	0	27120	0.0173	2.00e−04	17360	0.0238	1.00e−04	10560
	$\frac{1}{80}$	0.0063	0	207280	0.0173	0	136160	0.0239	1.00e−04	82320

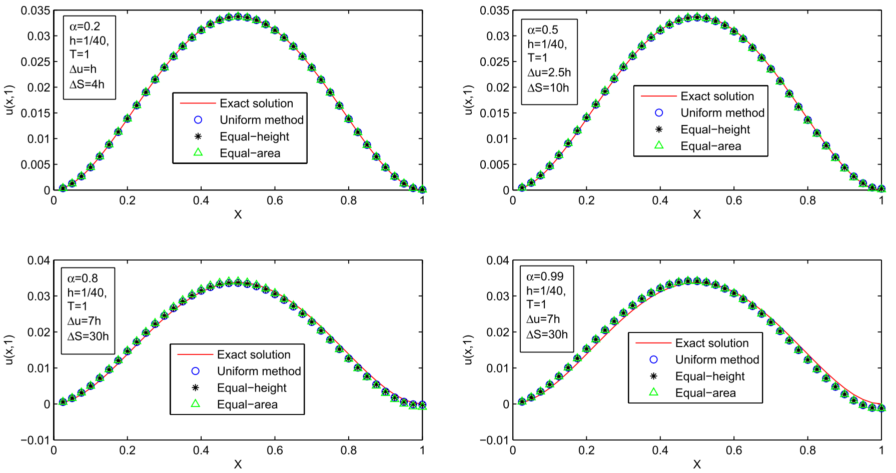

Fig. 3 The exact and numerical solutions by (16) and (36) (by using Algorithm 1 and 2), for example 2, with different Δu, Δs, α and $h = 1/40$, respectively

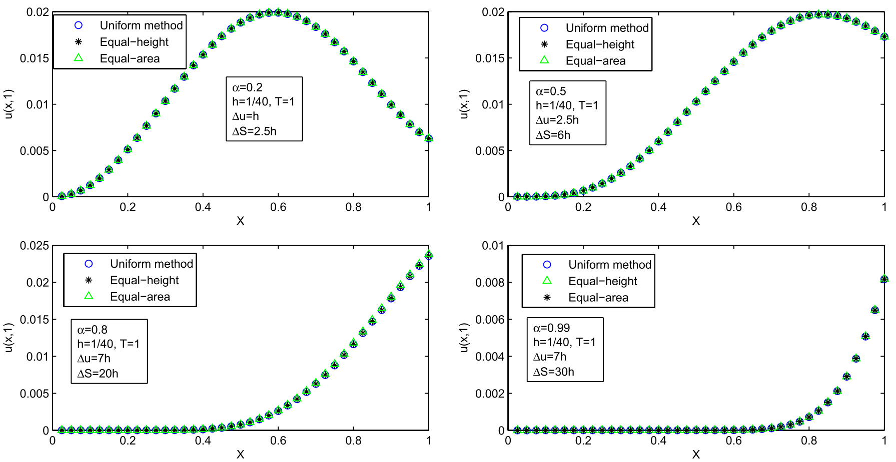

Fig. 4 The numerical solutions by (16) and (36) (by using Algorithm 1 and 2), for example 3, with different Δu, Δs, α and $h = 1/40$, respectively

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11227-023-05198-z.

Acknowledgements The authors would like to express special thanks to the referees for carefully reading, constructive comments, and valuable remarks which significantly improved the quality from this paper. This research is supported by a research grant of the University of Tabriz (Number 940).

Funding Not applicable.

Availability data and material Not applicable.

Declarations

Conflict of interest The authors have no conicts of interest and there is no fnancial interest to report.

Ethical Approval Not applicable.

References

1. Bagley RL, Calico R (1991) Fractional order state equations for the control of viscoelastically-damped structures. J Guid Control Dyn 14(2):304–311
2. Magin RL (2006) Fractional calculus in bioengineering. Begell House Redding
3. Marks R, Hall M (1981) Differintegral interpolation from a bandlimited signal's samples. IEEE Trans Acous Speech Signal Process 29(4):872–877
4. Wang Z, Huang X, Shi G (2011) Analysis of nonlinear dynamics and chaos in a fractional order financial system with time delay. Comput Math Appl 62(3):1531–1539
5. Gaul L, Klein P, Kemple S (1991) Damping description involving fractional operators. Mech Syst Signal Process 5(2):81–88
6. Gorenflo R (1997) Fractional calculus: some numerical methods. Courses and lectures-international centre for mechanical sciences. pp 277–290
7. Sebaa N, Fellah ZEA, Lauriks W, Depollier C (2006) Application of fractional calculus to ultrasonic wave propagation in human cancellous bone. Signal Process 86(10):2668–2677
8. Assaleh K, Ahmad WM (2007) Modeling of speech signals using fractional calculus. In: 2007 9th international symposium on signal processing and its applications, IEEE. 1–4
9. Magin R, Ovadia M (2008) Modeling the cardiac tissue electrode interface using fractional calculus. J Vibr Control 14(9–10):1431–1442
10. Fellah Z, Depollier C, Fellah M (2002) Application of fractional calculus to the sound waves propagation in rigid porous materials: validation via ultrasonic measurements. Acta Acust United Acust 88(1):34–39
11. Suárez JI, Vinagre BM, Calderón A, Monje C, Chen Y (2003) Using fractional calculus for lateral and longitudinal control of autonomous vehicles. In: International conference on computer aided systems theory. Springer, 337–348
12. Soczkiewicz E (2002) Application of fractional calculus in the theory of viscoelasticity. Mol Quantum Acoust 23:397–404
13. Mathieu B, Melchior P, Oustaloup A, Ceyral C (2003) Fractional differentiation for edge detection. Signal Process 83(11):2421–2432
14. Kulish VV, Lage JL (2002) Application of fractional calculus to fluid mechanics. J Fluids Eng 124(3):803–806
15. Ciuchi F, Mazzulla A, Scaramuzza N, Lenzi E, Evangelista L (2012) Fractional diffusion equation and the electrical impedance: experimental evidence in liquid-crystalline cells. J Phys Chem C 116(15):8773–8777
16. Chen W, Hu S, Cai W (2016) A causal fractional derivative model for acoustic wave propagation in lossy media. Arch Appl Mech 86(3):529–539
17. Momani S, Odibat Z (2007) Homotopy perturbation method for nonlinear partial differential equations of fractional order. Phys Lett A 365(5–6):345–350
18. Jafari H, Daftardar-Gejji V (2006) Solving a system of nonlinear fractional differential equations using adomian decomposition. J Comput Appl Math 196(2):644–651
19. Lesnic D (2006) The decomposition method for initial value problems. Appl Math Comput 181(1):206–213
20. Daftardar-Gejji V, Jafari H (2005) Adomian decomposition: a tool for solving a system of fractional differential equations. J Math Anal Appl 301(2):508–518
21. Zurigat M, Momani S, Odibat Z, Alawneh A (2010) The homotopy analysis method for handling systems of fractional differential equations. Appl Math Modell 34(1):24–35

22. Podlubny I (1998) Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications. Vol 198. Elsevier
23. Heris MS, Javidi M (2017) On fractional backward differential formulas for fractional delay differential equations with periodic and anti-periodic conditions. Appl Numer Math 118:203–220
24. Heris MS, Javidi M (2017) On fbdf5 method for delay differential equations of fractional order with periodic and anti-periodic conditions. Mediterranean J Math 14(3):134
25. Heris MS, Javidi M, Ahmad B (2019) Analytical and numerical solutions of Riesz space fractional advection-dispersion equations with delay. Comput Model Eng Sci 121(1):249–272
26. Diethelm K (1997) An algorithm for the numerical solution of differential equations of fractional order. Electron Trans Numer Anal 5(1):1–6
27. Diethelm K, Ford NJ, Freed AD (2004) Detailed error analysis for a fractional Adams method. Numer Algorithms 36(1):31–52
28. Diethelm K, Luchko Y (2004) Numerical solution of linear multi-term initial value problems of fractional order. J Comput Anal Appl 6(3):243–263
29. Blank L (1997) Numerical treatment of differential equations of fractional order. Nonlinear World 4:473–492
30. Garrappa R, Popolizio M (2011) On accurate product integration rules for linear fractional differential equations. J Comput Appl Math 235(5):1085–1097
31. Galeone L, Garrappa R (2006) On multistep methods for differential equations of fractional order. Mediterranean J Math 3(3–4):565–580
32. Garrappa R, Moret I, Popolizio M (2015) Solving the time-fractional Schrödinger equation by Krylov projection methods. J Comput Phys 293:115–134
33. Li C, Chen A, Ye J (2011) Numerical approaches to fractional calculus and fractional ordinary differential equation. J Comput Phys 230(9):3352–3368
34. Diethelm K, Ford NJ (2004) Multi-order fractional differential equations and their numerical solution. Appl Math Comput 154(3):621–640
35. Diethelm K (2010) The analysis of fractional differential equations: an application-oriented exposition using differential operators of Caputo type, vol 2004. Springer, Berlin
36. Heris MS, Javidi M (2019) A predictor-corrector scheme for the tempered fractional differential equations with uniform and non-uniform meshes. J Supercomput 75:8168–8206
37. Chen J, Liu F, Anh V (2008) Analytical solution for the time-fractional telegraph equation by the method of separating variables. J Math Anal Appl 338(2):1364–1377
38. Al-Khaled K, Momani S (2005) An approximate solution for a fractional diffusion-wave equation using the decomposition method. Appl Math Comput 165(2):473–483
39. Odibat Z, Momani S (2009) The variational iteration method: an efficient scheme for handling fractional partial differential equations in fluid mechanics. Comput Math Appl 58(11–12):2199–2208
40. Ganji D, Sadighi A (2007) Application of homotopy-perturbation and variational iteration methods to nonlinear heat transfer and porous media equations. J Comput Appl Math 207(1):24–34
41. Momani S, Odibat Z, Erturk VS (2007) Generalized differential transform method for solving a space-and time-fractional diffusion-wave equation. Phys Lett A 370(5–6):379–387
42. Momani S, Odibat Z (2008) Numerical solutions of the space-time fractional advection-dispersion equation. Numer Methods Partial Diff Equ Int J 24(6):1416–1429
43. Meerschaert MM, Tadjeran C (2006) Finite difference approximations for two-sided space-fractional partial differential equations. Appl Numer Math 56(1):80–90
44. Tadjeran C, Meerschaert MM, Scheffler H-P (2006) A second-order accurate numerical approximation for the fractional diffusion equation. J Comput Phys 213(1):205–213
45. Liu Q, Zeng F, Li C (2015) Finite difference method for time-space-fractional Schrödinger equation. Int J Comput Math 92(7):1439–1451
46. Ding H, Li C (2013) Numerical algorithms for the fractional diffusion-wave equation with reaction term. In: Abstract and applied analysis. Vol 2013. Hindawi
47. Heris MS, Javidi M (2018) Second order difference approximation for a class of riesz space fractional advection-dispersion equations with delay. arXiv preprint arXiv:1811.10513
48. Heris MS, Javidi M (2018) On fractional backward differential formulas methods for fractional differential equations with delay. Int J Appl Comput Math 4(2):72
49. Heris MS, Javidi M (2019) Fractional backward differential formulas for the distributed-order differential equation with time delay. Bullet Iran Math Soc 45(4):1159–1176

50. Javidi M, Heris MS (2019) Analysis and numerical methods for the Riesz space distributed-order advection-diffusion equation with time delay. SeMA J 76:533–551
51. Deng J, Zhao L, Wu Y (2017) Fast predictor-corrector approach for the tempered fractional differential equations. Numer Algorithms 74(3):717–754
52. Javidi M, Heris MS, Ahmad B (2019) A predictor-corrector scheme for solving nonlinear fractional differential equations with uniform and nonuniform meshes. Int J Model Simul Sci Comput 10:1950033
53. Kozyakin V (2009) On accuracy of approximation of the spectral radius by the Gelfand formula. Linear Algebra Appl 431(11):2134–2141
54. Thomas JW (2013) Numerical partial differential equations: finite difference methods, vol 22. Springer, New York