J. Math. Computer Sci., 37 (2025), 236–260

Online: ISSN 2008-949X

Journal of Mathematics and Computer Science

Journal Homepage: www.isr-publications.com/jmcs

The moving least square method for solving the time fractional partial integro-differential equation of Volterra type and its convergence analysis

Said Elbostani[a,*], Abdelali Mohib[b], Rachid El Jid[a], Anas Rachid[b], Zahra El Majouti[a]

[a] *MISI Laboratory, Faculty of Sciences and Technology, Hassan 1^er University, Settat, Morocco.*

[b] *Laboratoire LMAI, ENS de Casablanca, Hassan II university of Casablanca, B.P. 50069, Ghandi Casablanca, Morocco.*

Abstract

In this paper, the moving least square (MLS) approximation is implemented for the numerical solution of time fractional partial integro-differential equation (TFPIDE) on a bounded domain. To establish the scheme, we apply the finite difference scheme to approximate the time Caputo fractional derivative, and we employ the composite trapezoidal quadrature rule for estimating integrals. This approach is very convenient for solving TFPIDE since it does not require any need for mesh connectivity. Then, the problem solving turns into solution of a linear system. The applicability and the validity of this method is investigated. Furthermore, the error estimate of the proposed method is provided. Finally, several numerical problems are solved which confirmed the theoretical findings.

Keywords: Fractional partial integro-differential equation (FPIDE), moving least squares (MLS), Caputo fractional derivative, convergence analysis.

2020 MSC: 26A33, 35R09, 45D05, 45K05, 65A05, 65N35.

1. Introduction

The time fractional partial integro-differential equations occur as reformulations from some mixed value problems arising in many scientific fields such as dynamics of the population growth model with fractional temporal evolution [25], fractional order state equations for the control of viscoelastic structures [8], control strategy of the outbreak of dengue fever [17], the dynamics of motion for an accelerated mass-spring system within the framework of fractional calculus [9], a new mathematical model within a generalized fractional framework for investigating the dynamics of HIV/AIDS transmission [6], an efficient mathematical model to investigate the dynamics of COVID-19 within a generalized fractional framework [5], and the fractional modeling of diabetes and tuberculosis co-existence [18].

However, only limited problems with simple boundary conditions have analytical solutions and due to the mathematical complexities, the majority are generally difficult to be solved analytically. Therefore, numerical approaches become indispensable tools for numerical solutions of this kind of equations. For

[*] Corresponding author
Email address: elboustanisaid10@gmail.com (Said Elbostani)

doi: 10.22436/jmcs.037.02.08

Received: 2024-02-03 *Revised:* 2024-05-15 *Accepted:* 2024-08-24

instance, a few numerical approaches have been applied to solve different types of the time fractional partial integro-differential equations. Ziyang Luo proposed a new numerical scheme based on compact finite difference [23], Atta and Youssri developed an approximate spectral method based on shifted first-kind Chebyshev polynomials [4], and Fakhar-Izadi combined the Galerkin approximation with the Legendre polynomials as a basis [14].

Generally, the above-mentioned mesh methods are today the most powerful tool for solving this type of problem, the conditional stability of explicit finite difference procedures and the need to use a large amount of CPU time in implicit finite difference schemes limit the applicability of these methods. As well, these approaches provide the solution of the problem on mesh points only, and the efficiency of the methods is reduced in non-smooth and irregular domains.

To overcome the mesh problems, the mesh-free methods have been proposed and achieved remarkable progress in recent years. The collocation-based meshless method is more efficient since no background meshes and integration are needed. In this work, we consider a meshless method based on moving least square approximation (MLS) for solving the time fractional partial integro-differential equation (TFPIDE) of Volterra type:

$$\begin{cases} {}_0^C\mathcal{D}_t^{\alpha}u\left(x,t\right) - a\left(x\right)\partial_x^2 u\left(x,t\right) + \lambda\int_0^t k\left(x,t-s\right)u\left(x,s\right)ds = f\left(x,t\right), \quad (x,t)\in\Omega, \quad \alpha\in(0,1), \\ u\left(x,0\right) = g\left(x\right), \quad x\in[0,L], \\ u\left(0,t\right) = h_1\left(t\right) \text{ and } u\left(L,t\right) = h_2\left(t\right), \quad t\in(0,T], \end{cases} \tag{1.1}$$

where $\Omega := (0,L)\times(0,T]$, λ is a positive constant, ${}_0^C\mathcal{D}_t^{\alpha}$ is the Caputo fractional differential operator of order α for $t \geqslant 0$, a is continuous function on $\overline{\Omega}$, and there exist $a_0 > 0$ such that $a(z) > a_0$, $\forall z \in [0,L]$, the functions k and f are sufficiently smooth functions on $\overline{\Omega}$, also, g is smooth function on $[0,L]$, h_1 and h_2 are smooth functions on $[0,T]$.

Actually, the meshless methods have gained more attention, particularly the moving least squares method, it was introduced first in the late of 1968s by Shepard [30] and then developed by Lancaster and Salkauskas [19].

The moving least squares (MLS) approach is discussed in this work [31], along with its application, flexibility, limitations with field discontinuities, and various optimization options. The MLS method does not require domain elements or background cells. This method allows an easy adaptation of the nodal density, then the distribution of nodes could be chosen regularly or randomly in the consideration domain. MLS method is ideal for complex geometries and moving boundaries because it provides smooth and accurate approximations (see [22]). It has been applied in many branches of modern sciences, such as surface construction [1], function approximation [20], numerical solution of integral and a class of nonlinear fractional Fredholm integro-differential equations [11–13, 16].

However, as local systems need to be resolved at each point, this requires a complex and computationally expensive implementation (see [7]). Finite difference method (FDM) and finite element method (FEM) are easier to implement but have difficulty handling complex geometries, MLS offers superior adaptability. Spectral and Radial Basis Function (RBF) methods offer high accuracy but can be computationally intensive. For large-scale problems, the MLS method's computational cost and memory usage are significant, and efficient parallelization remains challenging.

The error analysis of the MLS method in several-dimensional spaces is well documented in the literature, Davoud and Armentano provided error analysis for the moving least squares approximation for functions in Sobolev spaces of fractional order [2, 24], Zuppa obtained error estimates for MLS approximations for the function and its derivatives [33] and The authors of [3, 28] obtained error estimates for MLS approximations in the one-dimensional and two-dimensional cases.

In this work we establish the new error analysis and the rate of convergence for MLS method when it is used for solving problem (1.1) numerically. The numerical technique used the composite trapezoidal quadrature rule for approximating integrals, and the finite difference scheme to approximate the first order time derivative, where the time derivative is defined in Caputo sense. Such problem exhibits a mild singularity at the initial time $t = 0$. For that we introduced the finite difference scheme to get rid of such singularity on a uniform mesh. Finally, the convergence of the used scheme is tested in various examples, which demonstrates the theoretical error estimates.

The structure of the present paper is as follows. In Section 2 we introduced some necessary definitions and theorems to prove the existence and uniqueness of the solution of the TFPIDE. In Sections 3 and 4 the MLS method is presented and discussed. In Section 5 the convergence analysis of the MLS method is investigated. Various examples are tested in Section 6. Finally, we conclude our results in Section 7.

2. Preliminaries on fractional calculus

In this section, we utilize the most common definitions and properties of operators in fractional calculus, specifically the Riemann-Liouville fractional derivative and the Caputo derivative [10, 27]. We then discuss the existence and uniqueness of solution using various definitions and theorems from analytical calculus [15, 29].

Definition 2.1. Let u be continuous on $\bar{\Omega}$. The Riemann-Liouville fractional integral of the function $u(x,t)$ is defined by:

$$I_a^{\alpha} u(x,t) = \frac{1}{\Gamma(\alpha)} \int_{s=a}^{t} (t-s)^{\alpha-1} u(x,s) ds,$$

where α is a positive real number, be the order of the integral.

Definition 2.2. The Caputo fractional derivative of the function $u(x,t)$ is defined by:

$${}_a^C\mathcal{D}_t^{\alpha} u(x,t) = \left[I_a^{n-\alpha} \left(\frac{\partial^n u}{\partial t^n} \right) \right] (x,t), \quad \text{for} \quad (x,t) \in \Omega,$$

where $\alpha \in \mathbb{R}^+$is the order of the derivative and $n = \lceil \alpha \rceil$, the smallest integer which is greater than or equal to α.

We now impose three properties of fractional integrals and its derivatives as follows. We have for all $t \geqslant a$:

1. ${}_a^C\mathcal{D}_t^{\alpha} u = 0$ for all $u = c \in \mathbb{R}$;
2. for $\alpha \in (0,1)$, we have ${}_a^C\mathcal{D}_t^{\alpha} I_a^{\alpha} u(x,t) = u(x,t)$ but $I_a^{\alpha}\,{}_a^C\mathcal{D}_t^{\alpha} u(x,t) = u(x,t) - u(x,a^+)$.
3. Fractional integrals and derivatives satisfy the linearity property:
   (a) $I_a^{\alpha}\{C_1 u_1 \pm C_2 u_2\} = C_1 I_a^{\alpha} u_1 \pm C_2 I_a^{\alpha} u_2$;
   (b) ${}_a^C\mathcal{D}_t^{\alpha}\{C_1 u_1 \pm C_2 u_2\} = C_1\,{}_a^C\mathcal{D}_t^{\alpha} u_1 \pm C_2\,{}_a^C\mathcal{D}_t^{\alpha} u_2$,
   C_1, C_2 are some positive constants.

2.0.1. *Existence and uniqueness of solution*

Definition 2.3. ([26]). Let E, S be two normed vector spaces over $\mathbb{R}$ or $\mathbb{C}$. A linear operator $L : E \to S$ is said to be bounded if there exists a positive constant c such that

$$\|Le\|_S \leqslant c\|e\|_E, \quad \forall e \in E.$$

Theorem 2.4. *Let E, S be normed vector spaces. The linear operator $L : E \to S$ is bounded if and only if L is continuous everywhere in E.*

Definition 2.5. Let $\Omega \subseteq \mathbb{R}^d$. The mapping $H : \Omega \to \Omega$ is a contraction mapping if there exists a positive constant $c \in [0,1)$ such that $\|H(\mathbf{x}) - H(\mathbf{y})\| \leqslant c\|\mathbf{x} - \mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in \Omega$.

Theorem 2.6. *Suppose $\Omega \subseteq \mathbb{R}^d$ be complete and $H : \Omega \to \Omega$ is a contraction mapping, then H has a unique fixed point $\mathbf{x}^*$ in Ω.*

Theorem 2.7. *If the assumptions*

1. *the partial derivatives* $\partial_x^\ell u(.,t)$ *are continuous for all* $\ell = 1,2$ *and* $t > 0$;
2. k *is a continuous function and* $\|k\| \leqslant C_0$, *where* $(C_0 > 0)$,

are satisfies and $\frac{[(\alpha+1)C\|a\|+\lambda C_0 T]T^\alpha}{\Gamma(\alpha+2)} < 1$, *then there exists a unique solution* $u(x,t) \in \bar{\Omega}$ *of Eq.* (1.1) *(*$(C > 0)$ *is an arbitrary constant which can take different values at different places).*

Proof. Now applying I_0^α to both sides of Eq. (1.1), we get $u(x,t) = Hu(x,t)$, $\forall (x,t) \in \bar{\Omega}$, where $Hu(x,t)$ is defined by:

$$Hu(x,t) = g(x) + I_0^\alpha f(x,t) + a(x) I_0^\alpha \partial_x^2 u - \lambda I_0^\alpha \left\{ \int_0^t k(x,t-s) u(x,s) ds \right\}.$$

Let $u_1, u_2 \in C(\bar{\Omega})$, using integration by parts for all $(x,t) \in \bar{\Omega}$, we have

$$\begin{aligned} Hu_1(x,t) - Hu_2(x,t) &= \frac{a(x)}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \partial_x^2 (u_1 - u_2)(x,s) ds \\ &\quad - \frac{\lambda}{\Gamma(\alpha)} \int_0^t (t-\rho)^{\alpha-1} \int_0^\rho k(x,\rho - s)(u_1 - u_2)(x,s) ds d\rho \\ &= \frac{a(x)}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \partial_x^2 (u_1-u_2)(x,s) ds - \frac{\lambda k(x,0)}{\Gamma(\alpha)} \int_0^t \frac{(t-\rho)^\alpha}{\alpha} (u_1 - u_2)(x,\rho) d\rho. \end{aligned}$$

Applying the norm and the triangle inequality

$$\begin{aligned} \|Hu_1 - Hu_2\| &\leqslant \left\| \frac{a(x)}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \partial_x^2 (u_1-u_2)(x,s) ds \right\| + \left\| \frac{\lambda k(x,0)}{\Gamma(\alpha)} \int_0^t \frac{(t-\rho)^\alpha}{\alpha} (u_1-u_2)(x,\rho) d\rho \right\| \\ &\leqslant \frac{\|a\|}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \left\| \partial_x^2 (u_1-u_2)(x,s) \right\| ds + \frac{\lambda C_0}{\Gamma(\alpha)} \int_0^t \frac{(t-\rho)^\alpha}{\alpha} \|(u_1-u_2)(x,\rho)\| d\rho. \end{aligned}$$

Using Definition 2.3 and Theorem 2.4, we obtain

$$\begin{aligned} \|Hu_1 - Hu_2\| &\leqslant \frac{C\|a\|}{\Gamma(\alpha)} \int_0^T (T-s)^{\alpha-1} ds \|u_1 - u_2\| + \frac{\lambda C_0}{\alpha \Gamma(\alpha)} \int_0^T (T-\rho)^\alpha d\rho \|u_1 - u_2\| \\ &\leqslant \left[\frac{C\|a\| T^\alpha}{\Gamma(\alpha+1)} + \frac{\lambda C_0 T^{\alpha+1}}{\Gamma(\alpha+2)} \right] \|u_1 - u_2\| \leqslant \left[\frac{[(\alpha+1)C\|a\| + \lambda C_0 T] T^\alpha}{\Gamma(\alpha+2)} \right] \|u_1 - u_2\|. \end{aligned}$$

Hence the operator H is a contraction if $\frac{[(\alpha+1)C\|a\|+\lambda C_0 T]T^\alpha}{\Gamma(\alpha+2)} < 1$ on the Banach space $(C(\bar{\Omega}), \|\cdot\|)$, then by using Theorem 2.6 one can conclude that Eq. (1.1) has a unique solution $u(x,t)$ in $\bar{\Omega}$. □

3. The MLS method in $\mathbb{R}^d$ $(d \geq 1)$

The MLS method is one of the most effective meshless techniques that allows an easy adaptation of the nodal density in the geometric domain, it does not require domain elements or background cells. Then, it can be easily generalized to the higher problems.

Let $X := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a set of all nodes in the bounded domain Ω on $\mathbb{R}^d$. Let a point $\bar{\mathbf{x}}$ in Ω, the neighborhood of this point $\bar{\mathbf{x}}$ is denoted by $\Omega_{\bar{\mathbf{x}}}$. Let $n_{\bar{\mathbf{x}}}$ be the number of nodes in $\Omega_{\bar{\mathbf{x}}}$ given by $\{\mathbf{x}_i,\ i = 1, \ldots, n_{\bar{\mathbf{x}}}\}$, see the Fig. 1 for more details.

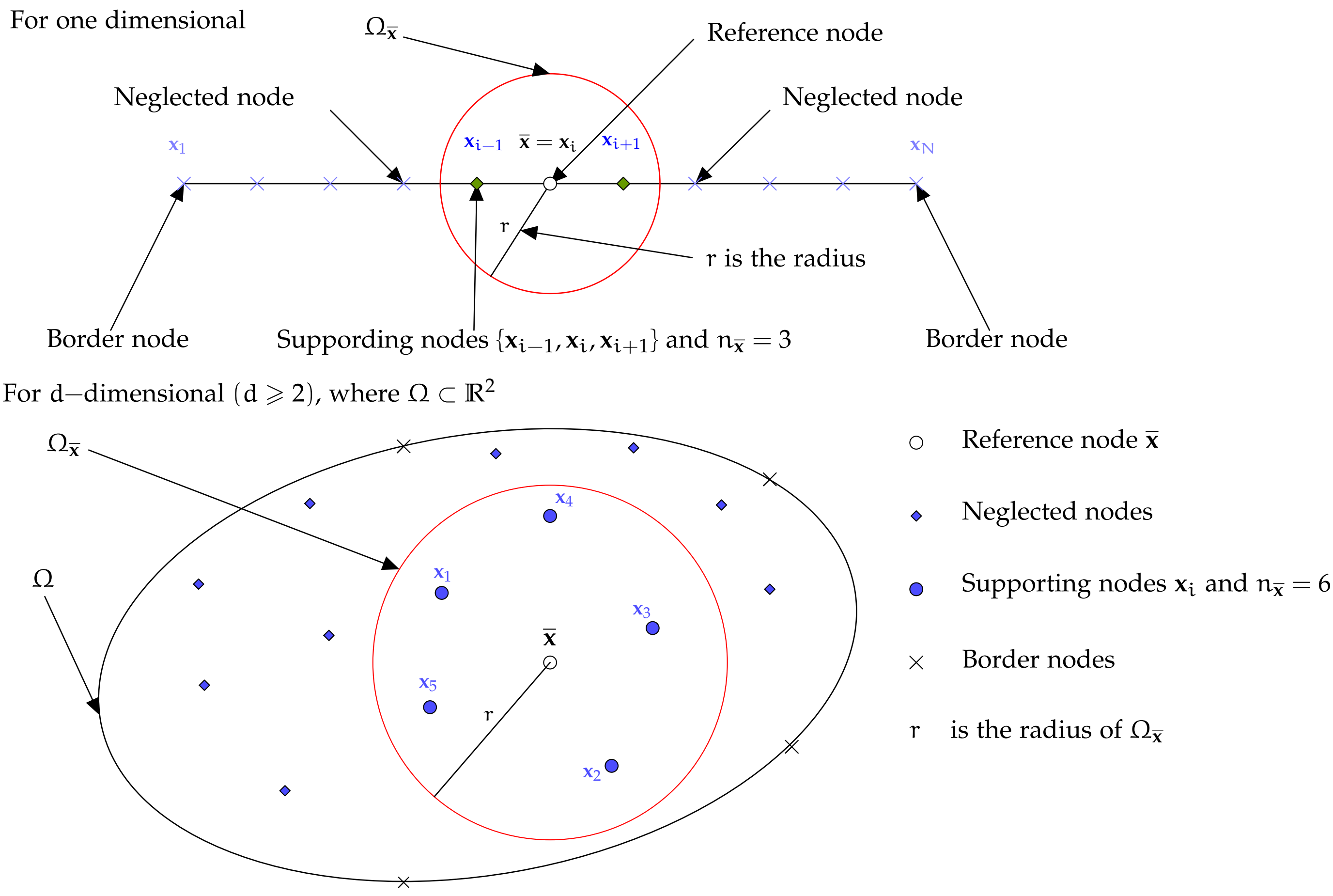
For one dimensional
$\Omega_{\overline{x}}$
Reference node
Neglected node
Neglected node
x_1
$\mathbf{x}_{i-1}$
$\overline{\mathbf{x}} = \mathbf{x}_i$
$\mathbf{x}_{i+1}$
x_N
r
r is the radius
Border node
Suppording nodes $\{\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}\}$ and $n_{\overline{x}} = 3$
Border node
For d−dimensional $(d \geqslant 2)$, where $\Omega \subset \mathbb{R}^2$
$\Omega_{\overline{x}}$
Ω
x_4
x_1
x_3
$\overline{\mathbf{x}}$
x_5
r
x_2
○ Reference node $\overline{\mathbf{x}}$
◆ Neglected nodes
● Supporting nodes $\mathbf{x}_i$ and $n_{\overline{x}} = 6$
× Border nodes
r is the radius of $\Omega_{\overline{x}}$

Figure 1: The construction of neighborhood of point $\overline{x}$ in $\mathbb{R}^d$, $d = 1, 2$.

The MLS approximation of the unknown function u in $\Omega_{\overline{x}}$ is written as follows:

$$\forall \mathbf{x} \in \Omega_{\overline{\mathbf{x}}}, \quad \overline{u}^r(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})\lambda_j(\overline{\mathbf{x}}) = \mathbf{p}^T(\mathbf{x})\lambda(\overline{\mathbf{x}}), \tag{3.1}$$

where $\mathbf{x} = [x_1, \dots, x_d]^T$, $\mathbf{p}(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x})]^T$ is a complete monomial basis function, m is the number of the basis and $\lambda(\overline{\mathbf{x}})$ is the unknown vector containing coefficients $\lambda_j(\overline{\mathbf{x}})$ with $j = 1, \dots, m$. The relation between the term m and the largest degree $\widehat{m}$ of $p_m(x)$ is $m = \frac{(\widehat{m}+d)!}{\widehat{m}!d!}$ see [21].
For example in one dimensional we have $\mathbf{x} = [x_1] = [x]$, the linear basis is given by $\mathbf{p}(\mathbf{x}) = \mathbf{p}(x) = [1, x]^T$ and the quadratic basis is given by $\mathbf{p}(\mathbf{x}) = \mathbf{p}(x) = [1, x, x^2]^T$.

Now the coefficient vector $\lambda(\overline{\mathbf{x}})$ is determined by minimizing a weighted discrete L_2 norm defined by:

$$F(\overline{\mathbf{x}}) = \sum_{j=1}^{n_{\overline{x}}} w_j^r(\overline{\mathbf{x}}) \left[\mathbf{p}^T(\mathbf{x}_j)\lambda(\overline{\mathbf{x}}) - \overline{u}_j\right]^2, \tag{3.2}$$

where $\overline{u}_j = u(\mathbf{x}_j)$, $j = 1, \dots, n_{\overline{x}}$, $w_j^r(\overline{\mathbf{x}}) = w\left(\frac{\|\overline{\mathbf{x}} - \mathbf{x}_j\|}{r}\right)$ is the weight function associated with the node j and r is the size of support domain. Then we introduce some weights functions w. The cubic weight function is as

$$w(z)=\begin{cases}\frac{2}{3}-4z^2+4z^3, & \text{if } 0\leqslant z\leqslant \frac{1}{2},\\ \frac{4}{3}-4z+4z^2-\frac{4}{3}z^3, & \text{if } \frac{1}{2}<z\leqslant 1,\\ 0, & \text{if } z>1.\end{cases}$$

The quartic weight function is

$$w(z)=\begin{cases}1-6z^2+8z^3-3z^4, & \text{if } 0\leqslant z\leqslant 1,\\ 0, & \text{if } z>1.\end{cases}$$

The Gaussian weight function (GWF) is

$$w(z)=\begin{cases}\dfrac{\exp\left(-\left(z\frac{r}{\mu}\right)^2\right)-\exp\left(-\left(\frac{r}{\mu}\right)^2\right)}{1-\exp\left(-\left(\frac{r}{\mu}\right)^2\right)}, & \text{if } 0\leqslant z<1,\\ 0, & \text{if } z\geqslant 1,\end{cases}$$

where μ is a constant to control the shape function. The Eq. (3.2) can be rewritten as

$$F(\overline{\mathbf{x}})=\sum_{j=1}^{n_{\overline{x}}} w_j^r(\overline{\mathbf{x}})\left[\sum_{\ell=1}^{m} p_\ell(\mathbf{x}_j)\lambda_\ell(\overline{\mathbf{x}})-\overline{u}_j\right]^2.$$

So the partial derivative of F for all $k=1,\dots,m$ is given by:

$$\frac{\partial F(\overline{\mathbf{x}})}{\partial \lambda_k}=2\sum_{j=1}^{n_{\overline{x}}} w_j^r(\overline{\mathbf{x}})p_k(\mathbf{x}_j)\left[\mathbf{p}^T(\mathbf{x}_j)\lambda(\overline{\mathbf{x}})-\overline{u}_j\right]=2\left[\sum_{j=1}^{n_{\overline{x}}} w_j^r(\overline{\mathbf{x}})p_k(\mathbf{x}_j)\mathbf{p}^T(\mathbf{x}_j)\lambda(\overline{\mathbf{x}})-\sum_{j=1}^{n_{\overline{x}}} w_j^r(\overline{\mathbf{x}})p_k(\mathbf{x}_j)\overline{u}_j\right]. \quad (3.3)$$

Define the matrices $\mathbf{A}(\overline{\mathbf{x}})$, $\mathbf{C}(\overline{\mathbf{x}})$, and the column vector $\overline{u}$ as follows:

$$\mathbf{A}(\overline{\mathbf{x}})=\sum_{j=1}^{n_{\overline{x}}} w_j^r(\overline{\mathbf{x}})\mathbf{p}(\mathbf{x}_j)\mathbf{p}^T(\mathbf{x}_j),\quad \mathbf{C}(\overline{\mathbf{x}})=\left[w_1^r(\overline{\mathbf{x}})\mathbf{p}(\mathbf{x}_1),\dots,w_{n_{\overline{x}}}^r(\overline{\mathbf{x}})\mathbf{p}(\mathbf{x}_{n_{\overline{x}}})\right], \text{ and } \overline{u}=[\overline{u}_1,\dots,\overline{u}_{n_{\overline{x}}}]^T. \quad (3.4)$$

Then, Eq. (3.3) can be rewritten by:

$$\frac{\partial F}{\partial \lambda}=2[\mathbf{A}(\overline{\mathbf{x}})\lambda(\overline{\mathbf{x}})-\mathbf{C}(\overline{\mathbf{x}})\overline{u}].$$

The unknown vector $\lambda(\overline{\mathbf{x}})$ can be determined by solving the linear system $\frac{\partial F}{\partial \lambda}=0$. Hence we obtain: $\lambda(\overline{\mathbf{x}})=\mathbf{A}^{-1}(\overline{\mathbf{x}})\mathbf{C}(\overline{\mathbf{x}})\overline{u}$ if $\mathbf{A}(\overline{\mathbf{x}})$ is non-singular. Now replacing $\lambda(\overline{\mathbf{x}})$ in Eq. (3.1) for all $\mathbf{x}\in\Omega_{\overline{x}}$ we get:

$$\overline{u}^r(\mathbf{x})=\mathbf{p}^T(\mathbf{x})\lambda(\overline{\mathbf{x}})=\mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\overline{\mathbf{x}})\mathbf{C}(\overline{\mathbf{x}})\overline{u}=\mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\overline{\mathbf{x}})\sum_{\ell=1}^{n_{\overline{x}}} w_\ell^r(\overline{\mathbf{x}})\mathbf{p}(\mathbf{x}_\ell)\overline{u}_\ell=\sum_{\ell=1}^{n_{\overline{x}}}\overline{\psi}_\ell(\mathbf{x})\overline{u}_\ell,$$

where $\overline{\psi}_\ell(\mathbf{x})=w_\ell^r(\overline{\mathbf{x}})\mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\overline{\mathbf{x}})\mathbf{p}(\mathbf{x}_\ell)$, $\overline{u}^r(\mathbf{x})$ is the MLS approximation in this subdomain $\Omega_{\overline{x}}$ of Ω. The global MLS approximation is given as

$$\forall \mathbf{x}\in\Omega,\ \ u^r(\mathbf{x})=\sum_{\ell=1}^{N}\psi_\ell(\mathbf{x})u_\ell,$$

where $N = n_x$ is the number of nodes in Ω, $\psi_\ell(\mathbf{x}) = w^r_\ell(\mathbf{x})\mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x}_\ell)$ is the shape function of this MLS method, the first and the second derivatives are obtained by the following expressions for all $i = 1, \dots, d$, $\ell = 1, \dots, N$:

$$\partial_{x_i}(\psi_\ell(\mathbf{x})) = \partial_{x_i}(w^r_\ell(\mathbf{x}))\mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x}_\ell) + w^r_\ell(\mathbf{x})\partial_{x_i}(\mathbf{p}^T(\mathbf{x}))\mathbf{A}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x}_\ell) + w^r_\ell(\mathbf{x})\mathbf{p}^T(\mathbf{x})\partial_{x_i}(\mathbf{A}^{-1}(\mathbf{x}))\mathbf{p}(\mathbf{x}_\ell),$$

$$\begin{aligned}\partial^2_{x_i}(\psi_\ell(\mathbf{x})) = {} & \partial^2_{x_i}(w^r_\ell(\mathbf{x}))\mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x}_\ell) + w^r_\ell(\mathbf{x})\partial^2_{x_i}(\mathbf{p}^T(\mathbf{x}))\mathbf{A}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x}_\ell) \\ & + w^r_\ell(\mathbf{x})\mathbf{p}^T(\mathbf{x})\partial^2_{x_i}(\mathbf{A}^{-1}(\mathbf{x}))\mathbf{p}(\mathbf{x}_\ell)2\Big[\partial_{x_i}(w^r_\ell(\mathbf{x}))\partial_{x_i}(\mathbf{p}^T(\mathbf{x}))\mathbf{A}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x}_\ell) \\ & + \partial_{x_i}(w^r_\ell(\mathbf{x}))\mathbf{p}^T(\mathbf{x})\partial_{x_i}(\mathbf{A}^{-1}(\mathbf{x}))\mathbf{p}(\mathbf{x}_\ell)\Big] + 2w^r_\ell(\mathbf{x})\partial_{x_i}(\mathbf{p}^T(\mathbf{x}))\partial_{x_i}(\mathbf{A}^{-1}(\mathbf{x}))\mathbf{p}(\mathbf{x}_\ell),\end{aligned} \tag{3.5}$$

where

$$\begin{aligned}&\partial_{x_i}(\mathbf{A}^{-1}(\mathbf{x})) = -\mathbf{A}^{-1}(\mathbf{x})\partial_{x_i}(\mathbf{A}(\mathbf{x}))\mathbf{A}^{-1}(\mathbf{x}),\\ &\partial^2_{x_i}(\mathbf{A}^{-1}(\mathbf{x})) = -\left[\partial_{x_i}(\mathbf{A}^{-1}(\mathbf{x}))\partial_{x_i}(\mathbf{A}(\mathbf{x}))\mathbf{A}^{-1}(\mathbf{x}) + \mathbf{A}^{-1}(\mathbf{x})\partial^2_{x_i}(\mathbf{A}(\mathbf{x}))\mathbf{A}^{-1}(\mathbf{x}) + \mathbf{A}^{-1}(\mathbf{x})\partial_{x_i}(\mathbf{A}(\mathbf{x}))\partial_{x_i}(\mathbf{A}^{-1}(\mathbf{x}))\right].\end{aligned}$$

In the MLS approximation, the function $\psi_\ell(.)$ is usually called the shape function corresponding to the nodal point $\mathbf{x}_\ell$ for all $\ell = 1, \dots, N$ and we have $\psi_\ell(\mathbf{x}_k) \neq \delta_{\ell,k}$, where $\delta_{\ell,k}$ is the Kronecker symbol.

4. Numerical approximation

Let N be a fixed positive integer, we define the grid points in the time interval $n = 0, \dots, N$ as $t^n = n\delta t$, where $\delta t = \frac{T}{N}$ is the time step size. Then the mesh $\{t^n, n = 0, \dots, N\}$ is uniform direction. Let $\{u(x, t^n)\}_{n=0}^N$ be the exact solution and denote $\{u^n\}_{n=0}^N$ as the approximate solution at each (x, t^n) of Eq. (1.1), where $x \in (0, L)$. For all $n = 0, \dots, N-1$ the Caputo fractional derivative of the function u at each (x, t^{n+1}), where $x \in (0, L)$ is given as:

$${}_0^C\mathcal{D}_t^\alpha u(x, t^{n+1}) := \frac{1}{\Gamma(1-\alpha)}\int_0^{t^{n+1}} (t^{n+1} - s)^{-\alpha}\partial_s u(x, s)ds = \frac{1}{\Gamma(1-\alpha)}\sum_{j=0}^{n}\int_{t^j}^{t^{j+1}} (t^{n+1} - s)^{-\alpha}\partial_s u(x, s)ds.$$

Now, we need to approximate the first order time derivative by finite difference scheme:

$${}_0^C\mathcal{D}_t^\alpha u\left(x, t^{n+1}\right) = \frac{1}{\Gamma(1-\alpha)}\sum_{j=0}^{n}\left[\frac{u^{j+1} - u^j}{\delta t} + \mathcal{O}(\delta t)\right]\int_{t^j}^{t^{j+1}} \left(t^{n+1} - s\right)^{-\alpha} ds. \tag{4.1}$$

Simplifying the integral in Eq. (4.1), we get

$$\int_{t^j}^{t^{j+1}} \left(t^{n+1} - s\right)^{-\alpha} ds = \left[\frac{-\left(t^{n+1} - s\right)^{1-\alpha}}{1-\alpha}\right]_{t^j}^{t^{j+1}} = \frac{d_\alpha(n-j)}{(1-\alpha)(\delta t)^{\alpha-1}}, \tag{4.2}$$

where $d_\alpha(j) = (j+1)^{1-\alpha} - (j)^{1-\alpha}$ for $j \geqslant 0$. Consequently, by substituting the Eq. (4.2) into Eq. (4.1), we obtain

$$\begin{aligned}{}_0^C\mathcal{D}_t^\alpha u\left(x, t^{n+1}\right) &= \frac{1}{\Gamma(1-\alpha)}\sum_{j=0}^{n}\left[\frac{u^{j+1} - u^j}{\delta t} + \mathcal{O}(\delta t)\right]\frac{d_\alpha(n-j)}{(1-\alpha)(\delta t)^{\alpha-1}} \\ &= \frac{1}{(\delta t)^\alpha\Gamma(2-\alpha)}\sum_{j=0}^{n}\left(u^{j+1} - u^j\right)d_\alpha(n-j) + \mathcal{O}((\delta t)^{2-\alpha}).\end{aligned}$$

Now, we can reformulate the above equality as

$$ {}_{0}^{C}\mathcal{D}_{t}^{\alpha}u\left(x,t^{n+1}\right)=\gamma_{\alpha}u^{n+1}+\ \gamma_{\alpha}\sum_{j=0}^{n}\left[d_{\alpha}(j+1)-d_{\alpha}(j)\right]u^{n-j}-\gamma_{\alpha}d_{\alpha}(n+1)u^{0}+\mathcal{O}((\delta t)^{2-\alpha}), \tag{4.3}$$

for all $x\in(0,L),\ n=0,\dots,N-1$, where

$$\gamma_{\alpha}=\frac{1}{(\delta t)^{\alpha}\Gamma(2-\alpha)}.$$

In the integral part, we apply the composite trapezoidal approximation at each $(x,t^{n+1}),\ x\in(0,L),\ n=0,\dots,N-1$, we obtain:

$$\begin{aligned}\int_{0}^{t^{n+1}}\lambda k\left(x,\ t^{n+1}-s\right)u\left(x,s\right)ds&=\lambda\sum_{j=0}^{n}\int_{t^{j}}^{t^{j+1}}k\left(x,t^{n+1}-s\right)u\left(x,s\right)ds\\&=\frac{\lambda\delta t}{2}\sum_{j=0}^{n}\left[k\left(x,t^{n+1}-t^{j+1}\right)u^{j+1}+k\left(x,t^{n+1}-t^{j}\right)u^{j}\right]+\mathcal{O}(\delta t)\\&=\frac{\lambda\delta t}{2}\left[\sum_{\ell=1}^{n+1}k\left(x,(n-\ell+1)\delta t\right)u^{\ell}+\sum_{\ell=0}^{n}k\left(x,(n-\ell+1)\delta t\right)u^{\ell}\right]+\mathcal{O}(\delta t)\\&=\frac{\lambda\delta t}{2}\left[k(x,0)u^{n+1}+2\sum_{\ell=1}^{n}k\left(x,(n-\ell+1)\delta t\right)u^{\ell}+k\left(x,(n+1)\delta t\right)u^{0}\right]+\mathcal{O}(\delta t)\\&=\frac{\lambda\delta t}{2}\left[k(x,0)u^{n+1}+2\sum_{\ell=0}^{n}k\left(x,(n-\ell+1)\delta t\right)u^{\ell}-k\left(x,(n+1)\delta t\right)u^{0}\right]+\mathcal{O}(\delta t).\end{aligned}$$

Using the transformation $\ell=n-j$, we get

$$\begin{aligned}\int_{0}^{t^{n+1}}\lambda k\left(x,\ t^{n+1}-s\right)u\left(x,s\right)ds=\frac{\lambda\delta t}{2}k(x,0)u^{n+1}+\lambda\delta t\sum_{j=0}^{n}k\left(x,(j+1)\delta t\right)u^{n-j}\\-\frac{\lambda\delta t}{2}k(x,(n+1)\delta t)u^{0}+\mathcal{O}(\delta t).\end{aligned}\tag{4.4}$$

From the Eqs. (4.3) and (4.4) into Eq. (1.1), we obtain the following recurrence relation for $x\in(0,L),\ n=0,\dots,N-1$:

$$\begin{aligned}\gamma_{\alpha}u^{n+1}-a\left(x\right)\partial_{x}^{2}u^{n+1}+\frac{\lambda\delta t}{2}k(x,0)u^{n+1}=\sum_{j=0}^{n}\left[\gamma_{\alpha}(d_{\alpha}(j)-d_{\alpha}(j+1))-\lambda\delta tk\left(x,(j+1)\delta t\right)\right]u^{n-j}\\+\left[\gamma_{\alpha}d_{\alpha}(n+1)+\frac{\lambda\delta t}{2}k(x,(n+1)\delta t)\right]u^{0}+f^{n+1},\end{aligned}\tag{4.5}$$

where $f^{n+1}=f(x,t^{n+1})$. Then, we use MLS shape functions to approach $u^{n}(x)$, $x\in(0,L)$. Let M be a positive integer, define the grid of space by $x_{m}=(m-1)\delta x$ for $m=1,\dots,M$, where the spatial step size of direction x is defined by $\delta x=\frac{L}{M-1}$. Hence the approximate solution $u^{n}(x)$ at $n=0,\dots,N-1$ can be given by:

$$u^{n}(x)=\sum_{j=1}^{M}\mathbf{c}_{j}^{n}\psi_{j}(x), \tag{4.6}$$

where $\psi_j(x)$, $j=1,\dots,M$ are the shape functions of the MLS approximation and $\{\mathbf{c}_j^n,\ j=1,\dots,M\}$ are unknown coefficients which need to be determined. Replacing M collocation points, thus we determine the values of coefficients $\{\mathbf{c}_j^n,\ j=1,\dots,M\}$ and we get the following system:

$$u^n(x_i)=\sum_{j=1}^{M}\mathbf{c}_j^n\psi_j(x_i),\quad i=1,\dots,M. \tag{4.7}$$

Rewritten Eq. (4.7) in a matrix form:

$$\mathbf{U}^n=\Psi\mathbf{C}^n,\quad n=0,\dots,N-1, \tag{4.8}$$

where $\mathbf{U}^n=\left[u_1^n,u_2^n,\dots,u_M^n\right]^T$, $\mathbf{C}^n=\left[c_1^n,c_2^n,\dots,c_M^n\right]^T$ and Ψ is an $M\times M$ matrix given by:

$$\Psi=\left[\psi_{ij}\right]=\begin{pmatrix}\psi_{11} & \cdots & \psi_{1M}\\ \vdots & \ddots & \vdots\\ \psi_{M1} & \cdots & \psi_{MM}\end{pmatrix}, \tag{4.9}$$

where $\psi_{ij}=\psi_j(x_i)$. In the Eq. (1.1) we have two boundary points and $M-2$ internal points. Thus we need to split the matrix Ψ by this form: $\Psi=\mathbf{T}+\mathbf{S}$, where $\mathbf{T}$ and $\mathbf{S}$ are defined by:

$$\mathbf{T}=\left[t_{ij}\right]=\begin{cases}\Psi_{ij}, & i=2,\dots,M-1,\ j=1,\dots,M,\\ 0, & \text{otherwise},\end{cases}\qquad \mathbf{S}=\left[s_{ij}\right]=\begin{cases}\Psi_{ij}, & i=1,M,\ j=1,\dots,M,\\ 0, & \text{otherwise}.\end{cases}$$

Using Eq. (4.6), we discrete $u_{xx}=\frac{d^2u}{dx^2}$ at $\{t^n,\ n=1,\dots,N\}$ by MLS approximation method:

$$u_{xx}^n(x)=\sum_{j=1}^{M}\mathbf{c}_j^n\frac{d^2\psi_j}{dx^2}(x), \tag{4.10}$$

where $\frac{d^2\psi_j}{dx^2}$ is obtained from Eq. (3.5) for all $j=1,\dots,M$. Substituting $M-2$ collocation points into Eq. (4.10), we have:

$$u_{xx}^n(x_i)=\sum_{j=1}^{M}\mathbf{c}_j^n\frac{d^2\psi_j}{dx^2}(x_i),\quad i=2,\dots,M-1. \tag{4.11}$$

Then Eq. (4.11) can be rewritten as

$$\mathbf{U}_{xx}^n=\mathbf{D}\mathbf{C}^n, \tag{4.12}$$

where $\mathbf{U}_{xx}^n=\left[u_{xx}{}_1^n,u_{xx}{}_2^n,\dots,u_{xx}{}_M^n\right]^T$ and $\mathbf{D}$ is an $M\times M$ matrix given by:

$$\mathbf{D}=\left[d_{ij}\right]=\begin{cases}\frac{d^2\psi_j}{dx^2}(x_i), & i=2,\dots,M-1,\ j=1,\dots,M,\\ 0, & i=1,M,\ j=1,\dots,M.\end{cases}$$

Now replacing $M-2$ internal collocation points in Eq. (4.5) and using Eqs. (4.8) and (4.12) together with boundary conditions in Eq. (1.1), we obtain the following recurrence relation:

$$\mathbf{L}\mathbf{C}^{n+1}=\sum_{j=0}^{n}\mathbf{B}(j)\mathbf{C}^{n-j}+\mathbf{E}(n+1)\mathbf{U}^0+\mathbf{F}^{n+1}+\mathbf{H}^{n+1}, \tag{4.13}$$

where

$$\mathbf{L} = \gamma_\alpha \mathbf{T} - \mathbf{AD} + \frac{\lambda\delta t}{2}\mathbf{K}(0)\mathbf{T} + \mathbf{S}, \tag{4.14}$$

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & a(x_2) & 0 & \dots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \dots & 0 & a(x_{M-1}) & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{K}(s) := \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & k(x_2, s) & 0 & \dots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \dots & 0 & k(x_{M-1}, s) & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{B}(s) := [\gamma_\alpha(d_\alpha(s) - d_\alpha(s+1))\mathbf{I} - \lambda\delta t\mathbf{K}\left((s+1)\delta t\right)]\,\mathbf{T}, \quad \mathbf{E}(s) := \gamma_\alpha d_\alpha(s)\mathbf{I} + \frac{\lambda\delta t}{2}\mathbf{K}(s\delta t), \tag{4.15}$$

$$\mathbf{I} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}, \tag{4.16}$$

$$\mathbf{F}^{n+1} = \left[0, f(x_2, t^{n+1}), \dots, f(x_{M-1}, t^{n+1}), 0\right]^T, \quad \mathbf{H}^{n+1} = \left[h_1(x_1, t^{n+1}), 0, \dots, 0, h_2(x_M, t^{n+1})\right]^T,$$

and in which $\mathbf{I}$ is an $M \times M$ matrix. Rewriting Eq. (4.13) by this form:

$$\mathbf{C}^{n+1} = \sum_{j=0}^{n} \mathbf{L}^{-1}\mathbf{B}(n-j)\mathbf{C}^j + \mathbf{L}^{-1}\mathbf{E}(n+1)\mathbf{U}^0 + \mathbf{L}^{-1}\left(\mathbf{F}^{n+1} + \mathbf{H}^{n+1}\right). \tag{4.17}$$

Finally, the approximate solution can be determined by this scheme at any time t^n, $n = 0, \dots, N-1$.

5. Convergence analysis

In this section, we show respectively the rate convergence of time fractional derivative, the trapezoidal approximation and the MLS method. We start by the formula expressed in Eq. (4.1), the time fractional derivative in Eq. (4.1) is approximated by the finite difference scheme, so the rate convergence of Eq. (4.3) is $\mathcal{O}\left((\delta t)^{2-\alpha}\right)$.

Lemma 5.1. *Assume that the solution $u(.,.)$ of Eq. (1.1) and the kernel $k(.,.)$ satisfied $\left|\partial_t^\ell u\right| \leqslant C'_\ell\left(1 + t^{\alpha-\ell}\right)$ and $\left|\partial_y^\ell k\right| \leqslant C_\ell$, where $C'_\ell > 0$ and $C_\ell > 0$ for $\ell = 0, 1$, respectively. Let $x \in (0, L)$ and for each $n = 0, \dots, N-1$, the remainder term of integral part is the order $\mathcal{O}(\delta t)$.*

Proof. Let $x \in (0, L)$ and for each $n = 0, \dots, N-1$, we have:

$$\begin{aligned} I_e := {} & \lambda \int_0^{t^{n+1}} k\left(x,\ t^{n+1} - s\right) u\left(x, s\right) ds - \frac{\lambda\delta t}{2}\sum_{j=0}^{n}\left[k\left(x, t^{n+1} - t^{j+1}\right) u(x, t^{j+1}) + k\left(x, t^{n+1} - t^j\right) u(x, t^j)\right] \\ = {} & \sum_{j=0}^{n} \lambda \int_{t^j}^{t^{j+1}} k\left(x,\ t^{n+1} - s\right) u\left(x, s\right) ds - \frac{\lambda\delta t}{2}\sum_{j=0}^{n}\left[k\left(x, t^{n+1} - t^{j+1}\right) u(x, t^{j+1}) + k\left(x, t^{n+1} - t^j\right) u(x, t^j)\right] \\ = {} & \sum_{j=0}^{n}\left[\lambda \int_{t^j}^{t^{j+1}} k\left(x,\ t^{n+1} - s\right) u\left(x, s\right) ds - \frac{\lambda\delta t}{2}\left(k\left(x, t^{n+1} - t^{j+1}\right) u(x, t^{j+1}) + k\left(x, t^{n+1} - t^j\right) u(x, t^j)\right)\right]. \end{aligned} \tag{5.1}$$

Using integration by parts of the above term on the left hand side:

$$\begin{aligned}\lambda\int_{t^j}^{t^{j+1}} k\left(x,t^{n+1}-s\right)u\left(x,s\right)ds &= \lambda\left[-\left(t^{j+1/2}-s\right)k\left(x,t^{n+1}-s\right)u\left(x,s\right)\right]_{s=t^j}^{t^{j+1}}\\ &\quad+\lambda\int_{t^j}^{t^{j+1}}\left(t^{j+1/2}-s\right)\frac{d}{ds}\left[k\left(x,t^{n+1}-s\right)u\left(x,s\right)\right]ds\\ &=\frac{\lambda\delta t}{2}\left(k\left(x,t^{n+1}-t^{j+1}\right)u(x,t^{j+1})+k\left(x,t^{n+1}-t^{j}\right)u(x,t^{j})\right)\\ &\quad+\lambda\int_{t^j}^{t^{j+1}}\left(t^{j+1/2}-s\right)\frac{d}{ds}\left[k\left(x,t^{n+1}-s\right)u\left(x,s\right)\right]ds,\end{aligned}\tag{5.2}$$

where $t^{j+1/2}=(j+\frac{1}{2})\delta t$. Replacing Eq. (5.2) into Eq. (5.1), we get:

$$I_e=\lambda\sum_{j=0}^{n}\int_{t^j}^{t^{j+1}}\left(t^{j+1/2}-s\right)\frac{d}{ds}\left[k\left(x,t^{n+1}-s\right)u\left(x,s\right)\right]ds.$$

So, we apply an absolute norm:

$$\begin{aligned}|I_e| &=\left|\lambda\sum_{j=0}^{n}\int_{t^j}^{t^{j+1}}\left(t^{j+1/2}-s\right)\frac{d}{ds}\left[k\left(x,t^{n+1}-s\right)u\left(x,s\right)\right]ds\right|\\ &\leqslant\lambda\sum_{j=0}^{n}\int_{t^j}^{t^{j+1}}\left|\left(t^{j+1/2}-s\right)\right|\left|\frac{d}{ds}\left[k\left(x,t^{n+1}-s\right)u\left(x,s\right)\right]\right|ds\\ &\leqslant\lambda\sum_{j=0}^{n}\int_{t^j}^{t^{j+1}}\left|\left(t^{j+1/2}-s\right)\right|\left(\left|\partial_y k\left(x,t^{n+1}-s\right)u\left(x,s\right)\right|+\left|k\left(x,t^{n+1}-s\right)\partial_s u\left(x,s\right)\right|\right)ds\\ &\leqslant\lambda\sum_{j=0}^{n}\int_{t^j}^{t^{j+1}}\left|\left(t^{j+1/2}-s\right)\right|\left(\left|\partial_y k\left(x,t^{n+1}-s\right)\right|\left|u\left(x,s\right)\right|+\left|k\left(x,t^{n+1}-s\right)\right|\left|\partial_s u\left(x,s\right)\right|\right)ds\\ &\leqslant\frac{\lambda\delta t}{2}\int_0^T C_1C_0'\left(1+s^{\alpha}\right)+C_0C_1'\left(1+s^{\alpha-1}\right)ds.\end{aligned}$$

Hence we obtained the estimate error of integral part by:

$$\|I_e\|\leqslant C\delta t,\quad\text{where}\quad C=\frac{\lambda}{2}\int_0^T C_1C_0'\left(1+s^{\alpha}\right)+C_0C_1'\left(1+s^{\alpha-1}\right)ds>0.\tag{5.3}$$

□

Using Eq. (4.8) we rewrite the Eq. (4.17) as:

$$\mathbf{U}^{n+1}=\sum_{j=0}^{n}\Psi\mathbf{L}^{-1}\mathbf{B}(n-j)\Psi^{-1}\mathbf{U}^j+\Psi\mathbf{L}^{-1}\mathbf{E}(n+1)\mathbf{U}^0+\Psi\mathbf{L}^{-1}\left(\mathbf{F}^{n+1}+\mathbf{H}^{n+1}\right).$$

So, we obtain the simple scheme by this expression:

$$\mathbf{U}^{n+1}=\sum_{j=0}^{n}\mathbf{V}(n-j)\mathbf{U}^j+\mathbf{Q}(n+1)\mathbf{U}^0+\mathbf{W}(n+1),\tag{5.4}$$

$$\mathbf{V}(s):=\Psi\mathbf{L}^{-1}\mathbf{B}(s)\Psi^{-1},\quad\mathbf{Q}(s):=\Psi\mathbf{L}^{-1}\mathbf{E}(s),\quad\mathbf{W}(s):=\Psi\mathbf{L}^{-1}\left(\mathbf{F}^s+\mathbf{H}^s\right).\tag{5.5}$$

$\mathbf{U}^{n+1}$ can be computed for $n = 0, \ldots, N-1$ by using the recursive formula expressed in Eq. (5.4) and $\mathbf{U}^0$ is computed by the initial condition, i.e., $[u(x_1, 0), \ldots, u(x_M, 0)]^T$.

Lemma 5.2. *For each $n = 0, \ldots, N-1$, the solution of Eq. (5.4) satisfies*

$$\mathbf{U}^{n+1} = \left[\beta_{n+1} + \sum_{j=1}^{n+1} \Theta_{n+1,j}\mathbf{Q}(j)\right] \mathbf{U}^0 + \sum_{j=1}^{n+1} \Theta_{n+1,j}\mathbf{W}(j), \tag{5.6}$$

where the sequences matrices β_n and $\Theta_{n,i}$ are defined by:

$$\begin{cases} \beta_n = \sum_{j=0}^{n-1} \mathbf{V}(n-1-j)\beta_j, & \forall n \geqslant 1, \\ \beta_0 := \mathbf{I}, \end{cases} \qquad \begin{cases} \Theta_{n,i} = \sum_{j=1}^{n-i} \mathbf{V}(j-1)\Theta_{n-j,i}, & n > i \geqslant 1, \\ \Theta_{n,n} := \mathbf{I}, & n \geqslant 1. \end{cases}$$

The matrices $\mathbf{I}$ and $\mathbf{V}(.)$ are defined in Eqs. (4.16) and (5.5), respectively.

Proof. For $n = 0$ we have in Eq. (5.6),

$$\mathbf{U}^1 = [\beta_1 + \mathbf{Q}(1)]\, \mathbf{U}^0 + \mathbf{W}(1), \tag{5.7}$$

Eq. (5.7) is identical to Eq. (5.4). We prove that, if the statement Eq. (5.4) is valid for $j = 1, \ldots, n$, then the statement holds for $j = n+1$:

$$\begin{aligned}
\mathbf{U}^{n+1} &= \sum_{j=0}^{n} \mathbf{V}(n-j)\mathbf{U}^j + \mathbf{Q}(n+1)\mathbf{U}^0 + \mathbf{W}(n+1) \\
&= \mathbf{V}(n)\mathbf{U}^0 + \sum_{j=1}^{n} \mathbf{V}(n-j)\mathbf{U}^j + \mathbf{Q}(n+1)\mathbf{U}^0 + \mathbf{W}(n+1) \\
&= \mathbf{V}(n)\mathbf{U}^0 + \sum_{j=1}^{n} \mathbf{V}(n-j)\left[\left(\beta_j + \sum_{k=1}^{j} \Theta_{j,k}\mathbf{Q}(k)\right)\mathbf{U}^0 + \sum_{k=1}^{j} \Theta_{j,k}\mathbf{W}(k)\right] + \mathbf{Q}(n+1)\mathbf{U}^0 + \mathbf{W}(n+1) \\
&= \left[\mathbf{V}(n) + \sum_{j=1}^{n} \mathbf{V}(n-j)\beta_j + \sum_{j=1}^{n} \mathbf{V}(n-j)\sum_{k=1}^{j} \Theta_{j,k}\mathbf{Q}(k) + \mathbf{Q}(n+1)\right]\mathbf{U}^0 \\
&\quad + \sum_{j=1}^{n} \mathbf{V}(n-j)\sum_{k=1}^{j} \Theta_{j,k}\mathbf{W}(k) + \mathbf{W}(n+1) \\
&= \left[\sum_{j=0}^{n} \mathbf{V}(n-j)\beta_j + \sum_{j=1}^{n} \mathbf{V}(n-j)\sum_{k=1}^{j} \Theta_{j,k}\mathbf{Q}(k) + \mathbf{Q}(n+1)\right]\mathbf{U}^0 \\
&\quad + \sum_{j=1}^{n} \mathbf{V}(n-j)\sum_{k=1}^{j} \Theta_{j,k}\mathbf{W}(k) + \mathbf{W}(n+1) \\
&= \left[\beta_{n+1} + \sum_{j=1}^{n} \mathbf{V}(n-j)\sum_{k=1}^{j} \Theta_{j,k}\mathbf{Q}(k) + \mathbf{Q}(n+1)\right]\mathbf{U}^0 + \sum_{j=1}^{n} \mathbf{V}(n-j)\sum_{k=1}^{j} \Theta_{j,k}\mathbf{W}(k) + \mathbf{W}(n+1).
\end{aligned}$$

Using the substitution $j = n+1-l$, we obtain

$$\mathbf{U}^{n+1} = \left[\beta_{n+1} + \sum_{l=1}^{n} \mathbf{V}(l-1)\sum_{k=1}^{n+1-l} \Theta_{n+1-l,k}\mathbf{Q}(k) + \mathbf{Q}(n+1)\right]\mathbf{U}^0$$

$$
\begin{aligned}
&+\sum_{l=1}^{n}\mathbf{V}(l-1)\sum_{k=1}^{n+1-l}\Theta_{n+1-l,k}\mathbf{W}(k)+\mathbf{W}(n+1)\\
&=\left[\beta_{n+1}+\sum_{l=1}^{n}\sum_{k=1}^{n+1-l}\mathbf{V}(l-1)\Theta_{n+1-l,k}\mathbf{Q}(k)+\mathbf{Q}(n+1)\right]\mathbf{U}^0\\
&+\sum_{l=1}^{n}\sum_{k=1}^{n+1-l}\mathbf{V}(l-1)\Theta_{n+1-l,k}\mathbf{W}(k)+\mathbf{W}(n+1)\\
&=\left[\beta_{n+1}+\sum_{k=1}^{n}\sum_{l=1}^{n+1-k}\mathbf{V}(l-1)\Theta_{n+1-l,k}\mathbf{Q}(k)+\mathbf{Q}(n+1)\right]\mathbf{U}^0\\
&+\sum_{k=1}^{n}\sum_{l=1}^{n+1-k}\mathbf{V}(l-1)\Theta_{n+1-l,k}\mathbf{W}(k)+\mathbf{W}(n+1)\\
&=\left[\beta_{n+1}+\sum_{k=1}^{n}\left(\sum_{l=1}^{n+1-k}\mathbf{V}(l-1)\Theta_{n+1-l,k}\right)\mathbf{Q}(k)+\mathbf{Q}(n+1)\right]\mathbf{U}^0\\
&+\sum_{k=1}^{n}\left(\sum_{l=1}^{n+1-k}\mathbf{V}(l-1)\Theta_{n+1-l,k}\right)\mathbf{W}(k)+\mathbf{W}(n+1)\\
&=\left[\beta_{n+1}+\sum_{k=1}^{n}\Theta_{n+1,k}\mathbf{Q}(k)+\mathbf{Q}(n+1)\right]\mathbf{U}^0+\sum_{k=1}^{n}\Theta_{n+1,k}\mathbf{W}(k)+\mathbf{W}(n+1)\\
&=\left[\beta_{n+1}+\sum_{k=1}^{n+1}\Theta_{n+1,k}\mathbf{Q}(k)\right]\mathbf{U}^0+\sum_{k=1}^{n+1}\Theta_{n+1,k}\mathbf{W}(k).
\end{aligned}
$$

Finally, lemma is proved using the principles of mathematical induction. □

Lemma 5.3. *Suppose that the expressions of matrices* Ψ, $\mathbf{L}$, $\mathbf{B}(.)$, *and* $\mathbf{V}(.)$ *are defined in Eqs.* (4.9), (4.14), (4.15), *and* (5.5), *respectively, we have the following estimation of* $\mathbf{V}(s)$ *for all* $s \geqslant 0$:

$$\|\mathbf{V}(s)\| \leqslant \left\|\Psi\mathbf{J}^{-1}\right\|\left[d_\alpha(s)-d_\alpha(s+1)\right],\ \forall s\geqslant 0,$$

where

$$\mathbf{L}=\gamma_\alpha\mathbf{J}\quad \textit{and}\quad \mathbf{J}=\mathbf{T}-\delta t^\alpha\Gamma(2-\alpha)\mathbf{A}\mathbf{D}+\frac{\lambda\Gamma(2-\alpha)\delta t^{1+\alpha}}{2}\mathbf{K}(0)\mathbf{T}+\delta t^\alpha\Gamma(2-\alpha)\mathbf{S}.$$

Proof. We have

$$
\begin{aligned}
\mathbf{V}(s)&=\Psi\mathbf{L}^{-1}\mathbf{B}(s)\Psi^{-1}=\Psi\mathbf{L}^{-1}\left[\gamma_\alpha(d_\alpha(s)-d_\alpha(s+1))\mathbf{I}-\lambda\delta t\mathbf{K}\left((s+1)\delta t\right)\right]\mathbf{T}\Psi^{-1}\\
&=\Psi\mathbf{L}^{-1}\left[\gamma_\alpha(d_\alpha(s)-d_\alpha(s+1))\mathbf{I}-\lambda\delta t\mathbf{K}\left((s+1)\delta t\right)\right].
\end{aligned}
$$

Now we apply the norm

$$
\begin{aligned}
\|\mathbf{V}(s)\|&=\left\|\Psi\mathbf{L}^{-1}\left[\gamma_\alpha(d_\alpha(s)-d_\alpha(s+1))\mathbf{I}-\lambda\delta t\mathbf{K}\left((s+1)\delta t\right)\right]\right\|\\
&\leqslant\left\|\Psi\mathbf{L}^{-1}\right\|\left\|\left[\gamma_\alpha(d_\alpha(s)-d_\alpha(s+1))\mathbf{I}-\lambda\delta t\mathbf{K}\left((s+1)\delta t\right)\right]\right\|\\
&\leqslant\left\|\Psi\mathbf{L}^{-1}\right\|\max_{1\leqslant i\leqslant M}\left|\gamma_\alpha(d_\alpha(s)-d_\alpha(s+1))-\lambda\delta t k(x_i,((s+1)\delta t)\right|,
\end{aligned}
$$

where $k>0$ and $d_\alpha(s)-d_\alpha(s+1)\geqslant 0$ for all $s\geqslant 0$, we get

$$\|\mathbf{V}(s)\|\leqslant\gamma_\alpha\left\|\Psi\mathbf{L}^{-1}\right\|\left[d_\alpha(s)-d_\alpha(s+1)\right]=\left\|\Psi\gamma_\alpha\mathbf{L}^{-1}\right\|\left[d_\alpha(s)-d_\alpha(s+1)\right]\leqslant\left\|\Psi\mathbf{J}^{-1}\right\|\left[d_\alpha(s)-d_\alpha(s+1)\right].$$

□

Lemma 5.4. *There exists a positive increasing sequence* θ_n *defined by:*

$$\begin{cases} \theta_n = 1 + \left\|\Psi \mathbf{J}^{-1}\right\| \theta_{n-1}, & n \geqslant 2, \\ \theta_1 = 1. \end{cases} \tag{5.8}$$

Hence we have

$$\sum_{k=1}^{n} \|\Theta_{n,k}\| \leqslant \theta_n, \quad n \geqslant 1. \tag{5.9}$$

Proof. For $n = 1$, we get in Eq. (5.9),

$$\sum_{k=1}^{1} \|\Theta_{1,k}\| = \|\Theta_{1,1}\| = 1 \leqslant 1 = \theta_1.$$

We assume that Eq. (5.9) is true for all $n = 1, \dots, m-1$, where $m = 2, \dots, N$ and we prove that Eq. (5.9) is true for $n = m$, we have

$$\begin{aligned} \sum_{k=1}^{m} \|\Theta_{m,k}\| &= \|\Theta_{m,m}\| + \sum_{k=1}^{m-1} \|\Theta_{m,k}\| \\ &= 1 + \sum_{k=1}^{m-1} \|\Theta_{m,k}\| \\ &= 1 + \sum_{k=1}^{m-1} \left\| \sum_{j=1}^{m-k} \mathbf{V}(j-1)\Theta_{m-j,k} \right\| \\ &\leqslant 1 + \sum_{k=1}^{m-1} \sum_{j=1}^{m-k} \|\mathbf{V}(j-1)\| \left\|\Theta_{m-j,k}\right\| = 1 + \sum_{j=1}^{m-1} \|\mathbf{V}(j-1)\| \sum_{k=1}^{m-j} \left\|\Theta_{m-j,k}\right\|. \end{aligned}$$

Now we apply the estimate of $\mathbf{V}(.)$ in Lemma 5.3 and the increasing sequence θ_n:

$$\begin{aligned} \sum_{k=1}^{m} \|\Theta_{m,k}\| &\leqslant 1 + \left\|\Psi \mathbf{J}^{-1}\right\| \sum_{j=1}^{m-1} \left[d_\alpha(j-1) - d_\alpha(j)\right] \theta_{m-j} \\ &\leqslant 1 + \left\|\Psi \mathbf{J}^{-1}\right\| \sum_{j=1}^{m-1} \left[d_\alpha(j-1) - d_\alpha(j)\right] \theta_{m-1} \\ &= 1 + \left\|\Psi \mathbf{J}^{-1}\right\| \left[d_\alpha(0) - d_\alpha(m-1)\right] \theta_{m-1} \\ &= 1 + \left\|\Psi \mathbf{J}^{-1}\right\| \left[1 - d_\alpha(m-1)\right] \theta_{m-1} \leqslant 1 + \left\|\Psi \mathbf{J}^{-1}\right\| \theta_{m-1} = \theta_m. \end{aligned}$$

Because $0 \leqslant 1 - d_\alpha(x) \leqslant 1$ for all $x \geqslant 0$. Thus Eq. (5.9) holds true for $n = m$. By the principle of induction, Eq. (5.9) is true for all n. □

The MLS method is used to approximate $u^n(\mathbf{x})$, the theoretical analysis of the MLS approximation has been started by Levin [20] and Zuppa [34]. We use the results obtained in [3, 32] for error estimation, stability, and convergence of the MLS method. Let $w \geqslant 0$ be a function such that $\text{supp}(w) \subset \overline{B}(0, r) = \{\mathbf{z}/\|\mathbf{z}\| \leqslant r\}$, where $r > 0$, $\mathbf{X} := \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ be a set of points in $\Omega \subset \mathbb{R}$ and $u_j = u\left(\mathbf{x}_j\right)$, $j = 1, \dots, N$. Let $\{p_1, \dots, p_m\}$ be a set of basis polynomials in the polynomial space with $m \ll n$. After the construction of the MLS method in Section 3, we need to ensure that the minimization problem has a unique solution, which ensures that the matrix $\mathbf{A}(\mathbf{x})$ defined in Eq. (3.4) is non-singular, in order to have a well-defined MLS approximation. We define

$$\langle u, v\rangle_{\mathbf{x}} = \sum_{i=1}^{N} w\left(\mathbf{x}-\mathbf{x}_i\right) u\left(\mathbf{x}_i\right) v\left(\mathbf{x}_i\right),$$

then $\|u\|_{\mathbf{x}} = \sqrt{\langle u, u\rangle_{\mathbf{x}}}$ is a discrete norm on the polynomial space $\mathbb{P}_m$. The error estimations are obtained on the system of nodes and the weights functions by using the following assumption.

Property 5.5 ([33]). *For any $\mathbf{x} \in \overline{\Omega}$, the matrix $A(\mathbf{x})$ defined in Eq. (3.4) is non-singular.*

Definition 5.6 ([33]). Let $\mathbf{x} \in \overline{\Omega}$. The star of $\mathbf{x}$, $\operatorname{st}(\mathbf{x})$ is defined as $\operatorname{st}(\mathbf{x}) = \{i \mid w\left(\mathbf{x}-\mathbf{x}_i\right) \neq 0\}$.

Theorem 5.7 ([33]). *A necessary condition for property 5.5 is that for any $\mathbf{x} \in \overline{\Omega}$,*

$$n = \operatorname{card}(\operatorname{st}(\mathbf{x})) \geqslant \operatorname{card}\left(\mathbb{P}_m\right) = m+1.$$

Theorem 5.8 ([3]). *Assume that $A(\mathbf{x})$ satisfies Property 5.5, then for any $\mathbf{x} \in \bar{\Omega}$, there exists $\hat{u}(\mathbf{x}) \in \mathbb{P}_m$, which satisfies*

$$\|u-\hat{u}(\mathbf{x})\|_{\mathbf{x}} \leqslant \|u-p\|_{\mathbf{x}}, \quad \forall p \in \mathbb{P}_m.$$

The aim is to obtain error estimation in terms of the parameter r, which plays the role of the support size of the weight function. For the error analysis, the following properties of the weight function and distribution of points are required, as introduced in [3].

(1) Given $\mathbf{x} \in \Omega$ there exist at least $m+1$ points $\mathbf{x}_j \in \mathbf{X} \cap B(\mathbf{x}, \frac{r}{2})$.
(2) $\exists c_0 > 0$ such that $w(z) \geqslant c_0, \forall z \in B(0, \frac{r}{2})$.
(3) $w \in C^1\left(B(0,r)\right) \cap W^{1,\infty}(\mathbb{R})$ and $\exists c_1$ such that $\|w'\|_{L^\infty(\mathbb{R})} \leqslant \frac{c_1}{r}$.
(4) $\exists c_p$ such that $\frac{r}{\sigma} \leqslant c_p$, where $\sigma = \min|\mathbf{x}_i - \mathbf{x}_k|$ is the minimum over the $m+1$ points in condition 1.
(5) $\exists c_k$ such that for all $\mathbf{x} \in \Omega$, $\operatorname{card}\{\mathbf{X} \cap B(0, 2r)\} < c_k$.
(6) $w \in C^2\left(B(0,r)\right) \cap W^{2,\infty}(\mathbb{R})$, and $\exists c_2$ such that $\|w''\|_{L^\infty(\mathbb{R})} \leqslant \frac{c_2}{r^2}$.

Theorem 5.9 ([3]). *Let $m \geqslant 1$, if $u \in C^{m+1}(\bar{\Omega})$ and properties* (1)-(6) *hold, then there exists $C = C(c_0, c_1, c_2, c_p, c_k, m)$ such that*

$$\|u'' - \hat{u}''\|_{L^\infty(\Omega)} \leqslant C \left\|u^{(m+1)}\right\|_{L^\infty(\Omega)} r^{m-1}.$$

Obviously, the error of the presented method will be affected by δt and the error of the second derivatives in Theorems 5.9. After applying the above theorems we assume that the scheme expressed in Eq. (5.4) is the q-th order accurate in space.

Theorem 5.10. *The solution $\mathbf{U}^n$of the scheme Eq. (5.4) satisfies*

$$\|\mathbf{E}^n\| \leqslant C\theta_n(\delta t + (\delta t)^{2-\alpha} + (\delta x)^q),$$

for some $C > 0$, where $\mathbf{E}^n = \mathbf{U}^n_{\text{exact}} - \mathbf{U}^n_{\text{app}}$ for all $n \geqslant 1$, $\mathbf{E}^0 = 0$, and θ_n is defined in Lemma 5.4.

Proof. Suppose that the exact solution of Eq. (1.1) at time $n\delta t$ is denoted by $\mathbf{U}^n_{\text{exact}}$ for each $n = 1, \ldots, N$. By the scheme in Eq. (5.4) we obtain:

$$\mathbf{U}^n_{\text{exact}} = \sum_{j=0}^{n-1} \mathbf{V}(n-1-j)\mathbf{U}^j_{\text{exact}} + \mathbf{Q}(n)\mathbf{U}^0 + \mathbf{W}(n) + \mathbf{R}(n), \tag{5.10}$$

and the approximate solution is denoted by:

$$\mathbf{U}_{\text{app}}^n = \sum_{j=0}^{n-1} \mathbf{V}(n-1-j)\mathbf{U}_{\text{app}}^j + \mathbf{Q}(n)\mathbf{U}^0 + \mathbf{W}(n), \tag{5.11}$$

where $\mathbf{R}(n)$ is the rest error at time $n\delta t$ and $\mathbf{U}_{\text{exact}}^0 = \mathbf{U}_{\text{app}}^0 = \mathbf{U}^0$. Now subtracting the Eq. (5.11) from the Eq. (5.10) and applying Lemma 5.2, we get:

$$\mathbf{E}^n = \beta_n \mathbf{E}^0 + \sum_{j=1}^{n} \Theta_{n,j}\mathbf{R}(j) = \sum_{j=1}^{n} \Theta_{n,j}\mathbf{R}(j). \tag{5.12}$$

Using the previous order of errors, the order of fractional derivative in Eq. (4.3), the order of integral part in Eq. (5.3) and the order of MLS approximation, we deduce that it exist a constant $C > 0$ for all $j = 1, \dots, N$ such that:

$$\|\mathbf{R}(j)\| \leqslant C(\delta t + (\delta t)^{2-\alpha} + (\delta x)^q). \tag{5.13}$$

Now applying the norm of Eq. (5.12) and using the inequality in Eq. (5.13),

$$\|\mathbf{E}^n\| \leqslant \sum_{j=1}^{n} \left\|\Theta_{n,j}\right\| \|\mathbf{R}(j)\| \leqslant C\left(\sum_{j=1}^{n} \left\|\Theta_{n,j}\right\|\right)(\delta t + (\delta t)^{2-\alpha} + (\delta x)^q) \leqslant C\theta_n(\delta t + (\delta t)^{2-\alpha} + (\delta x)^q).$$

□

Remark 5.11. The sequence θ_n in Eq. (5.8) can be written by this expression:

$$\theta_n = \sum_{i=0}^{n-1} \left\|\Psi\mathbf{J}^{-1}\right\|^i, \quad n \geqslant 1.$$

Then we have the following two bounded errors:

$$\|\mathbf{E}^n\| \leqslant Cn(\delta t + (\delta t)^{2-\alpha} + (\delta x)^q) \quad \text{if} \quad \left\|\Psi\mathbf{J}^{-1}\right\| = 1$$

or

$$\|\mathbf{E}^n\| \leqslant C\left(\frac{1 - \left\|\Psi\mathbf{J}^{-1}\right\|^n}{1 - \left\|\Psi\mathbf{J}^{-1}\right\|}\right)(\delta t + (\delta t)^{2-\alpha} + (\delta x)^q) \quad \text{if} \quad \left\|\Psi\mathbf{J}^{-1}\right\| \neq 1.$$

Finally, the analysis error of presented method of scheme in Eq. (5.4) is proven.

6. Numerical results

We used the above method to show the advantage and the sharpness of some errors for solving TFPIDE of Volterra. The errors L_∞, L_2, and Root-Mean-Square (RMS) of errors are measured based on the following formulas:

$$L_\infty = \max_{1\leqslant i\leqslant M} |u_i - u_{\text{exact}}\ (x_i)|, \quad L_2 = \sqrt{\sum_{i=1}^{M} |u_i - u_{\text{exact}}\ (x_i)|^2}, \quad \text{RMS} = \sqrt{\frac{1}{M}\left(\sum_{i=1}^{M} |u_i - u_{\text{exact}}\ (x_i)|^2\right)}.$$

We used the quadratic basis for the construction of shape function. The choice of support size for all weights functions is $r = 2.5\frac{L}{M}$ and we choose $\mu = \frac{L}{M}$ on Gaussian weight function for all problems.

Example 6.1. Consider the time fractional partial integro-differential equation of Volterra type:

$$\begin{cases} {}_0^C\mathcal{D}_t^{\alpha}u(x,t) - \partial_x^2 u(x,t) + \displaystyle\int_0^t x(t-s)u(x,s)ds = f(x,t), & (x,t) \in \Omega := (0,1) \times (0,1], \\ u(x,0) = 0, \quad \forall x \in [0,1], \\ u(0,t) = t + t^{\alpha} \text{ and } u(1,t) = 0, \quad \forall t \in (0,1], \end{cases}$$

where

$$f(x,t) = \left(1-x^2\right)\left(\frac{t^{1-\alpha}}{\Gamma(2-\alpha)} + \Gamma(1+\alpha)\right) + 2\left(t+t^{\alpha}\right) + x\left(1-x^2\right)\left(\frac{t^3}{6} + \frac{t^{2+\alpha}}{(1+\alpha)(2+\alpha)}\right),$$

and the exact solution (see [29]) of the above problem is $u(x,t) = (1-x^2)(t^{\alpha}+t)$.

Table 1: Comparison of methods for Example 6.1 using GWF with $t = 0.5$ and L_∞ error.

	Method of [29] in Table 2 with $M = N$			Proposed method with $M = 11$		
N	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
64	7.7015e − 5	1.0674e − 4	1.4035e − 4	7.6352e − 05	1.0567e − 04	1.3912e − 04
128	3.1966e − 5	4.5264e − 5	6.3712e − 5	3.1656e − 05	4.4847e − 05	6.3170e − 05
256	1.3638e − 5	1.9694e − 5	2.9251e − 5	1.3488e − 05	1.9525e − 05	2.9012e − 05
512	5.9723e − 6	8.7732e − 6	1.3547e − 5	5.8979e − 06	8.7019e − 06	1.3439e − 05
1024	2.6791e − 6	3.9924e − 6	6.3183e − 6	2.6417e − 06	3.9609e − 06	6.2692e − 06

Table 2: Different values of L_∞-error for Example 6.1 using GWF.

		$t = 0.5$			$t = 1$		
N	M	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
6	6	2.0000e − 03	2.7000e − 03	2.5000e − 03	2.4388e − 04	2.8955e − 04	4.5416e − 04
64	7	7.5905e − 05	1.0618e − 04	1.3986e − 04	1.3178e − 05	1.3405e − 05	3.2528e − 05
128	8	3.1938e − 05	4.5077e − 05	6.3423e − 05	5.8983e − 06	4.9165e − 06	1.4368e − 05
512	9	5.8913e − 06	8.7112e − 06	1.3453e − 05	1.4077e − 06	6.1253e − 07	2.5540e − 06
1024	10	2.6759e − 06	3.9781e − 06	6.2965e − 06	6.7279e − 07	2.0794e − 07	1.0479e − 06
2500	11	9.7078e − 07	1.4764e − 06	2.3691e − 06	2.7258e − 07	9.4940e − 08	3.2526e − 07
3000	12	8.0542e − 07	1.2183e − 06	1.9553e − 06	2.2060e − 07	7.5728e − 08	2.6205e − 07

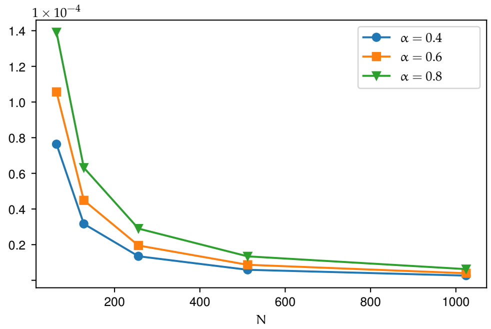

Figure 2: The L_∞-error at $t = 0.5$, where $M = 11$ with different values of α in Example 6.1.

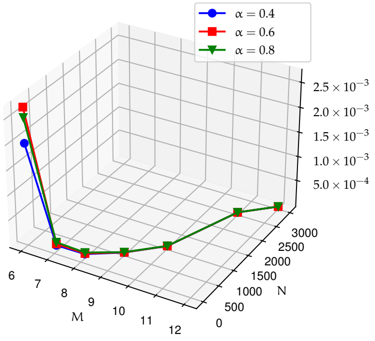

Figure 3: The L_∞-error at $t = 0.5$ with different values of α in Example 6.1.

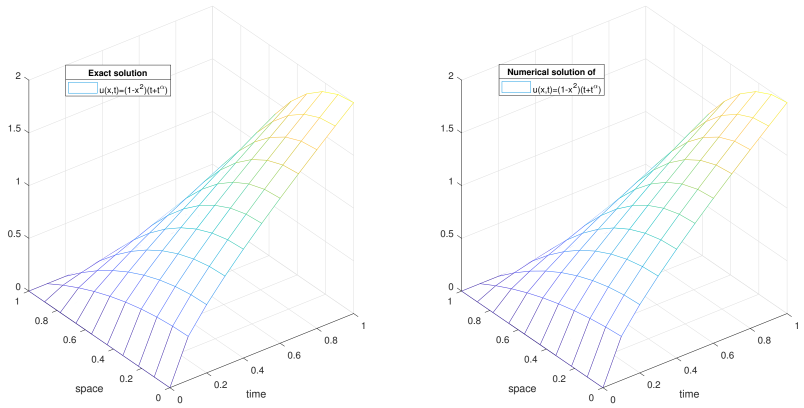

Figure 4: The exact and numerical solution for Example 6.1 with $M = N = 10$ and $\alpha = 0.5$.

In Example 6.1, the results are compared with the method in [29] based on finite difference method (FDM). The L_∞-error are reported in the Table 1 at $t = 0.5$ with different values of α and N, For the (FDM) chosen $N = M$ but in the proposed method we choose only $M = 11$ with the same values of N in [29], this results give better accuracy than those obtained in [29]. The L_∞-error at different values of α and t are reported in Table 2. Also the L_∞-error when $\alpha = 0.4,\ 0.6,\ 0.8$ are shown in Figs. 2 and 3 for $t = 0.5$ with different values of N and M, which is more sharp for small values of α. The approximate solution and the exact solution are illustrated in Fig. 4, where $\alpha = 0.5$ and $M = N = 10$, which confirms the convergence and the theoretical findings.

Example 6.2. In this example, considering the test problem to see the applicability of this method:

$$\begin{cases} {}_0^C\mathcal{D}_t^\alpha u(x,t) - \exp(x)\partial_x^2 u(x,t) + \displaystyle\int_0^t x^2(t-s)^3 u(x,s)\,ds = f(x,t), & (x,t) \in \Omega := (0,1) \times (0,1], \\ u(x,0) = 0, \quad \forall x \in [0,1], \\ u(0,t) = t^{\alpha+1} + t^{2\alpha+2} \text{ and } u(1,t) = (\cos(1)+1)\left(t^{\alpha+1} + t^{2\alpha+2}\right), \quad \forall t \in (0,1], \end{cases}$$

where

$$\begin{aligned} f(x,t) = {}& (\cos(x)+x^2)\left(\frac{2^{2\alpha+2}t^{\alpha+2}\Gamma(\frac{3}{2}+\alpha)}{\sqrt{\pi}(2+\alpha)} + t\Gamma(\alpha+2)\right) - \exp(x)(2-\cos(x))t^{\alpha+1}(1+t^{\alpha+1}) \\ & + 3(\cos(x)+x^2)x^2\left(\frac{(16\alpha^2+64\alpha+60)t^{\alpha+5} + (\alpha^2+9\alpha+20)t^{2\alpha+6}}{8\alpha^6+144\alpha^5+1046\alpha^4+3924\alpha^3+8018\alpha^2+8460\alpha+3600}\right), \end{aligned}$$

and the exact solution is $u(x,t) = (\cos(x)+x^2)t^{\alpha+1}(1+t^{\alpha+1})$.

Table 3: Comparison of errors for Example 6.2 using GWF, where $M = 20$ and $\delta t = 0.0005$.

	$\alpha = 0.1$			$\alpha = 0.3$		
t	L_∞-error	L_2-error	RMS-error	L_∞-error	L_2-error	RMS-error
0.1	$8.9482e-07$	$2.5376e-06$	$5.6742e-07$	$5.6778e-07$	$1.3018e-06$	$2.9109e-07$
0.2	$2.0769e-06$	$5.9428e-06$	$1.3289e-06$	$1.4636e-06$	$3.8330e-06$	$8.5709e-07$
0.3	$3.5082e-06$	$1.0072e-05$	$2.2522e-06$	$2.6541e-06$	$7.1989e-06$	$1.6097e-06$
0.4	$5.1894e-06$	$1.4929e-05$	$3.3382e-06$	$4.1503e-06$	$1.1446e-05$	$2.5593e-06$
0.5	$7.1255e-06$	$2.0527e-05$	$4.5900e-06$	$5.9738e-06$	$1.6640e-05$	$3.7208e-06$

Table 4: Comparison of errors for Example 6.2 using GWF, where $\delta t = 0.01$, $t = 0.1$.

	$\alpha = 0.1$			$\alpha = 0.2$		
M	L_∞-error	L_2-error	RMS-error	L_∞-error	L_2-error	RMS-error
6	$3.3325e-05$	$3.9546e-05$	$1.6145e-05$	$2.8583e-05$	$3.6180e-05$	$1.4770e-05$
9	$1.2601e-05$	$1.5966e-05$	$5.3221e-06$	$1.1701e-05$	$1.7678e-05$	$5.8926e-06$
12	$5.5395e-06$	$7.7223e-06$	$2.2292e-06$	$5.7391e-06$	$1.3767e-05$	$3.9741e-06$
15	$2.9205e-06$	$4.6663e-06$	$1.2048e-06$	$4.9269e-06$	$1.4491e-05$	$3.7416e-06$

Table 5: Comparison of errors for Example 6.2 using GWF, where $t = 0.1$ and $M = 23$.

	$\alpha = 0.1$			$\alpha = 0.2$		
N	L_∞-error	L_2-error	RMS-error	L_∞-error	L_2-error	RMS-error
10	$9.4335e-05$	$3.2541e-04$	$6.7852e-05$	$2.9893e-04$	$1.0000e-03$	$2.1503e-04$
50	$5.2021e-06$	$1.8222e-05$	$3.7996e-06$	$1.8608e-05$	$6.4328e-05$	$1.3413e-05$
120	$8.6328e-07$	$3.0893e-06$	$6.4416e-07$	$3.8863e-06$	$1.3623e-05$	$2.8406e-06$
160	$7.8372e-07$	$1.6742e-06$	$3.4909e-07$	$2.2285e-06$	$7.9335e-06$	$1.6542e-06$
200	$7.4460e-07$	$1.2400e-06$	$2.5855e-07$	$1.4149e-06$	$5.1165e-06$	$1.0669e-06$

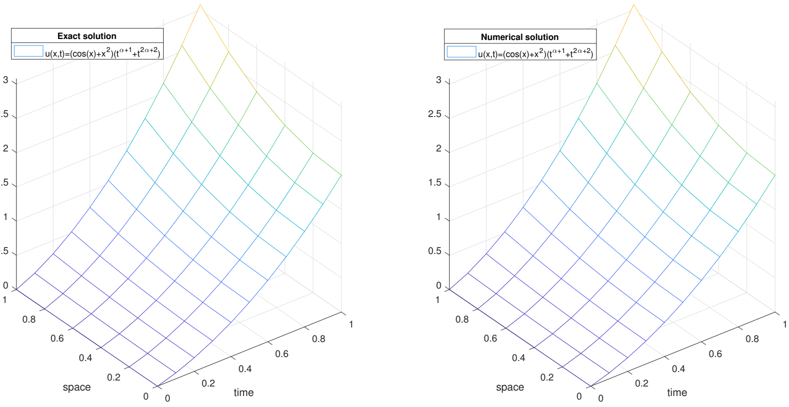

Figure 5: The exact and numerical solution for Example 6.2 with $\alpha = 0.1$, $M = 6$ and $N = 10$.

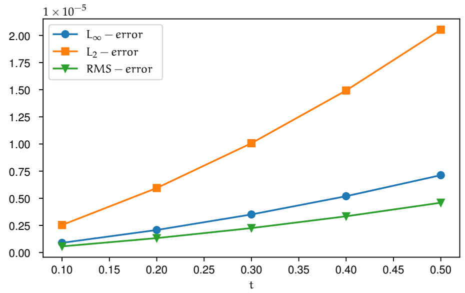

Figure 6: The error values at $\alpha = 0.1$, where $\delta t = 0.0005$ and $M = 20$ for Example 6.2 using GWF.

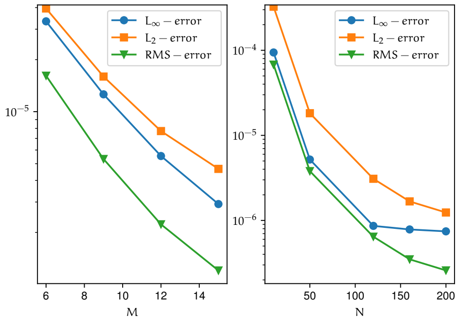

Figure 7: A comparison errors at $t = 0.1$ and $\alpha = 0.1$, where $\delta t = 0.01$ in (left) and $M = 23$ in (right) for Example 6.2 using GWF.

The numerical solution and the corresponding exact solution are illustrated in Fig. 5, where $\alpha = 0.1$, $N = 10$, and $M = 6$. The method provides good results with a small number of nodes in space. The L_∞, L_2 and RMS of errors at different values of t and α, where $\delta t = 0.0005$ are shown in Table 3. The different values of errors are presented in the Tables 4 and 5 at $t = 0.1$ and $\alpha = 0.1, 0.2$ when M and N increase, respectively. Graphs of those errors L_∞, L_2, and RMS are shown in Fig. 7, where $t = \alpha = 0.1$. The results converge to the exact values by increasing the numbers N and M. Fig. 6 confirms that the errors increase as $t \longrightarrow T$.

Example 6.3. We consider the following time fractional partial integro-differential problem:

$$\begin{cases} {}_0^C\mathcal{D}_t^\alpha u(x,t) - (1+\log(x+1))\partial_x^2 u(x,t) + \int_0^t \cos(x+t-s)u(x,s)ds = f(x,t), \quad (x,t)\in\Omega := (0,1)\times(0,T],\\ u(x,0)=\sqrt{x+1}+1, \quad \forall x\in[0,1],\\ u(0,t)=1+(t+1)^3 \text{ and } u(1,t)=\sqrt{2}+(t+1)^3, \quad \forall t\in(0,T],\end{cases}$$

where

$$f(x,t) = \frac{-3(\alpha^2-2\alpha t+2t^2-5\alpha+6t+6)t^{1-\alpha}}{\Gamma(1-\alpha)(\alpha^3-6\alpha^2+11\alpha-6)} + \frac{1+\log(1+x)}{4(x+1)\sqrt{x+1}} + 3\cos(x)(t^2+2t+\cos(t)-\frac{5}{3}\sin(t)-1)$$
$$+ ((\cos(t)-1)\sin(x)+\cos(x)\sin(t))\sqrt{x+1} + (-t^3-3t^2+3t-5\cos(t)-3\sin(t)+5)\sin(x),$$

and the exact solution is $u(x,t) = \sqrt{x+1} + (t+1)^3$.

Table 6: Comparison of errors for Example 6.3 using GWF, where $M = 23$ and $\delta t = 0.0001$ with $T = 2$.

	$\alpha = 0.3$			$\alpha = 0.7$		
t	L_∞-error	L_2-error	RMS-error	L_∞-error	L_2-error	RMS-error
0.1	3.9954e − 06	1.2158e − 05	2.5352e − 06	2.3151e − 05	7.7981e − 05	1.6260e − 05
0.5	5.6630e − 06	1.6649e − 05	3.4716e − 06	3.4939e − 05	1.1748e − 04	2.4497e − 05
1	9.0379e − 06	2.5886e − 05	5.3976e − 06	4.8305e − 05	1.6172e − 04	3.3722e − 05
1.5	1.4462e − 05	4.0949e − 05	8.5385e − 06	6.3716e − 05	2.1182e − 04	4.4168e − 05
2	2.2479e − 05	6.3348e − 05	1.3209e − 05	8.2240e − 05	2.7029e − 04	5.6358e − 05

Table 7: Comparison of weights functions and errors for Example 6.3, where $\alpha = 0.5$ and $t = 0.1$ with $T = 1$.

		Gaussian weight function		Cubic spline weight function		Quartic spline weight function	
N	M	L_2-error	RMS-error	L_2-error	RMS-error	L_2-error	RMS-error
100	6	8.9065e − 05	5.8545e − 05	1.8488e − 04	1.2098e − 04	2.9415e − 04	1.9198e − 04
200	9	4.7481e − 05	3.1758e − 05	1.0967e − 04	7.3394e − 05	1.4715e − 04	9.6265e − 05
600	15	1.2188e − 05	8.5246e − 06	3.1067e − 05	2.1189e − 05	3.4445e − 05	2.3523e − 05
800	18	7.4679e − 06	5.1989e − 06	1.9897e − 05	1.3656e − 05	2.0622e − 05	1.4394e − 05

Table 8: Comparison of errors for Example 6.3 using GWF, where $t = 0.1$ and $T = 1$.

		$\alpha = 0.3$			$\alpha = 0.7$		
N	M	L_∞-error	L_2-error	RMS-error	L_∞-error	L_2-error	RMS-error
10	6	8.4148e − 04	1.4000e − 03	5.6962e − 04	5.8000e − 03	9.8000e − 03	4.0000e − 03
100	9	3.8735e − 05	7.6270e − 05	2.5423e − 05	3.9251e − 04	8.0324e − 04	2.6775e − 04
200	12	1.9147e − 05	3.9557e − 05	1.1419e − 05	1.6428e − 04	3.9766e − 04	1.1480e − 04
300	15	9.9702e − 06	2.5425e − 05	6.5647e − 06	9.9308e − 05	2.7014e − 04	6.9750e − 05
500	18	6.0061e − 06	1.3976e − 05	3.2942e − 06	5.1542e − 05	1.5447e − 04	3.6409e − 05

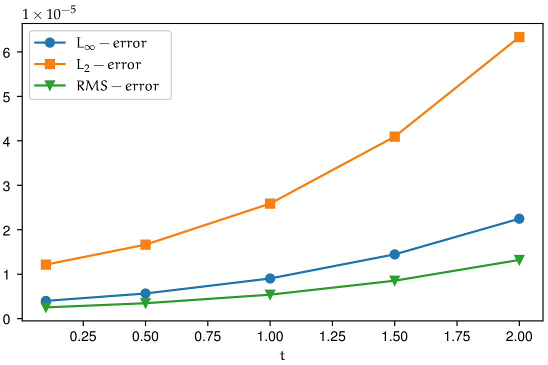

Figure 8: A comparison of errors at $\alpha = 0.3$, where $\delta t = 0.0001$ and $M = 23$ for Example 6.3.

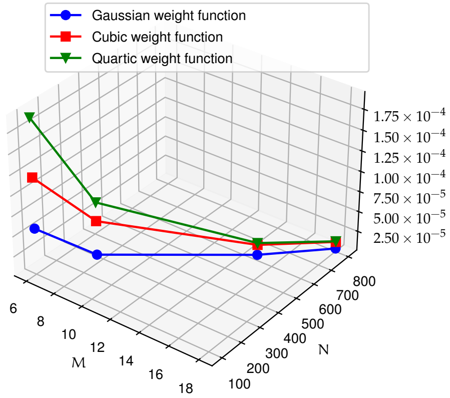

Figure 9: A comparison of weights functions at $t = 0.1$, where $\alpha = 0.5$ and $T = 2$ for RMS-error in Example 6.3.

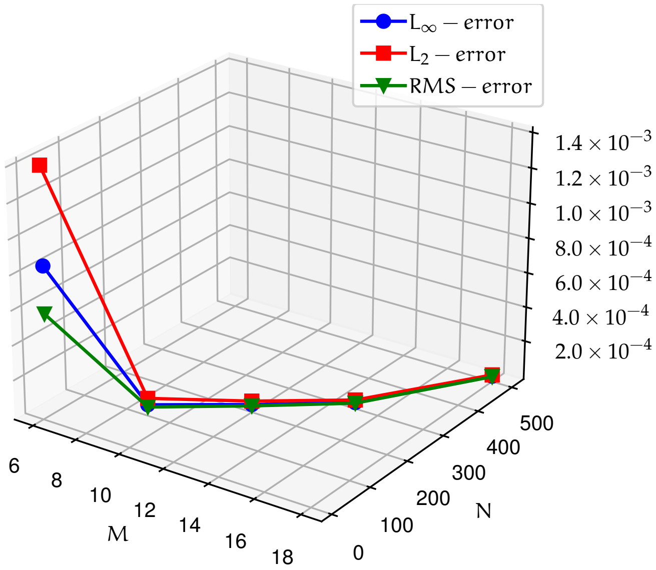

Figure 10: The error values at $t = 0.1$, where $\alpha = 0.3$ and $T = 1$ for Example 6.3.

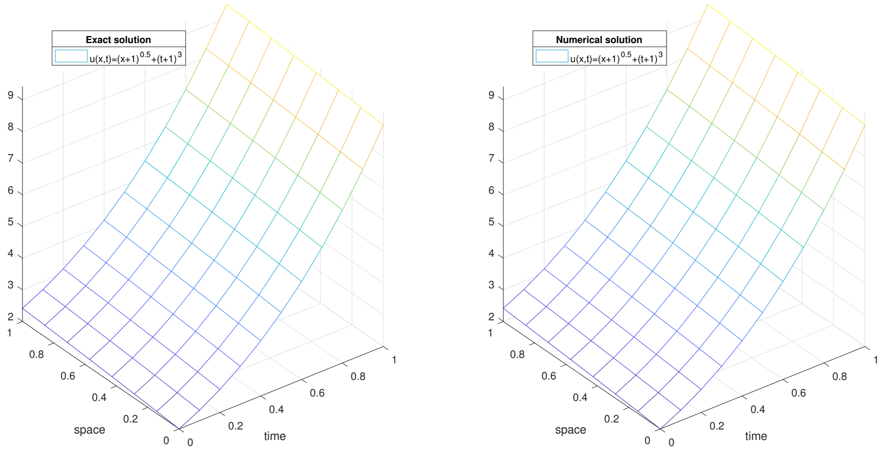

Figure 11: The exact and numerical solution for Example 6.3 with $\alpha = 0.5$, $M = 8$ and $N = 10$.

The L_∞-error, L_2-error, and RMS-error are shown in Table 6 at different values of t and α, where

$\delta t = 0.0001$ and $M = 23$. Fig. 8 displays the errors L_∞, L_2, and RMS at different values of t, where α $M = 23$ and $\delta t = 0.0001$. These numerical results provides the method with high accuracy in different values of α. Similarly, Table 7 and Fig. 9 display the comparison between the weights functions at different values of M and N when $\alpha = 0.5$ and $t = 0.1$, it is observed that the errors by Gaussian weight function is better than other weights. The numerical solution and the exact solution for Example 6.3 are shown in Fig. 11 with $\alpha = 0.5$, $M = 8$, and $N = 10$. It is observed that the numerical solutions converge to the exact solutions by using Gaussian weight function with only a small number of nodes. Also the errors are presented in Table 8 and Fig. 10 when $t = 0.1$ with one value in figure and two values of α in table. The results show that better accuracy is obtained with the present method by increasing values of M and N.

Example 6.4. Similarly we consider the following test problem:

$$\begin{cases} {}_0^C\mathcal{D}_t^\alpha u(x,t) - \partial_x^2 u(x,t) + \displaystyle\int_0^t x(t-s)^2 u(x,s)ds = f(x,t), \quad (x,t) \in \Omega := (0,1) \times (0,1], \\ u(x,0) = 0, \quad \forall x \in [0,1], \\ u(0,t) = \cos(1)t^{2\alpha+1} \text{ and } u(1,t) = \cos(\exp(1))t^{2\alpha+1}, \quad \forall t \in (0,1], \end{cases}$$

where the source function f is given by:

$$f(x,t) = \frac{\cos(\exp(x))2^{2\alpha+1}t^{\alpha+1}\Gamma(\alpha+\frac{3}{2})}{\sqrt{\pi}(\alpha+1)} + (\exp(x)\sin(\exp(x)) \\ + \exp(2x)\cos(\exp(x)))t^{2\alpha+1} + \frac{\cos(\exp(x))xt^{2\alpha+4}}{4\alpha^3+18\alpha^2+26\alpha+12},$$

and the exact solution for this choice of f is $u(x,t) = \cos(\exp(x))t^{2\alpha+1}$.

Table 9: Comparison errors for Example 6.4 using GWF, where $t = 0.2$.

		$\alpha = 0.2$			$\alpha = 0.4$		
N	M	L_∞-error	L_2-error	RMS-error	L_∞-error	L_2-error	RMS-error
10	6	1.0000e − 03	1.2000e − 03	5.0340e − 04	3.7841e − 04	4.4347e − 04	1.8105e − 04
100	12	1.5488e − 04	2.0668e − 04	5.9663e − 05	7.8292e − 05	1.0528e − 04	3.0391e − 05
200	20	3.2944e − 05	8.1689e − 05	1.8266e − 05	1.6790e − 05	4.2113e − 05	9.4167e − 06
300	30	1.0138e − 05	3.5818e − 05	6.5395e − 06	5.2803e − 06	1.8980e − 05	3.4652e − 06
500	43	6.4091e − 06	1.1545e − 05	1.7606e − 06	3.4882e − 06	6.1683e − 06	9.4066e − 07

Table 10: Comparison of errors for Example 6.4 using GWF, where $\delta t = 0.01$ and $M = 43$.

	$\alpha = 0.2$			$\alpha = 0.4$		
t	L_∞-error	L_2-error	RMS-error	L_∞-error	L_2-error	RMS-error
0.2	6.6992e − 06	1.1903e − 05	1.8152e − 06	5.4549e − 06	1.9137e − 05	2.9184e − 06
0.4	1.7084e − 05	3.0522e − 05	4.6545e − 06	1.3253e − 05	2.6405e − 05	4.0268e − 06
0.6	2.9941e − 05	5.3816e − 05	8.2069e − 06	2.5715e − 05	4.6028e − 05	7.0193e − 06
0.8	4.4687e − 05	8.0529e − 05	1.2281e − 05	4.2105e − 05	7.4614e − 05	1.1378e − 05
1	6.1008e − 05	1.1009e − 04	1.6788e − 05	6.2184e − 05	1.1050e − 04	1.6850e − 05

Table 11: Comparison of weights functions and errors for Example 6.4, where $\alpha = 0.5$, $\delta t = 0.001$, and $M = 37$.

	Gaussian weight function		Cubic spline weight function		Quartic spline weight function	
t	L_∞-error	RMS-error	L_∞-error	RMS-error	L_∞-error	RMS-error
0.1	$5.3889e-07$	$3.3906e-07$	$2.7744e-06$	$1.8449e-06$	$2.4267e-06$	$1.5487e-06$
0.3	$4.7419e-06$	$2.4799e-06$	$2.6280e-05$	$1.7950e-05$	$2.1493e-05$	$1.4399e-05$
0.5	$1.3279e-05$	$6.9968e-06$	$7.6026e-05$	$5.2161e-05$	$6.1098e-05$	$4.1465e-05$
0.9	$4.3125e-05$	$2.3259e-05$	$2.5673e-04$	$1.7651e-04$	$2.0305e-04$	$1.3935e-04$

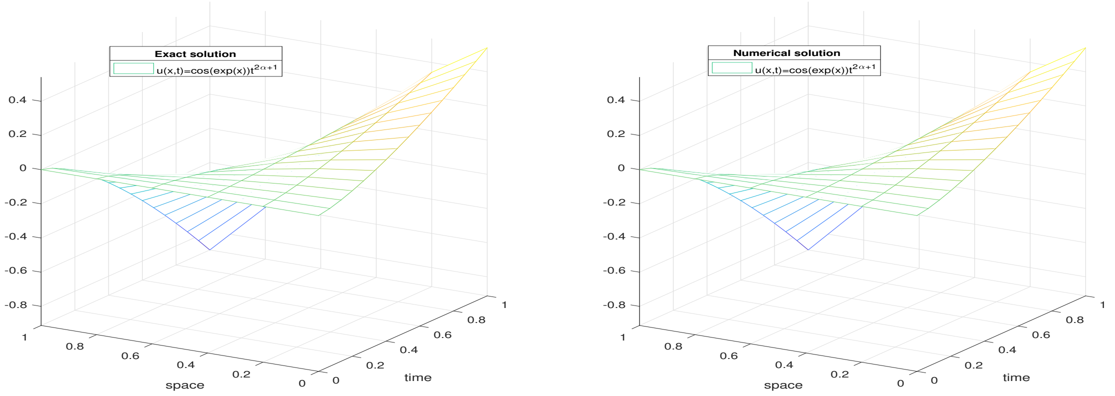

Figure 12: The exact and numerical solution for Example 6.4 with $\alpha = 0.2$, $M = 6$ and $N = 15$.

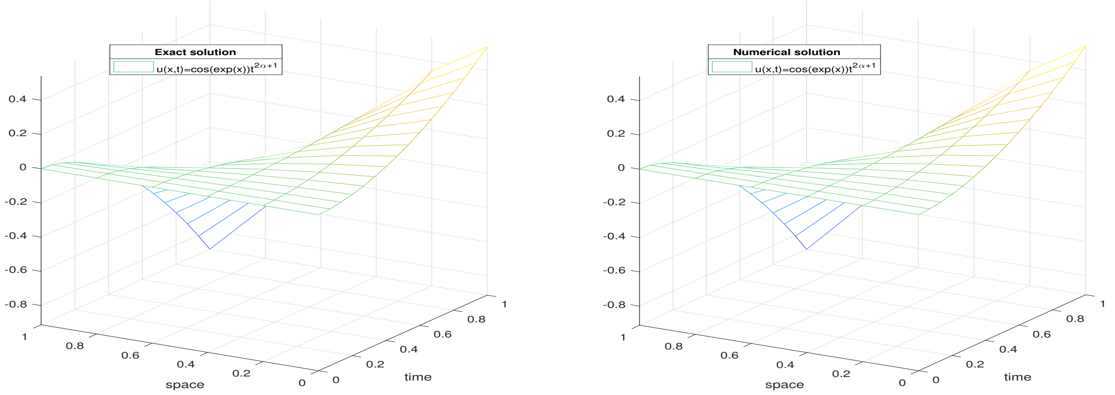

Figure 13: The exact and numerical solution for Example 6.4 with $\alpha = 0.4$, $M = 6$ and $N = 15$.

The L_∞-error, L_2-error and RMS-error are shown in Table 9 at $t = 0.2$, where N and M increase, Table 10 show the errors at different values of t and α, where $M = 43$ and $\delta t = 0.01$. The obtained errors results confirms that the proposed method provides the approximate solutions with high accuracy. The numerical solutions and the exact solutions for Example 6.4 are shown in Figs. 12 and 13 with $\alpha = 0.2$ and $\alpha = 0.4$, respectively. It is observed that the numerical solutions converge more quickly to the exact solutions by using Gaussian weight function. Also the comparison between the weights functions at different values of t, where $M = 37$ and $\delta t = 0.01$ presented in Table 11 which reveals that the Gaussian weight function results are more accurate than cubic and quartic weights functions. The errors results reported in Fig. 14 approve that the errors decrease along with the increase of M and N, this confirms the convergence of the method.

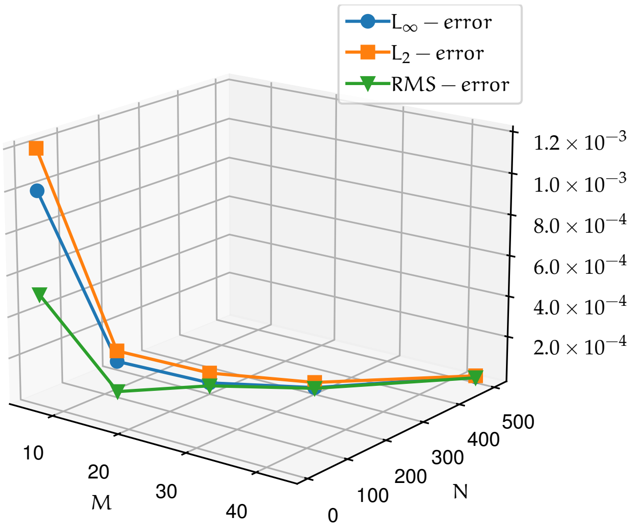

Figure 14: A comparison of errors for Example 6.4 with $\alpha = 0.2$ and $t = 0.5$.

7. Conclusion

In this study, we solve a TFPIDE of Volterra type by using the MLS method. The finite difference scheme is used to discretize the time fractional derivative. The second partial derivative and the integral part are approximated by MLS and the composite trapezoidal approximation, respectively. The error analysis is studied by more matrices sequences and it is shown that the numerical scheme is of order $\mathcal{O}(\delta t + (\delta t)^{2-\alpha})$ in time and of order $\mathcal{O}(h^q)$ over the entire domain. The numerical test problems reveal the applicability and the high accuracy of the proposed method on using less points in space. Experimentally the MLS approximations give better accuracy when we choose the Gaussian weight function. As a future research direction, this method can be extended for other kind of fractional partial integro-differential equations (FPIDEs) in a rectangular or a non-rectangular domain $\Omega \subset \mathbb{R}^d$ $(d \geqslant 2)$.

References

[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C. T. Silva, *Computing and rendering point set surfaces*, IEEE Trans. Vis. Comput. Graph., **9** (2003), 3–15. 1

[2] M. G. Armentano, *Error estimates in Sobolev spaces for moving least square approximations*, SIAM J. Numer. Anal., **39** (2001), 38–51. 1

[3] M. G. Armentano, R. G. Durán, *Error estimates for moving least square approximations*, Appl. Numer. Math., **37** (2001), 397–416. 1, 5, 5.8, 5, 5.9

[4] A. G. Atta, Y. H. Youssri, *Advanced shifted first-kind Chebyshev collocation approach for solving the nonlinear time-fractional partial integro-differential equation with a weakly singular kernel*, Comput. Appl. Math., **41** (2022), 19 pages. 1

[5] D. Baleanu, S. Arshad, A. Jajarmi, W. Shokat, F. Akhavan Ghassabzade, M. Wali, *Dynamical behaviours and stability analysis of a generalized fractional model with a real case study*, J. Adv. Res., **48** (2023), 157–173. 1

[6] D. Baleanu, M. Hasanabadi, A. Mahmoudzadeh Vaziri, A. Jajarmi, *A new intervention strategy for an HIV/AIDS transmission by a general fractional modelling and an optimal control approach*, Chaos Solitons Fractals, **167** (2023). 1

[7] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, *Meshless methods: An overview and recent developments*, Comput. Methods Appl. Mech. Eng., **139** (1996), 3–47. 1

[8] R. L. Bagley, R. A. Calico, *Fractional order state equations for the control of viscoelasticallydamped structures*, J. Guid. Control Dyn., **14** (1991), 304–311. 1

[9] O. Defterli, D. Baleanu, A. Jajarmi, S. S. Sajjadi, N. Alshaikh, J. H. Asad, *Fractional treatment: an accelerated mass-spring system*, Rom. J. Phys., **74** (2022), 1–13. 1

[10] K. Diethelm, *The analysis of fractional differential equations*, Springer-Verlag, Berlin, (2010). 2

[11] Z. El Majouti, R. El Jid, A. Hajjaj, *Numerical solution of two-dimensional Fredholm-Hammerstein integral equations on 2D irregular domains by using modified moving least-square method*, Int. J. Comput. Math., **98** (2021), 1574–1593. 1

[12] Z. El Majouti, R. El Jid, A. Hajjaj, *Solving Two-dimensional Linear and Nonlinear Mixed Integral Equations Using Moving Least Squares and Modified Moving Least Squares Methods*, IAENG Int. J. Appl. Math., **51** (2021), 57–65.

[13] Z. El Majouti, R. El Jid, A. Hajjaj, *Numerical solution for three-dimensional nonlinear mixed Volterra-Fredholm integral equations via modified moving least-square method*, Int. J. Comput. Math., **99** (2022), 1849–1867. 1

[14] F. Fakhar-Izadi, *Fully spectral-Galerkin method for the one- and two-dimensional fourth-order time-fractional partial integro-differential equations with a weakly singular kernel*, Numer. Methods Partial Differ. Equ., **38** (2022), 160–176. 1

[15] F. Ge, Y. Chen, C. Kou, *Regional analysis of time-fractional diffusion processes*, Springer, Cham, (2018). 2

[16] M. H. Heydari, H. Laeli Dastjerdi, M. Nili Ahmadabadi, *An efficient method for the numerical solution of a class of nonlinear fractional Fredholm integro-differential equations*, Int. J. Nonlinear Sci. Numer. Simul., **19** (2018), 165–173. 1

[17] A. Jajarmi, S. Arshad, D. Baleanu, *A new fractional modelling and control strategy for the outbreak of dengue fever*, Phys. A, **535** (2019), 14 pages. 1

[18] A. Jajarmi, B. Ghanbari, D. Baleanu, *A new and efficient numerical method for the fractional modeling and optimal control of diabetes and tuberculosis co-existence*, Chaos: An Interdisciplinary Journal of Nonlinear Science, **29** (2019), 15 pages. 1

[19] P. Lancaster, K. Salkauskas, *Surfaces generated by moving least squares methods*, Math. Comp., **37** (1981), 141–158. 1

[20] D. Levin, *The Approximation Power of Moving Least-Squares*, Math. Comp., **67** (1998), 1517–1531. 1, 5

[21] X. Li, S. Li, *On the stability of the moving least squares approximation and the element-free Galerkin method*, Comput. Math. Appl., **72** (2016), 1515–1531. 3

[22] G. R. Liu, Y. T. Gu, *An Introduction to Meshfree Methods and Their Programming*, Springer, New York, (2005). 1

[23] Z. Luo, X. Zhang, S. Wang, L. Yao, *Numerical approximation of time fractional partial integro-differential equation based on compact finite difference scheme*, Chaos Solitons Fractals, **161** (2022), 8 pages. 1

[24] D. Mirzaei, *Analysis of moving least squares approximation revisited*, J. Comput. Appl. Math., **282** (2015), 237–250. 1

[25] M. Mirzazadeh, M. Eslami, B. S. Ahmed, A. Biswas, *Dynamics of Population Growth Model With Fractional Temporal Evolution*, Life Sci. J., **11** (2014), 224–227. 1

[26] H. K. Pathak, *An introduction to nonlinear analysis and fixed point theory*, Springer, Singapore, (2018). 2.3

[27] I. Podlubny, *Fractional differential equations*, Academic Press, San Diego, CA, (1999). 2

[28] H. Ren, K. Pei, L. Wang, *Error analysis for moving least squares approximation in 2D space*, Appl. Math. Comput., **238** (2014), 527–546. 1

[29] S. Santra, J. Mohapatra, *A novel finite difference technique with error estimate for time fractional partial integro-differential equation of Volterra type*, J. Comput. Appl. Math., **400** (2022), 13 pages. 2, 6.1, 1, 6

[30] D. Shepard, *A two-dimensional interpolation function for irregularly-spaced data*, Proceedings of the 1968 23rd ACM national conference, (1968), 517–524. 1

[31] W. Y. Tey, N. A. C. Sidik, Y. Asako, M. W Muhieldeen, O. Afshar, *Moving Least Squares Method and its Improvement: A Concise Review*, J. Appl. Comput. Mech., **7** (2021), 883–889. 1

[32] M. Uddin, S. Haq, *RBFs approximation method for time fractional partial differential equations*, Commun. Nonlinear Sci. Numer. Simul., **16** (2011), 4208–4214. 5

[33] C. Zuppa, *Error estimates for moving least square approximations*, Bull. Braz. Math. Soc. (N.S.), **34** (2003), 231–249. 1, 5.5, 5.6, 5.7

[34] C. Zuppa, *Good quality point sets and error estimates for moving least square approximations*, Appl. Numer. Math., **47** (2003), 575–585. 5