Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 11, No. 4, 2023, pp. 716-726
DOI:10.22034/cmde.2023.55070.2286

A numerical approximation for the solution of a time–fractional telegraph equation by the moving least squares approach

Haniye Hajinezhad[1,*] **and Ali R. Soheili**[2]

[1]Department of Mathematics, Payame Noor University, Tehran, Iran.

[2]Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran.

Abstract

This paper focuses on the numerical solution of the time-fractional telegraph equation in Caputo sense with $1 < \beta < 2$. The time-fractional telegraph equation models neutron transport inside the core of a nuclear reactor. The proposed numerical solution consists of two stages. First, the time-discretized scheme of this equation is obtained by the Crank-Nicolson method. The stability and convergence of results from the semi-discretized scheme are presented. In the second stage, the numerical approximation of the unknown function at specific points is achieved through the collocation method using the moving least square method. The numerical experiments analyze the impact of some parameters of the proposed method.

Keywords. Time-Fractional telegraph equation, Moving least squares method, Stability, Convergence.

2010 Mathematics Subject Classification. 35R11, 65M06, 65M70.

1. Introduction

The classical telegraph equation is used in random walk theory [2] and is approximated by some authors [1, 13]. In recent decades, many studies have addressed fractional partial differential equations [6, 9, 17, 18]. The following partial differential equation is the time-fractional telegraph equation (TFTE) [7].

$$\frac{\partial^{\beta} u(x,y,t)}{\partial t^{\beta}} + \frac{\partial^{\beta-1} u(x,y,t)}{\partial t^{\beta-1}} + u(x,y,t) = \Delta u(x,y,t) + f(x,y,t), \quad (x,y) \in \Omega \subset \mathbb{R}^2, 0 < t \leq T. \tag{1.1}$$

It simulates the neutron transport process in a nuclear reactor [22]. In this work, equation (1) with the initial and boundary conditions

$$u(x,y,0) = \varphi(x,y), \qquad (x,y) \in \overline{\Omega} = \Omega \cup \partial\Omega, \tag{1.2}$$

$$\frac{\partial u(x,y,0)}{\partial t} = \psi(x,y), \qquad (x,y) \in \overline{\Omega} = \Omega \cup \partial\Omega, \tag{1.3}$$

$$u(x,y,t) = h(x,y,t), \qquad (x,y) \in \partial\Omega, \quad t > 0, \tag{1.4}$$

is considered. Where $u(x,y,t) \in C^2(\bar{\Omega} \times [0,T])$ is an unknown function, $\partial\Omega$ is the boundary of Ω, $1 < \beta < 2, \Delta$ is the Laplace operator, and $f(x,y), \varphi(x,y), \psi(x,y)$, and $h(x,y,t)$ are continuous functions. The fractional derivatives are

Received: 23 January 2023 ; Accepted: 06 April 2023.
$*$ Corresponding author. Email: H.Hajinezhad@pnu.ac.ir, haniye_hajinezhad@yahoo.com.

defined as follows.

$$\frac{\partial^{\beta-1}u(x,y,t)}{\partial t^{\beta-1}} = \frac{1}{\Gamma(2-\beta)}\int_0^t \frac{\partial u(x,y,s)}{\partial s}(t-s)^{1-\beta}ds, \qquad 1<\beta<2, \tag{1.5}$$

$$\frac{\partial^{\beta}u(x,y,t)}{\partial t^{\beta}} = \frac{1}{\Gamma(2-\beta)}\int_0^t \frac{\partial^2 u(x,y,s)}{\partial s^2}(t-s)^{1-\beta}ds, \qquad 1<\beta<2. \tag{1.6}$$

In recent years, many authors have studied the TFTE using mesh-free methods. Kumar et al. [7] used a time semi-discretization based on the finite difference method followed by a radial basis function for the spatial discretization to solve the TFTE. Nikan et al. [14] considered a mesh-free spectral approach based on the local radial basis function finite difference (LRBF-FD) to approximate the TFTE. Sepehrian and Shamohammadi [16] used a radial basis function collocation method to solve the nonlinear TFTE. Shivanian used the meshless local Petrov Galerkin (MLPG) scheme in [20], and the spectral meshless radial point interpolation (SMRPI) methods in [19] to solve the TFTE. Hosseini et al. [5] approximated the solution of the TFTE by the meshless local radial point interpolation (MLRPI) method. Mohebbi et al. [12] applied the radial basis function (RBF) technique for the TFTE. Sweilam et al. [21] used the shifted Chebyshev polynomials of the first kind in time and the Sinc function in space to approximate the solution of the TFTE.

When attempting to model real-world phenomena with fractional differential equations, the Riemann -Liouville derivative may present certain drawbacks, while Caputo's definition presents a modification of the Riemann-Liouville definition, and it possesses the benefit of suitably handling initial value problems [15]. Therefore, to estimate the solution of the time-fractional telegraph equation, the Caputo derivative is preferred. As the authors know, there is no research to approximate the solution of the time-fractional telegraph equation (1.1) using a hybrid of the finite difference method and the collocation method base on the moving least square (MLS) method. In this work, the time discretization of equation (1.1) is given by the Crank-Nicolson method. The stability and convergence of the semi-discretized scheme are obtained easily. Then the full discretization of equation (1.1) using the MLS method is presented, followed by the numerical tests for evaluating the proposed scheme. In the numerical tests, we evaluate the sensitivity of the approximate solution by choosing different values for some parameters of the proposed method, including the number of collocation nodes, the time step size, and the support domain's dimensionless size in the X and Y directions. According to the results of numerical tests, we suggest suitable values for them.

The outline of this work is as follows. Section 2 is dedicated to the semi-discretization of equation (1.1). In this section, the stability and convergence of the proposed semi-discretization scheme are addressed. In Section 3, the collocation method with the MLS method in detail, including the shape function and the weight function, is described. The full discretization of equation (1.1) is presented in Section 4. In Section 5, we analyze numerical test results to validate the proposed method. Finally, we summarize our findings and suggest future work in Section 6.

2. The time semi-discretization

In this section, we develop and analyze a semi–discrete scheme for equation (1.1) based on the Crank–Nicolson method by using our previous work [4].
Consider the grid size in time is $\Delta t = \frac{T}{N}$, where N is a positive integer. By using reference [4], the discretization of equations (1.5) and (1.6) at the time steps $\left(1-\frac{1}{2}\right)$ and $\left(n-\frac{1}{2}\right)$ are as follows.

$$\frac{\partial^{\beta-1}u(x,y,t)}{\partial t^{\beta-1}}\Big|^{1-\frac{1}{2}} = \frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left[u^1-u^0\right]+O(\Delta t)^{3-\beta}, \tag{2.1}$$

$$\begin{aligned}\frac{\partial^{\beta-1}u(x,y,t)}{\partial t^{\beta-1}}\Big|^{n-\frac{1}{2}} = {}& \frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}u^0-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})u^k-\left(\frac{1}{2^{2-\beta}}-b_1\right)u^{n-1}+\frac{1}{2^{2-\beta}}u^n\right\}\\ &+O(\Delta t)^{3-\beta}, \qquad n\geq 2,\end{aligned} \tag{2.2}$$

$$\left.\frac{\partial^{\beta}u(x,y,t)}{\partial t^{\beta}}\right|^{1-\frac{1}{2}}=\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left[\frac{u^{1}-u^{0}}{\Delta t}-\left.\frac{\partial u}{\partial t}\right|^{0}\right]+O(\Delta t)^{2-\beta}, \tag{2.3}$$

$$\begin{aligned}\frac{\partial^{\beta}u(x,y,t)}{\partial t^{\beta}}|^{n-\frac{1}{2}}=\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}&\left\{-b_{n-1}\left.\frac{\partial u}{\partial t}\right|^{0}-\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\frac{u^{k}-u^{k-1}}{\Delta t}\right.\\&\left.-(\frac{1}{2^{2-\beta}}-b_{1})\frac{u^{n-1}-u^{n-2}}{\Delta t}+\frac{1}{2^{2-\beta}}\frac{u^{n}-u^{n-1}}{\Delta t}\right\}+O(\Delta t)^{2-\beta},\qquad n\geq 2,\end{aligned} \tag{2.4}$$

where $b_s=\left(s+\frac{1}{2}\right)^{2-\beta}-\left(s-\frac{1}{2}\right)^{2-\beta},\quad s=1,2,\ldots,\quad 1<\beta<2$. By using Taylor expansion, we have

$$u(x,y,t_{n-\frac{1}{2}})=\frac{u^{n-1}+u^{n}}{2}+O(\Delta t)^{2},\qquad n\geq 1, \tag{2.5}$$

$$\Delta u(x,y,t_{n-\frac{1}{2}})=\frac{\Delta u^{n-1}+\Delta u^{n}}{2}+O(\Delta t)^{2},\qquad n\geq 1. \tag{2.6}$$

Using the relations (2.1), (2.3), (2.5), and (2.6), the discretization of equation (1.1) at the time step $\left(1-\frac{1}{2}\right)$ is as follows.

$$\begin{aligned}\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left[\frac{u^{1}-u^{0}}{\Delta t}-\left.\frac{\partial u}{\partial t}\right|^{0}\right]+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left(u^{1}-u^{0}\right)+\frac{1}{2}\left(u^{1}+u^{0}\right)\\=\frac{\Delta u^{1}+\Delta u^{0}}{2}+f^{1-\frac{1}{2}}+O(\Delta t)^{2-\beta}.\end{aligned} \tag{2.7}$$

Using the relations (2.2), (2.4), (2.5), and (2.6), the discretization of equation (1.1) at the time step $\left(n-\frac{1}{2}\right)$ for $n\geq 2$ is as follows.

$$\begin{aligned}&\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}\left.\frac{\partial u}{\partial t}\right|^{0}-\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)\frac{u^{k}-u^{k-1}}{\Delta t}-(\frac{1}{2^{2-\beta}}-b_{1})\frac{u^{n-1}-u^{n-2}}{\Delta t}+\frac{1}{2^{2-\beta}}\frac{u^{n}-u^{n-1}}{\Delta t}\right\}\\&+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}u^{0}-\sum_{k=1}^{n-2}\left(b_{n-k-1}-b_{n-k}\right)u^{k}-(\frac{1}{2^{2-\beta}}-b_{1})u^{n-1}+\frac{1}{2^{2-\beta}}u^{n}\right\}+\frac{u^{n-1}+u^{n}}{2}\\&=\frac{\Delta u^{n-1}+\Delta u^{n}}{2}+f^{n-\frac{1}{2}}+O(\Delta t)^{2-\beta},\qquad n\geq 2.\end{aligned} \tag{2.8}$$

Neglecting the truncation errors, equations (2.7) and (2.8) are as follows.

$$\begin{aligned}\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left[\frac{u^{1}-u^{0}}{\Delta t}-\left.\frac{\partial u}{\partial t}\right|^{0}\right]+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left(u^{1}-u^{0}\right)+\frac{1}{2}\left(u^{1}+u^{0}\right)\\=\frac{\Delta u^{1}+\Delta u^{0}}{2}+f^{1-\frac{1}{2}},\end{aligned} \tag{2.9}$$

and

$$\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}\frac{\partial u}{\partial t}\Big|^{0}-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,\frac{u^{k}-u^{k-1}}{\Delta t}-(\frac{1}{2^{2-\beta}}-b_1)\frac{u^{n-1}-u^{n-2}}{\Delta t}+\frac{1}{2^{2-\beta}}\frac{u^{n}-u^{n-1}}{\Delta t}\right\}$$
$$+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}u^{0}-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,u^{k}-(\frac{1}{2^{2-\beta}}-b_1)u^{n-1}+\frac{1}{2^{2-\beta}}u^{n}\right\}+\frac{u^{n-1}+u^{n}}{2}$$
$$=\frac{\Delta u^{n-1}+\Delta u^{n}}{2}+f^{n-\frac{1}{2}},\qquad n\geq 2, \tag{2.10}$$

respectively. Now, we consider the stability of this semi–discrete scheme. Let $\widetilde{u}^n$ $(n\geq 1)$ be the approximate solution of equations (2.9) and (2.10) with respect to the round-off error, and u^n $(n\geq 1)$ be the exact solution of equations (2.9) and (2.10). Define

$$e^{n}=u^{n}-\widetilde{u}^{n},\quad (n=0,1,\ldots).$$

We obtain the following round-off error equations.

$$\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\left\{\left[\frac{e^{1}-e^{0}}{\Delta t}-\delta e^{0}\right]+(e^{1}-e^{0})\right\}+\frac{1}{2}(e^{1}+e^{0})=\frac{1}{2}(\Delta e^{1}+\Delta e^{0}), \tag{2.11}$$

$$\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}\delta e^{0}-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,\frac{e^{k}-e^{k-1}}{\Delta t}-(\frac{1}{2^{2-\beta}}-b_1)\frac{e^{n-1}-e^{n-2}}{\Delta t}+\frac{1}{2^{2-\beta}}\frac{e^{n}-e^{n-1}}{\Delta t}\right\}$$
$$+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-b_{n-1}e^{0}-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,e^{k}-(\frac{1}{2^{2-\beta}}-b_1)e^{n-1}\right. \tag{2.12}$$
$$\left.+\frac{1}{2^{2-\beta}}e^{n}\right\}+\frac{e^{n-1}+e^{n}}{2}=\frac{\Delta e^{n-1}+\Delta e^{n}}{2},\qquad n\geq 2,$$

where $\delta e^{0}=\frac{\partial u}{\partial t}\Big|^{0}-\frac{\partial\widetilde{u}}{\partial t}\Big|^{0}$. Round-off error equations (2.11) and (2.12) in this work are the same as round-off error equations (19) and (20) in reference [4]. Therefore, similar to theorem 5 in reference [4] the following theorem can be easily proved.

Theorem 2.1. *If $e^k\in H_0^1(\Omega)$, then the solutions of the finite difference approaches (2.9) and (2.10) are unconditionally stable.*

Now, we consider the convergence of our proposed semi–discrete scheme. Let $u^n(n\geq 1)$ be the exact solution of (2.9) and (2.10), and let U^n $(n\geq 1)$ be the exact solution of (2.7) and (2.8). Define $\xi^n=U^n-u^n$, $(n\geq 1)$. Then, we obtain

$$\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\frac{\xi^{1}}{\Delta t}+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\times\frac{1}{2^{2-\beta}}\xi^{1}+\frac{1}{2}\xi^{1}=\frac{\Delta\xi^{1}}{2}+O(\Delta t)^{2-\beta}, \tag{2.13}$$

and

$$\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\frac{\xi^{k}-\xi^{k-1}}{\Delta t}-(\frac{1}{2^{2-\beta}}-b_1)\frac{\xi^{n-1}-\xi^{n-2}}{\Delta t}+\frac{1}{2^{2-\beta}}\frac{\xi^{n}-\xi^{n-1}}{\Delta t}\right\}$$
$$+\frac{(\Delta t)^{1-\beta}}{\Gamma(3-\beta)}\left\{-\sum_{k=1}^{n-2}(b_{n-k-1}-b_{n-k})\,\xi^{k}-(\frac{1}{2^{2-\beta}}-b_1)\xi^{n-1}+\frac{1}{2^{2-\beta}}\xi^{n}\right\}+\frac{\xi^{n-1}+\xi^{n}}{2} \tag{2.14}$$
$$=\frac{\Delta\xi^{n-1}+\Delta\xi^{n}}{2}+O(\Delta t)^{2-\beta},\quad n\geq 2.$$

Equations (2.13) and (2.14), in this work, are the same as equations (34) and (35) in reference [4], except for error terms. The right side of equations (34) and (35) in reference [4] contains $(O(\Delta x)^2 + O(\Delta y)^2 + O(\Delta t)^{(2-\beta)})$, and the right side of equations (2.13) and (2.14) in this work contains $O(\Delta t)^{(2-\beta)}$. Therefore, similar to theorem 6 in reference [4] the following theorem can be easily proved.

Theorem 2.2. *If $\xi^k \in H_0^1(\Omega)$, then the solutions of the finite difference approaches (2.9) and (2.10) are unconditionally convergent.*

3. The collocation method

In this section, the collocation method is explained generally. To approximate the solution of partial differential equations, using the mesh-free collocation method, the partial differential equation is usually discretized at collocation nodes by a form of collocation method. The collocation points could be different from the field nodes, but they are the same in our work.

In a collocation method, to approximate the unknown function and its derivatives at the collocation nodes $\mathbf{x} \in \Omega$ ($\mathbf{x}^T = [x, y]$), the following formulae are used.

$$\begin{aligned}\widehat{u}_{\mathbf{x}} &= \phi^T(\mathbf{x}).\mathbf{U}_{s_{\mathbf{x}}},\\ \frac{\partial^2 \widehat{u}_{\mathbf{x}}}{\partial x^2} &= \frac{\partial^2 \phi^T(\mathbf{x})}{\partial x^2}.\mathbf{U}_{s_{\mathbf{x}}},\\ \frac{\partial^2 \widehat{u}_{\mathbf{x}}}{\partial y^2} &= \frac{\partial^2 \phi^T(\mathbf{x})}{\partial y^2}.\mathbf{U}_{s_{\mathbf{x}}},\end{aligned} \tag{3.1}$$

where $\widehat{u}_{\mathbf{x}}$ is the approximation of u at node $\mathbf{x}$, $\mathbf{U}_{s_{\mathbf{x}}}$ is a vector of values u at nodes of support domain of $\mathbf{x}$, including $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n_{\mathbf{x}}}$, and ϕ is the vector of shape functions (The shape function is explained in subsection 3.1.) as follows.

$$\begin{aligned}\phi^T(\mathbf{x}) &= [\phi_1(\mathbf{x}) \quad \phi_2(\mathbf{x}) \quad ... \quad \phi_{n_{\mathbf{x}}}(\mathbf{x})],\\ \mathbf{U}^T_{s_{\mathbf{x}}} &= [u_{\mathbf{x}_1} \quad u_{\mathbf{x}_2} \quad ... \quad u_{\mathbf{x}_{n_{\mathbf{x}}}}],\end{aligned}$$

which $n_{\mathbf{x}}$ is the number of nodes in the support domain of interest node $\mathbf{x}$.

3.1. **The shape function.** This subsection explains the shape function and its derivatives for the MLS method. The MLS method was described by [8]. In the MLS method, the unknown function u at $\mathbf{x} \in \Omega$ is defined as follows.

$$\widehat{u}_{\mathbf{x}} = \Sigma_{j=1}^m p_j(\mathbf{x}).a_j(\mathbf{x}) = \mathbf{P}^T(\mathbf{x}).\mathbf{a}(\mathbf{x}), \tag{3.2}$$

where $\mathbf{x}^T = [x, y]$, $\mathbf{P}(\mathbf{x})$ is a vector of basis functions as

$$\mathbf{P}^T(\mathbf{x}) = [p_1(\mathbf{x}) \quad p_2(\mathbf{x}) \quad ... \quad p_m(\mathbf{x})],$$

$\mathbf{a}(\mathbf{x})$ is a vector of coefficients as

$$\mathbf{a}^T(\mathbf{x}) = [a_1(\mathbf{x}) \quad a_2(\mathbf{x}) \quad ... \quad a_m(\mathbf{x})],$$

and m is the number of the basis functions.
The coefficient $\mathbf{a}$ is obtained by minimizing the following weighted residual.

$$\mathbf{J} = \Sigma_{i=1}^{n_{\mathbf{x}}} \widehat{W}_i(\mathbf{x})[\mathbf{P}^T(\mathbf{x}_i).\mathbf{a}(\mathbf{x}) - u_i]^2,$$

where $\widehat{W}_i$ is a weight function (The weight function is described in subsection 3.2), $\mathbf{x}_i \quad (i = 1, 2, ..., n_{\mathbf{x}})$ is a node in the support domain of $\mathbf{x}$, $n_{\mathbf{x}}$ is the number of nodes in the support domain of $\mathbf{x}$ that the weight functions $\widehat{W}_i \neq 0$, u_i is the value of u at node $\mathbf{x}_i$.
The stationarity of J with respect to $\mathbf{a}(\mathbf{x})$ is

$$\frac{\partial \mathbf{J}}{\partial \mathbf{a}} = 0 \Rightarrow \Sigma_{i=1}^{n_{\mathbf{x}}} \widehat{W}_i(\mathbf{x}).\mathbf{P}^T(\mathbf{x}_i).[\mathbf{P}^T(\mathbf{x}_i).\mathbf{a}(\mathbf{x}) - u_i] = 0,$$

which results in

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{U}_{s_\mathbf{x}}, \tag{3.3}$$

where

$$\begin{aligned}
\mathbf{U}_{s_\mathbf{x}} &= [u_{\mathbf{x}_1} \quad u_{\mathbf{x}_2} \quad \dots \quad u_{n_\mathbf{x}}]^T,\\
\mathbf{A}(\mathbf{x}) &= \Sigma_{i=1}^{n_\mathbf{x}} \widehat{W}_i(\mathbf{x}).\mathbf{P}(\mathbf{x}_i).\mathbf{P}^T(\mathbf{x}_i),\\
\mathbf{B}(\mathbf{x}) &= [\widehat{W}_1(\mathbf{x})\mathbf{P}(\mathbf{x}_1) \quad \widehat{W}_2(\mathbf{x})\mathbf{P}(\mathbf{x}_2) \quad \dots \quad \widehat{W}_{n_\mathbf{x}}(\mathbf{x})\mathbf{P}(\mathbf{x}_{n_\mathbf{x}})].
\end{aligned}$$

Solving (3.3) for $\mathbf{a}(\mathbf{x})$ gives

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x}).\mathbf{B}(\mathbf{x}).\mathbf{U}_{s_\mathbf{x}}, \tag{3.4}$$

then by substituting $\mathbf{a}(\mathbf{x})$ in relation (3.2), we have

$$\widehat{u}(\mathbf{x}) = \phi^T.\mathbf{U}_{s_\mathbf{x}},$$

where

$$\phi^T = \mathbf{P}^T(\mathbf{x}).\mathbf{A}^{-1}(\mathbf{x}).\mathbf{B}(\mathbf{x}) = [\phi_1(\mathbf{x}) \quad \phi_2(\mathbf{x}) \quad \dots \quad \phi_{n_\mathbf{x}}(\mathbf{x})]. \tag{3.5}$$

For obtaining the partial derivatives of ϕ^T, by using euation (3.5), we have [3]

$$\phi^T = \gamma^T(\mathbf{x}).\mathbf{B}(\mathbf{x}),$$

where $\mathbf{A}.\gamma = \mathbf{P}$. Therefore we have [10]

$$\begin{aligned}
\phi^T_{xx} &= \gamma^T_{xx}.\mathbf{B} + 2\gamma^T_x.\mathbf{B}_x + \gamma^T.\mathbf{B}_{xx},\\
\phi^T_{yy} &= \gamma^T_{yy}.\mathbf{B} + 2\gamma^T_y.\mathbf{B}_y + \gamma^T.\mathbf{B}_{yy}.
\end{aligned}$$

3.2. **The weight function.** In this subsection, we describe our weight function. In this work, the rectangular support domains are used and the weight function is as follows [10].

$$\widehat{W}_i(\mathbf{x}) = \widehat{W}(r_{ix}).\widehat{W}(r_{iy}),$$

where

$$r_{ix} = \frac{|x - x_i|}{d_{s_x}}, \qquad r_{iy} = \frac{|y - y_i|}{d_{s_y}},$$

where

$$d_{s_x} = \alpha_{s_x}.d_{c_x}, \qquad d_{s_y} = \alpha_{s_y}.d_{c_y},$$

in which α_{s_x} and α_{s_y} are the dimensionless size of the support domain in the X and Y direction. When the nodes are uniformly distributed, d_{c_x} and d_{c_y} are distances between two neighboring nodes in the X and Y directions, respectively. We use the quadratic spline weight function which is [11]

$$\widehat{W}(r_i) = \begin{cases} 1 - 6r_i^2 + 8r_i^3 - 3r_i^4, & r_i \le 1,\\ 0, & r_i > 1. \end{cases}$$

4. The full discretization

In this section we want to present the full discretization of equation (1.1) with initial and boundary conditions (1.2)-(1.4). The time discretization of equation (1.1) with initial conditions (1.2) and (1.3) is given in relations (2.9) and (2.10). The time discretization of equation (1.4) is assumed as follows.

$$u^n(x,y) = h(x,y,n\Delta t), \qquad (x,y) \in \partial\Omega. \tag{4.1}$$

As it is explained in section 3.1, we have

$$\begin{aligned} \widehat{u}^n_{x_I} &= \phi^T \mathbf{U}^n_{s_{x_I}}, \qquad n \geq 1, \quad I = 1,2,...,n_d, \\ \Delta\widehat{u}^n_{x_I} &= \Delta\phi^T \mathbf{U}^n_{s_{x_I}}, \qquad n \geq 1, \quad I = 1,2,...,n_d, \end{aligned} \tag{4.2}$$

where n is the number of the time steps, x_I is a collocation node (which the value of u at x_I is unknown.), n_d is the number of collocation nodes, $\widehat{u}^n_{x_I}$ is the approximation value of u in x_I and at time step n, ϕ is the shape function. For $n = 1$, for all nodes $x_I (I = 1,2,...,n_d)$, by substituting (4.2) in equation (2.9) and using (4.1) the following system of equations resulted.

$$\mathbf{K}^1 \mathbf{U}^1_S = F^1, \tag{4.3}$$

where $\mathbf{U}^1_S$ is a $n_d \times 1$ vector of unknown values of the function u at collocation points at time step 1, $\mathbf{K}$ is a $n_d \times n_d$ coefficient matrix at time step 1, and F is a $n_d \times 1$ vector of known values. Solving the system of equations (4.3) results in the unknown $\mathbf{U}^1_S$. Similar to (4.3), for $n \geq 2$, for all nodes $x_I (I = 1,2,...,n_d)$, by substitution (4.2) in equation (2.10) and using (4.1), we have the following system of equations.

$$\mathbf{K}^n \mathbf{U}^n_S = F^n, \qquad n \geq 2, \tag{4.4}$$

where $\mathbf{U}^n_S$ is a $n_d \times 1$ vector of unknown values of the function u at collocation points at time step n, $\mathbf{K}$ is a $n_d \times n_d$ coefficient matrix at time step n, and F is a $n_d \times 1$ vector of known values. Solving the system of equations (4.4) results in the unknown $\mathbf{U}^n_S$ for $n \geq 2$.

5. Numerical experiments

In this section, some numerical tests are presented to analyze our proposed method. To evaluate the accuracy of our proposed scheme, we use the maximum absolute error as follows.

$$L_\infty = \max_{1 \leq i \leq I, 1 \leq j \leq J} \left| \tilde{U}_{i,j}(T) - U_{i,j}(T) \right|,$$

where $\tilde{U}_{i,j}(T)$ and $U_{i,j}(T)$ denote the numerical solution and the exact solution of equation (1.1) with initial and boundary conditions (1.2)-(1.4) at (x_i, y_j) and time T, respectively.

Example 5.1. Consider $u(x,y,t) = t^4 \sin(\pi x + \pi y)$ [7] is the exact solution of equation (1.1) on $\Omega = [0,1] \times [0,1]$ with

$$f(x,t) = \left(\frac{24t^{4-\beta}}{\Gamma(5-\beta)} + \frac{24t^{5-\beta}}{\Gamma(6-\beta)} + 2t^4\pi^2 \right) \sin(\pi x + \pi y) + t^4 \sin(\pi x + \pi y).$$

The function $\varphi(x,y)$, $\psi(x,y)$, and $h(x,y)$ in the initial and boundary conditions (1.2)-(1.4) are known using this exact solution.

For this example, we choose the complete polynomial basis of order 6 ($m = 6$) as follows.

$$\mathbf{P}^T(\mathbf{x}) = [1 \quad x \quad y \quad x^2 \quad y^2 \quad xy].$$

Figure 1 displays the rectangular domain $\Omega = [0,1]\times[0,1]$ with uniform points and a node with its rectangular support domain. In the following tests, the uniform points with rectangular support domain are used.
In the MLS approximation, the number of nodes in a support domain is usually larger than the number of unknown coefficients m [10]. If $1 < \alpha_{s_x}, \alpha_{s_y} \leq 1.5$, then there are nine points in the support domain of every x_I. Therefore, the number of nodes in a support domain is larger than the number of unknown coefficients ($m = 6$). If $2 < \alpha_{s_x}, \alpha_{s_y} < 3$, then the number of nodes in a support domain is 25. According to our experience, it does not give valid results. Reference [10] states that $\alpha_{s_x} = \alpha_{s_y} \in [2,3]$ results in a good approximation for many problems. In the following tests, we show that $1 < \alpha_{s_x}, \alpha_{s_y} < 1.5$ are suitable. Also, we show that in some tests, $2.1 < \alpha_{s_x}, \alpha_{s_y} \leq 2.9$ are not suitable. The next tests show that the size of the time step and the number of collocation points are effective for an accurate solution. The following experiments demonstrate that a higher number of collocation points on Ω is required for smaller time steps.

Test 1. In this example, our proposed method is considered by 11×11 uniform points on Ω and with $\beta = 1.3$ and $\beta = 1.9$. Table 1 shows with the different values α_{s_x}, α_{s_y} ($1 < \alpha_{s_x}, \alpha_{s_y} \leq 1.5$), different values Δt, and $\beta = 1.3$, the maximum absolute errors are small at $T = 1.0$. Table 2 shows with different values α_{s_x}, α_{s_y} ($1 < \alpha_{s_x}, \alpha_{s_y} \leq 1.5$), different values Δt, and $\beta = 1.9$, the maximum absolute errors are small at $T = 1.0$ except for $\alpha_{s_x} = \alpha_{s_y} = 1.5$ and $\Delta t = \frac{1}{160}$. For $\alpha_{s_x} = \alpha_{s_y} = 1.5$ and $\Delta t = \frac{1}{160}$, the maximum absolute error is larger than 1. Increasing the number of time steps theoretically increases the accuracy, but numerically choosing an appropriate time step is necessary. The numerical computations need to be more accurate for smaller time steps, and hence we need a larger number of collocation points. We repeat this test by assuming 21×21 uniform points on Ω and $\beta = 1.9$. Table 3 shows by assuming 21×21 uniform points on Ω and $\beta = 1.9$ for different values Δt, α_{s_x}, and α_{s_y} ($1 < \alpha_{s_x}, \alpha_{s_y} \leq 1.5$), the maximum absolute errors are small enough. These tests reveal that an increase in the number of collocation points on Ω is necessary for accurate results with smaller time steps.

Test 2. This example is based on our proposed method by 21×21 uniform nodes on Ω, $\Delta t = \frac{1}{80}$, $\beta = 1.9$, and different values α_{s_x} and α_{s_y}. Table 4 shows for $1 < \alpha_{s_x} = \alpha_{s_y} \leq 1.5$ and $\alpha_{s_x} = \alpha_{s_y} = 2.1$, the maximum absolute errors are small enough but for $2.1 < \alpha_{s_x} = \alpha_{s_y} \leq 2.9$, the maximum absolute errors are extremely large. The following test shows that with suitable $\alpha_{s_x}, \alpha_{s_y}, \Delta t$, and a suitable number of collocation points on Ω, the proposed method is sufficiently accurate.

Test 3. Kumar et al. [7] considered this example with $\beta = 1.7$ and 1.9 by assuming 2025 uniform points on Ω with different values Δt using a local meshless method. Now, this test is repeated by our proposed method. We assume $\alpha_{s_x} = \alpha_{s_y} = 1.1$. Table 5 presents the maximum absolute errors with $\beta = 1.7$ and 1.9, obtained by Kumar et al. [7] and our proposed method at time $T = 1.0$. As this table shows, the errors of these two methods are very close.
According to Tests 1, 2, and 3, for our proposed method, it is recommended to choose $1 < \alpha_{s_x} = \alpha_{s_y} < 1.5$, and to increase the number of collocation points for smaller time steps to achieve accurate approximations.

Table 1. The maximum absolute errors with 11×11 uniform points on Ω, $\beta = 1.3$, different values α_{s_x}, α_{s_y}, and different values Δt, at $T = 1.0$.

$\alpha_{s_x}, \alpha_{s_x}$	$\Delta t = \frac{1}{10}$	$\Delta t = \frac{1}{20}$	$\Delta t = \frac{1}{40}$	$\Delta t = \frac{1}{80}$	$\Delta t = \frac{1}{160}$
1.1	$1.1052e-2$	$9.4889e-3$	$6.9677e-3$	$5.2616e-3$	$4.2948e-3$
1.2	$1.0418e-2$	$8.8640e-3$	$6.3530e-3$	$4.6576e-3$	$3.7075e-3$
1.3	$9.4537e-3$	$7.9162e-3$	$5.4287e-3$	$3.7686e-3$	$2.8907e-3$
1.4	$8.8707e-3$	$7.3515e-3$	$4.8975e-3$	$3.3054e-3$	$2.5854e-3$
1.5	$8.8958e-3$	$7.4554e-3$	$5.0415e-3$	$3.5485e-3$	$3.0725e-3$

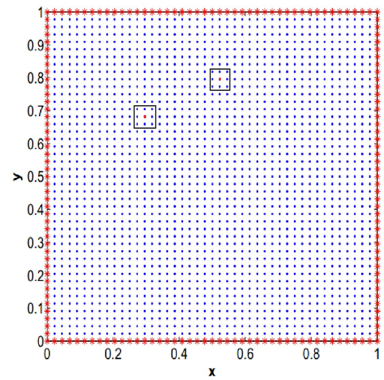

FIGURE 1. Rectangular domain with uniform points on Ω and rectangular support domain with nine points.

TABLE 2. The maximum absolute errors with 11×11 uniform points on Ω, $\beta = 1.9$, different values α_{s_x}, α_{s_y} , and different values Δt, at $T = 1.0$.

$\alpha_{s_x}, \alpha_{s_x}$	$\Delta t = \frac{1}{10}$	$\Delta t = \frac{1}{20}$	$\Delta t = \frac{1}{40}$	$\Delta t = \frac{1}{80}$	$\Delta t = \frac{1}{160}$
1.1	$1.9166e-2$	$1.2923e-2$	$8.3161e-3$	$5.6519e-3$	$4.2409e-3$
1.2	$1.8604e-2$	$1.2395e-2$	$7.8368e-3$	$5.3025e-3$	$4.3441e-3$
1.3	$1.7761e-2$	$1.1635e-2$	$7.2644e-3$	$5.3359e-3$	$6.5805e-3$
1.4	$1.7282e-2$	$1.1286e-2$	$7.3047e-3$	$6.7523e-3$	$1.3210e-2$
1.5	$1.7425e-2$	$1.1595e-2$	$8.2041e-3$	$9.8328e-3$	Not valid

TABLE 3. The maximum absolute errors with 21×21 uniform points on Ω, $\beta = 1.9$, different values α_{s_x}, α_{s_y} , and different values Δt, at $T = 1.0$.

$\alpha_{s_x}, \alpha_{s_x}$	$\Delta t = \frac{1}{10}$	$\Delta t = \frac{1}{20}$	$\Delta t = \frac{1}{40}$	$\Delta t = \frac{1}{80}$	$\Delta t = \frac{1}{160}$
1.1	$1.7005e-3$	$1.0820e-3$	$6.2425e-3$	$3.5889e-3$	$2.1729e-3$
1.2	$1.6862e-3$	$1.0684e-3$	$6.1123e-3$	$3.4695e-3$	$2.0817e-3$
1.3	$1.6645e-3$	$1.0478e-3$	$5.9229e-3$	$3.3207e-3$	$2.0723e-3$
1.4	$1.6513e-3$	$1.0359e-3$	$5.8313e-3$	$3.3171e-3$	$2.3856e-3$
1.5	$1.6535e-3$	$1.0391e-3$	$5.8995e-3$	$3.5196e-3$	$3.0839e-3$

TABLE 4. The maximum absolute errors with 21×21 uniform points on Ω, $\beta = 1.9$, $\Delta t = \frac{1}{80}$, and different values $\alpha_{s_x}, \alpha_{s_y}$ at $T = 1.0$.

$\alpha_{s_x} = \alpha_{s_y}$	L_∞	$\alpha_{s_x} = \alpha_{s_y}$	L_∞	$\alpha_{s_x} = \alpha_{s_y}$	L_∞
1.1	$3.5889-3$	2.1	$2.5555e-3$	2.6	$1.7780e+103$
1.2	$3.4695e-3$	2.2	$2.7249e+14$	2.7	$9.1916e+122$
1.3	$3.3207e-3$	2.3	$1.2501e+39$	2.8	$7.7079e+140$
1.4	$3.3171e-3$	2.4	$1.0461e+61$	2.9	$3.6443e+157$
1.5	$3.5196e-3$	2.5	$1.7204e+83$		

TABLE 5. Comparison of the maximum absolute errors with $\beta = 1.7, 1.9$, different values Δt, $\alpha_{s_x} = \alpha_{s_y} = 1.1$, and 2025 points on Ω at $T = 1.0$.

	$\beta = 1.7$		$\beta = 1.9$	
Δt	L_∞	L_∞ [7]	L_∞	L_∞ [7]
$\frac{1}{5}$	$7.2126e-3$	$3.0420e-2$	$1.6646e-2$	$5.7458e-2$
$\frac{1}{10}$	$1.2581e-2$	$1.2917e-2$	$1.6492e-2$	$2.7619e-2$
$\frac{1}{20}$	$8.7169e-3$	$5.4532e-3$	$1.0311e-2$	$1.3079e-2$
$\frac{1}{40}$	$5.0766e-3$	$2.3351e-3$	$5.7494e-3$	$6.1953e-3$

6. Conclusion

In this work, the Crank-Nicolson method was used for the time semi-discretization of the time-fractional telegraph equation in Caputo sense with $1 < \beta < 2$. The stability and convergence of this semi-discretized scheme were valid. Then, the strong form of the moving least square method was used for full discretization. According to our experience, we suggest that the dimensionless size of the support domain in the X and Y directions should be in $(0, 1.5)$; sometimes, larger values for them make inaccurate approximations. Our tests reveal that an increase in the number of collocation points is necessary for accurate results with smaller time steps. The strong form of the moving least square method for the time-fractional telegraph equation depends on the number of collocation nodes, the time step size, and the support domain's dimensionless size in the X and Y directions. These four parameters should be chosen proportional to each other. Then, the numerical tests of our proposed method were sufficiently accurate. These results are related to the Crank-Nicolson method with the collocation method based on the MLS method and can be examined for the other meshless methods as well.

References

[1] A. Babu, et al., *Numerical solution of the hyperbolic telegraph equation using cubic B-spline-based differential quadrature of high accuracy*, Computational Methods for Differential Equations, *10*(4) (2022), 837–859.

[2] J. Banasiak and J. R. Mika, *Singularly perturbed telegraph equations with applications in the random walk theory*, Journal of Applied Mathematics and Stochastic Analysis, *11*(1) (1998), 9–28.

[3] T. Belytschko, et al., *Smoothing and accelerated computations in the element free Galerkin method*, Journal of computational and applied mathematics, *74*(1-2) (1996), 111–126.

[4] H. Hajinezhad and A. R. Soheili, *A numerical approximation for the solution of a time–fractional telegraph equation based on the Crank–Nicolson method*, Iranian journal of numerical analysis and optimization, *12*(3) (Special Issue) (2022), 607–628.

[5] V. R. Hosseini, et al., *Numerical solution of fractional telegraph equation by using radial basis functions*, Engineering Analysis with Boundary Elements, *38* (2014), 31–39.

[6] M. Javidi, et al., (2021). *Numerical solution of fractional Riesz space telegraph equation*, Computational Methods for Differential Equations, *9*(1) (2021), 187–210.

[7] A. Kumar, et al., *A local meshless method to approximate the time-fractional telegraph equation*, Engineering with Computers, *37*(4) (2021), 3473–3488.

[8] P. Lancaster and K. Salkauskas, *Surfaces generated by moving least squares methods*, Mathematics of computation, *37*(155) (1981), 141–158.

[9] C. Li and A. Chen, *Numerical methods for fractional partial differential equations*, International Journal of Computer Mathematics, *95*(6-7) (2018), 1048-1099.

[10] G. R. Liu and Y. T. Gu, *An introduction to meshfree methods and their programming*, Springer Science and Business Media, (2005).

[11] G. R. Liu and M. B. Liu, *Smoothed particle hydrodynamics– a meshfree particle method*, World Scientific, Singapore, (2003).

[12] A. Mohebbi, et al., *The meshless method of radial basis functions for the numerical solution of time fractional telegraph equation*, International Journal of Numerical Methods for Heat and Fluid Flow, *24*(8) (2014), 1636–1659.

[13] S. Niknam and H. Adibi, *A numerical solution of two-dimensional hyperbolic telegraph equation based on moving least square meshless method and radial basis functions*, Computational Methods for Differential Equations, *10*(4) (2022), 969–985.

[14] O. Nikan, et al., *Numerical approximation of the nonlinear time-fractional telegraph equation arising in neutron transport*, Communications in Nonlinear Science and Numerical Simulation, *99* (2021).

[15] Z. M. Odibat and S. Momani, *An algorithm for the numerical solution of differential equations of fractional order*, Journal of Applied Mathematics and Informatics *26*(1-2) (2008), 15–27.

[16] B. Sepehrian and Z. Shamohammadi, *Numerical solution of nonlinear time-fractional telegraph equation by radial basis function collocation method*, Iranian Journal of Science and Technology, Transactions A: Science, *42*(4) (2018), 2091–2104.

[17] K. Shah, et al., *Stable numerical results to a class of time-space fractional partial differential equations via spectral method*, Journal of Advanced Research, *2* (2020), 39–48.

[18] K. Shah, et al., *Evaluation of one dimensional fuzzy fractional partial differential equations*, Alexandria Engineering Journal, *59*(5) (2020), 3347–3353.

[19] E. Shivanian, *Spectral meshless radial point interpolation (SMRPI) method to two-dimensional fractional telegraph equation*, Mathematical Methods in the Applied Sciences, *39*(7) (2016), 1820–1835.

[20] E. Shivanian, et al., *Local integration of 2-D fractional telegraph equation via moving least squares approximation*, Engineering Analysis with Boundary Elements, *56* (2015), 98–105.

[21] N. H. Sweilam, et al., (2020). *Sinc-Chebyshev collocation method for time-fractional order telegraph equation*, Appl. Comput. Math, *19*(2) (2020), 162–174.

[22] V. A. Vyawahare and P. Nataraj, *Fractional-order modeling of neutron transport in a nuclear reactor*, Applied Mathematical Modelling, *37*(23) (2013), 9747–9767.