A novel numerical implementation for solving time fractional telegraph differential equations having multiple space and time delays via Delannoy polynomial

Ömür Kıvanç Kürkçü

Department of Engineering Basic Sciences, Konya Technical University, Konya, Turkey,
omurkivanc@outlook.com, ORCID: 0000-0002-3987-7171

ABSTRACT

This paper is concerned with solving numerically the time fractional telegraph equations having multiple space and time delays by proposing a novel matrix-collocation method dependent on the Delannoy polynomial. This method enables easy and fast approximation tool consisting of the matrix expansions of the functions using only the Delannoy polynomial. Thus, the solutions are obtained directly from a unique matrix system. Also, the residual error computation, which involves the same procedure as the method, provides the improvement of the solutions. The method is evaluated under some valuable error tests in the numerical applications. To do this, a unique computer module is devised. The present results are compared with those of the existing methods in the literature, in order to oversee the precision and efficiency of the method. One can express that the proposed method admits very consistent approximation for the equations in question.

1. Introduction

In this study, we aim to implement a novel matrix-collocation method dependent on the Delannoy polynomial to solve efficiently and precisely the time fractional telegraph differential equations having multiple space and time delays

$$
{}_t^C\mathcal{D}_0^\alpha u(x,t) + {}_t^C\mathcal{D}_0^\beta u(x-\tau_1, t-\tau_2) + p_1 u(x-\tau_3, t-\tau_4) = p_2\,{}_xD^2 u(x,t) + g(x,t), \tag{1}
$$

subject to the initial condition $u(x,c) = \lambda_1(x)$ in Ω and the boundary conditions $u(a,t) = \lambda_2(t)$ and $u(b,t) = \lambda_3(t)$ on $\partial\Omega$,

where $u(x,t)$ and $g(x,t)$ are analytic functions on the domain

$$
\Omega = \{(x,t) : a \le x \le b,\ c \le t \le d\},
$$

${}_t^C\mathcal{D}_0^\alpha$ (similarly ${}_t^C\mathcal{D}_0^\beta$) ($1 < \alpha \le 2$, $0 < \beta \le 1$) is the time fractional derivative operator in the Caputo sense [1]; $\{p_1, p_2\}$ are the real constants; $\{\tau_1, \tau_3\}$ and $\{\tau_2, \tau_4\}$ are the spatial and temporal delays, respectively.

It is explicit to state that the partial differential equations (PDEs) govern physical and mathematical modelling of many natural phenomena arising in real world applications. PDEs can easily built a model into a mathematical formation

by enclosing time and space variables on specific domains of this model, such as rectangular plate or elliptic domain. In particular, the fields of application of PDEs have been spread due to the inclusion of the time or space fractional manner of partial derivatives. In fact, there is a growing interest and attention in the fractional PDEs (FPDEs), with the developments of real world hybrid models occurring especially in fluid mechanics [2], electrical circuits [3], viscoelasticity [4], heat transfer [5], finance [6], diffusion phenomena [7] and so on. In addition to these composite structured equations, temporal or spatial lags may also be encountered due to the indeterminable physical conditions. These complexities can be detected via delay arguments in FPDEs. Inherently, the variational combinations of these instruments make FPDEs too severe problems to handle them analytically. Therefore, the numerical implementations have been proposed to acquire the approximate solutions of FPDEs of delay types so far. For example, Kürkçü et al. [8] have applied a matrix method based on the matching polynomial of complete graph to solve the space-time FPDEs of delay type. Dehestani et al. [9] have employed the Genocchi hybrid collocation method for solving several FPDEs containing delay arguments. Rihan [10] has constructed the semi-discrete and ϑ-methods for delay FPDEs of parabolic type. Usman et al. [11] have utilized an operational matrix method dependent on the shifted Gegenbauer polynomials to treat FPDEs of delays. Motivating by these mentioned studies, we introduce a novel methodology consisting of the matrix expansions of the functions via the Delannoy polynomial for solving equation (1). The main advantage of the proposed method is that it directly converts the functions in equation (1) into the matrix expansions, applying the Delannoy polynomial base. This inventive tool provides easy and fast computation to get the approximate solutions.

On the other hand, the telegraph differential equations (TDEs) can be appeared differently in the study of the wave propagation theory on the behaviour of electrical signals found in a cable transmission system [12, 13], particle transport applications [14] and Brownian motion processes [15]. As a fractional sense of TEs, some numerical studies can be detailed as follows: Kumar et al. [16] have studied a local meshless method to acquire the solutions of time fractional TDEs with one and two dimensional forms. Hosseini et al. [3] have deployed the radial basis function method for time fractional TDEs. Wang et al. [17] have solved a class of time fractional TDE subjected to the initial conditions by implementing the reproducing kernel method. Pandey and Mishra [18] have established the numerical simulation of the time-space fractional TDEs using the homotopy analysis method and Sumudu transform. For the numerical solutions of TDEs of integer order, the Galerkin-type method [19], the thin plate splines radial basis method [20], the Chebyshev cardinal function method [21], the Crank–Nicolson finite difference scheme and Haar wavelet method [22], and the cubic B-spline collocation method [23] have been deployed up to now. We here draw attention to the fact that TEs of equation (1) type are hardly found in the literature. Indeed, we establish and solve equation (1) efficiently via this study.

The layout of this paper is shaped as follows: Section 1.1 presents a brief information about the Delannoy numbers and their polynomial form. Section 2 admits the details of the proposed method and its matrix expansions related to the equation (1). Section 3 reveals the properties of the residual error computation to improve the obtained solutions emerged from Section 2. Section 4 applies three specific problems to exhibit the efficiency and precision of the method. Section 5 emphasizes on a competitive innovation and developments via the method, interpreting the results obtained in Section 4.

1.1. Preliminary to Delannoy numbers and their polynomial form

The Delannoy numbers $D(n, k)$ ($n, k \in \mathbb{N}$), which are named after Henri Delannoy [24], are related to the lattice path enumeration (n, k) using the number of walk from the initial point $(0, 0)$ to (n, k) with $(0, 1)$, $(1, 1)$ or $(1, 0)$ increments [24, 25]. The generating function of the Delannoy walks is formulated as

$$F[x, y, t] = \sum_{n \geq 0} (xt + yt + xyt)^n = \frac{1}{1 - t(x + y + xy)},$$

where x, y and xy imply the three steps $(0, 1)$, $(1, 1)$ and $(1, 0)$, respectively, and t implies the length of the walk [25].

The main implicit formulation of the Delannoy numbers can be stated as follows [25, 26]:

$$D(n, k) = \sum_{i=0}^{n} \binom{n}{i}\binom{k}{i} 2^i = \binom{n+k}{k} {}_2F_1\left(-n, -k; -(n+k); -1\right),$$

where ${}_2F_1(a, b;\, c;\, t)$ is a hypergeometric function.

A fundamental recurrence relation of $D(n, k)$ is defined to be

$$D(n, k) = D(n-1, k) + D(n-1, k-1) + D(n, k-1),$$

where $D(0, 0) = 1$ [26, 27] and this relation can provide the generating function (see [27])

$$\sum_{n,k=0}^{\infty} D(n, k) x^n y^k = (1 - x - y - xy)^{-1}.$$

The central Delannoy numbers, which are denoted by $D(n, n)$ or briefly $D(n)$ instead of k, can be defined to be (see [28])

$$D(n, n) = D(n) = \sum_{i=0}^{n} \binom{n}{i}\binom{n+i}{i},$$

or another expression (see [28])

$$D(n) = \sum_{i=0}^{n} \binom{n}{i}^2 2^i.$$

Their recurrence relation stands for (see [29])

$$nD(n) = 3(2n-1)D(n-1) - (n-1)D(n-2),$$

and also their generating function has the form (see [26, 27])

$$\sum_{n=0}^{\infty} D(n) x^n = (1 - 6x + x^2)^{-1/2}.$$

$D(n)$ yields an algebraical relation between the Legendre polynomial $P_n(x)$ as (see [25])

$$D(n) = P_n(3),$$

and more uniquely, $D(n)$ stands for the equality

$$D(n + \alpha, n) = P_n^{(\alpha, 0)}(3), \ \forall \alpha \in \mathbb{Z}, \ni \alpha \geq -n,$$

where $P_n^{(\alpha, 0)}(x)$ is the Jacobi polynomial with the parameters $(\alpha, 0)$ [30].

As a combinatorial purpose, $D(n)$ is equivalent to the generalized central trinomial coefficients $T_n(3, 2)$, where $T_n(b, c)$ implies the coefficient of x^n when $(x^2 + bx + c)^n$ is expanded algebraically [31].

Indeed, the Delannoy numbers can occur in the applications of lattice and posets [25], number theory [32] and the formulation of the alignments between DNA sequences [33].

In view of the Delannoy numbers, their polynomial implicit form was established by Sun [31] as

$$D_n(x) = \sum_{i=0}^{n} \binom{n}{i}\binom{n+i}{i}x^i. \tag{2}$$

In particular, the Delannoy polynomial can be associated with the Legendre polynomial $P_n(x)$ and the generalized central trinomial coefficients $T_n(b,c)$ [31], so

$$D_n(x) = P_n(2x+1) = T_n(2x+1, x^2 + x), \ \forall x \in \mathbb{Z}.$$

In addition to these coincides, the Delannoy polynomial has also a natural connection with the Schmidt polynomial $S_n(\alpha, x)$ for $\alpha = 1$ [34].

2. Foundation of the matrix expansions for method of solution

In this section, the matrix expansions of the functions in equation (1) are derived to form a fundamental basis of the proposed method. Supposing that $u(x,t)$ be an analytic function on Ω and $_xD^n$ (or $_tD^n$) is a differential operator of integer order n with respect to x (or t). Firstly, the Delannoy polynomial solution of equation (1) is found to be

$$u(x,t) \cong u_N(x,t) = \sum_{m=0}^{N}\sum_{n=0}^{N} z_{mn}D_m(x)D_n(t), \tag{3}$$

where z_{mn}'s are the unknown Delannoy coefficients to be identified via the method and $D_m(x)$ represents the Delannoy polynomial (2).

The matrix expansion of the Delannoy polynomial solution (3) can be constructed as

$$u_N(x,t) = \boldsymbol{D}(x)\overline{\boldsymbol{D}(t)}\boldsymbol{A}, \tag{4}$$

where

$$\boldsymbol{D}(x) = \begin{bmatrix} D_0(x) & D_1(x) & \cdots & D_N(x) \end{bmatrix},$$

$$\overline{\boldsymbol{D}(t)} = \begin{bmatrix} \boldsymbol{D}(t) & 0 & \cdots & 0 \\ 0 & \boldsymbol{D}(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{D}(t) \end{bmatrix},$$

and

$$\boldsymbol{A} = \begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0N} & a_{10} & \cdots & a_{1N} & a_{N0} & \cdots & a_{NN} \end{bmatrix}^T,$$

such that $\overline{\boldsymbol{D}(t)}$ is spanned by $\boldsymbol{D}(t)$ with dimensional $1 \times N + 1$.

Inserting properly the spatial delay $x \to x - \tau_3$ and temporal delay $t \to t - \tau_4$ into the matrix expansion (4), it holds

$$u\left(x - \tau_3, t - \tau_4\right) = \boldsymbol{D}\left(x - \tau_3\right)\overline{\boldsymbol{D}\left(t - \tau_4\right)}\boldsymbol{A}, \tag{5}$$

where

$$\boldsymbol{D}\left(x - \tau_3\right) = \begin{bmatrix} D_0(x - \tau_3) & D_1(x - \tau_3) & \cdots & D_N(x - \tau_3) \end{bmatrix},$$

and

$$\overline{\boldsymbol{D}\left(t - \tau_4\right)} = \begin{bmatrix} \boldsymbol{D}\left(t - \tau_4\right) & 0 & \cdots & 0 \\ 0 & \boldsymbol{D}\left(t - \tau_4\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{D}\left(t - \tau_4\right) \end{bmatrix}.$$

Now, using the time fractional Caputo derivative operator in terms of α, which is of the form:

Definition 2.0.1 *[1] Suppose that $t > a$ and $\alpha, t \in \mathbb{R}$. Then the time fractional Caputo derivative of order α is defined to be*

$$_t^C\mathcal{D}_0^\alpha u\left(x, t\right) = \begin{cases} \frac{1}{\Gamma(\lceil\alpha\rceil-\alpha)} \int_0^t (t - s)^{\lceil\alpha\rceil-\alpha-1} \left(_sD^{\lceil\alpha\rceil}u\left(x, s\right)\right) ds, & \lceil\alpha\rceil - 1 < \alpha < \lceil\alpha\rceil \\ \partial^\alpha u\left(x, t\right)/\partial t^\alpha, & \alpha \in \mathbb{N} \end{cases}.$$

We can insert this operator on the matrix expansion (4) as

$$_t^C\mathcal{D}_0^\alpha u\left(x, t\right) = \boldsymbol{D}\left(x\right)\overline{\boldsymbol{D}^{(\alpha)}\left(t\right)}\boldsymbol{A}, \ 1 < \alpha \le 2, \tag{6}$$

where

$$\boldsymbol{D}^{(\alpha)}\left(t\right) = \begin{bmatrix} _t^C\mathcal{D}_0^\alpha(D_0(t)) & _t^C\mathcal{D}_0^\alpha(D_1(t)) & \cdots & _t^C\mathcal{D}_0^\alpha(D_N(t)) \end{bmatrix},$$

and

$$\overline{\boldsymbol{D}^{(\alpha)}\left(t\right)} = \begin{bmatrix} \boldsymbol{D}^{(\alpha)}\left(t\right) & 0 & \cdots & 0 \\ 0 & \boldsymbol{D}^{(\alpha)}\left(t\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{D}^{(\alpha)}\left(t\right) \end{bmatrix}.$$

Using analogously the matrix expansions (5) and (6), we can write

$$_t^C\mathcal{D}_0^\beta u\left(x - \tau_1, t - \tau_2\right) = \boldsymbol{D}\left(x - \tau_1\right)\overline{\boldsymbol{D}^{(\beta)}\left(t - \tau_2\right)}\boldsymbol{A}, \ 0 < \beta \le 1, \tag{7}$$

where

$$\overline{D^{(\beta)}(t-\tau_2)} = \begin{bmatrix} D^{(\beta)}(t-\tau_2) & 0 & \cdots & 0 \\ 0 & D^{(\beta)}(t-\tau_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D^{(\beta)}(t-\tau_2) \end{bmatrix}.$$

In addition, by making use of the matrix expansion (4), we can inherently state the following matrix expansion with respect to the spatial partial derivative of integer order:

$$_xD^2u(x,t) = D^{(2)}(x)\overline{D(t)}A, \tag{8}$$

where

$$D^{(2)}(x) = \begin{bmatrix} D_0^{(2)}(x) & D_1^{(2)}(x) & \cdots & D_N^{(2)}(x) \end{bmatrix}.$$

We are now ready to constitute a fundamental matrix form of the equation (1) by means of the matrix expansions (5)-(8). Thereby, collecting them under a unique form, it follows that

$$W(x_i,t_j)A = G(x_i,t_j), \tag{9}$$

where

$$W(x_i,t_j) = \left[D(x_i)\overline{D^{(\alpha)}(t_j)} + D(x_i-\tau_1)\overline{D^{(\beta)}(t_j-\tau_2)} + p_1D(x_i-\tau_3)\overline{D(t_j-\tau_4)} - p_2D^{(2)}(x_i)\overline{D(t_j)} \right],$$

and

$$G(x_i,t_j) = \begin{bmatrix} G(x_0,t_0) & \cdots & G(x_0,t_N) & \cdots & G(x_N,t_0) & \cdots & G(x_N,t_N) \end{bmatrix}^T,$$

such that x_i and t_i are the Chebyshev-Lobatto collocation points on Ω defined to be (see [8])

$$x_i = \frac{a+b}{2} + \frac{a-b}{2}\cos\left(\frac{\pi i}{N}\right), \ t_j = \frac{c+d}{2} + \frac{c-d}{2}\cos\left(\frac{\pi j}{N}\right), \ i,j = 0,1,\ldots,N.$$

For more brief form of the matrix form (9), it can be summarized as

$$WA = G \Rightarrow \begin{bmatrix} W & : & G \end{bmatrix}, \tag{10}$$

where

$$W = \begin{bmatrix} W(x_0,t_0) & \cdots & W(x_0,t_N) & \cdots & W(x_N,t_0) & \cdots & W(x_N,t_N) \end{bmatrix}^T.$$

On the other hand, using the main matrix expansion (4), we can form the matrix expansions of the initial and boundary conditions at the Chebyshev-Lobatto collocation points, respectively, as

$$u\left(x_i, c\right) = \boldsymbol{D}\left(x_i\right)\overline{\boldsymbol{D}\left(c\right)}\boldsymbol{A} = \lambda_1(x_i),$$
$$u\left(a, t_j\right) = \boldsymbol{D}\left(a\right)\overline{\boldsymbol{D}\left(t_j\right)}\boldsymbol{A} = \lambda_2(t_j),$$
$$u\left(b, t_j\right) = \boldsymbol{D}\left(b\right)\overline{\boldsymbol{D}\left(t_j\right)}\boldsymbol{A} = \lambda_3(t_j),$$

(11)

where $i, j = 0, 1, \ldots, N$.

After replacing the conditional matrix expansions (11) by the last $3N + 3$ rows in the matrix system (10), we can now state the augmented matrix system as

$$\left[\widetilde{\boldsymbol{W}} : \widetilde{\boldsymbol{G}}\right] = \begin{bmatrix} \boldsymbol{W}^* & ; & \boldsymbol{G}^* \\ \boldsymbol{D}\left(x_i\right)\overline{\boldsymbol{D}\left(c\right)} & : & \lambda_1(x_i) \\ \boldsymbol{D}\left(a\right)\overline{\boldsymbol{D}\left(t_j\right)} & : & \lambda_2(t_j) \\ \boldsymbol{D}\left(b\right)\overline{\boldsymbol{D}\left(t_j\right)} & : & \lambda_3(t_j) \end{bmatrix},$$

where $\boldsymbol{W}^*$ and $\boldsymbol{G}^*$ are of $(N^2 - N - 2) \times 1$ matrix structures, this system is consistent as long as $\mathrm{rank}\,\widetilde{\boldsymbol{W}} = \mathrm{rank}\left[\widetilde{\boldsymbol{W}} : \widetilde{\boldsymbol{G}}\right] = (N + 1)^2$. If this situation is satisfied, then the system can be solved via Gaussian elimination method or

$$\boldsymbol{A} = \widetilde{\boldsymbol{G}}(\widetilde{\boldsymbol{W}})^{-1}, \ \det\widetilde{\boldsymbol{W}} \neq 0.$$

Eventually, the unknown Delannoy coefficients (z_{mn}) are obtained and then substituted into the solution form (3), the Delannoy polynomial solution of equation (1) is acquired.

3. Residual error computation

Formerly, the residual error analysis technique was employed to improve the approximate solutions of the space-time fractional PDEs involving multi delays [8] and one dimensional TDE of integer order [19]. In this paper, we shall construct a residual error computation based on the Delannoy polynomial. Basically, the residual error computation is made up of the same procedure as the proposed method and a residual function obtained by substituting the Delannoy polynomial solution into equation (1). A residual function corresponding to N can be described on Ω as

$$R_N\left(x, t\right) = {}^C_t\mathcal{D}^\alpha_0 u_N(x, t) + {}^C_t\mathcal{D}^\beta_0 u_N(x - \tau_1, t - \tau_2) + p_1 u_N(x - \tau_3, t - \tau_4) - p_{2\,x}D^2 u_N(x, t) - g(x, t).$$

(12)

After taking the initial and boundary conditions, the error problem can be stated as (see [8])

$$L\left[e_N\left(x, t\right)\right] = L\left[u\left(x, t\right)\right] - L\left[u_N\left(x, t\right)\right] = -R_N\left(x, t\right),$$

(13)

where $e_N\left(x, t\right)$ is an error function under the linear operator.

Equation (13) can be solved by following the algorithmic structure of the proposed method. Then, the estimated error function $e_{N,K}\left(x, t\right)$ is derived by

$$e_{N,K}\left(x, t\right) = \sum_{m=0}^{K} \sum_{n=0}^{K} z^*_{mn} D_m\left(x\right) D_n\left(t\right), \ K > N,$$

and using this function, the corrected Delannoy polynomial solution and the corrected error function can be constructed, respectively, as

$$u_{N,K}(x,t) = u_N(x,t) + e_{N,K}(x,t) \text{ and } E_{N,K}(x,t) = u(x,t) - u_{N,K}(x,t).$$

4. Numerical problems

In this section, three problems are dealt numerically with the proposed method. To do this, a practical module is uniquely devised on Mathematica 11.3 run on a standard personal computer with 6 GB RAM and 2.50 GHz i5-2450M CPU. For this clearness, the computation results can be acquired sensitively from the mentioned problems. In addition, graphical and numerical tools, such as a root mean squared error (E_{RMS}) computation, are deployed to interpret the precision and accuracy of the method. E_{RMS} is formulated as follows [21]:

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}[u(x_i,T) - u_N(x_i,T)]^2},$$

where $u(x_i,T)$ and $u_N(x_i,T)$ are the exact and Delannoy polynomial solutions at the spatial collocation points x_i and specific time T, respectively.

Problem 4.1. [3, 16, 17] Consider the time fractional TDE having non-polynomial exact solution

$$^C_t\mathcal{D}_0^\alpha u(x,t) + {}^C_t\mathcal{D}_0^\beta u(x,t) + u(x,t) = {}_xD^2u(x,t) + g(x,t),$$

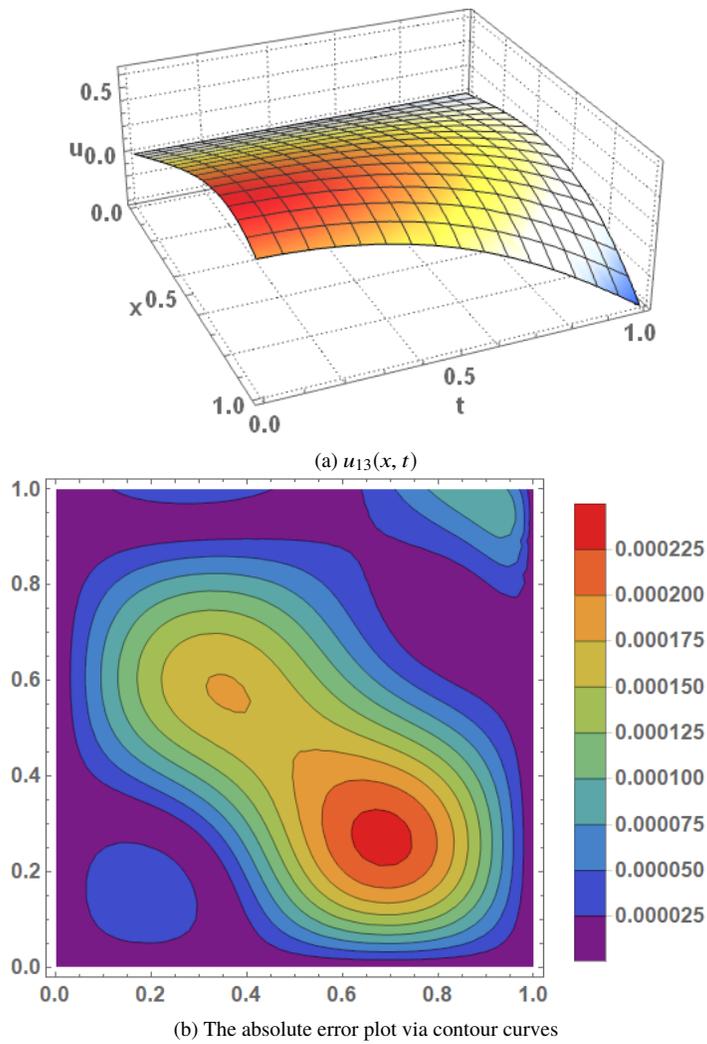with $1 < \alpha \le 2$, $0 < \beta = \alpha - 1 \le 1$ and $\Omega = [0,1]^2$,

subject to the initial condition $u(x,0) = x\cos(x^2)$ in Ω and boundary conditions $u(0,t) = 0$ and $u(1,t) = \cos(1 + t^2)$ on $\partial\Omega$. The exact solution is formulated as $u(x,t) = x\cos(x^2 + t^2)$. The force function $g(x,t)$ can be easily determined in accordance with α. Solving this problem by employing the proposed method, the Delannoy polynomial solutions are acquired. At first, the solution $u_{13}(x,t)$ is demonstrated along with its contour curves, which represent the absolute error values, in Figure 1. From Figure 1, this solution takes five decimal accuracies (10^{-5}). Besides, Table 1 compares the computation results of E_{RMS} in terms of the proposed (PM), the local meshless (LMM) [16], the radial basis function (RBF) [3] and the reproducing kernel (RKM) [17] methods. It can be noticed from Table 1 that the proposed method achieves far more accurate approximation than the other methods with respect to the low computation limit (N) and each specified fractional derivative. Also, in Table 2, the proposed method ($N = 20$) takes 10^{-9} of accuracy, while RKM ($h = 0.001$) [17] enables 10^{-5} of accuracy on average.

Table 1. *Numerical computation results of E_{RMS} at T = 1 with respect to the proposed and existing methods for Problem 4.1.*

Methods	$\alpha = 1.25$	$\alpha = 1.5$	$\alpha = 1.75$	$\alpha = 1.95$
PM ($N = 20$)	$2.1583e{-}12$	$5.2895e{-}12$	$1.9302e{-}13$	$2.7054e{-}10$
LMM ($M = 50$) [16]	$2.2992e{-}05$	$1.0880e{-}04$	$6.8631e{-}04$	$2.4189e{-}03$
RBF ($N = 50$) [3]	$2.6551e{-}03$	$4.1743e{-}03$	$5.4164e{-}03$	$3.8800e{-}03$
RKM ($h = 0.001$) [17]	-	-	-	$4.3309e{-}04$

Problem 4.2. [16] Consider the time fractional TDE having non-polynomial exact solution

$$^C_t\mathcal{D}_0^\alpha u(x,t) + {}^C_t\mathcal{D}_0^\beta u(x,t) + u(x,t) = {}_xD^2u(x,t) + g(x,t),$$

(a) $u_{13}(x, t)$

(b) The absolute error plot via contour curves

Figure 1. *The Delannoy polynomial solution and its error plot for Problem 4.1 having $\alpha = 1.9$ and $\beta = 0.9$.*

Table 2. *Absolute error history in terms of PM and RKM for Problem 4.1 having $\alpha = 1.95$.*

x_i	t_i	$|e_{20}(x_i, t_i)|$ $N = 20$	RKM $h = 0.001$ [17]
0.1	0.1	$3.69731e - 10$	$1.49994e - 07$
0.2	0.2	$1.33009e - 09$	$2.39840e - 06$
0.3	0.3	$2.75515e - 09$	$1.21090e - 05$
0.4	0.4	$4.15103e - 09$	$3.79919e - 05$
0.5	0.5	$4.46054e - 09$	$9.13299e - 05$
0.6	0.6	$3.41082e - 09$	$1.84091e - 04$
0.7	0.7	$1.89897e - 09$	$3.25310e - 04$
0.8	0.8	$7.62644e - 10$	$5.15381e - 04$
0.9	0.9	$1.40729e - 10$	$7.38682e - 04$
1.0	1.0	$9.33087e - 13$	$9.56449e - 04$

with $1 < \alpha \leq 2$, $0 < \beta = \alpha - 1 \leq 1$, $\Omega = [0,1]^2$ and

$$g(x,t) = \sin^2 x \left(\frac{6t^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{6t^{4-\alpha}}{\Gamma(5-\alpha)} \right) - 2t^3 \cos 2x + t^3 \sin^2 x,$$

subject to the initial condition $u(x,0) = 0$ and boundary conditions $u(0,t) = 0$ and $u(1,t) = t^3 \sin^2 1$. The exact solution of the problem is given as $u(x,t) = t^3 \sin^2 x$. The problem is immediately and accurately solved by the proposed method for different N and α. As seen in Figure 2, the Delannoy polynomial solution $u_{10}(x,t)$ has a good approximation to the exact solution, yielding 10^{-8} of accuracy for $\alpha = 1.7$. Besides, Table 3 reveals that the proposed method (PM) performs much more accurate computations than LMM [16], as well as much faster. It is explicit to say from Figure 3 that the L_2 and L_∞ errors are decreased when N increases and so this situation confirms the precision of the method in accordance with the results given in Table 3.

(a) $u_{10}(x,t)$

(b) The absolute error plot via contour curves
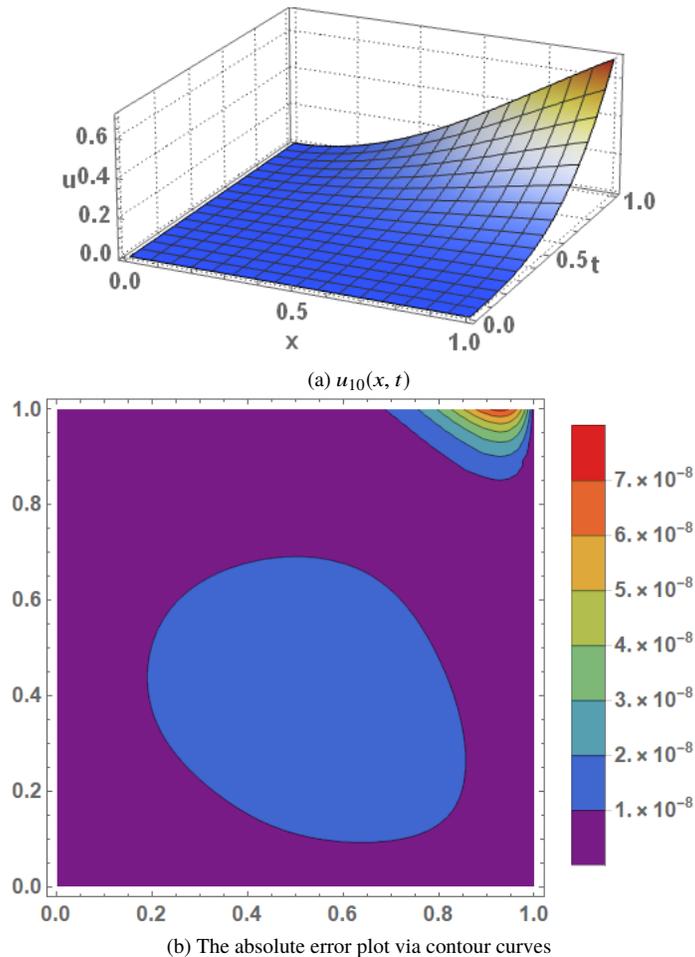
**Figure 2.** *The Delannoy polynomial solution and its error plot for Problem 4.2 having $\alpha = 1.7$ and $\beta = 0.7$.*

Problem 4.3. Consider the time fractional TDE with multiple space and time delays

$$^C_t \mathcal{D}^{1.6}_0 u(x,t) + ^C_t \mathcal{D}^{0.9}_0 u(x-1, t-1) + 0.3u(x-0.5, t-0.1) = 0.5 \,_x D^2 u(x,t) + g(x,t),$$

with $\Omega = [0,1]^2$ and

Table 3. *Numerical comparisons of E_{RMS} (at $T = 1$) and CPU running time (in seconds) in terms of PM and LMM for Problem 4.2 having $\alpha = 1.9$.*

PM			LMM [16] m & $N = 1000$		
N	E_{RMS}	Timing	m	E_{RMS}	Timing
6	4.0353e–05	0.906	3	6.5786e–05	34.249
7	2.8692e–05	1.250	7	6.5785e–05	35.542
9	5.0556e–07	2.359	9	6.5804e–05	37.021
11	3.2466e–09	4.672	11	6.5794e–05	37.380
13	1.8760e–09	9.625	15	6.5796e–05	38.332
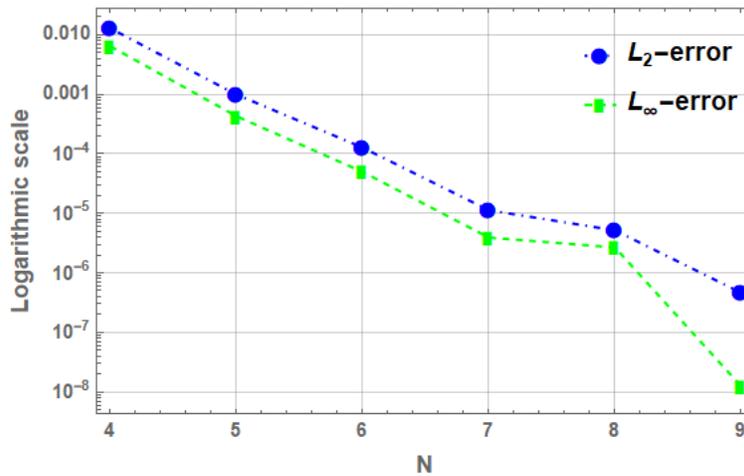
Figure 3. *The decreasing error computations on logarithmic scale versus N for Problem 4.2 having $\alpha = 1.2$.*

$$g(x, t) = e^t(-0.135726 + 0.271451x + t^{0.1}(-0.386692 + 0.386692x)\,_1F_1(0.1; 1.1; -t)$$
$$+ 1.12706xt^{0.4}\,_1F_1(0.4; 1.4; -t)),$$

such that $_1F_1(a_1; a_2; t)$ is the confluent hypergeometric function of Kummer type [35],

subject to the initial condition $u(x, 0) = x$ and boundary conditions $u(0, t) = 0$ and $u(1, t) = e^t$. The exact solution of the problem is $u(x, t) = xe^t$. After solving this problem by arranging $N = 4(1)11$ in the method, the Delannoy polynomial solution is illustrated along with its contour curves specifying 10^{-9} of accuracy in Figure 4. Besides, Figure 5 indicates the L_2 and L_∞ error computations and CPU running time versus N. It is worth clarifying from Figure 5 that the method gets much clearer precision to the exact solution and returns the solutions by consuming short time at most 12 seconds, as N is increased from 4 to 11. On the other hand, the Delannoy polynomial solution for $N = 6$ is easily improved by employing the residual error computation with its limit $N = 7$, as seen in Table 4.
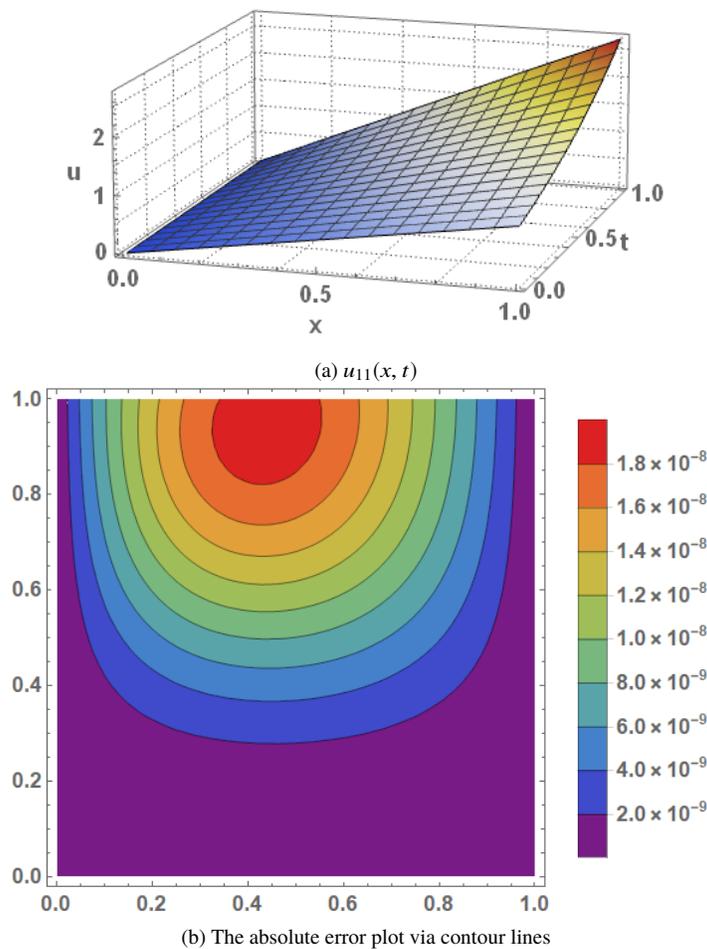

(a) $u_{11}(x, t)$

(b) The absolute error plot via contour lines

Figure 4. *The Delannoy polynomial solution and its error plot for Problem 4.3.*

5. Conclusions

A novel matrix-collocation method dependent on the Delannoy polynomial has been productively implemented to treat the time fractional telegraph differential equations of space and time delays. To do this, the inventive and pure matrix structures containing the fractional derivative forms have been developed to easily extend the functions in equation (1)
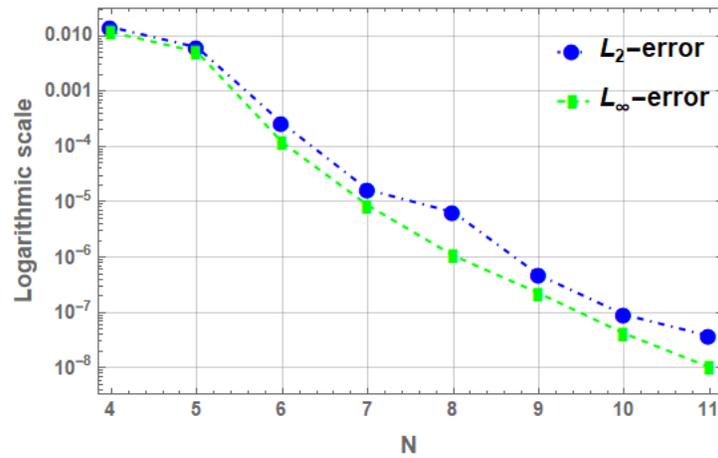
(a) The error computations on logarithmic scale
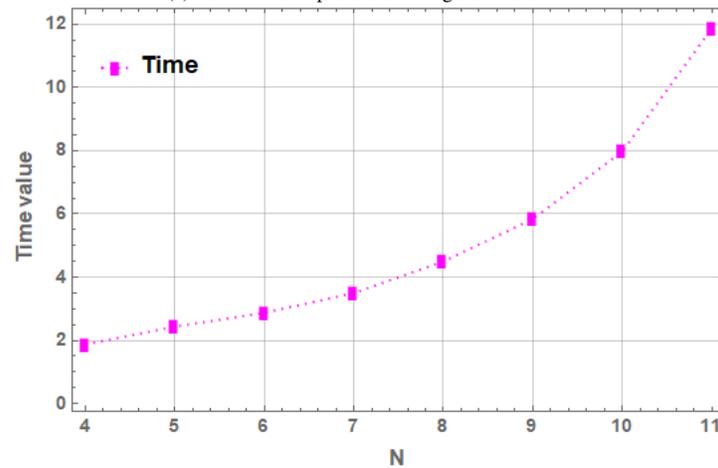
(b) CPU running time

Figure 5. *Displacements of the error computations and timing tool versus N for Problem 4.3.*

Table 4. *History of the absolute and corrected absolute error results for Problem 4.3.*

(x_i, t_i)	$|e_6(x_i, t_i)|$	$|E_{6,7}(x_i, t_i)|$
$(0.2, 0.2)$	$3.4034e{-}05$	$1.9525e{-}05$
$(0.4, 0.4)$	$9.2624e{-}05$	$6.2166e{-}05$
$(0.6, 0.6)$	$1.1806e{-}04$	$1.0044e{-}04$
$(0.8, 0.8)$	$1.0941e{-}04$	$8.3094e{-}05$

using the collocation points. With the involvement of the Delannoy polynomial in these matrix profiles, the proposed method has enabled to obtain persistent numerical solutions, which are often in good agreement with the analytical solutions corresponding to the specific problems. In addition, as noticed in the error and time complexity comparisons given in Tables 1, 2 and 3, the method has yielded very remarkable results compared to the existing methods. As N increases, the precision of the method becomes much clear as in Table 3, Figures 3 and 5.

When all computations are investigated, the findings reveal that the method is very competitive and provides a novel approximation into the literature by treating the delay form of TDEs. In fact, we can specify that the method would be applicable tool for some generations of the fractional partial differential equations. Inherently, this specification is regarded as a next vision for incoming works.

References

[1] Caputo, M., Elasticit$\grave{a}$e Dissipazione, Bologna, Zanichelli, 1969.

[2] Moaddy, K., Momani, S., Hashim, I., The non-standard finite difference scheme for linear fractional PDEs in fluid mechanics, Comput. Math. Appl., 61, (2011), 1209–1216.

[3] Hosseini, V.R., Chen, W., Avazzadeh, Z., Numerical solution of fractional telegraph equation by using radial basis functions, Eng. Anal. Bound. Elem., 38, (2014), 31–39.

[4] Faraji, M, Ansari, O.R., Linear and nonlinear vibrations of fractional viscoelastic Timoshenko nanobeams considering surface energy effects, Appl. Math. Model., 43, (2017), 337–350.

[5] Arqub, O.A., Numerical solutions for the Robin time-fractional partial differential equations of heat and fluid flows based on the reproducing kernel algorithm, Int. J. Numer. Method H., 28(4), (2018), 828–856.

[6] Koleva, M.N., Vulkov, L.G., Numerical solution of time-fractional Black–Scholes equation, Comp. Appl. Math., 36, (2017), 1699–1715.

[7] Soori, Z., Aminataei, A., A new approximation to Caputo-type fractional diffusion and advection equations on non-uniform meshes, Appl. Numer. Math., 144, (2019), 21–41.

[8] Kürkçü, Ö.K., Aslan, E., Sezer, M., An advanced method with convergence analysis for solving space-time fractional partial differential equations with multi delays, Eur. Phys. J. Plus, 134, (2019), 393.

[9] Dehestani, H., Ordokhani, Y., Razzaghi, M., A numerical technique for solving various kinds of fractional partial differential equations via Genocchi hybrid functions, Rev. R. Acad. Cienc. Exactas, Fis. Nat. Madr., 113, (2019), 3297–3321.

[10] Rihan, F.A., Computational methods for delay parabolic and time-fractional partial differential equations, Numer. Methods Partial Differ. Equ., 26, (2010), 1556–1571.

[11] Usman, M., Hamid, M., Zubair, T., Haq, R.U., Wang, W., Liu, M.B., Novel operational matrices-based method for solving fractional-order delay differential equations via shifted Gegenbauer polynomials, Appl. Math. Comput., 372, (2020), 124985.

[12] Pozar, D.M., Microwave Engineering in Addison-Wesley, 1990.

[13] Mainardi, F., Paradisi, P., Fractional diffusive waves, J. Comput. Acoust., 9(4), (2001), 1417–1436.

[14] Abdou, M.A., Adomian decomposition method for solving the telegraph equation in charged particle transport, J. Quant. Spectrosc. Radiat. Transf., 95, (2005), 407–414.

[15] Orsingher, E., Beghin, L., Time-fractional telegraph equations and telegraph processes with brownian time, Probab. Theory Relat. Fields, 128, (2004), 141–160.

[16] Kumar, A., Bhardwaj, A., Dubey, S., A local meshless method to approximate the time-fractional telegraph equation, Eng. Comput., (2020), https://doi.org/10.1007/s00366-020-01006-x.

[17] Wang, Y.L., Du, M.-J., Temuer, C.-L., Tian, D., Using reproducing kernel for solving a class of time-fractional telegraph equation with initial value conditions, Int. J. Comput. Math., 95(8), (2018), 1609–1621.

[18] Pandey, R.K., Mishra, H.K., Numerical simulation for solution of space-time fractional telegraphs equations with local fractional derivatives via HAFSTM, New Astronomy, 57, (2017), 82–93.

[19] Yüzbaşı, Ş., Karaçayır, M., A Galerkin-type method to solve one-dimensional telegraph equation using collocation points in initial and boundary conditions, Int. J. Comput. Methods, 15(1), (2018), 1850031, (16 pages).

[20] Dehghan, M., Shokri, A., A numerical method for solving the hyperbolic telegraph equation, Numer. Methods Partial Differ. Equ., 24, (2008), 1080–1093.

[21] Dehghan, M., Lakestani, M., The use of Chebyshev cardinal functions for solution of the second-order one-dimensional telegraph equation, Numer. Methods Partial Differ. Equ., 25, (2009), 931–938.

[22] Pandit, S., Kumar, M., Tiwari, S., Numerical simulation of second-order hyperbolic telegraph type equations with variable coefficients, Comput. Phys. Commun., 187, (2015), 83–90.

[23] Sharifi, S., Rashidinia, J., Numerical solution of hyperbolic telegraph equation by cubic B-spline collocation method, Appl. Math. Comput., 281, (2016), 28–38.

[24] Delannoy, H., Emploi de l'echiquier pour la resolution de certains problμemes de probabilites, Assoc. Franc. Bordeaux, 24, (1895), 70–90.

[25] Banderier, C., Schwer, S., Why Delannoy numbers?, J. Stat. Plan. Infer., 135(1), (2005), 40–54.

[26] Weisstein, E.W., Delannoy Number, from MathWorld–A Wolfram Web Resource, https://mathworld.wolfram.com/DelannoyNumber.html.

[27] Comtet, L., Advanced Combinatorics: The Art of Finite and Infinite Expansions, Rev. enl. ed. Dordrecht, Netherlands, Reidel, (1974), 80-81.

[28] Sulanke, R.A., Objects counted by the central Delannoy numbers, J. Integer Seq., 6, (2003), Article 03.1.5.

[29] Peart, P., Woan, W.-J., A bijective proof of the Delannoy recurrence, Congressus Numerantium, 158, (2002), 29–33.

[30] Hetyei, G., Shifted Jacobi polynomials and Delannoy numbers, (2009), ArXiv e-prints arXiv:0909.5512v2.

[31] Sun, Z., Congruences involving generalized central trinomial coefficients, Sci. China Math., 57, (2014), 1375–1400.

[32] Sun, Z.-W., On Delannoy numbers and Schröder numbers, J. Number Theory, 131, (2011), 2387–2397.

[33] Torres, A., Cabada, A., Nieto, J.J., An exact formula for the number of alignments between two DNA sequences, DNA Seq., 14, (2003), 227–430.

[34] Guo, V.J.W., Zeng, J., Proof of some conjectures of Z.-W. Sun on congruences for Apery polynomials, J. Number Theory, 132, (2012), 1731–1740.

[35] Abramowitz, M., Stegun, I.A., Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables, Nat. Bureau of Standards, Appl. Math. Series, 55, 1964.