Turkish Journal of Mathematics

http://journals.tubitak.gov.tr/math/

Research Article

Turk J Math
(2019) 43: 373 – 392
© TÜBİTAK
doi:10.3906/mat-1806-87

# A novel graph-operational matrix method for solving multidelay fractional differential equations with variable coefficients and a numerical comparative survey of fractional derivative types

**Ömür Kıvanç KÜRKÇÜ**[1*], **Ersin ASLAN**[2], **Mehmet SEZER**[3]
[1]Department of Mathematics, Faculty of Arts and Sciences, İzmir University of Economics, İzmir, Turkey
[2]Department of Software Engineering, Faculty of Engineering, Manisa Celal Bayar University, Manisa, Turkey
[3]Department of Mathematics, Faculty of Sciences and Arts, Manisa Celal Bayar University, Manisa, Turkey

**Received:** 07.07.2018 • **Accepted/Published Online:** 17.12.2018 • **Final Version:** 18.01.2019

**Abstract:** In this study, we introduce multidelay fractional differential equations with variable coefficients in a unique formula. A novel graph-operational matrix method based on the fractional Caputo, Riemann–Liouville, Caputo–Fabrizio, and Jumarie derivative types is developed to efficiently solve them. We also make use of the collocation points and matrix relations of the matching polynomial of the complete graph in the method. We determine which of the fractional derivative types is more appropriate for the method. The solutions of model problems are improved via a new residual error analysis technique. We design a general computer program module. Thus, we can explicitly monitor the usefulness of the method. All results are scrutinized in tables and figures. Finally, an illustrative algorithm is presented.

**Key words:** Collocation points, fractional derivative, graph theory, matching polynomial, matrix method

## 1. Introduction

In this study, we introduce multidelay fractional differential equations with variable coefficients in a unique formula:

$$\sum_{k=0}^{m} P_{k,n}\left(t\right) {}_{t}^{M}D_{a}^{n} y\left(\lambda_{k,n}t+\mu_{k,n}\right)=f\left(t\right),\ n\geq 0,\ c\leq t\leq d, \tag{1.1}$$

subject to the mixed conditions

$$\sum_{j=0}^{\lceil n\rceil-1} c_{j,n}D^{j}y\left(c\right)+d_{j,n}D^{j}y\left(d\right)=\tau_{j}. \tag{1.2}$$

Here, $P_{k,n}(t)$, $f(t)$ are analytic functions on $c\leq t\leq d$; $\lambda_{k,n}$, $\tau_j$, and $\mu_{k,n}$ are proper constants; ${}_{t}^{M}D_{a}^{n}y(t)$, $(a\in\mathbb{Z},\ n\in\mathbb{R}^{+})$ gives the $n$th integer and fractional order derivatives of $y(t)$ with respect to $t$, and $M$ represents well-known fractional Caputo [14], Riemann–Liouville [17, 33, 41], Caputo–Fabrizio [15, 38], and Jumarie [20, 32] derivative types. From now on, we denote them respectively as C, RL, CF, and J in this study.

Our first object in this study is to efficiently solve Eq. (1.1) by considering different fractional derivative types. The second object is to determine which of the fractional derivative types is more appropriate for the

*Correspondence: omurkivanc@outlook.com
2010 *AMS Mathematics Subject Classification:* 34K37, 65L60, 68R10, 05C31

method. We employ the graph-operational matrix method to find a matching polynomial solution of Eq. (1.1) in the form

$$y\left(t\right) \cong y_N\left(t\right) = \sum_{r=0}^{N} y_r M_r\left(K_r, t\right), \tag{1.3}$$

where $y_r$ is the unknown coefficients to be found and $K_r$ is a complete graph with $r$ vertices and $r(r\text{-}1)/2$ edges. The standard collocation points used in matrices are defined by

$$t_i = c + \left(\frac{d-c}{N}\right) i, \tag{1.4}$$

where $i = 0, 1, \ldots, N$ and $c = t_0 < t_1 < \ldots < t_N = d$.

Since 1695, fractional differential equations (FDEs) and their classes continue to be of importance in mathematics, physics, electroanalytical chemistry, capacitor theory, electrical circuits, biology, control theory, and fluid dynamics [9, 13–15, 17, 20, 32, 33, 38, 39, 41, 45]. FDEs consist of broad classes of ordinary and fractional differential equations. Thus, we can establish the modeling of stiff physical phenomena. For example, Bagley and Torvik [9] constructed a fractional differential equation that estimates the placement of a rigid plate in Newtonian fluid. Bode [13, 41] established a system that investigates the feedback of amplifier design. It is necessary to specify that there is a problem while finding the analytical solution of FDEs. Most of the time, solving FDEs analytically is too hard. Efficient numerical methods are developed in order to eliminate this hardship. Some of them are the Taylor matrix [23, 24], Bernstein matrix collocation [51], quadrature [44], differential transform [47], Fermat tau [50], multiscale collocation [40], and homotopy perturbation methods [3].

Recently, Gülsu et al. [23, 24] obtained the approximate solutions of fractional relaxation–oscillation and Bagley–Torvik equations. Heris and Javidi [30] studied fractional delay differential equations with periodic and antiperiodic conditions. Rahimkhani et al. [42] proposed the Bernoulli wavelet to solve fractional pantograph differential equations. Abd-Elhameed and Youssri [1, 2] introduced the fifth-kind orthonormal Chebyshev polynomial and the generalized Lucas polynomial sequence methods to obtain the numerical solution of FDEs. The stability and asymptotics of solutions of fractional differential equations with constant delay were investigated by Cermák et al. [16].

In the literature there is no study introducing a generalized form of delay FDEs with single fractional term. We thus develop a novel graph-operational matrix method based on simplified matrices and collocation points along with the matching polynomial of the complete graph in order to solve Eq. (1.1).

Readers are encouraged to see Sections 1.1 and 1.2 for brief information about fractional calculus, matching polynomials, and graph theory.

### 1.1. A brief introduction to fractional calculus

Let us give some definitions and notations that form a basis of fractional calculus. Let $f\left(t\right)$ be a continuous function on $I = \{[a, b] : a, b \in \mathbb{R}\}$, $D$ be a differential operator, and $\Gamma\left(n\right)$ be a Euler gamma function. We give the following definitions.

**Definition 1.1** *[17, 33, 41] Assume that $n > 0$ and $t > a$ $(n, t \in \mathbb{R})$. Then the fractional integral of $f\left(t\right)$ is defined by*

$$J_a^n f(t) = \frac{1}{\Gamma(n)} \int_a^t (t-s)^{n-1} f(s)\, ds,$$

*where $J_a^n$ is a fractional integral operator.*

**Definition 1.2** *[17, 33, 41] Assume that $t > a$ and $n, t \in \mathbb{R}$. Then the fractional Riemann–Liouville derivative of order $n$ is defined by*

$${}_t^{RL}D_a^n f(t) = \begin{cases} \frac{1}{\Gamma(\lceil n \rceil - n)} D^{\lceil n \rceil} \int_a^t (t-s)^{\lceil n \rceil - n - 1} f(s)\, ds, & \lceil n \rceil - 1 < n < \lceil n \rceil \\ f^{(n)}(t), & n \in \mathbb{N}. \end{cases}$$

Note that there is an noncompliance while taking the derivative of a constant. For example, ${}_t^{RL}D_0^{1/2}(1) = 1/\sqrt{t\pi}$, but it should have been zero. To eliminate this noncompliance, Caputo [14] introduced the following fractional derivative type.

**Definition 1.3** *[14] Assume that $t > a$ and $n, t \in \mathbb{R}$. Then the fractional Caputo derivative of order $n$ is defined by*

$${}_t^{C}D_a^n f(t) = \begin{cases} \frac{1}{\Gamma(\lceil n \rceil - n)} \int_a^t (t-s)^{\lceil n \rceil - n - 1} f^{(\lceil n \rceil)}(s)\, ds, & \lceil n \rceil - 1 < n < \lceil n \rceil \\ f^{(n)}(t), & n \in \mathbb{N}. \end{cases}$$

In recent years, Jumarie [32] modified the fractional Riemann–Liouville derivative to eliminate errors in the calculation of derivative of constants. Caputo and Fabrizio [15] established a new fractional derivative without singular kernel. Both are as follows.

**Definition 1.4** *[32] Assume that $t > 0$ $(t \in \mathbb{R})$. Then the fractional Jumarie derivative of order $n$ is defined by*

$${}_t^{J}D_0^n f(t) = \frac{1}{\Gamma(1-n)} D^1 \int_0^t (t-s)^{-n} (f(s) - f(0))\, ds, \; 0 < n < 1.$$

Then Ghosh et al. [20] generalized the above formulation as

$${}_t^{J}D_a^n f(t) = \begin{cases} \frac{1}{\Gamma(1-n)} D^1 \int_a^t (t-s)^{-n} (f(s) - f(a))\, ds, & 0 < n < 1 \\ D^m \left( {}_t^{J}D_a^{n-m} f(t) \right), & m \geq 1 \;\&\; m \leq n < m+1. \end{cases}$$

**Definition 1.5** *[15, 38] Assume that $t \geq 0$. Then the fractional Caputo–Fabrizio derivative of order $n$ is defined by*

$${}_t^{CF}D_0^n f(t) = \frac{1}{1-n} \int_0^t \exp\left( -\frac{n(t-s)}{1-n} \right) f'(s)\, ds, \; 0 < n < 1.$$

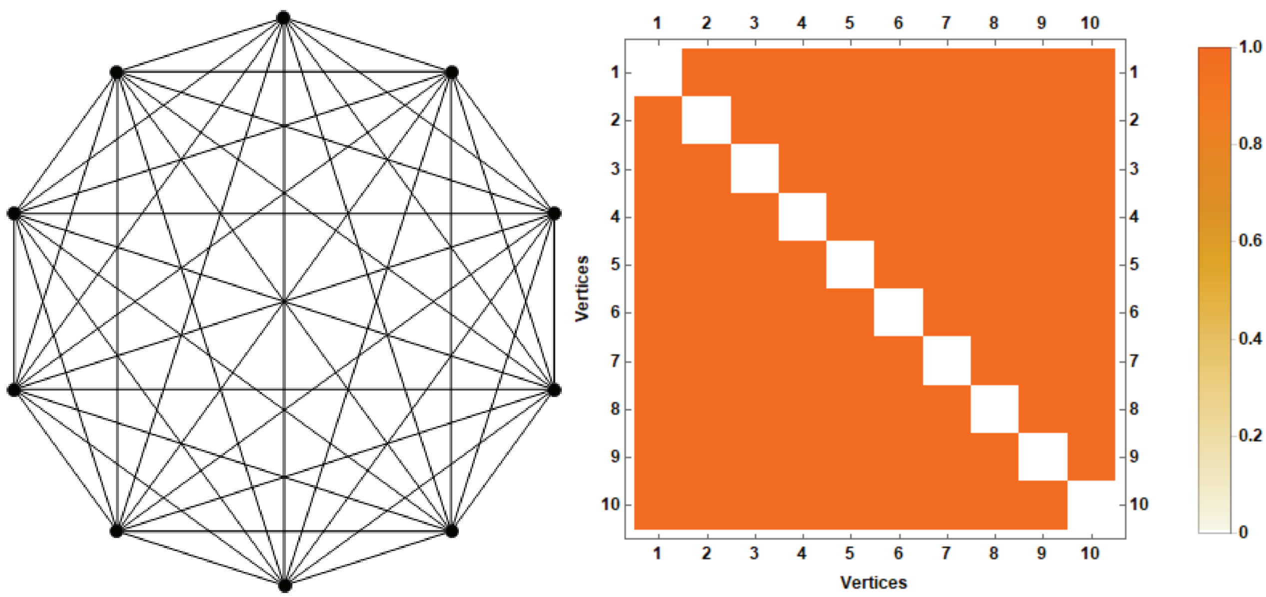
When $m \geq 1$ and $n \in [0,1]$, ${}_{t}^{CF}D_{0}^{m+n}f(t)$ is of the following form [10, 15, 38]:

$$ {}_{t}^{CF}D_{0}^{n+m}f(t) = {}_{t}^{CF}D_{0}^{n}\left({}_{t}^{C}D_{0}^{m}f(t)\right). $$

Using Definitions 1.2, 1.3, 1.4, and 1.5 we merge them with the graph-operational matrix collocation method.

### 1.2. Some properties of graph theory and matching polynomials

Graph theory is used in vulnerability, modeling of real life and networks, transportation, algorithms, and mathematical chemistry [7, 8, 22, 25, 28, 31]. A finite simple graph is made up of a vertex set $V(G)$ and an edge set $E(G)$. Edges in a graph connect vertices to each other. Complete, complete bipartite, cycle, wheel, path, tree, and star graphs are fundamental structures for graph theory. Our terminology and notations are taken from [22, 28]. Complete graph $K_r$ is illustrated in Figure 1a for $r$=10. The reason why we choose $K_r$ for our method is that $K_r$ has a durable structure, because its vertices are incident to each other. Therefore, the connection on $K_r$ is strictly sustainable when it encounters a damage or an interruption. In Figure 1b, the shaded (orange) area represents the durableness of $K_{10}$, while the diagonal white area shows that $K_{10}$ is a simple graph.




(a) (b)

**Figure 1**. A complete graph (a) $K_{10}$ and its adjacency matrix plot (b).

In 1972, Heilman and Lieb [29] introduced a polynomial, which has no specific name, for the theory of monomer-dimer systems. In 1979, Farrell [19] denominated it as the matching polynomial, which is obtained by gathering $k$-matching numbers of independent edges in a graph. Some authors have used this polynomial with different names, such as acyclic [26, 27] and reference [5]. Let $G_n$ be a graph with $n$ vertices and $m$ edges. It is generally defined by

$$ M_n(G_n,t) = \sum_{k=0}^{m} (-1)^k p(G_n,k)\, t^{n-2k}, $$

where $p(G,0)=1$ and $p(G_n,k)$ is the matching number to be obtained by $k$-matching numbers [19, 26, 27, 29]. Aihara [5] used the reference (matching) polynomial for monocyclic conjugated system. Hosoya [31] established the mathematical properties and physicochemical interpretations of the matching and some other polynomials in chemistry. Godsil and Gutman [21] constructed the properties of the matching polynomial. Yan and Yeh [49] studied the matching polynomial of subdivision graphs. Araujo et al. [6] generalized the matching polynomial to make a comparison between generalized matching polynomial and hypergeometric functions. Bian et al. [12] constructed the recurrences of the matching polynomial of ortho-chains and meta-chains.

The matching polynomial of a complete graph is of the following explicit form [26, 27]:

$$M_n(K_n,t)=\sum_{k=0}^{\lfloor n/2\rfloor}(-1)^k\frac{(2k)!}{2^k k!}\begin{pmatrix} n \\ 2k \end{pmatrix}t^{n-2k},$$

or the recurrence relation

$$M_n(K_n,t)=tM_n(K_{n-1},t)-(n-1)M_n(K_{n-2},t),$$

which are equivalent to the modified Hermite polynomials (see [26, 27, 48]). The readers can refer to more studies on the matching polynomial in [5, 6, 12, 19, 21, 26, 27, 29, 31, 48, 49].

## 2. Construction of the method

In this section, we introduce new matrix relations to constitute the method. In addition, fractional derivatives C, CF, RL, and J are separately applied to the method. We construct the matrix relation of the matching polynomial solution (1.3) of Eq. (1.1):

$$y(t)\cong y_N(t)=\boldsymbol{M}_r(K_r,t)\,\boldsymbol{Y},\ \ \boldsymbol{M}_r(K_r,\ t)=\boldsymbol{X}(t)\,\boldsymbol{S}\Rightarrow y(t)=\boldsymbol{X}(t)\,\boldsymbol{S}\boldsymbol{Y}$$

and

$$ {}_{t}^{M}D_a^n y(t)={}^{M}\boldsymbol{X}_{k,n}(t)\,\boldsymbol{S}\boldsymbol{Y},\ \ n\in\mathbb{R}^+,\ \ k=0,1,\ldots,m,\ \ \text{M=\{C, CF, RL, J\}}, \tag{2.1}$$

where

$$\boldsymbol{X}_{k,0}(t)=\begin{bmatrix} 1 & t & t^2 & \cdots & t^N \end{bmatrix}\ \text{for } n=0,$$

$${}^{M}\boldsymbol{X}_{k,n}(t)=\begin{bmatrix} {}_{t}^{M}D_a^n(1) & {}_{t}^{M}D_a^n(t) & \cdots & {}_{t}^{M}D_a^n\left(t^N\right) \end{bmatrix}\ \text{for } n\neq 0,$$

$$\boldsymbol{Y}=\begin{bmatrix} y_0 & y_1 & \cdots & y_N \end{bmatrix}^T,$$

and ($N_{e,o}$ represents even or odd number of $N$)

$$\boldsymbol{S}^T=\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
0 & \begin{pmatrix} 1 \\ 0 \end{pmatrix} & 0 & \cdots & 0 \\
(-1)^1\frac{2!}{2^1 1!}\begin{pmatrix} 2 \\ 2 \end{pmatrix} & 0 & \begin{pmatrix} 2 \\ 0 \end{pmatrix} & \cdots & 0 \\
0 & (-1)^1\frac{2!}{2^1 1!}\begin{pmatrix} 3 \\ 2 \end{pmatrix} & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
(-1)^{\frac{N}{2}}\frac{N!}{2^{\frac{N}{2}}\frac{N}{2}!}\begin{pmatrix} N \\ N \end{pmatrix} & 0 & (-1)^{\frac{N}{2}-1}\frac{(N-1)!}{2^{\frac{N}{2}-1}\left(\frac{N}{2}-1\right)!}\begin{pmatrix} N \\ N-2 \end{pmatrix} & \cdots & \left.\begin{pmatrix} N \\ 0 \end{pmatrix}\right\} N_e \\
0 & (-1)^{\left\lfloor\frac{N}{2}\right\rfloor}\frac{\left(2\left\lfloor\frac{N}{2}\right\rfloor\right)!}{2^{\left\lfloor\frac{N}{2}\right\rfloor}\left\lfloor\frac{N}{2}\right\rfloor!}\begin{pmatrix} N \\ 2\left\lfloor\frac{N}{2}\right\rfloor \end{pmatrix} & 0 & \cdots & \left.\begin{pmatrix} N \\ 0 \end{pmatrix}\right\} N_o
\end{bmatrix}.$$

Now, taking $t \to \lambda_{k,n}t + \mu_{k,n}$ into the form (2.1), we write

$$ {}_{t}^{M}D_{a}^{n} y\left(\lambda_{k,n}t+\mu_{k,n}\right) = {}^{M}\boldsymbol{X}_{k,n}\left(\lambda_{k,n}t+\mu_{k,n}\right)\boldsymbol{S}\boldsymbol{Y}, \tag{2.2} $$

where

$$ {}^{M}\boldsymbol{X}_{k,n}\left(\lambda_{k,n}t+\mu_{k,n}\right) = \left[\begin{array}{cccc} {}_{t}^{M}D_{a}^{n}(1) & {}_{t}^{M}D_{a}^{n}\left(\lambda_{k,n}t+\mu_{k,n}\right) & \cdots & {}_{t}^{M}D_{a}^{n}\left(\lambda_{k,n}t+\mu_{k,n}\right)^{N} \end{array}\right], $$

and similarly, we obtain the following form for $n = 0$:

$$ \boldsymbol{X}_{k,0}\left(\lambda_{k,0}t+\mu_{k,0}\right) = \left[\begin{array}{cccc} 1 & \left(\lambda_{k,0}t+\mu_{k,0}\right) & \cdots & \left(\lambda_{k,0}t+\mu_{k,0}\right)^{N} \end{array}\right]. $$

It is crucial to state from (2.2) that the derivative of a constant is different from zero when $M$=RL. For example, ${}_{t}^{RL}D_{0}^{1/2}(1) = 1/\sqrt{t\pi}$. This leads to obtaining infinity when the initial point is $c$=0. In some model problems, we therefore take $c = 10^{-16}$, which is very close to zero.

The fundamental matrix equation of Eq. (1.1) is obtained by substituting Eqs. (1.4) and (2.2) into Eq. (1.1) as

$$ \boldsymbol{W} = \sum_{k=0}^{m} \boldsymbol{P}_{k,n}\boldsymbol{X}_{k,n}\boldsymbol{S} \Rightarrow \boldsymbol{W}\boldsymbol{Y} = \boldsymbol{F}, \tag{2.3} $$

where

$$ \boldsymbol{W} = \left[\begin{array}{c} W(t_0) \\ W(t_1) \\ \vdots \\ W(t_N) \end{array}\right], \quad \boldsymbol{P}_{k,n} = \left[\begin{array}{cccc} P_{k,n}(t_0) & 0 & \cdots & 0 \\ 0 & P_{k,n}(t_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_{k,n}(t_N) \end{array}\right], $$

$$ \boldsymbol{X}_{k,n} = \left[\begin{array}{c} {}^{M}\boldsymbol{X}_{k,n}\left(\lambda_{k,n}t_0+\mu_{k,n}\right) \\ {}^{M}\boldsymbol{X}_{k,n}\left(\lambda_{k,n}t_1+\mu_{k,n}\right) \\ \vdots \\ {}^{M}\boldsymbol{X}_{k,n}\left(\lambda_{k,n}t_N+\mu_{k,n}\right) \end{array}\right] = \left[\begin{array}{cccc} {}_{t}^{M}D_{a}^{n}(1) & {}_{t}^{M}D_{a}^{n}\left(\lambda_{k,n}t_0+\mu_{k,n}\right) & \cdots & {}_{t}^{M}D_{a}^{n}\left(\lambda_{k,n}t_0+\mu_{k,n}\right)^{N} \\ {}_{t}^{M}D_{a}^{n}(1) & {}_{t}^{M}D_{a}^{n}\left(\lambda_{k,n}t_1+\mu_{k,n}\right) & \cdots & {}_{t}^{M}D_{a}^{n}\left(\lambda_{k,n}t_1+\mu_{k,n}\right)^{N} \\ \vdots & \vdots & \ddots & \vdots \\ {}_{t}^{M}D_{a}^{n}(1) & {}_{t}^{M}D_{a}^{n}\left(\lambda_{k,n}t_N+\mu_{k,n}\right) & \cdots & {}_{t}^{M}D_{a}^{n}\left(\lambda_{k,n}t_N+\mu_{k,n}\right)^{N} \end{array}\right], $$

and

$$ \boldsymbol{F} = \left[\begin{array}{cccc} f(t_0) & f(t_1) & \cdots & f(t_N) \end{array}\right]^{T}. $$

Eq. (2.3) can be written as

$$ \boldsymbol{W}\;\boldsymbol{Y} = \boldsymbol{F} \text{ or } [\boldsymbol{W}\,;\,\boldsymbol{F}]. \tag{2.4} $$

The matrix relation of the conditions (1.2) is

$$ \boldsymbol{U}_j\boldsymbol{Y} = \tau_j \Rightarrow [\boldsymbol{U}_j\,;\,\tau_j],\; j = 0,1,\ldots,\lceil n\rceil - 1, \tag{2.5} $$

where

$$ \boldsymbol{U}_j = \left[\begin{array}{cccc} u_{j0} & u_{j1} & \ldots & u_{jN} \end{array}\right] = \sum_{k=0}^{\lceil n\rceil-1}\left[c_{k,n}\boldsymbol{X}_{k,n}(c) + d_{k,n}\boldsymbol{X}_{k,n}(d)\right]\boldsymbol{S}\;\boldsymbol{Y}. $$

By removing $m$ rows of (2.4), the rows of the condition matrix (2.5) are then written into (2.4). Thus, it follows that

$$\left[\tilde{\boldsymbol{W}}\,;\,\tilde{\boldsymbol{F}}\right]=\left[\begin{array}{cccccc} w_{00} & w_{01} & \cdots & w_{0N} & ; & f\left(t_0\right) \\ w_{10} & w_{11} & \cdots & w_{1N} & ; & f\left(t_1\right) \\ \vdots & \vdots & \vdots & \vdots & \vdots ; & \vdots \\ w_{N-\lceil n\rceil+1,0} & w_{N-\lceil n\rceil+1,1} & \cdots & w_{N-\lceil n\rceil+1,N} & ; & f\left(t_{N-\lceil n\rceil+1}\right) \\ u_{00} & u_{01} & \cdots & u_{0N} & ; & \tau_0 \\ u_{10} & u_{11} & \cdots & u_{1N} & ; & \tau_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots ; & \vdots \\ u_{\lceil n\rceil-1,0} & u_{\lceil n\rceil-1,1} & \cdots & u_{\lceil n\rceil-1,N} & ; & \tau_{\lceil n\rceil-1} \end{array}\right],$$

which is the augmented matrix system. There is $\operatorname{rank}\tilde{\boldsymbol{W}}=\operatorname{rank}\left[\tilde{\boldsymbol{W}}\,;\,\tilde{\boldsymbol{F}}\right]=N+1$. After solving the system, the coefficients matrix $\boldsymbol{Y}$ can be easily determined as

$$\boldsymbol{Y}=\left(\tilde{\boldsymbol{W}}\right)^{-1}\tilde{\boldsymbol{F}}.$$

Inserting these coefficients into the matching polynomial solution (1.3), we thereby obtain the matching polynomial solution of Eq. (1.1).

## 3. Residual error analysis based on fractional derivative

Kürkçü et al. [34, 35, 37] recently applied residual error analysis to ordinary integro-differential-(difference) equations and they established the residual function-based convergence analysis for model problems [36]. Our aim is to improve the obtained approximate solutions by employing the residual error analysis based on the fractional derivative. In order to apply it, an algorithm is formed as follows:

**Step 1:** $R_N(t)\leftarrow\sum_{k=0}^{m}P_{k,n}(t)\,{}_t^M D_a^n y\left(\lambda_{k,n}t+\mu_{k,n}\right)-f(t)$, where $M=\{\text{C, CF, RL, J}\}$ and $R_N(t)$ is the residual function;

**Step 2:** $L\left[e_N(t)\right]\leftarrow L\left[y(t)\right]-L\left[y_N(t)\right]=-R_N(t)$, where $e_N(t)=y(t)-y_N(t)$;

**Step 3:** $\sum_{j=0}^{\lceil n\rceil-1}\left[c_{j,n}D^j e_N(c)+d_{j,n}D^j e_N(d)\right]\leftarrow 0$;

**Step 4:** $e_{N,M}(t)\leftarrow\sum_{n=0}^{M}y_n^* M_n\left(K_n,t\right)$, where $e_{N,M}(t)$ is the estimated error function;

**Step 5:** $y_{N,M}(t)\leftarrow y_N(t)+e_{N,M}(t)$, where $y_{N,M}(t)$ is the corrected matching polynomial solution;

**Step 6:** $E_{N,M}(t)\leftarrow y(t)-y_{N,M}(t)$, where $E_{N,M}(t)$ is the corrected error.

It follows from the above algorithm that

$$\left.\begin{array}{l}\left|y(t)-y_N(t)\right|<\varepsilon_N \\ \left|y(t)-y_{N,M}(t)\right|<\varepsilon_{N,M}\end{array}\right\}\Rightarrow\varepsilon_{N,M}<\varepsilon_N,\ \forall t\in[c,d], \tag{3.1}$$

where positive integers $\varepsilon_N$ and $\varepsilon_{N,M}$ are very close to zero. That is, we can state that the distance between $\varepsilon_{N,M}$ and $\varepsilon_N$ increases as $N$ and $M$ are increased. Thereby, we can estimate the behavior of the improved matching polynomial solution of Eq. (1.1).

## 4. Fractional model problems

Let us solve some fractional model problems using the present method based on C, CF, RL, and J. The numerical results are tabulated in tables in order to compare them. The behaviors and displacements of the matching polynomial solutions are illustrated in figures for different values of $n$. All computations are performed with a general computer program developed by the authors. The program enables us to obtain consistent results. The CPU time of the method is provided by a personal computer equipped with 8 GB RAM and 3.30 GHz CPU. ${}^{M}L_{\infty}$, ${}^{M}\boldsymbol{X}_{k,n}$, ${}^{M}y_N(t)$, ${}^{M}e_N(t)$, and ${}^{M}E_{N,M}(t)$ mean that the results of model problems are obtained by fractional derivatives $M$={C, CF, RL, J}.

**Model 4.1.** [30] Consider the fractional delay differential equation with antiperiodic condition

$$ {}_{t}^{M}D_{0}^{1/5}y(t)+y(t-1)=f(t),\ 0\leq t\leq 2, $$

subject to the conditions $y(0)=-y(2)$. Here, $P_{0,0}(t)=P_{1,0.2}(t)=1$, $\lambda_{0,0}=1$, $\mu_{0,0}=-1$, $\lambda_{1,0.2}=1$, $\mu_{1,0.2}=0$, and, by using the Caputo derivative, $f(t)$ is calculated as

$$ f(t)=\frac{\Gamma(3)\,t^{1.8}}{\Gamma(2.8)}-\frac{\Gamma(2)\,t^{0.8}}{\Gamma(1.8)}+t^2-3t+1. $$

We apply our method to solve the problem by using $N$=3 and different fractional derivative types. By Eq. (2.3), the fundamental matrix equation is

$$ \underbrace{\left\{\boldsymbol{P}_{0,0}\boldsymbol{X}_{0,0}+\boldsymbol{P}_{1,0.2}{}^{M}\boldsymbol{X}_{1,0.2}\right\}\boldsymbol{S}}_{\boldsymbol{W}}\,\boldsymbol{Y}=\boldsymbol{F}, \tag{4.1} $$

where

$$ \boldsymbol{P}_{0,0}=\boldsymbol{P}_{1,0.2}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},\boldsymbol{X}_{0,0}=\begin{bmatrix}1&-1&1\\1&0&0\\1&1&1\end{bmatrix},\ {}^{C}\boldsymbol{X}_{1,0.2}={}^{J}\boldsymbol{X}_{1,0.2}=\begin{bmatrix}0&0&0\\0&1.0737&1.1930\\0&1.8694&4.1542\end{bmatrix}, $$

$$ {}^{CF}\boldsymbol{X}_{1,0.2}=\begin{bmatrix}0&0&0\\0&1.1060&1.1520\\0&1.9673&4.2612\end{bmatrix},\ \boldsymbol{S}=\begin{bmatrix}1&0&-1\\0&1&0\\0&0&1\end{bmatrix}, $$

and if the interval of the problem is taken as $10^{-16}\leq t\leq 2$, then we obtain

$$ {}^{RL}\boldsymbol{X}_{1,0.2}=\begin{bmatrix}1361.32&1.70e-13&1.89e-29\\0.85894&1.07367&1.19297\\0.74775&1.86937&4.15416\end{bmatrix}. $$

We calculate Eq. (4.1) including ${}^{C}\boldsymbol{X}_{0.2}$, ${}^{CF}\boldsymbol{X}_{0.2}$, ${}^{RL}\boldsymbol{X}_{0.2}$, and ${}^{J}\boldsymbol{X}_{0.2}$. By using ${}^{C}\boldsymbol{X}_{0.2}$ in Eq. (4.1), the augmented matrix is of the form

$$ \left[\tilde{\boldsymbol{W}}\,;\,\tilde{\boldsymbol{F}}\right]=\begin{bmatrix}1&-1&0&;&1\\1&1.07370&0.1930&;&-0.8807\\1&0&-1&;&-1\end{bmatrix}. $$

It is noticed that the above matrix is a numerically different form when ${}^{CF}\boldsymbol{X}_{0.2}$, ${}^{RL}\boldsymbol{X}_{0.2}$, and ${}^{J}\boldsymbol{X}_{0.2}$ are used. When we solve the augmented matrix, we obtain

$$\boldsymbol{Y} = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}^T.$$

Then the solution is obtained as

$$\begin{aligned} y_2(t) &= \sum_{n=0}^{2} y_n M_n(K_n, t) = y_0 M_0(K_0, t) + y_1 M_1(K_1, t) + y_2 M_2(K_2, t) \\ &= (0)\cdot 1 + (-1)\cdot t + 1\cdot\left(t^2 - 1\right) \\ &= t^2 - t - 1, \end{aligned}$$

which is the exact solution. Note that the same solution is obtained when ${}^{CF}\boldsymbol{X}_{0.2}$, ${}^{RL}\boldsymbol{X}_{0.2}$, and ${}^{J}\boldsymbol{X}_{0.2}$ are used. Heris and Javidi [30] obtained the best approximate value of $y(1)$ as $7.2526e-05$ using the fractional backward differential formula method with step size $h = 0.0001$ while the present method directly finds the exact solution.

**Model 4.2.** **[9, 18, 50]** Consider the Bagley–Torvik differential equation estimating a rigid plate in Newtonian fluid:

$$D^2 y(t) + {}_t^M D_0^{3/2} y(t) + y(t) = f(t), \ 0 < t < 1,$$

subject to the initial conditions $y(0) = 0$ and $y'(0) = 1$. When we use the fractional C, CF, RL, J derivatives, $f(t)$ is obtained respectively as

$$f(t) = \sqrt{2}\left(\cos(t)\, S\left(\sqrt{\frac{2t}{\pi}}\right) - \sin(t)\, C\left(\sqrt{\frac{2t}{\pi}}\right)\right) \text{ and } f(t) = -e^{-t} + \cos(t) - \sin(t),$$

$$f(t) = \frac{1}{\sqrt{t\pi}} + \sqrt{2}\cos(t)\, S\left(\sqrt{\frac{2t}{\pi}}\right) - \sqrt{2}\sin(t)\, C\left(\sqrt{\frac{2t}{\pi}}\right),$$

and

$$\begin{aligned} f(t) = {}& 0.02779 t^{3.5}\, {}_1F_2\left(3; 3.75,\ 3.25; -t^2/4\right) - 0.687774 t^{1.5}\, {}_1F_2\left(2; 2.75,\ 2.25; -t^2/4\right) \\ &+ 0.56419\, {}_1F_2\left(1; 1.75,\ 1.25; -t^2/4\right)/\sqrt{t}, \end{aligned}$$

where $S(t) = \int_0^t \sin(\frac{\pi x^2}{2})dx$ and $C(t) = \int_0^t \cos(\frac{\pi x^2}{2})dx$ are Fresnel integrals [4], and ${}_1F_2$ is a form of the generalized hypergeometric function ${}_pF_q$ [4]. The exact solution is $y(t) = \sin(t)$. We apply our method based on different fractional derivatives to solve the problem by taking $N$=4 to 32. We also employ the residual error analysis to improve the solutions. In Table 1, we compare the $L_\infty$ errors and the corrected $L_\infty$ errors together with the shifted Chebyshev tau (SCT) and the Fermat tau operational matrix (FTM) methods [18, 50]. As seen there, the present results are far better than those in [18, 50]. In addition, we compare CPU time according to $N$ and the FTM [50]. Although our method includes all fractional derivative types, it has better processing time than the FTM as seen in Table 2. One notices that when $N$=32 our method with CF slows down due to the fact that CF includes compact exponential function. Therefore, we use C, RL, J for the applicability of the present method. In addition, the behaviors of the obtained solutions are illustrated along with the exact solution in Figure 2.

**Table 1**. Comparison of the $L_\infty$ errors and corrected $L_\infty$ errors in terms of $N$, $M$, fractional C, CF, RL, and J derivatives for Model 4.2.

| $N, M$ | $^{C}L_\infty$ | $^{CF}L_\infty$ | $^{RL}L_\infty$ | $^{J}L_\infty$ | FTM [50] | SCT [18] |
|---|---|---|---|---|---|---|
| 4 | 5.1$e$-04 | 4.0$e$-04 | 5.1$e$-04 | 5.1$e$-04 | 2.7$e$-04 | 3.4$e$-04 |
| 4, 5 | 2.2$e$-05 | 1.9$e$-05 | 2.6$e$-05 | 2.2$e$-05 | n.a. | n.a. |
| 8 | 6.5$e$-09 | 6.1$e$-09 | 6.1$e$-04 | 6.5$e$-09 | 3.5$e$-07 | 4.3$e$-07 |
| 8, 9 | 1.5$e$-10 | 1.4$e$-10 | 1.5$e$-10 | 1.5$e$-10 | n.a. | n.a. |
| 16 | 1.6$e$-15 | 7.8$e$-16 | 5.6$e$-14 | 3.8$e$-15 | 4.2$e$-10 | 1.8$e$-08 |
| 32 | 9.9$e$-16 | 4.2$e$-14 | 4.4$e$-16 | 9.9$e$-16 | 5.8$e$-12 | 7.1$e$-10 |

**Table 2**. Comparison of CPU time in terms of $N$ and fractional C, CF, RL, and J derivatives for Model 4.2.

| $N$ | Ours with C | Ours with CF | Ours with RL | Ours with J | FTM [50] |
|---|---|---|---|---|---|
| 4 | 0.9375 | 0.4375 | 1.3125 | 0.2344 | 23.16 |
| 8 | 2.5156 | 1.4375 | 3.2344 | 0.3438 | 110.78 |
| 16 | 9.0313 | 32.094 | 10.063 | 0.6406 | 313.27 |
| 32 | 34.391 | 1469.0 | 36.203 | 1.2188 | 1023.85 |

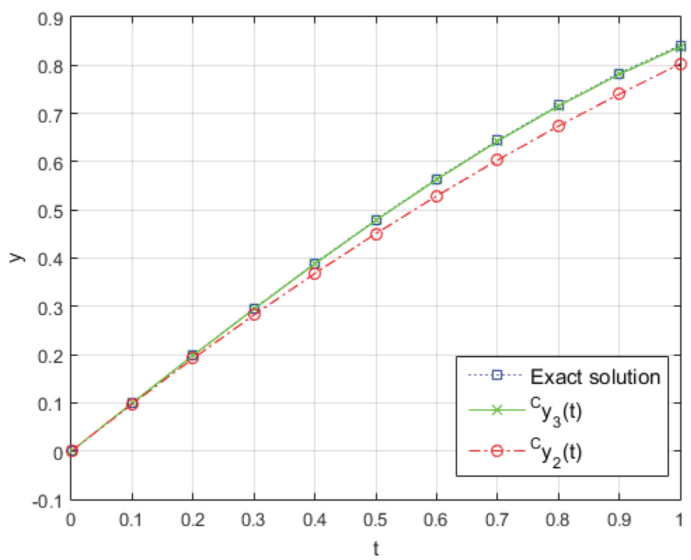



**Figure 2**. Behavior of the exact solution and matching polynomial solutions with C for Model 4.2.

**Model 4.3.** [**43**] Consider the fractional Mathieu differential equation modeled for the parametric resonance of mechanical, optical, and electrical systems:

$$ {}_t^M D_0^n y(t) + (1 - 0.32\cos(3t))\, y(t) = 0, 1 < n \le 2, 0 \le t \le 2, $$

subject to $y(0) = 1$ and $y'(0) = 0$. We solve the problem for $N$=10 and different $n$ and fractional derivative types and then we apply the residual error analysis for $M = 12$. Table 3 illustrates the numerical results of the present, Runge–Kutta [46], and shifted (first and second kind) Chebyshev polynomial methods (SCP) [46] for $n$=2 and time interval [0,2]. It is easily seen from Table 3 that the present results coincide well with the mentioned methods. In terms of different fractional derivative types and $n$, we demonstrate the displacement of the matching polynomial solution in Figure 3. As seen in Figure 3, the matching polynomial solution shows different displacements according to fractional derivatives. Thus, we can interpret Figures 3a–3d as follows:

**Table 3**. Comparison of the present and existing solutions at time [0,2] for $n$=2 and Model 4.3.

| Time $t_i$ | ${}^{C}y_{10}(t_i)$ | ${}^{C}y_{10,12}(t_i)$ | SCP (first kind) $m$=10 [46] | SCP (second kind) $m$=10 [46] | Runge–Kutta method [46] |
|---|---|---|---|---|---|
| 0.1 | 0.996591 | 0.996590 | 0.996589 | 0.996582 | 0.996589 |
| 0.2 | 0.986247 | 0.986242 | 0.986247 | 0.986245 | 0.986242 |
| 0.4 | 0.943250 | 0.943236 | 0.943232 | 0.943233 | 0.943238 |
| 0.6 | 0.866936 | 0.866915 | 0.866922 | 0.866917 | 0.866917 |
| 0.8 | 0.753746 | 0.753718 | 0.753723 | 0.753722 | 0.753721 |
| 1.0 | 0.603763 | 0.603730 | 0.603727 | 0.603730 | 0.603734 |
| 1.2 | 0.422373 | 0.422336 | 0.422343 | 0.422342 | 0.422340 |
| 1.4 | 0.219367 | 0.219329 | 0.219338 | 0.219335 | 0.219333 |
| 1.6 | 0.006062 | 0.006024 | 0.006022 | 0.006025 | 0.006028 |
| 1.8 | −0.207745 | −0.207781 | −0.207773 | −0.207771 | −0.207776 |
| 2.0 | −0.415250 | −0.415124 | −0.415133 | −0.415136 | −0.415130 |

- As $n$ approaches 2, the displacement of the solution ${}^{C}y_{10}(t)$ in Figure 3a is more consistent than that of ${}^{CF}y_{10}(t)$ in Figure 3b.
- As $n$ approaches 2, the displacement of the solution ${}^{J}y_{10}(t)$ in Figure 3d is more consistent than that of ${}^{RL}y_{10}(t)$ in Figure 1c.
- ${}^{C}y_{10}(t)$ is the most proper solution for Model 4.3 in comparison to the others.

On the other hand, the 3D plot of absolute errors with respect to $N$ is depicted in Figure 4.

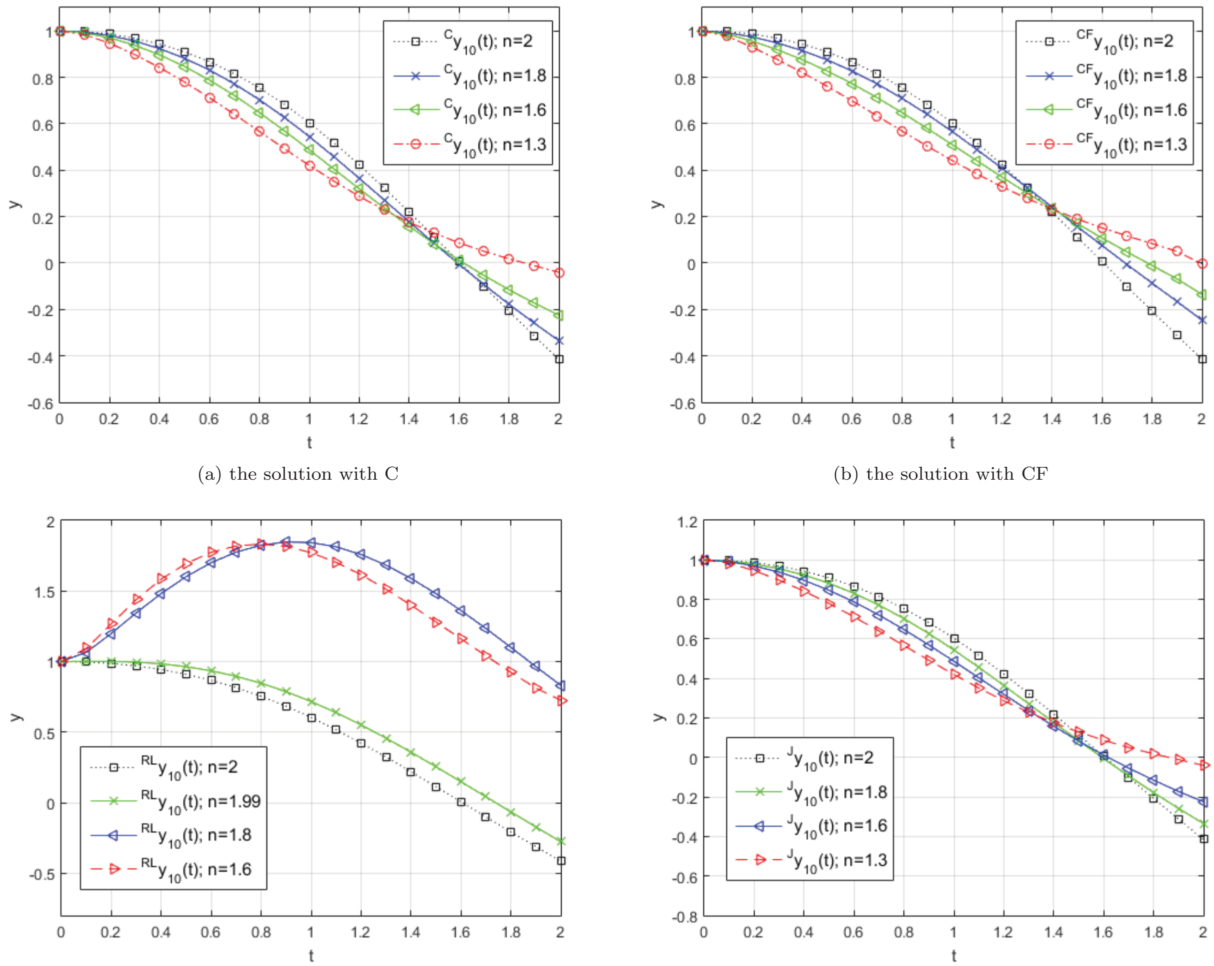
**Model 4.4.** [11] Consider the generalized fractional differential equation

$$ {}_{t}^{M}D_{0}^{5/2}y(t)+y(t)+y(t-0.5)=f(t),\ 0\leq t\leq 1, $$

subject to the initial conditions $y(0)=y'(0)=y''(0)=0$. The exact solution of this problem is $y(t)=t^3$. When our method based on C is applied to the problem it is observed from Table 4 that the obtained results for $N$=3 are far better than those obtained by Laguerre–Gauss method (LGM) ($N$=22) in [11]. The same results are also obtained when CF, RL, and J are used.

**Model 4.5.** [1] Consider the linear fractional oscillator equation modeled for the behavior of a physical pendulum:

$$ {}_{t}^{M}D_{0}^{n}y(t)+w^2y(t)=0,\ 0\leq t\leq L,\ 1<n\leq 2, $$




(a) the solution with C

(b) the solution with CF

(c) the solution with RL

(d) the solution with J

**Figure 3**. Displacement of the matching polynomial solution in terms of $n$ on [0,2] for Model 4.3.

**Table 4**. Comparison of the present and existing results at time interval [0,1] for Model 4.4.

| Time $t_i$ | ${}^{C}y_3\,(t_i)$ | LGM $N$=22 [11] |
|---|---|---|
| 0.1 | $8.3e-17$ | $6.3e-06$ |
| 0.2 | $8.3e-17$ | $3.9e-05$ |
| 0.3 | $8.3e-17$ | $1.0e-04$ |
| 0.4 | $8.3e-17$ | $1.9e-04$ |
| 0.5 | $8.3e-17$ | $2.9e-04$ |
| 0.6 | $8.3e-17$ | $4.1e-04$ |
| 0.7 | $5.6e-17$ | $5.3e-04$ |
| 0.8 | $1.1e-16$ | $6.6e-04$ |
| 0.9 | $1.1e-16$ | $8.0e-04$ |
| 1.0 | $1.1e-16$ | $9.5e-04$ |

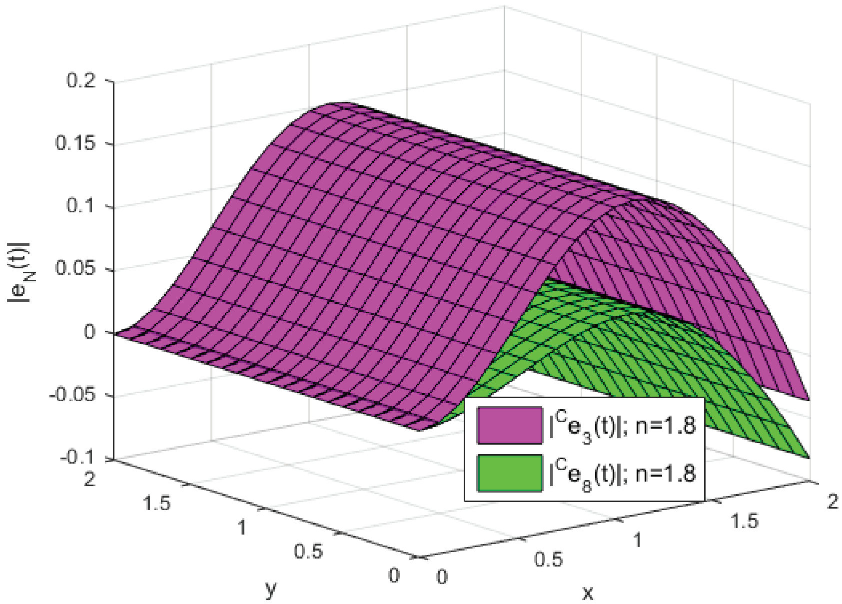



**Figure 4**. Comparison of the absolute errors with C in terms of $N$ for Model 4.3.

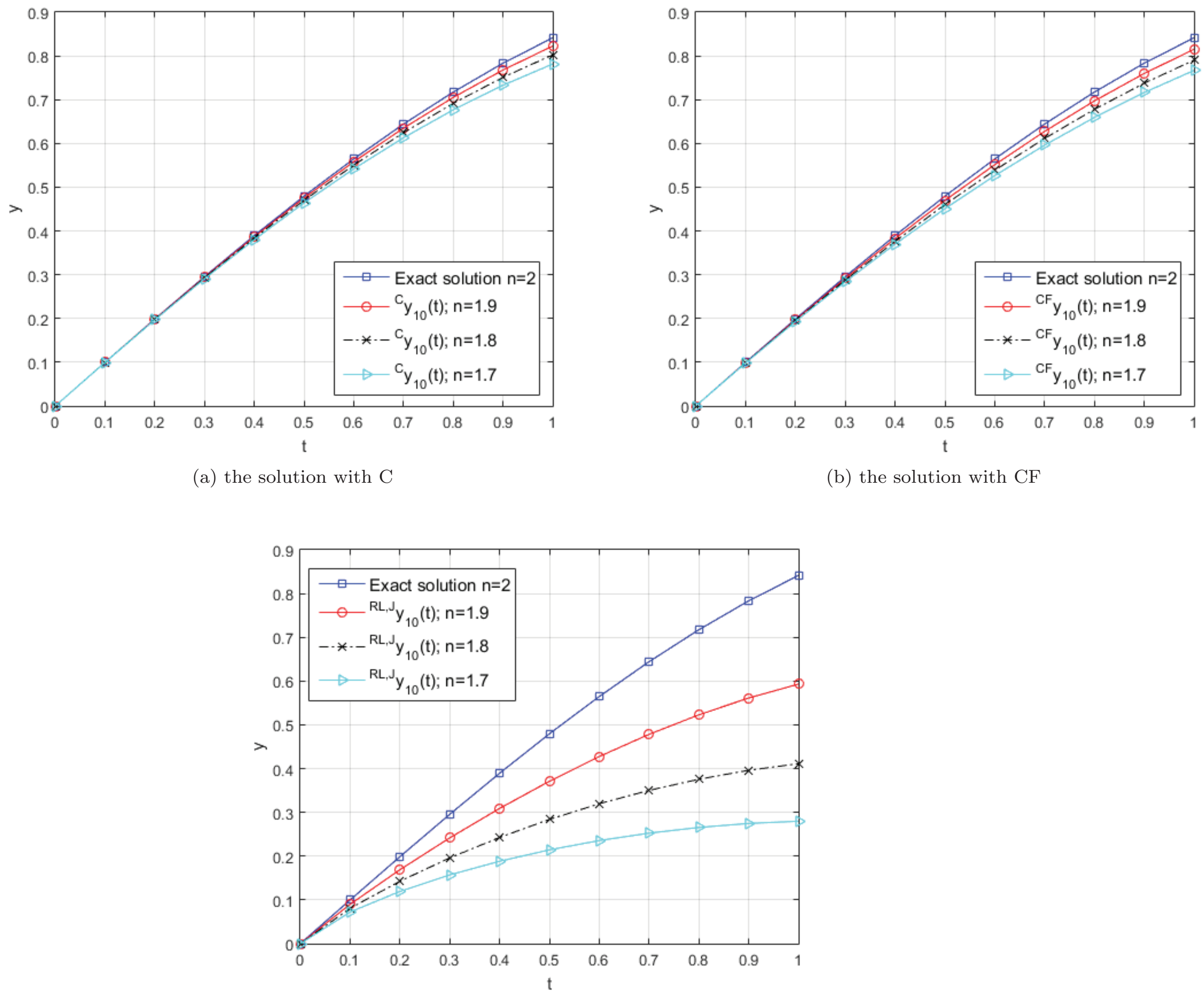
subject to the initial conditions $y(0) = 0$ and $y'(0) = w$. Here, $w$=1, $L = \{1, 15\}$, and the exact solution of the problem is $y(t) = \sin t$ for $n$=2. After employing the present method and residual error analysis based on C, CF, RL, and J, Table 5 indicates that we obtain remarkable CPU time (s) and highly accurate results compared to Lucas and Legendre tau methods (LuTM and LeTM) [1]. Figure 5 shows the displacement of the solution $y_{10}(t)$ when $L$=1 and $n$ is increased. In addition, Figure 6 shows the behavior of the solution $y_{20}(t)$ when $L = 15$. Thereby, we infer from Figures 5 and 6 that the Caputo derivative provides the most proper fractional derivative for this model when $L = 1$, and similarly, the Caputo–Fabrizio derivative does so when $L = 15$.

**Table 5**. Comparison of CPU time and $L_\infty$ errors in terms of $N$, $M$, $L = 1$, and $n = 2$ for Model 4.5.

| $N, M$ | ${}^{C}L_\infty$ | CPU time | LuTM $L_\infty$ [1] | LuTM CPU time [1] | LeTM $L_\infty$ [1] | LeTM CPU time [1] |
|---|---|---|---|---|---|---|
| 4 | $1.3e-03$ | 0 | $1.3e-04$ | 12.35 | $8.3e-03$ | 19.47 |
| $4, 6$ | $5.0e-06$ | 0 | n.a. | n.a. | n.a. | n.a. |
| 8 | $1.1e-08$ | 0 | $5.4e-07$ | 33.58 | $2.5e-06$ | 45.17 |
| $8, 10$ | $1.6e-11$ | 0 | n.a. | n.a. | n.a. | n.a. |
| 16 | $1.1e-16$ | 0 | $2.2e-14$ | 67.59 | $5.4e-14$ | 95.24 |
| 32 | $1.1e-16$ | 0.05 | $4.4e-16$ | 91.27 | $9.3e-15$ | 174.12 |

**Model 4.6.** Consider the multidelay fractional differential equation with variable coefficients:

$$\sin(t)\, {}_t^M D_0^{8/5} y(t-1) + {}_t^M D_0^{13/10} y(0.5t) - t^2 D^1 y(t) + ty(2t) = f(t), \; 0 \le t \le 1,$$




(a) the solution with C

(b) the solution with CF

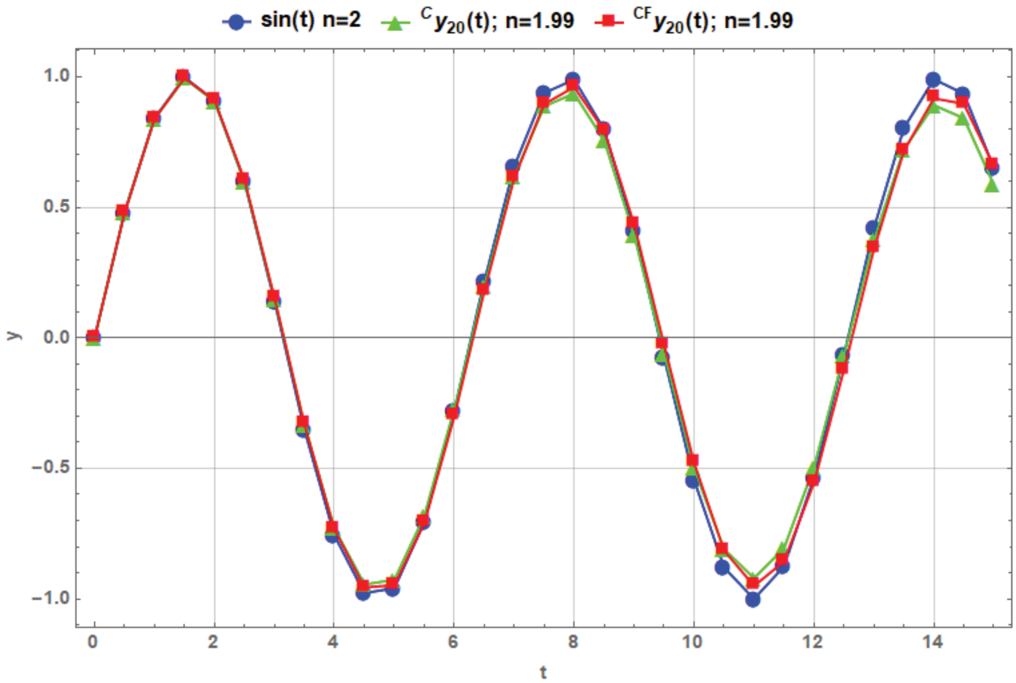
(c) the solution with RL and J

**Figure 5**. Displacement of the exact solution and matching polynomial solution in terms of $n$ on [0,1] for Model 4.5.

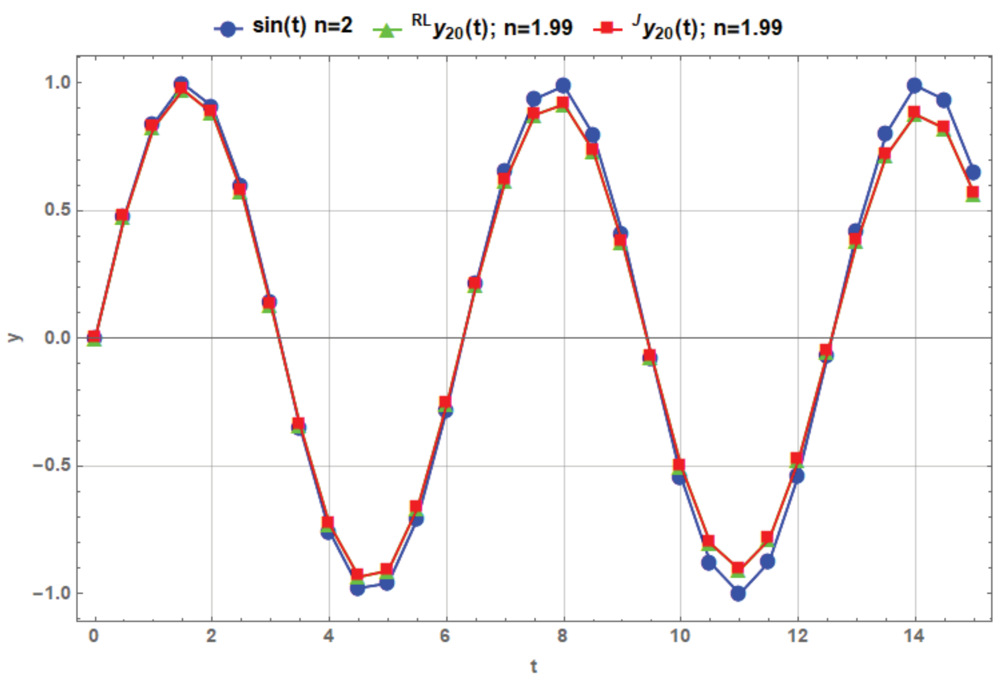
subject to the initial conditions $y(0) = y'(0) = 1$. The exact solution of the problem is $y(t) = e^t$ and $f(t)$ is differently obtained when the fractional derivative types are changed. Let us solve this tough problem by using the present method based on C, CF, RL, and J. Then we improve the obtained solutions via the residual error analysis with $N = 8$ and $M = 10$. It is observed from Table 6 that our solution with respect to C, CF, RL, and J is improved via the residual error analysis for $N = 8$ and $M = 10$. Table 6 clearly indicates that the residual error analysis is convenient for the present method based on different fractional derivative types. In addition, we point out that our estimation (3.1) is satisfied in Table 6. The present method based on CF and RL leads to better results. We indicate CPU times in Table 7. It is noticed from Table 7 that using RL in our method causes a long processing time. Hence, we can choose a CF-based method for efficiently solving this model.

## 5. An illustrative algorithm

We describe an algorithm that summarizes the formulation of the method. It can be easily adopted to well-known computer programs.




(a) the solution with C and CF




(b) the solution with RL and J

**Figure 6**. Behavior of the exact solution and matching polynomial solution in terms of $n$ on [0,15] for Model 4.5.

**Step 1:** Start

**Step 2:** Input($N$, $c$, $d$)

**Table 6**. Comparison of the corrected absolute and absolute errors in terms of C, CF, RL, and J for Model 4.6.

| $\mathrm{t_i}$ | $\left\|{}^{\mathrm{C}}\mathrm{e}_8(\mathrm{t})\right\|$ | $\left\|{}^{\mathrm{C}}\mathrm{E}_{8,10}(\mathrm{t})\right\|$ | $\left\|{}^{\mathrm{CF}}\mathrm{e}_8(\mathrm{t})\right\|$ | $\left\|{}^{\mathrm{CF}}\mathrm{E}_{8,10}(\mathrm{t})\right\|$ |
|---|---|---|---|---|
| 0.1 | 2.25e − 06 | 3.96e − 10 | 6.31e − 08 | 1.08e − 09 |
| 0.2 | 1.10e − 05 | 3.06e − 09 | 8.32e − 07 | 3.75e − 09 |
| 0.4 | 5.79e − 05 | 2.34e − 08 | 7.79e − 06 | 1.03e − 08 |
| 0.6 | 1.51e − 04 | 7.33e − 08 | 2.63e − 05 | 1.31e − 08 |
| 0.8 | 2.85e − 04 | 1.55e − 07 | 5.97e − 05 | 8.32e − 09 |
| 1.0 | 4.37e − 04 | 2.61e − 07 | 1.10e − 04 | 5.00e − 09 |
| $\mathrm{t_i}$ | $\left\|{}^{\mathrm{RL}}\mathrm{e}_8(\mathrm{t})\right\|$ | $\left\|{}^{\mathrm{RL}}\mathrm{E}_{8,10}(\mathrm{t})\right\|$ | $\left\|{}^{\mathrm{J}}\mathrm{e}_8(\mathrm{t})\right\|$ | $\left\|{}^{\mathrm{J}}\mathrm{E}_{8,10}(\mathrm{t})\right\|$ |
| 0.1 | 1.28e − 06 | 1.12e − 10 | 1.34e − 06 | 1.32e − 09 |
| 0.2 | 5.23e − 06 | 4.63e − 10 | 5.14e − 06 | 5.07e − 09 |
| 0.4 | 2.05e − 05 | 1.80e − 09 | 1.77e − 05 | 1.74e − 08 |
| 0.6 | 4.17e − 05 | 3.55e − 09 | 3.03e − 05 | 2.98e − 08 |
| 0.8 | 6.12e − 05 | 4.99e − 09 | 3.44e − 05 | 3.78e − 08 |
| 1.0 | 7.09e − 05 | 5.54e − 09 | 2.30e − 05 | 2.27e − 08 |

**Table 7**. Comparison of CPU time in terms of $N$, C, CF, RL, and J for Model 4.6.

| $N$ | The method with C | The method with CF | The method with RL | The method with J |
|---|---|---|---|---|
| 4 | 2.25 | 0.42 | 3.53 | 0.56 |
| 16 | 22.1 | 5.25 | 27.5 | 2.78 |

**Step 3:** $M_r(K_r,t) \leftarrow p(K_r,0), p(K_r,1), \ldots, p(K_r,m)$

**Step 4:** $\boldsymbol{S}^T \leftarrow$ takecoeffcients $\left(t^0, t^1, \ldots, t^n\right)$

**Step 5:** Compose(C, CF, RL, J)

**Step 6:** Recall the collocation points

**Step 7:** $\boldsymbol{W} \leftarrow$write($\boldsymbol{P}_{k,n}$, ${}^{M}\boldsymbol{X}_{k,n}$, $\boldsymbol{Y}$, $\boldsymbol{F}$)

**Step 8:** $\tilde{\boldsymbol{W}} \leftarrow$ join ($\boldsymbol{W}$, $\boldsymbol{U}_j$)

**Step 9:** $\tilde{\boldsymbol{F}} \leftarrow$ join ($\boldsymbol{F}$, $\tau_j$)

**Step 10:** $\boldsymbol{Y} \leftarrow$ linearsolve $\left(\tilde{\boldsymbol{W}}\,;\,\tilde{\boldsymbol{F}}\right)$

**Step 11:** $y_N(t) \leftarrow \boldsymbol{M}_r(K_r,t)\,\boldsymbol{Y}$

**Step 12:** Stop

## 6. Conclusions

The multidelay FDEs under the unique formulation (1.1) have been practically solved by a novel method consisting of different fractional derivative types. Taking into account Eq. (1.1), we can construct the following:

- Proportional delay FDEs: $\sum_{k=0}^{m} P_{k,n}(t)\, {}_{t}^{M}D_{a}^{n} y(\lambda_{k,n} t) = f(t)$ for $\mu_{k,n} = 0$,

- Constant delay or difference FDEs: $\sum_{k=0}^{m} P_{k,n}(t)\, {}_{t}^{M}D_{a}^{n} y(t + \mu_{k,n}) = f(t)$ for $\lambda_{k,n} = 1$.

We have merged well-known fractional derivative types with our method. It is observed from the results that each fractional derivative type is in good agreement with the method. Even so, we would like to express the estimation of our method with the fractional Caputo, Riemann–Liouville, Caputo–Fabrizio, and Jumarie derivative types as follows:

**i)** As seen in the tables and figures, the fractional Caputo and Caputo–Fabrizio derivatives are more appropriate than Riemann–Liouville and Jumarie derivatives for our method.

**ii)** Since the fractional Caputo–Fabrizio derivative includes a compact exponential function, the method based on it can take a long processing time, as seen in Table 2.

**iii)** We know that the fractional Jumarie derivative is a modified version of the Riemann–Liouville derivative [32]. This situation is also observed in Figures 3c and 3d.

**iv)** In order to use the fractional Jumarie and Riemann–Liouville derivatives in the method, we are required to take the initial point as $c \neq 0$.

It is worth specifying that the above differences are due to including different kernels of the fractional derivative types. That is, the Caputo derivative includes a singular kernel and a differentiated function (see Definition 1.3), while the Caputo–Fabrizio derivative includes a smooth kernel. The Riemann–Liouville derivative is relatively equivalent to the Caputo type except for derivative of a constant. The Jumarie derivative type is the fixed formulation of the Riemann–Liouville type, but the Jumarie type holds for $n \in (0,1)$.

As a result, we demonstrate here that the fractional Caputo derivative is the best selection to employ in our method. Therefore, the practicability and efficiency of the method depend on determining the fractional derivative type.

The processing time of the method is very reasonable in comparison with other methods, as seen from Tables 2 and 5. It is clearly noticed from Figures 2–6 that the matching polynomial solutions coincide well with the exact solutions as $n$ and $N$ are increased. The durable structure of the complete graph has given rise to the use of high values of $N$. New residual error analysis based on the fractional derivative has improved the obtained solutions, as seen in Tables 1, 3, 5, and 6.

We show that the present method based on a different graph structure can be used for solving delay fractional integro-differential and (fractional) partial differential equations. All of them are planned as a future work.

**Acknowledgment**

The authors would like to thank the reviewer for his/her constructive and valuable comments to improve the paper.

## References


[1] Abd-Elhameed WM, Youssri YH. Generalized Lucas polynomial sequence approach for fractional differential equations. Nonlinear Dyn 2017; 89: 1341–1355.

[2] Abd-Elhameed WM, Youssri YH. Fifth-kind orthonormal Chebyshev polynomial solutions for fractional differential equations. Comput Appl Math 2018; 37: 2897–2921.

[3] Abdulaziz O, Hashim I, Momani S. Application of homotopy-perturbation method to fractional IVPs. J Comput Appl Math 2008; 216: 574-584.

[4] Abramowitz M, Stegun IA. Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables. Gaithersburg, MD, USA: National Bureau of Standards, 1964.

[5] Aihara J. Matrix representation of an olefinic reference structure for monocyclic conjugated compounds. Bull Chem Soc Japan 1979; 52: 1529-1530.

[6] Araujo O, Estrada M, Morales DA, Rada J. The higher-order matching polynomial of a graph. Int J Math Math Sci 2005; 10: 1565-1576.

[7] Aslan E. A measure of graphs vulnerability: Edge scattering number. Bull Int Math Virtual Inst 2014; 4: 53-60.

[8] Bacak-Turan G, Kırlangıç A. Neighbor rupture degree and the relations between other parameters. Ars Combin 2011; 102: 333-352.

[9] Bagley RL, Torvik PJ. On the appearance of the fractional derivative in the behaviour of real materials. J Appl Mech 1984; 51: 294-298.

[10] Baleanu D, Mousalou A, Rezapour S. A new method for investigating approximate solutions of some fractional integro-differential equations involving the Caputo-Fabrizio derivative. Adv Differ Equ 2017; 2017: 51.

[11] Bhrawy AH, Al-Zahrani AA, Alhamed YA, Baleanu D. A new generalized Laguerre-Gauss collocation scheme for numerical solution of generalized fractional pantograph equations. Rom J Phys 2014; 59: 646-657.

[12] Bian H, Zhang F, Wang G, Yu H. Matching polynomials for chains of cycles. Discrete Math 20111; 311: 319-323.

[13] Bode HW. Network Analysis and Feedback Amplifier Design. Shanghai, China: Tung Hwa Book Company, 1949.

[14] Caputo M. Elasticit'a e Dissipazione. Bologna, Italy: Zanichelli, 1969.

[15] Caputo M, Fabrizio M. A new definition of fractional derivative without singular kernel. Prog Fract Differ Appl 2015; 1: 1-13.

[16] Cermá J, Došláb Z, Kisela T. Fractional differential equations with a constant delay: stability and asymptotics of solutions. Appl Math Comput 2017; 298: 336-350.

[17] Diethelm K. The Analysis of Fractional Differential Equations. Berlin, Germany: Springer-Verlag, 2010.

[18] Doha EH, Bhrawy AH, Ezz-Eldien SS. Efficient Chebyshev spectral methods for solving multi-term fractional orders differential equations. Appl Math Model 2011; 35: 5662-5672.

[19] Farrell EJ. An introduction to matching polynomials. J Combin Theory Series B 1979; 27: 75-86.

[20] Ghosh U, Sengupta S, Sarkar S, Das S. Characterization of non-differentiable points in a function by fractional derivative of Jumarrie type. Eur J Acad Essays 2015; 2: 70.

[21] Godsil CD, Gutman I. On the theory of the matching polynomial. J Graph Theor 1981; 5: 137-145.

[22] Gross JL, Yellen J, Zhang PF. Handbook of Graph Theory. New York, NY, USA: CRC Press, 2013.

[23] Gülsu M, Öztürk Y, Anapalı A. Numerical approach for solving fractional relaxation–oscillation equation. Appl Math Model 2013; 37: 5927-5937.

[24] Gülsu M, Öztürk Y, Anapalı A. Numerical solution the fractional Bagley–Torvik equation arising in fluid mechanics. Int J Comput Math 2017; 94: 173-184.

[25] Gunther G, Hartnell BL. On minimizing the effects of betrayals in a resistance movement. In: Proceedings of the Eighth Manitoba Conference on Numerical Mathematics and Computing; 1978. pp. 285-306.

[26] Gutman I. The acyclic polynomial of a graph. Publ Inst Math Beograd 1977; 22: 63-69.

[27] Gutman I. The matching polynomial. Match Commun Math Co 1979; 6: 75-91.

[28] Harary F. Graph Theory. New York, NY, USA: Addison-Wesley, 1994.

[29] Heilmann OJ, Lieb EH. Theory of monomer-dimer systems. Commun Math Phys 1972; 25: 190-232.

[30] Heris MS, Javidi M. On fractional backward differential formulas for fractional delay differential equations with periodic and anti-periodic conditions. Appl Numer Math 2017; 118: 203-220.

[31] Hosoya H. On some counting polynomials in chemistry. Discrete Appl Math 1988; 19: 239-257.

[32] Jumarie G. Modified Riemann-Liouville derivative and fractional Taylor series of non-differentiable functions further results. Comput Math Appl 2006; 51: 1367-1376.

[33] Kilbas AA, Srivastava HM, Trujillo JJ. Theory and Applications of Fractional Differential Equations. San Diego, CA, USA: Elsevier, 2006.

[34] Kürkçü ÖK, Aslan E, Sezer M. A numerical approach with error estimation to solve general integro-differential–difference equations using Dickson polynomials. Appl Math Comput 2016; 276: 324-339.

[35] Kürkçü ÖK, Aslan E, Sezer M. A novel collocation method based on residual error analysis for solving integro-differential equations using hybrid Dickson and Taylor polynomials. Sains Malays 2017; 46: 335-347.

[36] Kürkçü ÖK, Aslan E, Sezer M. A numerical method for solving some model problems arising in science and convergence analysis based on residual function. Appl Numer Math 2017; 121: 134-148.

[37] Kürkçü ÖK, Aslan E, Sezer M, İlhan Ö. A numerical approach technique for solving generalized delay integro-differential equations with functional bounds by means of Dickson polynomials. Int J Comput Meth 2018; 15: 1850039.

[38] Losada J, Nieto JJ. Properties of a new fractional derivative without singular kernel. Prog Fract Differ Appl 2015; 1: 87-92.

[39] Mainardi F. Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models. London, UK: Imperial College Press, 2010.

[40] Pezza L, Pitolli F. A multiscale collocation method for fractional differential problems. Math Comput Simul 2018; 147: 210-219.

[41] Podlubny I. Fractional Differential Equations. New York, NY, USA: Academic Press, 1999.

[42] Rahimkhani P, Ordokhani Y, Babolian E. Numerical solution of fractional pantograph differential equations by using generalized fractional-order Bernoulli wavelet. J Comput Appl Math 2017; 309: 493-510.

[43] Rand RH, Sah SM, Suchorsky MK. Fractional Mathieu equation. Commun Nonlinear Sci Numer Simulat 2010; 15: 3254-3262

[44] Rehman M, Idrees A, Saeed U. A quadrature method for numerical solutions of fractional differential equations. Appl Math Comput 2017; 307: 38-49.

[45] Rida S, Arafa A, Abedl-Rady A, Abdl-Rahaim H, Fractional physical differential equations via natural transform. Chinese J Phys 2017; 55: 1569-1575.

[46] Sinha SC, Wu DH. An efficient computational scheme for the analysis of periodic systems. J Sound Vib 1991; 151: 91-117.

[47] Ünal E, Gökdoğan A. Solution of conformable fractional ordinary differential equations via differential transform method. Optik 2017; 128: 264-273.

[48] Weisstein EW. Matching Polynomial. MathWorld: A Wolfram Web Resource. Available at http://mathworld.wolfram.com/MatchingPolynomial.html.

[49] Yan W, Yeh YN. On the matching polynomial of subdivision graphs. Discrete Appl Math 2009; 157: 195-200.

[50] Youssri YH. A new operational matrix of Caputo fractional derivatives of Fermat polynomials: an application for solving the Bagley-Torvik equation. Adv Differ Equ 2017; 2017: 73.

[51] Yüzbaşı Ş. Numerical solutions of fractional Riccati type differential equations by means of the Bernstein polynomials. Appl Math Comput 2013; 219: 6328-6343.