A COLLOCATION METHOD FOR SOLVING FRACTIONAL ORDER LINEAR SYSTEM

M. Mashoof\(^1\), A.H. Refahi Sheikhani\(^2\*\), and H. Saberi Najafi\(^3\)

\(^1\)Department of Applied Mathematics, Faculty of Mathematical Sciences, Lahijan Branch, Islamic Azad University, Lahijan, Iran.
mashoof.mohammad@yahoo.com

\(^2\)Department of Applied Mathematics, Faculty of Mathematical Sciences, Lahijan Branch, Islamic Azad University, Lahijan, Iran.
*Corresponding author: ah_refahi@liau.ac.ir,

\(^3\)Department of Applied Mathematics, Faculty of Mathematical Sciences, Lahijan Branch, Islamic Azad University, Lahijan, Iran.
hnajafi@guilan.ac.ir

Abstract. In the current paper we propose a collocation method to achieve an algorithm for numerically solving of fractional order linear systems where a fractional derivative is defined in the Caputo form. We have used the Taylor collocation method for solving fractional differential equations; a collocation method which is based on taking the truncated Taylor expansions of the vector function’s solution in the fractional order linear system and substituting their matrix forms into the system. Through using collocation points we have obtained a system of linear algebraic equation. The method has been tested by some numerical examples.

Key words and Phrases: Fractional system, collocation method, numerical simulation.

Abstrak. In the current paper we propose a collocation method to achieve an algorithm for numerically solving of fractional order linear systems where a fractional derivative is defined in the Caputo form. We have used the Taylor collocation method for solving fractional differential equations; a collocation method which is based on taking the truncated Taylor expansions of the vector function’s solution in the fractional order linear system and substituting their matrix forms into the system. Through using collocation points we have obtained a system of linear algebraic equation. The method has been tested by some numerical examples.

Kata kunci: Sistem fraksional, metode kolokasi, simulasi numerik.
1. Introduction

Fractional differential equations have been generalized from integer order derivatives through replacing the integer order derivatives by fractional ones. The field of fractional differential equations has received attention and interest only in the past 20 years or so [2, 4, 3]. In recent years, studies concerning the application of the fractional differential equations in science has attracted more interest among scholars [9, 5]; readers can refer to [7, 8] for the theory and applications of fractional calculus in this regard. For instance, [10, 15] formulated the motion of a rigid plate immersing in a Newtonian fluid. They show that the use of fractional derivatives for the mathematical modelling of viscoelastic materials is quite natural [19]. It should be mentioned that the main reason for the theoretical development here is the wide use of polymers in various fields of engineering [19]. Moreover, in 1991, S. Westerlund in a paper on electrochemically polarizable media proposed the use of fractional derivatives for the description of propagation of plane electromagnetic waves in an isotropic and homogeneous lossy dielectric [19]. Caputo suggested the fractional order version of the relationship between the electric field and electric flux density [19].

Recently, fractional derivatives have been used to new applications in neural networks and control system [5, 6, 20]. Several methods such as Haar-wavelet operational matrix method have exited to solve the fractional linear system [16]. In [16] the authors introduced Haar-wavelet operational matrix method for fractional control system and translated the control system with initial condition into a Sylvester equation. A typical n-term linear non-homogeneous fractional order differential equation in time domain can be described as the following form

$$a_0(D_t^{\alpha_n}y(t)) + \cdots + a_1(D_t^{\alpha_1}y(t)) + a_0(D_t^{\alpha_0}y(t)) = u(t).$$  \hspace{1cm} (1)

A fractional order system described by n-term fractional differential (1) can be rewritten to the state-space representation in the form [12, 24]:

$$D_t^\beta x(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t).$$  \hspace{1cm} (2)

For this reason, the behavior of output in system (2) is useful. Recently, the collocation method has become a very useful technique for solving differential equations [18, 13, 1]. In the current study, we present a numerical solution of the fractional order system through the use of Taylor collocation method for a system of the form

$$D_t^\alpha x(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t),$$  \hspace{1cm} (3)

with $0 < \alpha \leq 1$, where $A$, $B$, $C$ and $D$ are $n \times n$, $n \times m$, $p \times n$ and $p \times m$ matrices, and $u(t)$ is an $m$-vector function, and $D_t^\alpha x(t)$ is the $\alpha$th-order (always fractional) derivative of $x(t)$ in the Caputo form. By using this method, we can translate (3) into an algebraic linear equation that can be solved through the use of existing methods [21, 22].
This paper is organized as follows. We review some basic definitions of fractional derivative operators. The function approximations and two applicable algorithms are presented in section 3. Section 4 contains four examples, and finally, conclusions are presented in section 5.

2. Basic Definition and Theorem

Here and in this part of the study, we deal with fractional calculus, definitions and theorems; see [23, 14, 17]. for more details in this regard.

Definition 2.1. A real function \( f(x) \), \( x \geq 0 \) is said to be in space \( C_{\mu} \), \( \mu \in \mathbb{R} \) if there exists a real number \( p(>\mu) \), such that
\[
f(x) = x^p f_1(x) \quad \text{where} \quad f_1(x) \in [0, \infty),
\]
and it is said to be in the space \( C_m^{\mu} \) iff \( f \in C_{\mu} \), \( m \in \mathbb{N} \).

Definition 2.2. The Riemann-Liouville fractional derivative of order \( \alpha \) with respect to the variable \( x \) and with the starting point at \( x = a \) is
\[
aD_t^\alpha f(x) = \begin{cases}
\frac{1}{\Gamma(\alpha + m + 1)} \times \frac{d^{m+1}}{dx^{m+1}} \times \int_a^x (x - \tau)^{m-\alpha} f(\tau)d\tau ; & 0 \leq m \leq \alpha < m + 1, \\
\frac{d^{m+1}}{dx^{m+1}} ; & \alpha = m + 1 \in \mathbb{N}.
\end{cases}
\]

Definition 2.3. The Riemann-Liouville fractional integral of order \( \alpha \) is
\[
aD_t^{-\alpha} f(x) = \frac{1}{\Gamma(\alpha)} \int_a^x (x - \tau)^{\alpha-1} f(\tau)d\tau ; \alpha > 0.
\]

Definition 2.4. The fractional derivative of \( f(x) \) in the Caputo form is defined as
\[
D_t^\alpha f(x) = \frac{1}{\Gamma(n - \alpha)} \int_0^x (x - \tau)^{n-\alpha-1} f^{(n)}(\tau)d\tau; n-1 < \alpha \leq n, n \in \mathbb{N}, x > 0, f \in C_{-1}^n.
\]

For the Caputo’s derivative we have \( D_t^n C = 0 \), and \( C \) is a constant. Moreover, we have
\[
D_t^\alpha x^t = \begin{cases}
0 ; & n \in \mathbb{N}, n < \lfloor \alpha \rfloor, \\
\frac{\Gamma(n+1)}{\Gamma(n+1-\alpha)} x^n ; & n \in \mathbb{N}, n \leq \lfloor \alpha \rfloor.
\end{cases}
\]

Definition 2.5. (Fractional Derivative of a Vector) If \( X(x) = (X_1(x) \cdots X_n(x))^T \) is a vector function, we define
\[
D_t^\alpha X(x) = \begin{pmatrix}
D_t^\alpha X_1(x) \\
D_t^\alpha X_2(x) \\
\vdots \\
D_t^\alpha X_n(x)
\end{pmatrix}.
\]
Now consider the fractional differential equations as
\[ D_α^t x(t) = Ax(t) + q(t), \]  
where \( 0 < α < 1 \), \( A \) is an \( N \times N \) matrix, and \( q : [0,h] \rightarrow \mathbb{C}^N \). Two following theorems show the form of the general solution of (9) where \( E_α(t) \) is the Mittag-Leffler function.

**Definition 2.6.** The Mittag-Leffler function with parameter \( α \) is given by
\[ E_α(z) = \sum_{k=0}^{∞} \frac{z^k}{Γ(αk + 1)}, \quad ℜ(α) > 0, \quad z \in \mathbb{C}. \]

It is obvious that if \( α = 1 \), then \( E_α(z) = e^z \).

**Theorem 2.7.** If \( λ_1, \ldots, λ_N \), be the eigenvalues of \( A \) and \( u^{(1)}, \ldots, u^{(N)} \) will be the corresponding eigenvectors; then the general solution of the homogeneous differential equation \( D_α^t x(t) = Ax(t) \) will be
\[ x(t) = \sum_{l=1}^{N} c_l u^{(l)} E_α(λ_l t^α), \]  
with certain constants \( c_l \in \mathbb{C} \). The unique solution of this differential equation subject to the initial condition \( x(0) = x_0 \) is characterized by the linear system
\[ x_0 = (u^{(1)}, \ldots, u^{(N)})(c_1, \ldots, c_N)^T. \]

**Proof.** See [11].

For the inhomogeneous boundary value problem, we have the following theorem.

**Theorem 2.8.** The general solution of the boundary value problem (9) has the form \( x = x_{hom} + x_{inhom} \) where \( x_{hom} \) is the general solution of the associated homogeneous problem and \( x_{inhom} \) is a particular solution of the inhomogeneous problem.

**Proof.** See [11].

Following we recall the generalization of Taylor formula which forms the basis of our numerical method.

**Theorem 2.9.** (generalized Taylor formula) Suppose that \( D_0^{kα} f(x) \in C(a,b) \) for \( k = 0,1, \ldots, n + 1 \) where \( 0 < α \leq 1 \); then one has
\[ f(x) = \sum_{i=0}^{n} \frac{(x-a)^{iα}}{Γ(iα + 1)} D_x^{iα} f(a) + \frac{D_x^{n+1} f(ζ)}{Γ((n+1)α + 1)} (x-a)^{(n+1)α}, \]  
with \( a \leq ζ \leq x \), for all \( x \in (a,b) \), where \( D_x^{nα} = D_t^n D_x^n \cdots D_x^n \) (n times).

**Proof.** See [17].
We can use the generalized Taylor formula in the matrix form as follows

\[ f(x) = X(x)M_0v + \frac{D_x^{N+1} f(\zeta)}{\Gamma((N+1)\alpha+1)} (x-a)^{(N+1)\alpha}, \]  

where

\[ X(x) = \begin{pmatrix} 1 & (x-a)^{\alpha} & (x-a)^{2\alpha} & \cdots & (x-a)^{N\alpha} \end{pmatrix}_{1 \times (N+1)}, \]

and

\[ M_0 = \begin{pmatrix} \Gamma(1) & 0 & 0 & \cdots & 0 \\ 0 & \Gamma(\alpha+1) & 0 & \cdots & 0 \\ 0 & 0 & \Gamma(2\alpha+1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \Gamma((N\alpha+1)) \end{pmatrix}, \]

and

\[ v = \begin{pmatrix} D_x^{0\alpha} f(a) & D_x^{\alpha} f(a) & D_x^{2\alpha} f(a) & \cdots & D_x^{N\alpha} f(a) \end{pmatrix}; \]

hence we can approximate \( f(x) \) as follows

\[ f(x) \simeq X(x)M_0v. \]  

Moreover, we have

\[ D_x^{\alpha} f(x) \simeq D_x^{\alpha} (X(x)M_0v) = (D_x^{\alpha} X(x))M_0v, \]

and

\[ D_x^{\alpha} (X(x)) = \begin{pmatrix} D_x^{\alpha} 1 & D_x^{\alpha} (x-a)^{\alpha} & D_x^{\alpha} (x-a)^{2\alpha} & \cdots & D_x^{\alpha} (x-a)^{N\alpha} \end{pmatrix}_{1 \times (N+1)} \]

\[ = \begin{pmatrix} \Gamma(\alpha+1) & \cdots & \Gamma(N\alpha+1) \end{pmatrix}_{1 \times (N+1)} \]

\[ = \begin{pmatrix} 0 & \Gamma(\alpha+1) & \cdots & \Gamma((N-1)\alpha+1) \end{pmatrix}_{1 \times (N+1)} (t-a)^{N\alpha}; \]

thus

\[ D_x^{\alpha} (X(x)) = X(x)M_1, \]

where

\[ M_1 = \begin{pmatrix} 0 & \Gamma(\alpha+1) & 0 & \cdots & 0 \\ 0 & 0 & \Gamma(2\alpha+1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \Gamma((N\alpha+1)) \end{pmatrix}; \]

and

\[ D_x^{\alpha} f(x) \simeq X(x)M_1M_0v. \]
3. Numerical method

In this section, we present two algorithms for numerical solution of system (3) based on Taylor collocation method. For this, we use equations (14) and (19) for each array of vector function \( x(t) \), and it’s fractional derivative, namely as

\[
x_j(t) \simeq x_j^N = X(t)M_0v_j,
\]

and

\[
D_t^\alpha x_j(t) \simeq D_t^\alpha x_j^N(t) = X(t)M_1M_0v_j.
\]

If we use (22) and (23) to approximate \( x_j(t) \) and \( D_t^\alpha x_j(t) \) for \( j = 1, \ldots, n \); then we have a system of linear equations.

**Theorem 3.1.** If we approximate \( x_j(t) \) and \( D_t^\alpha x_j(t) \) for \( j = 1, \ldots, n \), by (22) and (23) respectively; then we have a system of linear equations as \( \tilde{X}\tilde{F}\tilde{v} = \tilde{b} \) where \( \tilde{v} = (\bar{v}, \bar{v}, \ldots, \bar{v})^T \).

**Proof.** By substituting equations (22) and (23) in equation (3) we have

\[
X(t)( \sum_{i=1}^{n} a_{ij}M_0v_i + (a_{jj}M_0 - M_1M_0)v_j) = -\sum_{i=1}^{m} b_{ji}u_i(t); \quad j = 1, \ldots, n,
\]

or

\[
\overline{X(t)}(A \ast T - M_1)M_0\overline{v} = \overline{\beta(t)},
\]

where

\[
\overline{X(t)} = \begin{pmatrix} X(t) & 0 & \cdots & 0 \\ 0 & X(t) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & X(t) \end{pmatrix},
\]

\[
T = \begin{pmatrix} I_{N+1} & 0 & \cdots & 0 \\ 0 & I_{N+1} & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & I_{N+1} \end{pmatrix},
\]

\[
\overline{M}_0 = \begin{pmatrix} M_0 & 0 & \cdots & 0 \\ 0 & M_0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & M_0 \end{pmatrix},
\]

\[
\overline{M}_1 = \begin{pmatrix} M_1 & 0 & \cdots & 0 \\ 0 & M_1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & M_1 \end{pmatrix},
\]
A Collocation Method

\[ \overline{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}, \]

\[ \overline{\beta(t)} = -Bu(t), \quad (26) \]

and

\[ A \ast \overline{T} = \begin{pmatrix} a_{11}I_{N+1} & \cdots & a_{1n}I_{N+1} \\ \vdots & \ddots & \vdots \\ a_{n1}I_{N+1} & \cdots & a_{nn}I_{N+1} \end{pmatrix}. \quad (27) \]

Dispersing equation (24) by the collocation points \( t_i, i = 0, 1, \ldots, N \), we can obtain

\[ \overline{X(t_i)Fv} = \overline{\beta(t_i)}, \]

where

\[ \overline{F} = (A \ast \overline{T} - \overline{M_1})\overline{M_0}. \quad (28) \]

Now we find the matrix representation of the initial condition \( x(a) \),

\[ \overline{X}(a)M_0v_j = \lambda_j \text{ for } j = 1, 2, \ldots, n, \quad (29) \]

or

\[ \left\{ \begin{array}{l} \overline{X}(a)M_0\overline{v} = \overline{\lambda} \\ \overline{\lambda} = (\lambda_1 \cdots \lambda_n)^T \end{array} \right. \quad (30) \]

To obtain the numerical solution of (3), by replacing \( \overline{X(t_N)Fv} = \overline{\beta(t_N)} \) with equation (29), we have the new algebraic equation,

\[ \left\{ \begin{array}{l} \overline{X(t_i)Fv} = \overline{\beta(t_i)} \text{ for } i = 1, \ldots, N - 1 \\ \overline{X}(a)M_0\overline{v} = \overline{\lambda} \end{array} \right. \quad (31) \]

or in blocked matrix form

\[ \overline{X} \overline{Fv} = \overline{b}, \quad (32) \]

where

\[ \overline{X} = \begin{pmatrix} \overline{X(t_0)} & 0 & \cdots & 0 & 0 \\ 0 & \overline{X(t_1)} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \overline{X(t_{N-1})} & 0 \\ 0 & 0 & \cdots & 0 & \overline{X(a)} \end{pmatrix}_{(N+1) \times (N+1)} \]
\[ \tilde{F} = \begin{pmatrix} F & 0 & \cdots & 0 & 0 \\ 0 & F & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & F & 0 \\ 0 & 0 & \cdots & 0 & M_0 \end{pmatrix}_{(N+1) \times (N+1)} \]

\[ \tilde{v} = \begin{pmatrix} v \\ \vdots \\ v \end{pmatrix}_{(N+1) \times 1}, \quad \tilde{b} = \begin{pmatrix} \beta(t_0) \\ \vdots \\ \beta(\beta(t_{N-1})) \end{pmatrix}_{(N+1) \times 1} \] (33)

From above discussions we found that \( v \) is \( 1 \times (N+1) \) and \( \tilde{v} \) is \( 1 \times n(N+1)^2 \); hence if in (31) we set

\[
\begin{cases}
\tilde{v} = \begin{pmatrix} y_1 & y_2 & \cdots & y_{n(N+1)} & \cdots & y_{n(N+1)^2-n} & \cdots & y_{n(N+1)^2} \end{pmatrix}^T, \\
y_i = y_{jn(N+1)+i} \text{ for } i = 1, 2, \ldots, n(N+1) \text{ and } j = 1, 2, \ldots, N;
\end{cases}
\] (34)

hence we obtain a system of \( n(N+1)^2 \) linear algebraic equations with \( n(N+1)^2 \) unknown Taylor coefficients for \( x_i(t), i = 1, 2, \ldots, n \).

From above discussions we find that

\[ v_i = \begin{pmatrix} y_{(i-1)(N+1)+1} & y_{(i-1)(N+1)+2} & \cdots & y_{(i-1)(N+1)+N} & y_{i(N+1)} \end{pmatrix}, \] (35)

which is the vector of Taylor coefficients for \( x_i(t), i = 1, 2, \ldots, n \).

The algorithm is

**Proposition 3.2.**

**Input** \( A, B, x(a), u(t), N, a, b \)

**Output** \( x(t) = (x_1(t), \cdots, x_n(t))^T \)

1. **Compute** \( M_0, M_1 \)
2. **Set** \( \overline{M_0} = \text{diag}(M_0, \cdots, M_0), \overline{M_1} = \text{diag}(M_1, \cdots, M_1), \overline{M} = \text{diag}(I_{N+1}, \cdots, I_{N+1}) \)
3. **Compute** \( \overline{F} = (A * \overline{I} - \overline{M}) \)
4. **for** \( i = 0, \cdots, N \) **do**
   
   Set \( t_i = a + \frac{i b}{N} \) compute \( X(t_i) \)
   
   Construct \( \overline{X}(t_i) = \text{diag}(X(t_i), \cdots, X(t_i)) \), \( \overline{\beta}(t_i) = -Bu(t_i) \)
   
   **end do**
5. **Set** \( \tilde{b} = (\beta(t_0), \cdots, \beta(t_{N-1}), x(a))^T, \overline{X} = \text{diag}(\overline{X}(t_0), \cdots, \overline{X}(t_{N-1}), \overline{X}(a)), \tilde{F} = \text{diag}(\overline{F}, \cdots, \overline{F}, \overline{M}) \)
6. **Solve** \( \overline{X} \tilde{F} \tilde{v} = \tilde{b} \) such that \( \tilde{v}_i = \tilde{v}_j \text{ for } i = 1, 2, \cdots, n(N+1) \text{ and } j = 1, 2, \cdots, N \)
7. **for** \( i = 1, 2, \cdots, n \) **do**
   
   \( x_i(t) \approx X(t)M_0(\tilde{v}_{i-1}(N+1), \cdots, \tilde{v}_{i(N+1)}) \)
   
   **end do.**
If \( a \leq t \leq b \) and \( (b-a) \) are large, we can divide \( I = [a,b] \) into \( I_1 = [a,t_1] \), \( I_2 = [t_1,t_2] \), \( \cdots \), \( I_N = [t_{N-1},b] \) and approximate \( x(t) \) on subinterval \( I_{k+1} \) and use the approximated solution at \( t_{k+1} \) to initial condition of next subinterval. The following Lemma shows that by this procedure error can be decreased.

**Lemma 3.3.** Let \( x(t) \) be bounded on \( [a,b] \). If we use \( I_1 = [a,t_1] \), \( I_2 = [t_1,t_2] \), \( \cdots \), \( I_N = [t_{N-1},b] \) and approximate \( x(t) \) on subinterval \( I_{k+1} \) and use approximated solution at \( t_{k+1} \) to initial condition of next subinterval, absolute error at time \( t = b \) decreases.

**Proof.** From (12), for interval \( I \), if \( |D^{N+1}_I x_i(\zeta)| < l_i \) then we have

\[
e = |x_i(b) - x_i^N(b)| = |D^{N+1}_I x_i(\zeta)| \frac{(b-a)^{(N+1)\alpha}}{\Gamma((N+1)\alpha+1)} < l_i \frac{(b-a)^{(N+1)\alpha}}{\Gamma((N+1)\alpha+1)}
\]

where \( h = \frac{b-a}{N} \). Moreover, for subinterval \( I_j \) we have

\[
e_j = |x_i(t_j) - x_i^N(t_j)| < l_i \frac{(t_j - t_{j-1})^{(N+1)\alpha}}{\Gamma((N+1)\alpha+1)} = L_i h^{(N+1)\alpha},
\]

thus

\[
e_1 + e_2 + \cdots + e_N < L_i Nh^{(N+1)\alpha} < L_i (Nh)^{(N+1)\alpha} = e.
\]

This shows that the following algorithm can be more efficient than algorithm 3.2 at \( t = b \). ■

**Proposition 3.4.** Input \( A, B, x(a), u(t), N, a, b \) 
Output \( x(b) \)

1. for \( i = 0, 1, \cdots, N-1 \) do  
   Set \( I_i = [t_i, t_i+1] \), \( a = t_i, b = t_{i+1} \) 
   Utilize algorithm 3.2 and approximate \( x(t_{i+1}) \) 
   Set \( x(a) = x(t_{i+1}) \)
2. Set \( x(b) = x(t_N) \)  
end do.

4. Numerical Examples

To illustrate the method and algorithms referred to in the previous section, we consider the following examples. In order to show the efficiency of the method for solving fractional order linear systems, we apply it to solve different types of
fractional linear systems whose exact solutions are known. We use $\| \cdot \|_2$ to compare the exact and numerical solutions.

**Example 4.1.** Let us consider the following system with the initial conditions

$$D^\alpha_t x(t) = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix} x(t) + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x(t); 0 \leq t \leq 1; x(0) = \begin{pmatrix} 1 \\ -3 \end{pmatrix},$$

(37)

the exact solution when $\alpha = 1$ is

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \frac{1}{8} e^{-2t} \begin{pmatrix} 22 - 11e^{2t} - 3e^{6t} \\ -22 + e^{2t} - 3e^{6t} \end{pmatrix}.$$ 

(38)

We approximately solve the above system for $N = 10$ and obtain the approximate solution for $\alpha = 1$ as

$$x_1(t) = 1 - 7\frac{t}{\Gamma(2)} + 4.9990\frac{t^2}{\Gamma(3)} - 45.9461\frac{t^3}{\Gamma(4)} - 53.7703\frac{t^4}{\Gamma(5)} - 4.3006\times10^2\frac{t^5}{\Gamma(6)} - 2.1043\times10^3\frac{t^6}{\Gamma(7)} + 3.3841\times10^3\frac{t^7}{\Gamma(8)} - 1.1878\times10^5\frac{t^8}{\Gamma(9)} + 5.1080\times10^5\frac{t^9}{\Gamma(10)} - 2.5584\times10^6\frac{t^{10}}{\Gamma(11)},$$

and

$$x_2(t) = -3 + 4\frac{t}{\Gamma(2)} - 17.0010\frac{t^2}{\Gamma(3)} - 1.9462\frac{t^3}{\Gamma(4)} - 141.7694\frac{t^4}{\Gamma(5)} - 254.0886\frac{t^5}{\Gamma(6)} - 2.4559\times10^3\frac{t^6}{\Gamma(7)} + 4.0813\times10^3\frac{t^7}{\Gamma(8)} - 1.2011\times10^5\frac{t^8}{\Gamma(9)} + 5.1302\times10^5\frac{t^9}{\Gamma(10)} - 2.5607\times10^6\frac{t^{10}}{\Gamma(11)}.$$ 

In figure 1 we can see the numerical and exact solutions.

**Example 4.2.** Let us consider following fractional equation

$$D^\alpha_t x(t) = x(t); x(0) = 1; 0 \leq t \leq 1;$$

(39)

the exact solution, when $\alpha = 1$, is

$$x(t) = e^t,$$

(40)

and for $\alpha = \frac{1}{2}$ is

$$x(t) = e^t(\text{erf}\sqrt{t} + 1).$$

(41)

We approximately solve fractional equation (39) for $\alpha = 1$ with $N = 3$ and obtained the approximate solution,

$$x(t) = 1.0000 + 1.0000\frac{t}{\Gamma(2)} + 0.9474\frac{t^2}{\Gamma(3)} + 1.4211\frac{t^3}{\Gamma(4)},$$

also for $\alpha = 0.5$ and $N = 9$ we have,

$$x(t) = 1.0000 + 1.0000\frac{t^{1.5}}{\Gamma(1.5)} + 0.9916\frac{t}{\Gamma(2)} + 1.1163\frac{t^2}{\Gamma(2.5)} +$$
Figure 1. Numerical and exact solution of example 4.1 for $\alpha = 1$ and $N = 10$.

Figure 2. Numerical and exact solution of example 4.2 for $\alpha = 1$ and $N = 3$.

\[
+0.1837 \frac{t^2}{\Gamma(3)} + 4.6131 \frac{t^{\frac{3}{2}}}{\Gamma(3.5)} - 9.6736 \frac{t^3}{\Gamma(4)} + \\
+21.8928 \frac{t^{\frac{5}{2}}}{\Gamma(4.5)} - 24.2722 \frac{t^4}{\Gamma(5)} + 16.2662 \frac{t^{\frac{7}{2}}}{\Gamma(5.5)}.
\]

In figure 2 we can see the numerical and exact solutions of example 4.2 for $\alpha = 1$ with $N = 3$; Moreover, figure 3 shows the results for $\alpha = 0.5$ with $N = 9$ in example 4.2.

Example 4.3. In this example, we consider a fractional system with three equations,

\[
D_0^\alpha x(t) = \begin{pmatrix} -1 & 0 & 0 \\ 2 & 1 & -9 \\ 3 & 6 & 1 \end{pmatrix} x(t); 0 \leq t \leq 1; 0 < \alpha \leq 1; x(0) = \begin{pmatrix} -3 \\ 5 \\ 0 \end{pmatrix}.
\]

(42)
The general solution of this system according to theorems 2.7 and 2.8 is given by
\[ x(t) = c_1 u_1 E_\alpha(\lambda_1 t^\alpha) + c_2 u_2 E_\alpha(\lambda_2 t^\alpha) + c_3 u_3 E_\alpha(\lambda_3 t^\alpha), \]
where \( c_1, c_2, c_3 \) are constants, \( \lambda_1, \lambda_2, \lambda_3 \) are eigenvalues, and \( u_1, u_2, u_3 \) are the corresponding eigenvectors of \( A \).

In the figures 4 and 5 the numerical and exact solutions of \( x(t) \) for \( \alpha = 0.975 \) are shown for \( N = 5 \) and \( N = 15 \), respectively. From figure 4 we can see that the numerical solution with \( N = 5 \) is not a mismatch with the exact solution, but from figure 5 we can see that the numerical solution is in very good agreement with the exact solution for \( N = 15 \). Moreover, the errors of numerical solutions for different \( N \) are shown in table 1. From table 1 we can see that the numerical solutions are more and more close to the exact solution when the value of \( N \) becomes large.

**Example 4.4.** Consider example 4.3 with \( t \in [0, 3], N = 16 \) and divide \([0, 3]\) to subintervals \([0,1],[1,2],[2,3]\). Now we solve it with algorithms 3.1 and 3.2, and
A Collocation Method

Figure 5. Numerical and exact solution of example 4.3 for $\alpha = 0.975$ and $N = 15$.

Table 1. Error of approximated solution in example 4.3

<table>
<thead>
<tr>
<th>$N$</th>
<th>error at $t = \frac{1}{4}$</th>
<th>error at $t = \frac{1}{2}$</th>
<th>error at $t = \frac{3}{4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.0966</td>
<td>1.6726</td>
<td>2.2574</td>
</tr>
<tr>
<td>8</td>
<td>0.0352</td>
<td>0.0366</td>
<td>0.0488</td>
</tr>
<tr>
<td>16</td>
<td>1.5671e-07</td>
<td>4.4075e-07</td>
<td>0.0016</td>
</tr>
<tr>
<td>32</td>
<td>4.1904e-12</td>
<td>3.8083e-07</td>
<td>0.0016</td>
</tr>
</tbody>
</table>

Table 2. Error of approximated solution in example 4.4 at $t = b$ by algorithm 3.2 and algorithm 3.4.

<table>
<thead>
<tr>
<th>subinterval</th>
<th>algorithm</th>
<th>[0, 1]</th>
<th>[1, 2]</th>
<th>[2, 3]</th>
<th>[0, 3]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.2</td>
<td>9.1339e-07</td>
<td>6.7420e-05</td>
<td>0.0014</td>
<td>690.6396</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td>9.1339e-07</td>
<td>6.7420e-05</td>
<td>0.0014</td>
<td>690.6396</td>
</tr>
</tbody>
</table>

compare the results. The numerical results will presented in table 2. From table 2 we see that if we use algorithm 3.2, the error at time $t = 3$ is 690.6396 which is unacceptable, but with dividing interval $[0, 3]$ into three intervals $[0, 1], [1, 2], [2, 3]$ and use algorithm 3.4, the error at time $t = 1, 2$ and 3 is 9.1359e-07, 6.7420e-05 and 0.0014, respectively. This example confirms Lemma 3.2, and so we see that algorithm 3.4 is useful.

In figure 6 we can see the numerical and exact solutions of example 4.4 with $N = 16$ by algorithm 3.2. Figures 7, 8, and 9 show the numerical and exact solutions of example 4.4 with $N = 16$ by algorithm 3.4.
5. Conclusions

The fractional order linear system plays an important role in physics, chemical mixing, chaos theory, and biological systems. In this paper we presented a collocation method to solve fractional order linear system with initial conditions. These methods approximate the numeric solution of fractional order linear system through taking the truncated Taylor expansions of the vector function’s solution in the fractional order linear system and then substituting their matrix forms into the system. This method transforms fractional linear system into an algebraic equation. Examples show that the Taylor collocation method has been successfully applied to find the approximate solutions of the fractional linear system. Numerical
results show that this method is extremely effective and practical for this sort of approximate solutions. This method will be applicable in large domains as well.

REFERENCES


