Numerical study on fractional differential-algebraic systems by means of Chebyshev Pseudo spectral method

Hoda F. Ahmed
Faculty of Science, Department of Mathematics, Minia University, Minia, Egypt

ABSTRACT
A numerical treatment to a system of Caputo fractional order differential-algebraic equations (SFDAEs) is presented throughout this article. The suggested method based upon the shifted Chebyshev pseudo-spectral method (SCPSM). The shifted Chebyshev polynomials (SCPs) are handled to reduce the SFDAEs into the solution of linear/nonlinear systems of algebraic equations. By using some tested applications, the effectiveness and the accuracy of the suggested approach are demonstrated graphically. Also numerical comparisons between the proposed technique with other numerical methods in the existing literature are held. The numerical results show that the proposed technique is computationally efficient, accurate and easy to implement.

1. Introduction
Nowadays, the fractional differential equations (FDEs) have gained increased appearances in varied problems in various fields of physics, chemistry, biology, applied science and engineering, this is due to their accuracy in various fields of physics, chemistry, biology, applied science and engineering, this is due to their accuracy in various fields. Consequently, the development of analytical and numerical algorithms for FDEs is an interested topic for many researchers [9–16]. Many physical phenomena’s are obviously designated by a system of differential-algebraic equations (SDAEs). These types of systems follow in the displaying of the mechanical systems subject to constraints, power systems, electrical networks, optimal control, chemical process and in other numerous applications [17]. SFDAEs have currently confirmed to be a suitable devise in the displaying of the numerous engineering and physical problems such as non-integer order optimal controller design, electrochemical processes, complex biochemical [18]. The approximate and numerical solutions of these types of systems have been a focus of several researchers especially the nonlinear systems, because most of these systems don’t have exact solutions. Numerical methods to solve SDAEs have been given such as, the numerical algorithms for computing the matrix Green’s operator [19], implicit Runge–Kutta method [20], Padé approximation method [21,22], homotopy perturbation method [23], Adomain decomposition method [24] and variation iteration method (VIM) [25].

While methods like, the homotopy analysis method (HAM) [26], the Adomian decomposition method (ADM) [27], the variational iteration method [27], fractional differential transform method [28], the generalized triangular function operational matrices [29], the Bezier curves method [30], and Gegenbauer tau method [31] have been successfully applied for SFDAEs.

In this paper, we apply the SCPSM to solve the following SFDAEs [31]:

\[
D_i^\alpha y_i(t) = f_i(t, y_1, y_2, \cdots y_n, y'_1, y'_2, \cdots, y'_n), \quad i = 1, 2, \cdots, l - 1, \quad t \geq 0, \quad 0 < \alpha_i \leq 1, \quad \text{and} \quad y_i = g_i(t, y_1, y_2, \cdots y_n), \quad i = l, l + 1, \cdots, n \tag{1}
\]

With the initial conditions:

\[
y_i(0) = a_i, \quad i = 1, 2, \cdots, n. \tag{2}
\]

where \(D_i^\alpha\) is the Caputo fractional derivative of order \(\alpha_i\). When \(\alpha_i = 1\), the SFDAEs is reduced to the classical SDAEs. The proposed technique is one of the known spectral methods, which have been widely used in many fields. These methods are characterized by their speed of convergence and the high level of accuracy.

For FDEs, the spectral collocation method (also called pseudo-spectral method) is more applicable and commonly applied to numerically solve different types of the fractional differential equations [15,32]. In collocation technique, expansion coefficients are determined by constructing the approximate solution to satisfy the differential equation at some applicably selected points from the domain identified as collocation points. Recently, various types of orthogonal
polynomials and collocation points are used in spectral collocation approximations [15,32].

CPs have many useful properties. These polynomials present, among others, very good properties in the approximation of functions. This encourages many researchers for using these polynomials for solving different types of differential equations and FDEs [32–35].

The proposed technique used the properties of Chebyshev polynomials (CPs) to reduce the SFDAEs into a system of algebraic equation which is greatly simplifying the problem. To the best of our knowledge, the numerical treatment of SFDAEs has not been established by using SCPSM yet.

To check the accuracy of the suggested method, five numerical applications including comparisons between our obtained results with those achieved by using other existing methods are presented. This article is prescribed as follows: The basic definition of the Caputo fractional derivative and the main properties of the CPs are summarized in Section 2. In Section 3, the necessary theorems of the upper bound of errors and the convergence analysis of the fractional derivatives of the SCPs are explained. Section 4 contains the procedures for the implementation of the suggested method to nonlinear FDEs. Some applications are discussed in Section 5. Finally, a brief conclusion finishes the paper in Section 6.

2. Preliminaries and notations

2.1. Caputo fractional derivatives

Definition 2.1: A real function \( f(t) \), \( t > 0 \), is assumed to be in the space \( C_{\mu,\mu} \in \mathbb{N} \), if there exists a real number \( p > \mu \) such that \( f(t) = t^p g(t) \) where \( g(t) \in C(0,\infty) \), and is assumed to be in space \( C^m_{\mu} \) if and only if \( f^{(m)} \in C_{\mu, m} \in \mathbb{N} \).

Definition 2.2: The fractional derivative \( \mathcal{D}^\alpha_c \) of \( f(t) \) in the Caputo sense is defined as

\[
\mathcal{D}^\alpha_c f(t) = \frac{1}{\Gamma(m - \alpha)} \int_0^t (t - s)^{m-\alpha-1} f^{(m)}(s) \, ds \quad (3)
\]

For \( m - 1 < \alpha \leq m, m \in \mathbb{N}, t > 0, f(t) \in C^m_{\mu} \)

Caputo derivatives have the following properties:

\begin{align}
\mathcal{D}_c^{\alpha} (\lambda f(t) + \mu g(t)) &= \lambda \mathcal{D}_c^{\alpha} f(t) + \mu \mathcal{D}_c^{\alpha} g(t), \\
\mathcal{D}_c^{\alpha} C &= 0, \quad C \text{ is a constant} \\
\mathcal{D}_c^{\alpha} t^n &= \begin{cases} 
0 & \text{for } n \in \mathbb{N}_0 \text{ and } n < \lfloor \alpha \rfloor \\
\Gamma(n + 1) & \text{for } n \in \mathbb{N}_0 \text{ and } n \geq \lfloor \alpha \rfloor 
\end{cases}
\end{align}

(5)

where \( \lfloor \alpha \rfloor \) is the ceiling function and \( \mathbb{N}_0 = \{0,1,2,\ldots\} \).

For \( \alpha \in \mathbb{N} \), the Caputo fractional derivative agrees with the integer order derivative. More details in [12,13].

2.2. Main properties of the CPs

The CPs, \( \mathcal{T}_n(z) \) of degree \( n \) are determined by the following recurrence relation [32–34]

\[
\mathcal{T}_n+1(z) = 2z\mathcal{T}_n(z) - \mathcal{T}_{n-1}(z), \quad \mathcal{T}_0(z) = 1, \\
\mathcal{T}_1(z) = z, \quad n = 1, 2, \ldots
\]

(6)

The analytic form of \( \mathcal{T}_n(z) \) is defined by

\[
\mathcal{T}_n(z) = n \sum_{i=0}^{\left\lfloor \frac{n}{2} \right\rfloor} (-1)^{n-j} \frac{(n-j)!}{j!((n-2j)!)} z^{n-2j},
\]

where \( \left\lfloor \frac{n}{2} \right\rfloor \) is the integer part of \( \frac{n}{2} \).

The CPs, \( \mathcal{T}_n(z) \) of degree \( n \) defined in the interval \([0,L]\), constructed by offering the change of variable \( z = (\frac{1}{2} t - 1) \) and defined by

\[
\mathcal{T}_n(t) = n \sum_{k=0}^{\left\lfloor \frac{n}{2} \right\rfloor} (-1)^{n-k} 2^{nk} (n+k-1)! \frac{t^k}{k!(n-k)!}, \quad n = 1, 2, \ldots
\]

(7)

The square integrable function \( f(t) \) in \([0,L]\), can be approximated using the first \((m+1)\) terms of the CPs as

\[
f_m(t) = \sum_{i=0}^{m} c_i \mathcal{T}_i^m(t)
\]

(8)

Where the coefficients \( c_i \) are given by

\[
c_0 = \frac{1}{\pi} \int_0^L \frac{f(t) \mathcal{T}_1^m(t)}{\sqrt{Lt-t^2}} \, dt, \\
c_i = \frac{2}{\pi} \int_0^L \frac{f(t) \mathcal{T}_i^m(t)}{\sqrt{Lt-t^2}} \, dt, \quad i = 1, 2, \ldots m
\]

(9)

3. The approximation of the fractional derivatives of the SCPs and its convergence analysis

The approximate formulation of the non-integer derivatives, the truncating error and the convergence analysis of the SCPs are considered in the following theorems.

Theorem 3.1: (Chebyshev truncation theorem) [32]

The error in approximating \( x(t) \) by the sum of its \( m \)-terms is bounded by the sum of the absolute values of all neglected coefficient. If

\[
x_m(t) = \sum_{k=0}^{m} c_k \mathcal{T}_k(t),
\]

then

\[
E_T(m) = |x(t) - x_m(t)| \leq \sum_{k=m+1}^{\infty} |c_k|, \quad t \in [-1,1]
\]

(11)
Theorem 3.2: Let \( f(t) \) be approximated by SCPs as in (8) and suppose that \( \alpha > 0 \), then

\[
D^\alpha_x (f_m(t)) = \sum_{j=0}^{m} \sum_{k=\alpha}^{j-\alpha} c_j \psi^{(\alpha)}_{i,k} t^{k-\alpha}
\]

(12)

Where \( \psi^{(\alpha)}_{i,k} \) is given by

\[
\psi^{(\alpha)}_{i,k} = (-1)^{i-k} \frac{2^{2k}(i+k+1)!\Gamma(k+1)}{k!(i-k)!(2k)!}\Gamma(k+1-\alpha)
\]

(13)

For Proof, see [32].

Theorem 3.3: [32,35]

The Caputo fractional derivative of order \( \alpha \) for the SCPs can be expressed in terms of the SCPs themselves as in the following form

\[
D^\alpha_x (T_j^i(t)) = \sum_{k=\alpha}^{j-\alpha} \theta_{i,k} T_j^i(t)
\]

(14)

where

\[
\theta_{i,j} = \frac{(-1)^{i-k}2(i+k-1)!\Gamma(k-\alpha+1)}{h_j \Gamma(k+1)\Gamma(k-\alpha+j+1)}
\]

(15)

Step 2: Use Eq. (12) to approximate the Caputo fractional derivatives, then the SFDAEs (17) is reduced to

\[
\begin{aligned}
&\sum_{j=0}^{m} c_{ij} \psi^{(\alpha)}_{i,k} t^{k-\alpha} = f_i(t, \sum_{j=0}^{m} c_{ij} T_j^i(t), \ldots, T_j^i(t)) \\
&\sum_{j=0}^{m} c_{ij} T_j^i(t, \sum_{j=0}^{m} c_{ij} T_j^i(t), \ldots, \sum_{j=0}^{m} c_{ij} T_j^i(t)) \\
&\text{for } i = 1, 2, \ldots, l-1, \quad 0 < \alpha_i \leq 1, \quad m = 0, 1, \ldots, m,
\end{aligned}
\]

(20)

Step 3: Approximate the initial conditions (18) by using SCPs:

\[
\sum_{j=0}^{m} c_{ij} T_j^i(0) = a_i, \quad i = 1, 2, \ldots, n.
\]

(21)

Step 4: Collocate Eq. (20) at the \( \sum_{j=0}^{m} n(1-\lfloor \alpha_i \rfloor) \) points. For a suitable collocation points, use the of the SCPs roots, \( T_{i+1-\lfloor \alpha_i \rfloor}(t) \).

Step 5: The obtained equations of the previous step 4 with Eq. (21) represent a system of linear/nonlinear algebraic equations which contain \( \sum_{j=0}^{m} n + m \) of unknowns; \( c_{ij}, i = 1, 2, \ldots, n, j = 0, 1, \ldots, m \).

Step 6: Solve the algebraic system by using the Newton iteration method to obtain the unknowns.

Step 7: The approximated solutions will be

\[
y_{i,m}(t) = c_{0}T_{0}^i(t) + c_{1}T_{1}^i(t) + \cdots + c_{m}T_{m}^i(t),
\]

\[
i = 1, 2, \ldots, n
\]

5. Numerical applications

In this section five numerical applications of SFDAEs are solved by the proposed technique; the applications include variable and constant coefficient linear nonlinear SFDAEs

Application 5.1: Consider the following variable coefficient linear SFDAE [27–29]

\[
\begin{aligned}
&\sum_{j=0}^{m} c_{ij} \psi^{(\alpha)}_{i,k} t^{k-\alpha} = f_i(t, \sum_{j=0}^{m} c_{ij} T_j^i(t), \ldots, T_j^i(t)) \\
&y(t) - \sin(t) = 0.
\end{aligned}
\]

(22)

With the initial conditions

\[
x(0) = 1, y(0) = 0.
\]

For the special case at \( \alpha = 1 \), system (22) has exact solution:

\[
x(t) = e^{-t} + tsin(t), y(t) = sin(t).
\]

(24)

By implementing the proposed technique using \( m_1 = \ldots \)
$$m_2 = 5, x(t) = \sum_{j=0}^{5} c_j T_j^*(t)$$ and $$y(t) = \sum_{j=0}^{5} b_j T_j^*(t)$$ to Eq. (22), we obtain

$$\begin{align*}
\sum_{n=1}^{5} \sum_{r=0}^{n-1} c_n \psi_{n,r}(t) - t \sum_{j=0}^{5} b_j T_j^*(t) + \sum_{j=0}^{5} c_j T_j^*(t) \\
-(1+t) \sum_{j=0}^{5} b_j T_j^*(t) = 0, \quad \sum_{j=0}^{5} b_j T_j^*(t) - \sin(t) = 0,
\end{align*}$$

(25)

And the approximated equations for the initial conditions will be

$$\begin{align*}
\sum_{j=0}^{5} c_j T_j^*(0) &= 1 \quad \text{and} \quad \sum_{j=0}^{5} b_j T_j^*(0) = 0. \quad (26)
\end{align*}$$

The equations obtained by collocating system (25) at the first five roots of the SCPs $$T_j^*(t)$$ with Eqs. (26) represent a system of linear algebraic equations which contains twelve equations for twelve unknowns; $$c_j$$ and $$b_j, j = 0, 1, \ldots, 5$$. These unknowns are obtained by using Newton iteration method.

For $$\alpha = 1$$. The estimated solutions are

$$x(t) = 1 + t (-0.9999 + t (1.498 + t (-0.154 + (-0.1527 + 0.018 t) t)))$$

$$y(t) = 2.34188 \times 10^{-17} + t(0.9999 + t(0.00005 + t(-0.1672 + (0.00090 + 0.0075t)t)))$$

Figure 1(a,b) shows the performance of the approximate solutions; $$x(t)$$ and $$y(t)$$ respectively, of the Application 5.1 at $$m = 5$$ for different values of $$\alpha$$ ($$\alpha = 0.75, \alpha = 0.85, \alpha = 0.95, \alpha = 1$$) with the exact solutions at $$\alpha = 1$$. It is easy to see that the obtained solutions are continuously dependent on the fractional derivative and the estimated solutions are in good agreement with the exact solution at $$\alpha = 1$$. The Absolute errors

### Table 1. Absolute error of $$x(t)$$ of Application 5.1 for different values of $$m$$.

| $$t$$ | $$m = 5$$ | $$m = 10$$ | $$m = 15$$ | $$m = 20$$ |
|------|----------|----------|----------|----------|
| 0.0  | 3.3307 \times 10^{-16} | 4.4409 \times 10^{-16} | 1.1102 \times 10^{-16} | 0 |
| 0.1  | 3.8569 \times 10^{-6} | 1.3878 \times 10^{-16} | 2.2205 \times 10^{-16} | 1.1102 \times 10^{-16} |
| 0.2  | 9.0051 \times 10^{-6} | 9.5480 \times 10^{-15} | 1.1102 \times 10^{-16} | 1.1102 \times 10^{-16} |
| 0.4  | 3.9128 \times 10^{-6} | 1.0769 \times 10^{-13} | 3.3307 \times 10^{-16} | 1.1102 \times 10^{-16} |
| 0.5  | 5.1621 \times 10^{-6} | 1.2346 \times 10^{-13} | 3.3307 \times 10^{-16} | 0 |
| 0.6  | 9.0323 \times 10^{-6} | 2.6978 \times 10^{-14} | 2.2204 \times 10^{-16} | 1.1102 \times 10^{-16} |
| 0.7  | 4.0753 \times 10^{-6} | 1.2734 \times 10^{-13} | 1.1102 \times 10^{-16} | 0 |
| 0.8  | 7.9293 \times 10^{-6} | 4.7074 \times 10^{-14} | 4.4409 \times 10^{-16} | 0 |
| 0.9  | 3.5209 \times 10^{-6} | 5.3291 \times 10^{-15} | 4.4409 \times 10^{-16} | 0 |
| 1.0  | 3.0737 \times 10^{-6} | 2.1094 \times 10^{-14} | 4.4409 \times 10^{-16} | 2.2204 \times 10^{-16} |
| CPU  | 0.0312   | 0.0624   | 0.140401 | 0.265202 |

### Figure 1. The behaviours of $$x(t)$$ and $$y(t)$$ of Application 5.1 for different values of $$\alpha$$ with the exact solutions at $$\alpha = 1$$. (a) $$x(t)$$, (b) $$y(t)$$.}

### Table 2. Approximate solution of $$x(t)$$ obtained by our proposed method with the results in [27–29].

| $$t$$ | VIM [27] | HAM [28] | TF [29] | SCPSM | VIM [27] | HAM [28] | TF [29] | SCPSM | Exact |
|------|-----------|-----------|---------|-------|-----------|-----------|---------|-------|-------|
| 0.0  | 0.8492996 | 0.8492995 | 0.8492996 | 0.8495402 | 0.9148207 | 0.9148208 | 0.9148208 | 0.9148208 | 0.9148208 |
| 0.2  | 0.8016669 | 0.8016696 | 0.8016674 | 0.8015908 | 0.8584646 | 0.8584646 | 0.8584646 | 0.8584646 | 0.8584646 |
| 0.4  | 0.7978998 | 0.7979000 | 0.7978979 | 0.7973782 | 0.8294742 | 0.8294743 | 0.8294743 | 0.8294743 | 0.8294743 |
| 0.5  | 0.8250871 | 0.8250873 | 0.8250855 | 0.8251602 | 0.8260873 | 0.8260874 | 0.8260874 | 0.8260874 | 0.8260874 |
| 0.6  | 0.8760143 | 0.8760146 | 0.8760129 | 0.8760693 | 0.8462434 | 0.8462434 | 0.8462434 | 0.8462434 | 0.8462434 |
| 0.8  | 0.9454581 | 0.9454582 | 0.9454568 | 0.9451350 | 0.8875971 | 0.8875971 | 0.8875971 | 0.8875971 | 0.8875971 |
| 1.0  | 1.0209756 | 1.0209755 | 1.0209744 | 1.0291829 | 0.9475376 | 0.9475377 | 0.9475377 | 0.9475377 | 0.9475377 |
| 1.2  | 1.1229594 | 1.1229592 | 1.1229582 | 1.1230337 | 1.0232138 | 1.0232138 | 1.0232138 | 1.0232138 | 1.0232138 |
| 1.4  | 1.2234365 | 1.2234363 | 1.2234354 | 1.2233991 | 1.1115638 | 1.1115639 | 1.1115639 | 1.1115639 | 1.1115639 |
| 1.6  | 1.3269759 | 1.3269757 | 1.3269747 | 1.3269025 | 1.2093504 | 1.2093505 | 1.2093498 | 1.2093504 | 1.2093504 |

$$\alpha = 0.75$$ and $$\alpha = 1$$.
between the exact solution and the approximate solution of Application 5.1 at different values of $m$ ($m = 5, 10, 15, 20$) with their CPU time are tabulated in Table 1. Executing time of this problem is measured using Mathematica 10 software on CPU Intel(R) Core(TM) i3, it’s apparent that the solution doesn’t require much CPU time. It is remarkable that the proposed technique is very effective even with using few terms of SCPS and the overall errors can be made smaller by adding more terms of SCPS.

In Table 2 the approximate numerical solutions for $x(t)$ for $\alpha = 0.75$ and $1$ are compared with the solutions given by VIM [27], HAM [28] and TF [29]. These numerical results demonstrate the harmony between our method and the other numerical methods used in the comparisons.

**Application 5.2:** Consider the following nonlinear SFDAE [27–29]

\[
\begin{align*}
D_α^2 x(t) &= 1 + x(t) + z(t)x(t), \\
D_α z(t) &= y(t) - x^2(t) - z(t) \\
y(t) &= x^2(t)
\end{align*}
\]  
(27)

With $0 < \alpha \leq 1$ and initial conditions

\[
x(0) = y(0) = z(0) = 1.
\]  
(28)

For the special case when $\alpha = 1$, the exact solution is $x(t) = e^t, y(t) = e^{2t}, z(t) = e^{-t}$.

By employing the proposed technique using $m_1 = m_2 = m_3 = 5$, $x(t) = \sum_{j=0}^{5} c_j T_j^{\alpha}(t)$, $y(t) = \sum_{j=0}^{5} b_j T_j(t)$ and $z(t) = \sum_{j=0}^{5} a_j T_j^{\alpha}(t)$

We obtain a system of nonlinear algebraic equations which contains eighteen equations for eighteen unknowns; $c_j, b_j$ and $a_j, i = 0, 1, \ldots, 5$. These unknowns are obtained by using Newton iteration method.

For $\alpha = 1$. The estimated solutions will be

\[
x(t) = 0.9999 + t(3.0001 + t(2.4967 + t(0.85017 + (0.0904 + 0.0737t)))
\]

![Figure 2](image-url) The behaviours of $x(t), y(t)$ and $z(t)$ of Application 5.2 for different values of $\alpha$ with the exact solutions at $\alpha = 1, m = 5$; (a) $x(t)$, (b) $y(t)$, (c) $z(t)$.

**Table 3.** (a) Absolute errors of $x(t)$ of Application 5.2 for different values of $m$.

| $m$  | $t$  | $x$    | $m = 5$          | $m = 10$         | $m = 15$         | $m = 20$         |
|------|------|--------|-----------------|-----------------|-----------------|-----------------|
| 0.0  | 0.0  | 1.11   | $10^{-16}$      | 0.0             | 1.11            | $10^{-16}$      |
| 0.1  | 1.047| $10^{-6}$ | 2.89            | $10^{-15}$      | 2.22            | $10^{-16}$      |
| 0.2  | 2.565| $10^{-6}$ | 1.11            | $10^{-15}$      | 0.0             | 2.22            | $10^{-16}$      |
| 0.3  | 1.299| $10^{-6}$ | 3.86            | $10^{-14}$      | 0.0             | 2.22            | $10^{-16}$      |
| 0.4  | 1.208| $10^{-6}$ | 3.44            | $10^{-14}$      | 0.0             | 2.22            | $10^{-16}$      |
| 0.5  | 2.393| $10^{-6}$ | 3.109           | $10^{-15}$      | 0.0             | 2.22            | $10^{-16}$      |
| 0.6  | 1.083| $10^{-6}$ | 4.707           | $10^{-14}$      | 0.0             | 2.22            | $10^{-16}$      |
| 0.7  | 1.576| $10^{-6}$ | 2.443           | $10^{-14}$      | 4.44            | $10^{-16}$      | 0.0             |
| 0.8  | 2.854| $10^{-6}$ | 7.105           | $10^{-14}$      | 0.0             | 2.22            | $10^{-16}$      |
| 0.9  | 9.906| $10^{-7}$ | 1.599           | $10^{-14}$      | 4.44            | $10^{-16}$      | 8.88            | $10^{-16}$      |
| 1.0  | 2.289| $10^{-7}$ | 1.243           | $10^{-14}$      | 4.44            | $10^{-16}$      | 0.0             |
|      | CPU  | 0.0468 | 0.156001        | 0.343202        | 0.665204        |

(b) Absolute errors of $y(t)$ of Application 5.2 for different values of $m$.

| $m$  | $t$  | $y$    | $m = 5$          | $m = 10$         | $m = 15$         | $m = 20$         |
|------|------|--------|-----------------|-----------------|-----------------|-----------------|
| 0.0  | 0.0  | 1.11   | $10^{-16}$      | 0.0             | 1.11            | $10^{-16}$      |
| 0.1  | 3.961| $10^{-5}$ | 2.31            | $10^{-11}$      | 1.55            | $10^{-15}$      |
| 0.2  | 1.208| $10^{-5}$ | 4.694           | $10^{-11}$      | 6.66            | $10^{-16}$      |
| 0.3  | 1.008| $10^{-4}$ | 4.060           | $10^{-10}$      | 0.0             | 2.22            | $10^{-16}$      |
| 0.4  | 1.401| $10^{-4}$ | 4.213           | $10^{-10}$      | 4.44            | $10^{-16}$      |
| 0.5  | 0.789| $10^{-5}$ | 1.247           | $10^{-10}$      | 4.44            | $10^{-16}$      |
| 0.6  | 2.118| $10^{-4}$ | 6.500           | $10^{-11}$      | 1.33            | $10^{-15}$      |
| 0.7  | 2.768| $10^{-4}$ | 1.016           | $10^{-10}$      | 2.66            | $10^{-15}$      |
| 0.8  | 0.146| $10^{-4}$ | 2.075           | $10^{-10}$      | 3.55            | $10^{-15}$      |
| 0.9  | 4.097| $10^{-4}$ | 2.374           | $10^{-10}$      | 8.88            | $10^{-16}$      |
|      | CPU  | 0.0468 | 0.156001        | 0.343202        | 0.665204        |

(c) Absolute errors of $z(t)$ of Application 5.2 for different values of $m$.

| $m$  | $t$  | $z$    | $m = 5$          | $m = 10$         | $m = 15$         | $m = 20$         |
|------|------|--------|-----------------|-----------------|-----------------|-----------------|
| 0.0  | 0.0  | 1.11   | $10^{-16}$      | 0.0             | 1.11            | $10^{-16}$      |
| 0.1  | 4.177| $10^{-7}$ | 1.665           | $10^{-15}$      | 1.11            | $10^{-16}$      |
| 0.2  | 9.684| $10^{-7}$ | 7.77            | $10^{-15}$      | 1.11            | $10^{-16}$      |
| 0.3  | 4.224| $10^{-7}$ | 1.266           | $10^{-14}$      | 0.0             | 1.11            | $10^{-16}$      |
| 0.4  | 5.275| $10^{-7}$ | 1.366           | $10^{-14}$      | 0.0             | 1.11            | $10^{-16}$      |
| 0.5  | 9.102| $10^{-7}$ | 1.665           | $10^{-15}$      | 1.11            | $10^{-16}$      |
| 0.6  | 3.772| $10^{-7}$ | 1.499           | $10^{-14}$      | 0.0             | 1.11            | $10^{-16}$      |
| 0.7  | 5.693| $10^{-7}$ | 1.160           | $10^{-14}$      | 1.66            | $10^{-16}$      |
| 0.8  | 9.534| $10^{-7}$ | 2.165           | $10^{-15}$      | 5.55            | $10^{-17}$      |
| 0.9  | 2.568| $10^{-7}$ | 3.053           | $10^{-15}$      | 5.55            | $10^{-17}$      |
|      | CPU  | 0.0468 | 0.156001        | 0.343202        | 0.665204        |

Absolute errors of $z(t)$ for Application 5.2.
The behavior of \( y(t) \), \( z(t) \) and \( w(t) \) of Application 5.3 for different values of \( \alpha \) with exact solutions \( \alpha = 1 \); (a) \( y(t) \), (b) \( z(t) \), (c) \( w(t) \).}

\[
y(t) = 1 + t(6.00725 + t(13.647 + t(19.38 + t(4.417 + 11.9554)))})
\]

\[
z(t) = 1 + t(-0.99999 + t(0.4997 + t(-0.1652 + (0.0385 - 0.0051)t)))
\]

The numerical solutions of Application 5.2 are shown graphically through Figure 2 and tabulated in Table 3 (a–c) and Table 4(a) and 4b). From Figure 2, it is easy to conclude that the obtained solutions are continuously depend on the fractional derivative and the estimated solutions are in good agreement with the exact solution at \( \alpha = 1 \). Table 3(a–c) show the effectiveness of the suggested technique even with using few terms of SCPs and the accuracy of the method is increased by adding more terms of the SCPs. Also the solutions do not need much CPU time to produce very accurate numerical solutions.

Table 4(a,b) demonstrate the accuracy of the proposed technique when compared with the numerical methods in [27–29].
\begin{align}
\text{Table 5. (a) The absolute error of } y(t) \text{ of Application 5.3 for different values of } m. \\
| t | m = 5 | m = 10 | m = 15 | m = 20 |
\hline
0. & 1.041 \times 10^{-17} & 1.332 \times 10^{-17} & 3.284 \times 10^{-17} & 1.400 \times 10^{-17} \\
0.1 & 1.050 \times 10^{-5} & 4.109 \times 10^{-10} & 1.746 \times 10^{-12} & 5.187 \times 10^{-16} \\
0.2 & 3.803 \times 10^{-5} & 2.770 \times 10^{-9} & 6.319 \times 10^{-12} & 4.025 \times 10^{-15} \\
0.4 & 4.376 \times 10^{-5} & 2.431 \times 10^{-8} & 1.306 \times 10^{-11} & 6.314 \times 10^{-14} \\
0.5 & 1.750 \times 10^{-5} & 3.329 \times 10^{-9} & 7.657 \times 10^{-12} & 6.883 \times 10^{-14} \\
0.6 & 2.615 \times 10^{-5} & 1.368 \times 10^{-8} & 1.426 \times 10^{-11} & 1.280 \times 10^{-14} \\
0.7 & 5.738 \times 10^{-5} & 3.409 \times 10^{-8} & 4.268 \times 10^{-12} & 8.267 \times 10^{-16} \\
0.8 & 4.610 \times 10^{-5} & 3.292 \times 10^{-8} & 8.064 \times 10^{-12} & 2.110 \times 10^{-15} \\
0.9 & 1.020 \times 10^{-4} & 3.692 \times 10^{-8} & 2.085 \times 10^{-11} & 1.477 \times 10^{-14} \\
1. & 9.876 \times 10^{-5} & 1.840 \times 10^{-8} & 1.604 \times 10^{-11} & 1.188 \times 10^{-14} \\
CPU & 0.0468 & 0.124801 & 0.280802 & 0.624004
\end{align}

\begin{align}
\text{Table 6. (a) The absolute error of } x(t) \text{ of Application 5.4 for different values of } m. \\
| t | m = 5 | m = 10 | m = 15 | m = 20 |
\hline
0. & 1.282 \times 10^{-16} & 1.127 \times 10^{-17} & 1.635 \times 10^{-16} & 1.132 \times 10^{-17} \\
0.1 & 1.197 \times 10^{-16} & 5.204 \times 10^{-17} & 1.769 \times 10^{-16} & 1.735 \times 10^{-18} \\
0.2 & 9.021 \times 10^{-17} & 6.940 \times 10^{-18} & 1.735 \times 10^{-17} & 1.388 \times 10^{-17} \\
0.4 & 8.327 \times 10^{-17} & 3.188 \times 10^{-18} & 1.388 \times 10^{-17} & 1.388 \times 10^{-17} \\
0.5 & 8.327 \times 10^{-17} & 0.0 & 1.110 \times 10^{-16} & 0.0 \\
0.6 & 1.110 \times 10^{-16} & 0.0 & 1.110 \times 10^{-16} & 0.0 \\
0.7 & 1.665 \times 10^{-16} & 0.0 & 1.110 \times 10^{-16} & 0.0 \\
0.8 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.9 & 1.110 \times 10^{-16} & 0.0 & 1.110 \times 10^{-16} & 0.0 \\
1. & 0.0 & 0.0 & 0.0 & 0.0 \\
CPU & 0.0312 & 0.124801 & 0.296402 & 0.608404
\end{align}

where \(0 < \alpha \leq 1\) and initial conditions:

\[y(0) = z(0) = w(0) = 0\]  \(\text{(30)}\)

At \(\alpha = 1\). The exact solution is:

\[y(t) = t \sin(t), z(t) = \tan(t), w(t) = t \cos(t)\]

By applying our proposed technique using \(m_1 = m_2 = m_3 = 5\), \(y(t) = \sum_{j=0}^{5} c_j t^j\), \(z(t) = \sum_{j=0}^{5} b_j t^j\) and \(w(t) = \sum_{j=0}^{5} a_j t^j\) to the fractional system (5-8), we obtain a system of linear algebraic equations with unknowns; \(c_j, b_j\) and \(a_j, j = 0, 1, \ldots, 5\). These unknowns are attained by using Newton iteration method. For \(\alpha = 1\), the estimated solutions will be

\[y(t) = -1.56 \times 10^{-17} + t(-0.0002 + t(1.004) + t(-0.013 + (-0.1584 + 0.0097t)))\]

\[z(t) = -5.898 \times 10^{-17} + t(1.007 + t(-0.17 + t(1.17 + t(-1.478 + 0.0257t))))\]

\[w(t) = -2.776 \times 10^{-17} + t(0.9998 + t(0.005 + t(-0.52(0.03146 + 0.0257t))))\]

The numerical solutions of Application 5.3 are shown graphically through Figure 3 and the absolute errors between our approximate solutions and the exact solutions for different values of \(m\) with their CPU time are given in Table 5-(a-c)

\textbf{Application 5.4:} Consider the following nonlinear SFDAEs \(\text{[26]}\)

\[D_\alpha^* x(t) = x^2(t) - y(t) + 2t\]

\[D_\alpha^* y(t) = 2z(t) - 2(t + 1), \quad 0 < \alpha \leq 1\]

\[D_\alpha^* z(t) = -y(t) + 3x^2(t) + t\]
Table 7. (a) Numerical comparison between the approximate solutions of $y_1(t)$ of Application 5.5 with SGOM [31] at $m = 3$ and $m = 5$.

**Approximate solution of $y_1(t)$ of Application 5.5**

| $t$ | $m = 3$ | $m = 5$ |
|-----|---------|---------|
|     | SGOM [31] | SCPSM |
| 0.0 | 1.4665 × 10^{-16} | 1.30 × 10^{-17} |
| 0.1 | 2.31 × 10^{-3} | 2.14 × 10^{-4} |
| 0.2 | 1.84 × 10^{-3} | 1.70 × 10^{-3} |
| 0.4 | 7.94 × 10^{-4} | 2.54 × 10^{-3} |
| 0.8 | 8.99 × 10^{-5} | 2.21 × 10^{-4} |
| 1.2 | 2.42 × 10^{-5} | 8.08 × 10^{-5} |
| 0.6 | 5.07 × 10^{-1} | 1.20 × 10^{-4} |
| 0.7 | 1.18 × 10^{-1} | 3.10 × 10^{-4} |
| 0.8 | 1.47 × 10^{-3} | 3.91 × 10^{-3} |
| 1.0 | 6.46 × 10^{-4} | 2.21 × 10^{-3} |
| CPU | 0.0156 | 0.0312 |

For $\alpha = 1$, the estimated solutions are

$$x(t) = -5.974 \times 10^{-17} + (2.884 \times 10^{-17} + t(-1.44 \times 10^{-15} + (1.835 \times 10^{-15} + (7.275 \times 10^{-16})t)))$$

$$y(t) = -5.55 \times 10^{-17} + t(-2.33 \times 10^{-15} + (1.2 \times 10^{-14} + (1.896 \times 10^{-14} + (1 - 3.4 \times 10^{-16})t)))$$

$$z(t) = 1.0 + t(0.99 + (4.619 \times 10^{-13} + t(1.99999 + (1.546 \times 10^{-12}t - 6.17 \times 10^{-13}t)))$$

The numerical solutions of Application 5.4 are shown graphically through Figure 4 and the absolute errors for different values of $m$ are tabulated in Table 6(a–c).

**Application 5.5:** Consider the following SFDAEs [31]

$$D_{a}^{0.5} f_1(t) + 2y_1(t) = \frac{\Gamma\left(\frac{7}{5}\right)}{\Gamma(3)} f_2(t) + y_3(t)$$

$$D_{a}^{0.5} f_2(t) + y_2(t) + y_3(t) = \frac{\Gamma\left(\frac{3}{5}\right)}{\Gamma(2)} f_3(t) + t^2 + \sin(t)$$

with the initial conditions

$$y_1(0) = y_2(0) = y_3(0) = 0$$

The exact solution of this problem is

$$y_1(t) = t^{\frac{5}{7}}, y_2(t) = t^2, y_3(t) = \sin(t)$$

By using our proposed technique using $m_1 = m_2 = m_3 = 5$, $y_1(t) = \sum_{j=0}^{5} c_j T_j(t), y_2(t) = \sum_{j=0}^{5} b_j j! T_j(t)$ and $y_3(t) = \sum_{j=0}^{5} a_j j! T_j(t)$ to the fractional system (33), we obtain a system of linear algebraic equations with unknowns; $c_j, b_j$ and $a_j, j = 0, 1, \ldots, 5$. These unknowns are obtained by solving the algebraic system. Then the estimated solutions are

$$y_1(t) = -4.13 \times 10^{-16} + t(-0.003 + t(-0.003 + t(0.25 + (1.199 + (-0.645 + 0.199999)t))))$$

$$y_2(t) = 1.28 \times 10^{-16} + t(0.0002 + t(0.0002 + t(0.099 + t(0.016 + (-0.023 + 0.011999999)t)) + t(0.99999 + t(0.0198 + t(-0.25 + (0.115 - 0.04588)t))))$$

The numerical results of Application 5.5 are graphically illustrated in Figure 5. A numerical comparison between our attained solutions with the results in [31] is tabulated in Table 7(a–c). The numerical results demonstrate the effectiveness and the accuracy of the proposed technique even by using a few terms of SCPs, and our obtained results are quite similar to the results given by SGOM [31].
Figure 4. The behaviours of \(y(t), z(t)\) and \(w(t)\) Application 5.4 for different values of \(\alpha\) with exact solutions at \(\alpha = 1\); (a) \(x(t)\), (b) \(y(t)\), (c) \(z(t)\).

Figure 5. The behaviours of the approximate and exact solutions of Application 5.5. (a) \(y_1(t)\) (b) \(y_2(t)\) (c) \(y_3(t)\).

6. Conclusion

Through this paper, The SCPSM method has been extended to solve the linear and nonlinear SFDAEs. The numerical results of some tested applications were a good evidence for the applicability and efficiency of the suggested method. A specific advantage of the suggested implementation is that it transfers the fractional differential equations into a system of algebraic equations which is easier to solve. Also, satisfactory results are obtained by using a few terms of the SCPs, and the efficiency of the anticipated method is increased by using more terms of SCPs. The obtained solutions are continuously depended on the fractional derivative and this note confirms the physical meaning of the behaviour of the solution for the proposed real problems. Also the solutions do not require much CPU time.

Acknowledgement

The author is very grateful to the referees for carefully reading the paper and for their comments and suggestions which have improved the paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Hoda F. Ahmed © http://orcid.org/0000-0001-8930-1296

References

[1] Butzer PL, Westphal U. An introduction to fractional calculus. Ch. 1. In: Hilfer R, editor. Applications of fractional calculus in physics. Singapore: World Scientific; 2000. p. 87–130.
[2] Debnath L. Fractional integral and fractional differential equations in fluid mechanics. Fract. Calc. Appl. Anal. 2003;6:119–155.
[3] Samko SG, Kilbas AA, Marichev OI. Fractional integrals and derivatives: Theory and applications. Amsterdam: Gordan and Breach. 1993.
[4] Ullah R, Ellahi R, Sait SM, et al. On the fractional-order model of HIV-1 infection of CD4\(^+\) T-cells under the influence of antiviral drug treatment. J Taibah Univ Sci. 2020;14(1):50–59.
[5] Coronel-Escamilla A, Torres F, Gómez-Aguilar JF, et al. On the trajectory tracking control for an SCARA robot manipulator in a fractional model driven by induction motors with PSO tuning. Multibody Syst Dyn. 2018;43(3):257–277.
[6] Gómez-Aguilar JF. Chaos in a nonlinear Bloch system with Atangana–Baleanu fractional derivatives. Numer Methods Partial Differ Equ. 2018;34(5):1716–1738.
[7] Pérez JE, Gómez-Aguilar JF, Baleanu D, et al. Chaotic attractors with fractional conformable derivatives in the Liouville–Caputo sense and its dynamical behaviors. Entropy. 2018;20(5):384.
[8] Morales-Delgado VF, Gómez-Aguilar JF, Escobar-Jiménez RF, et al. Fractional conformable derivatives of Liouville–Caputo type with low-fractionality. Physica A. 2018;503:424–438.
[9] Daftardar-Gejji V, Babakhani A. Analysis of a system of fractional differential equations. J Math Anal Appl. 2004;293(2):511–522.
[10] Ahmed HF. A numerical technique for solving multi-dimensional fractional optimal control problems. J Taibah Univ Sci. 2018;12(5):494–505.

[11] Atangana A, Gómez-Aguilar JF. Numerical approximation of Riemann-Liouville definition of fractional derivative: from Riemann-Liouville to Atangana-Baleanu. Numer Methods Partial Differ Equ. 2018;34(5):1502–1523.

[12] Miller KS, Ross B. An introduction to the fractional calculus and fractional differential equations. New York: Wiley; 1993.

[13] Podlubny I. Fractional differential equations. San Diego: Academic Press; 1999.

[14] Al-Sadi W, Zhenyou H, Alkhazzan A. Existence and stability of a positive solution for nonlinear hybrid fractional differential equations with singularity. J Taibah Univ Sci. 2019;13(1):951–960.

[15] Mallawi F, Alzaidy JF, Hafez RM. Application of a Legendre collocation method to the space–time variable fractional-order advection–dispersion equation. J Taibah Univ Sci. 2019;13(1):324–330.

[16] Belmor S, Ravichandran C, Jarad F. Nonlinear generalized fractional differential equations with generalized fractional integral conditions. J Taibah Univ Sci. 2020;14(1):114–123.

[17] Campbell S, Ilchmann A, Mehrmann V, et al., editors. Applications of differential-algebraic equations: examples and benchmarks. Cham: Springer International Publishing; 2019.

[18] Shiri S, Baleanu D. System of fractional differential algebraic equations with applications. Chaos Solitons Fractals. 2019;120:203–212.

[19] Thota S, Kumar SD. Symbolic algorithm for a system of differential-algebraic equations. Kyungpook Math J. 2016;56(4):1141–1160.

[20] Ascher UM, Petzold LR. Projected implicit Runge–Kutta methods for differential-algebraic equations. SIAM J Numer Anal. 1991;28(4):1097–1120.

[21] Cilnik E, Bayram M. The numerical solution of physical problems modeled as a systems of differential-algebraic equations (DAEs). J Franklin Inst. 2005;342(1):1–6.

[22] Guzel N, Bayram M. On the numerical solution of differential-algebraic equations with index-3. Appl Math Comput. 2006;175(2):1320–1331.

[23] Soltanian F, Dehghan M, Karbassi SM. Solution of the differential algebraic equations via homotopy perturbation method and their engineering applications. Int J Comput Math. 2010;87(9):1950–1974.

[24] Hosseini MM. Adomian decomposition method for solution of differential-algebraic equations. J Comput Appl Math. 2006;197(2):495–501.

[25] Soltanian F, Karbassi SM, Hosseini MM. Application of He’s variational iteration method for solution of differential-algebraic equations. Chaos Solitons Fractals. 2009;41(1):436–445.

[26] Zurigat M, Momani S, Alawneh A. Analytical approximate solutions of systems of fractional algebraic–differential equations by homotopy analysis method. Comput Math Appl. 2010;59(3):1227–1235.

[27] İbiş B, Bayram M. Numerical comparison of methods for solving fractional differential–algebraic equations (FDAs). Comput Math Appl. 2011;62(8):3270–3278.

[28] İbiş B, Bayram M, Aşgörün AG. Applications of fractional differential transform method to fractional differential-algebraic equations. Eur J Pure Appl Math. 2011;4(2):129–141.

[29] Damarla SK, Kundu M. Numerical solution to fractional order differential-algebraic equations using generalized triangular function operational matrices. JFCA. 2015;6(2):31–52.

[30] Ghomanjani F. A new approach for solving fractional differential-algebraic equations. J Taibah Univ Sci. 2017;11(6):1158–1164.

[31] Ahmed HF, Melad MB. New numerical approach for solving fractional differential-algebraic equations. JFCA. 2018;9(2):141–162.

[32] Khader MM, Sweilam NH. On the approximate solutions for system of fractional integro-differential equations using Chebyshev pseudo-spectral method. Appl Math Model. 2013;37(24):9819–9828.

[33] Snyder MA. Chebyshev methods in numerical approximation. Englewood Cliffs: Prentice-Hall; 1966.

[34] Öztürk Y. Numerical solution of systems of differential equations using operational matrix method with Chebyshev polynomials. J Taibah Univ Sci. 2018;12(2):155–162.

[35] Doha EH, Bhrawy AH, Ezz-Eldien SS. A Chebyshev spectral method based on operational matrix for initial and boundary value problems of fractional order. Comput Math Appl. 2011;62(5):2364–2373.