Advances in Continuous and Discrete Models

RESEARCH Open Access

A computational method based on the generalized Lucas polynomials for fractional optimal control problems

Sh. Karami[1], A. Fakharzadeh Jahromi[1] and M.H. Heydari[1*]

*Correspondence:
heydari@sutech.ac.ir
[1]Department of Mathematics, Shiraz University of Technology, Shiraz, Iran

Abstract

Nonorthogonal polynomials have many useful properties like being used as a basis for spectral methods, being generated in an easy way, having exponential rates of convergence, having fewer terms and reducing computational errors in comparison with some others, and producing most important basic polynomials. In this regard, this paper deals with a new indirect numerical method to solve fractional optimal control problems based on the generalized Lucas polynomials. Through the way, the left and right Caputo fractional derivatives operational matrices for these polynomials are derived. Based on the Pontryagin maximum principle, the necessary optimality conditions for this problem reduce into a two-point boundary value problem. The main and efficient characteristic behind the proposed method is to convert the problem under consideration into a system of algebraic equations which reduces many computational costs and CPU time. To demonstrate the efficiency, applicability, and simplicity of the proposed method, several examples are solved, and the obtained results are compared with those obtained with other methods.

Keywords: Generalized Lucas polynomials; Fractional optimal control problems; Spectral collocation method; Operational matrices; Caputo fractional derivative; Pontryagin's maximum principle

1 Introduction and background

Fractional optimal control problems (FOCPs) are indeed generalizations of classical optimal control problems (OCPs) in which either the dynamic constraints or the performance index or both include at least one fractional derivative term. In recent years, this kind of problems has received much attention, since many real-world phenomena can be demonstrated or modeled by fractional differential equations (FDEs) much better than by integer order ones. There is a growing body of literature recognizing the importance of FOCPs, like [1–4]. It should be also emphasized that obtaining exact analytical solutions for nonlinear FOCPs is difficult, and in most cases impossible. Therefore, there exists a critical need to introduce numerical methods to solve these problems.

In spite of the fact that a number of numerical methods have been extensively used for solving FOCPs, considerable attention is still directed to finding some alternative and new

methods. It should be remarked that numerical methods for solving FOCPs may be classified into two main categories: indirect and direct methods. The indirect methods are generally based upon the generalization of Pontryagin maximum principle (PMP) for FOCPs and usually need the numerical solution of two-point boundary value problem resulting from the related necessary optimality conditions. Nevertheless, the direct methods are based upon discretization-then-optimization of the main FOCPs. For indirect numerical methods, Agrawal in [5, 6], with dependence on Riemann–Liouville and Caputo operators, introduced a general formulation and solution method for FOCPs. Sweilam et al. in [7] studied two distinct numerical methods based on Chebyshev polynomials for solving FOCP in the sense of Caputo. Pooseh et al. in [8] achieved the necessary optimality conditions for FOCPs with free terminal time. Moreover, one can refer to Legendre spectral collocation method [9], Bessel collocation method [10], Jacobi spectral collocation method [11], fractional Chebyshev pseudospectral method [12], Legendre wavelet collocation method [13], variational iterative method [14], and predictor–corrector method [15]. For direct numerical methods, one can address Legendre orthogonal polynomials [16], Bernoulli polynomials [17], shifted Legendre orthogonal polynomials [18], wavelets methods [19, 20], Boubaker polynomials [21], shifted Chebyshev schemes [22], Hermite polynomials [23], Bernoulli wavelet basis [24], fractional-order Dickson functions [25], generalized shifted Legendre polynomials [26], and generalized Bernoulli polynomials [27].

The above literature review indicates that many researchers have widely used orthogonal basis functions to obtain approximate solutions of FOCPs, but little attention has been directed toward nonorthogonal polynomials such as Fibonacci and Lucas polynomials. The main two advantages of Lucas polynomials in comparison with shifted Legendre and shifted Chebyshev polynomials to approximate an arbitrary function defined in $[0, 1]$ are as follows:

- The Lucas polynomials have fewer terms than shifted Legendre and shifted Chebyshev polynomials; for example, the sixth Lucas polynomial has four terms, whereas the sixth shifted Legendre and shifted Chebyshev polynomials have seven terms, and this difference will increase by increasing the degree of polynomials. Therefore, Lucas polynomials take less CPU time as compared to shifted Legendre and shifted Chebyshev polynomials to approximate an arbitrary function.
- The coefficients of the individual terms in Lucas polynomials are smaller than the corresponding ones in shifted Legendre and shifted Chebyshev polynomials. Due to the fact that computational errors in the product are related to the coefficients of individual terms, using Lucas polynomials reduces computational errors.

Recently, by regarding the advantages of Lucas polynomials, attention to these polynomials in the literature has grown. The authors in [28, 29] established numerical algorithms based on Lucas polynomials and generalized Lucas polynomials (GLPs) to solve multiterm fractional differential equations. In [30] the GLPs are utilized to obtain numerical solution of fractional initial value problems. Oruç provided numerical solutions for nonlinear sinh-Gordon and generalized Benjamin–Bona–Mahony–Burgers equations based on a hybridization method of Fibonacci and Lucas polynomials [31, 32]. Dehestani et al. solved variable-order fractional reaction-diffusion and sub-diffusion equations using Lucas multiwavelet functions [33]. The authors in [34] applied Lucas wavelets for solving fractional Fredholm–Volterra integro-differential equations. In [35], a numer-

ical optimization method based on fractional Lucas functions is developed for evaluating the approximate solution of the multidimensional fractional differential equations. Kumar et al. used normalized Lucas wavelets to solve Lane–Emden and pantograph equations [36]. Ali et al. [37] numerically solved multidimensional Burgers-type equations using Lucas polynomials. Furthermore, the authors in [38] investigated the GLPs to solve certain types of fractional pantograph differential equations numerically.

Up to now, great attention has been paid to numerical solutions of fractional differential equations taking Lucas polynomials as basis functions. This gives us a strong motivation to test their ability to solve FOCPs and introduce an efficient numerical method. The principal aim of this research is to construct an indirect numerical method for solving FOCPs using the GLPs in which high accuracy of the obtained approximate solution is one of the remarkable features of the method. To this end, first, we establish the necessary optimality conditions for FOCPs and obtain the operational matrices. Then, we use the necessary optimality conditions, the spectral collocation method, and operational matrices based on the GLPs to reduce the given problem into a nonlinear (or linear) system of algebraic equations that can be simply solved through the Newton iterative technique. Numerical test examples are also given to illustrate the accuracy and simplicity of the proposed method.

This article is organized in the following way. Some preliminaries of fractional calculus are presented in Sect. 2. The problem formulation and also the necessary optimality conditions are introduced in Sect. 3. Section 4 is devoted to introducing the GLPs and some of their properties. In Sect. 5, the GLPs operational matrices of the integer and Caputo fractional derivatives are determined. The proposed scheme is described in Sect. 6 to solve the given FOCP, and numerical examples are considered to show the efficiency of the new approach in Sect. 7. Finally, the conclusions and remarks are given in Sect. 8.

2 Some basic preliminaries of fractional calculus

In this section, we remind some notations and definitions for Caputo fractional derivatives, Riemann–Liouville fractional derivatives, and integrals. These concepts are common in fractional differential equations references and are used frequently (see for instance [39, 40]).

Definition 2.1 Assume that $\mathcal{G} : [0, T] \to \mathbb{R}$ is a function and $\alpha > 0$ is the order of derivative or integral. For $\tau \in [0, T]$, we define

- The left-side and right-side Caputo fractional derivatives by

$$ {}_0^C D_\tau^\alpha \mathcal{G}(\tau) = \frac{1}{\Gamma(p-\alpha)} \int_0^\tau (\tau - s)^{p-\alpha-1} \mathcal{G}^{(p)}(s)\, ds \tag{2.1} $$

and

$$ {}_\tau^C D_T^\alpha \mathcal{G}(\tau) = \frac{(-1)^p}{\Gamma(p-\alpha)} \left(\int_\tau^T (s - \tau)^{p-\alpha-1} \mathcal{G}^{(p)}(s)\, ds \right), \tag{2.2} $$

respectively;

- The left-side and right-side Riemann–Liouville fractional derivatives by

$$ {}_0D_\tau^\alpha \mathcal{G}(\tau) = \frac{1}{\Gamma(p-\alpha)} \frac{d^p}{d\tau^p} \left(\int_0^\tau (\tau - s)^{p-\alpha-1} \mathcal{G}(s)\, ds \right) \tag{2.3} $$

and

$$ {}_\tau D_T^\alpha \mathcal{G}(\tau) = \frac{(-1)^p}{\Gamma(p-\alpha)} \frac{d^p}{d\tau^p} \left(\int_\tau^T (s - \tau)^{p-\alpha-1} \mathcal{G}(s)\, ds \right), \tag{2.4} $$

respectively;

- The left-side and right-side Riemann–Liouville fractional integrals by

$$ {}_0I_\tau^\alpha \mathcal{G}(\tau) = \frac{1}{\Gamma(\alpha)} \int_0^\tau (\tau - s)^{\alpha-1} \mathcal{G}(s)\, ds \tag{2.5} $$

and

$$ {}_\tau I_T^\alpha \mathcal{G}(\tau) = \frac{1}{\Gamma(\alpha)} \int_\tau^T (s - \tau)^{\alpha-1} \mathcal{G}(s)\, ds, \tag{2.6} $$

respectively;

where $\Gamma(.)$ denotes the gamma function and $p = [\alpha] + 1$ ($[\alpha]$ is the integer part of α).

The Caputo and Riemann–Liouville fractional derivatives are linked with each other as follows:

$$ {}_0^C D_\tau^\alpha \mathcal{G}(\tau) = {}_0D_\tau^\alpha \mathcal{G}(\tau) - \sum_{i=0}^{p-1} \frac{\mathcal{G}^{(i)}(0)}{\Gamma(i-\alpha+1)} \tau^{i-\alpha} \tag{2.7} $$

and

$$ {}_\tau^C D_T^\alpha \mathcal{G}(\tau) = {}_\tau D_T^\alpha \mathcal{G}(\tau) - \sum_{i=0}^{p-1} \frac{\mathcal{G}^{(i)}(T)}{\Gamma(i-\alpha+1)} (T - \tau)^{i-\alpha}. \tag{2.8} $$

As a consequence, if $\mathcal{G}$ and $\mathcal{G}^{(k)}$, $k = 1, 2, \ldots p-1$, vanish at $\tau = 0$, then

$$ {}_0D_\tau^\alpha \mathcal{G}(\tau) = {}_0^C D_\tau^\alpha \mathcal{G}(\tau), \tag{2.9} $$

and if they vanish at $\tau = T$, then

$$ {}_\tau D_T^\alpha \mathcal{G}(\tau) = {}_\tau^C D_T^\alpha \mathcal{G}(\tau). \tag{2.10} $$

Caputo fractional derivatives of the power functions are yielded in the following forms:

$$ {}_0^C D_\tau^\alpha \tau^\beta = \begin{cases} 0, & \beta \in \mathbb{N}_0 \text{ and } \beta < \lceil \alpha \rceil, \\ \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)} \tau^{\beta-\alpha}, & \beta \in \mathbb{N}_0 \text{ and } \beta \geq \lceil \alpha \rceil, \\ & \text{or } \beta \notin \mathbb{N} \text{ and } \beta > \lfloor \alpha \rfloor, \end{cases} \tag{2.11} $$

and

$$ {}_{\tau}^{C}D_{T}^{\alpha}(T-\tau)^{\beta} = \begin{cases} 0, & \beta \in \mathbb{N}_0 \text{ and } \beta < \lceil \alpha \rceil, \\ \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)}(T-\tau)^{\beta-\alpha}, & \beta \in \mathbb{N}_0 \text{ and } \beta \geq \lceil \alpha \rceil, \\ & \text{or } \beta \notin \mathbb{N} \text{ and } \beta > \lfloor \alpha \rfloor, \end{cases} \tag{2.12} $$

where $\lfloor \alpha \rfloor$ and $\lceil \alpha \rceil$ are the largest integer less than or equal to α and the smallest integer greater than or equal to α, respectively. Also $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$ and $\mathbb{N} = \{1, 2, 3, \ldots\}$.

Theorem 2.2 *Let $\alpha \in (0,1)$ and $f, g : [0,T] \to R$ be two functions of class C^1. Then the formula for fractional integration by parts is derived as follows* [41]:

$$ \int_0^T f(\tau)\,{}_0^{C}D_{\tau}^{\alpha} g(\tau)\,d\tau = \int_0^T g(\tau)\,{}_{\tau}D_{T}^{\alpha} f(\tau) + \left[{}_{\tau}I_{T}^{1-\alpha} f(\tau) g(\tau) \right]_0^T. \tag{2.13} $$

3 Necessary optimality conditions for FOCPs

In this study, we consider a class of FOCPs in the sense of Caputo as follows:

$$ \begin{aligned} &\text{Min } \mathfrak{J}(\mathfrak{W}) = \int_0^T \mathcal{F}\big(\tau, \mathfrak{V}(\tau), \mathfrak{W}(\tau)\big)\,d\tau, \\ &\text{subject to :} \quad {}_0^{C}\mathfrak{D}_{\tau}^{\alpha}\mathfrak{V}(\tau) = \mathcal{G}\big(\tau, \mathfrak{V}(\tau), \mathfrak{W}(\tau)\big), \\ &\mathfrak{V}(0) = \mathfrak{V}_0, \end{aligned} \tag{3.1} $$

where $0 < \alpha \leq 1$, $\mathfrak{V} \in R^n$, $\mathfrak{W} \in R^s$, $\mathcal{F} : R \times R^n \times R^s \to R$, and $\mathcal{G} : R \times R^n \times R^s \to R^n$. The scalar function $\mathcal{F}$ and the vector function $\mathcal{G}$ are generically nonlinear and supposed to be differentiable functions; also $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ are the state and the control variables, respectively. Obviously, when $\alpha = 1$, this problem is transformed to the standard OCPs.

In 2014, a general formulation of FOCPs in the sense of Caputo was presented by Pooseh et al. In order to obtain the necessary optimality conditions for problem (3.1), we follow the method of [8]. First, the Hamiltonian scalar function is defined as

$$ \begin{aligned} &\mathcal{H}\big(\tau, \mathfrak{V}(\tau), \mathfrak{W}(\tau), \lambda(\tau)\big) \\ &\quad = \mathcal{F}\big(\tau, \mathfrak{V}(\tau), \mathfrak{W}(\tau)\big) + \lambda^{T}(\tau)\mathcal{G}\big(\tau, \mathfrak{V}(\tau), \mathfrak{W}(\tau)\big), \end{aligned} \tag{3.2} $$

where $\lambda(\tau)$ is a Lagrange multiplier. Then the necessary optimality conditions for problem (3.1) are determined by the following theorem [8].

Theorem 3.1 *If $(\mathfrak{V}(\tau), \mathfrak{W}(\tau))$ is a minimizer of (3.1), then there exists a co-state vector $\lambda(\tau)$ for which the triple $(\mathfrak{V}(\tau), \mathfrak{W}(\tau), \lambda(\tau))$ satisfies the following relations*:

$$ \begin{aligned} &{}_0^{C}\mathfrak{D}_{\tau}^{\alpha}\mathfrak{V}(\tau) = \frac{\partial \mathcal{H}}{\partial \lambda}\big(\tau, \mathfrak{V}(\tau), \mathfrak{W}(\tau), \lambda(\tau)\big), \\ &{}_{\tau}\mathfrak{D}_{T}^{\alpha}\lambda(\tau) = \frac{\partial \mathcal{H}}{\partial \mathfrak{V}}\big(\tau, \mathfrak{V}(\tau), \mathfrak{U}(\tau), \lambda(\tau)\big), \\ &\frac{\partial \mathcal{H}}{\partial \mathfrak{W}}\big(\tau, \mathfrak{V}(\tau), \mathfrak{W}(\tau), \lambda(\tau)\big) = 0, \end{aligned} \tag{3.3} $$

$$\big[\lambda(\tau)\big]_{\tau=T} = 0$$

for all $\tau \in [0, T]$, *where* $\mathcal{H}$ *is described by* (3.2).

4 The generalized Lucas polynomials and their properties

Lucas polynomials $L_n(\tau)$ of degree n defined over $[0, 1]$, originally studied by Bicknell in 1970, can be generated through the following recurrence relation [42]:

$$\begin{aligned} &L_{j+2}(\tau) = \tau L_{j+1}(\tau) + L_j(\tau), \quad j \geq 0, \\ &L_0(\tau) = 2, \qquad L_1(\tau) = \tau. \end{aligned} \tag{4.1}$$

Also, the Binet form of the Lucas polynomials is given by [42]

$$L_j(\tau) = \frac{(\tau + \sqrt{\tau^2+4})^j + (\tau - \sqrt{\tau^2+4})^j}{2^j}, \quad j \geq 0.$$

Moreover, these polynomials can be represented by the following explicit form as well [29]:

$$L_j(\tau) = j \sum_{i=0}^{\lfloor \frac{j}{2} \rfloor} \frac{1}{j-i} \binom{j-i}{i} \tau^{j-2i}, j \geq 1. \tag{4.2}$$

It has been shown that these polynomials satisfy the following properties:

- $L_n(\tau) = \mathrm{F}_{n+1}(\tau) + \mathrm{F}_{n-1}(\tau)$,
- $\tau L_n(\tau) = \mathrm{F}_{n+2}(\tau) - \mathrm{F}_{n-2}(\tau)$,
- $L_{-n}(\tau) = (-1)^n L_n(\tau)$,
- $\frac{dL_n(\tau)}{d\tau} = \frac{n}{\tau^2+4}(\tau L_n(\tau) + 2L_{n-1}(\tau))$,
- $L_n(0) = 1 + (-1)^n$,
- $L_n(1) = \mathcal{L}_n$,

where $\mathrm{F}_n(\tau)$ is the Fibonacci polynomials of order n and $\mathcal{L}_n$ is the Lucas number [42].

Besides, if a and b are nonzero real constants, the sequence of generalized Lucas polynomials (GLPs) defined over $[0, 1]$ is given by the following recurrence relation [29]:

$$\begin{aligned} &\mu^{a,b}_{j+2}(\tau) = a\tau \mu^{a,b}_{j+1}(\tau) + b\mu^{a,b}_j(\tau), \quad j \geq 0, \\ &\mu^{a,b}_0(\tau) = 2, \qquad \mu^{a,b}_1(\tau) = a\tau. \end{aligned} \tag{4.3}$$

In this regard, the first few GLPs $\mu^{a,b}_j(\tau)$ can be computed as follows:

$$\begin{aligned} &\mu^{a,b}_0(\tau) = 2, \qquad \mu^{a,b}_1(\tau) = a\tau, \\ &\mu^{a,b}_2(\tau) = a^2\tau^2 + 2b, \\ &\mu^{a,b}_3(\tau) = a^3\tau^3 + 3ab\tau, \\ &\mu^{a,b}_4(\tau) = a^4\tau^4 + 4a^2b\tau^2 + 2b^2, \\ &\mu^{a,b}_5(\tau) = a^5\tau^5 + 5a^3b\tau^3 + 5ab^2\tau. \end{aligned}$$

It is also shown that the GLPs can be described with two equivalent forms [29]

$$\mu_j^{a,b}(\tau) = \begin{cases} 2, & j = 0, \\ j\sum_{n=0}^{\lfloor \frac{j}{2} \rfloor} \frac{a^{j-2n} b^n \binom{j-n}{n}}{j-n} \tau^{j-2n}, & j \geq 1, \end{cases} \tag{4.4}$$

and

$$\mu_j^{a,b}(\tau) = \begin{cases} 2, & j = 0, \\ 2j\sum_{m=0}^{j} \frac{a^m b^{\frac{j-m}{2}} \xi_{j+m} \binom{\frac{j+m}{2}}{\frac{j-m}{2}}}{j+m} \tau^m, & j \geq 1, \end{cases} \tag{4.5}$$

where

$$\xi_l = \begin{cases} 1, & l \text{ even}, \\ 0, & l \text{ odd}. \end{cases} \tag{4.6}$$

The Binet form for these polynomials is [29]

$$\mu_j^{a,b}(\tau) = \frac{(a\tau + \sqrt{a^2\tau^2 + 4b})^j + (a\tau - \sqrt{a^2\tau^2 + 4b})^j}{2^j}, \quad j \geq 0.$$

It is worthy to mention here that from the GLPs for special values of a and b, we can extract some of the well-known polynomials; some specific cases of these values are shown in Table 1 [29].

Any continuous function $W(\tau)$ defined over $[0, 1]$ can be expanded in terms of the GLPs in the following form [29]:

$$W(\tau) = \sum_{i=0}^{\infty} c_i \mu_i^{a,b}(\tau). \tag{4.7}$$

By truncating the infinite series in equation (4.7), it can be written as follows:

$$W(\tau) \approx W_N(\tau) = \sum_{i=0}^{N} c_i \mu_i^{a,b}(\tau) = C^T \Phi(\tau), \tag{4.8}$$

where

$$C^T = [c_0, c_1, \ldots, c_N] \tag{4.9}$$

Table 1 The relation between the GLPs and some other polynomials

a	b	Polynomials	
1	$-\beta$	First kind Dickson	$D_n(\tau) = \mu_n^{1,-\beta}(\tau)$
3	−2	Fermat–Lucas	$f_n(\tau) = \mu_n^{3,-2}(\tau)$
2	−1	First-kind Chebyshev	$T_n(\tau) = \frac{1}{2}\mu_n^{2,-1}(\tau)$
2	1	Pell–Lucas	$P_n(\tau) = \mu_n^{2,1}(\tau)$
1	1	Lucas	$L_n(\tau) = \mu_n^{1,1}(\tau)$

and

$$\Phi(\tau) = \left[\mu_0^{a,b}(\tau), \mu_1^{a,b}(\tau), \ldots, \mu_N^{a,b}(\tau)\right]^T. \tag{4.10}$$

Now, the following two theorems state the convergence and error estimate of the generalized Lucas expansion.

Theorem 4.1 *Suppose that $h(\tau)$ is defined over $[0,1]$ and $|h^{(j)}(0)| \leq K^j$, $j \geq 0$, where K is a positive constant. Also, suppose that $h(\tau)$ has the expansion $h(\tau) = \sum_{j=0}^{\infty} c_j \mu_j^{a,b}(\tau)$; then it holds that*

1. $|c_j| \leq \frac{|a|^{-j} K^j \cosh(2|a|^{-1} b^{\frac{1}{2}} K)}{j!}$;
2. *The series converges absolutely.*

Proof The proof is given in [29]. □

Theorem 4.2 *If $h(\tau)$ satisfies the assumptions stated in Theorem 4.1 and $e_N(\tau) = \sum_{j=N+1}^{\infty} c_j \mu_j^{a,b}(\tau)$, then the global error estimate is given as follows*:

$$\left|e_N(\tau)\right| < \frac{2e^{K(1+\sqrt{1+a^{-2}b})} \cosh(2K(1+\sqrt{1+a^{-2}b}))(1+\sqrt{1+a^{-2}b})^{N+1}}{(N+1)!}.$$

Proof The proof is given in [29]. □

5 Operational matrices of the GLPs

This section is devoted to deriving operational matrices of derivatives for the GLPs. Based on the GLPs vector $\Phi(\tau)$ mentioned in equation (4.10), we can determine the operational matrix of integer derivative as follows [29, 30]:

$$\frac{d\Phi(\tau)}{d\tau} = \mathcal{S}^{(1)} \Phi(\tau), \tag{5.1}$$

where $\mathcal{S}^{(1)} = (\mathcal{S}_{ij}^{(1)})$ is the $(N+1) \times (N+1)$ operational matrix of the first derivative; then, the elements of this matrix can be obtained explicitly in the following form:

$$\mathcal{S}_{ij}^{(1)} = \begin{cases} (-1)^{\frac{i-j+1}{2}} iab^{\frac{i-j-1}{2}} \delta_j, & i > j \text{ and } (i+j) \text{ odd}, \\ 0, & \text{otherwise}, \end{cases}$$

where

$$\delta_m = \begin{cases} \frac{1}{2}, & m = 0, \\ 1, & \text{otherwise}. \end{cases} \tag{5.2}$$

Equation (5.1) enables one to obtain $\frac{d^i \Phi(\tau)}{d\tau^i}$ for $i \geq 1$ as

$$\frac{d^i \Phi(\tau)}{d\tau^i} = \mathcal{S}^{(i)} \Phi(\tau) = \left(\mathcal{S}^{(1)}\right)^i \Phi(\tau). \tag{5.3}$$

Now, we elicit the left Caputo fractional derivative operational matrix of the GLPs of order α, which generalizes the integer differentiation operator matrix. This matrix will be derived in the next theorem [29, 30].

Theorem 5.1 *Let $\Phi(\tau)$ be the GLPs vector defined in equation (4.10); then, for any $\alpha > 0$, we have*

$$ {}_0^C D_\tau^\alpha \Phi(\tau) = \frac{d^\alpha \Phi(\tau)}{d\tau^\alpha} = \tau^{-\alpha} \mathcal{S}^{(\alpha)} \Phi(\tau), \tag{5.4} $$

where $\mathcal{S}^{(\alpha)}$ is the $(N+1)\times(N+1)$ lower triangular generalized Lucas operational matrix of order α for the left Caputo fractional derivative. This matrix is obtained explicitly in the form

$$ \mathcal{S}^{(\alpha)} = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_\alpha(\lceil\alpha\rceil,0) & \cdots & \gamma_\alpha(\lceil\alpha\rceil,\lceil\alpha\rceil) & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_\alpha(i,0) & \cdots & \gamma_\alpha(i,\lceil\alpha\rceil) & \cdots & \gamma_\alpha(i,i) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_\alpha(N,0) & \cdots & \gamma_\alpha(N,\lceil\alpha\rceil) & \cdots & \gamma_\alpha(N,i) & \cdots & \gamma_\alpha(N,N) \end{bmatrix}, \tag{5.5} $$

where

$$ \gamma_\alpha(i,j) = \sum_{n=\lceil\alpha\rceil}^{i} \frac{(-1)^{\frac{n-j}{2}} i n! \xi_{i+n} \xi_{j+n} \delta_j b^{\frac{i-j}{2}} (\frac{i+n}{2}-1)!}{(\frac{i-n}{2})!(\frac{n-j}{2})!(\frac{j+n}{2})!\Gamma(1+n-\alpha)}, $$

also, ξ_i and δ_j are given in equations (4.6) and (5.2).

Moreover, we can express the right Caputo fractional derivative operational matrix of order α of the GLPs vector $\Phi(\tau)$ in the following form:

$$ {}_\tau^C D_l^\alpha \Phi(\tau) = \mathcal{S}_{(\alpha)} \Phi(\tau), \tag{5.6} $$

which is constructed with the help of the following lemmas.

Lemma 5.2 *Let $\Phi(\tau)$ and $T_N(\tau) = [1, \tau, \ldots, \tau^N]^T$ be the vectors of generalized Lucas and Taylor polynomials, respectively; then $\Phi(\tau) = A T_N(\tau)$, where $A = (a_{i+1,j+1})_{i,j=0}^N$ is a lower triangular $(N+1)\times(N+1)$ matrix, and*

$$ a_{i+1,j+1} = \begin{cases} 2, & i = j = 0, \\ \dfrac{2i a^j b^{\frac{i-j}{2}} \xi_{i+j+2} (\frac{\frac{i+j}{2}}{\frac{i-j}{2}})}{(i+j)}, & i \geq j, i \neq 0, \\ 0, & \text{otherwise}, \end{cases} $$

where ξ_i is mentioned in equation (4.6).

Proof Regarding the definition expressed in (4.5), we have

$$\Phi(\tau) = AT_N(\tau), \tag{5.7}$$

where

$$A = \begin{bmatrix} 2 & 0 & 0 & & \\ 0 & a & 0 & & \\ 2b & 0 & a^2 & & \\ 0 & 3ab & 0 & & \\ \vdots & \vdots & \vdots & & \\ 2b^{\frac{N}{2}}\xi_{N+2} & \frac{2Nab^{\frac{N-1}{2}}\xi_{N+3}(\frac{\frac{N+1}{2}}{\frac{N-1}{2}})}{(N+1)} & \frac{2Na^2b^{\frac{N-2}{2}}\xi_{N+4}(\frac{\frac{N+2}{2}}{\frac{N-2}{2}})}{(N+2)} & & \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ a^3 & 0 & \cdots & 0 \\ \ddots & \vdots & & \\ \frac{2Na^3b^{\frac{N-3}{2}}\xi_{N+5}(\frac{\frac{N+3}{2}}{\frac{N-3}{2}})}{(N+3)} & \frac{2Na^4b^{\frac{N-4}{2}}\xi_{N+6}(\frac{\frac{N+4}{2}}{\frac{N-4}{2}})}{(N+4)} & \cdots & a^N \end{bmatrix}.$$

So, the desired result is obtained. □

Lemma 5.3 *Suppose that* $\Phi(\tau)$ *and* $T_N(\tau)$ *are defined in Lemma* 5.2. *Then there is a lower triangular matrix* L *such that* $T_N(l-\tau) = LT_N(\tau)$, *where* $T_N(l-\tau) = [1, l-\tau, \dots, (l-\tau)^N]^T$, *and the entries of the matrix* L *are given in the form*

$$L_{i+1,j+1} = \begin{cases} (-1)^j l^{i-j}\binom{i}{j}, & i \geq j, \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, we can conclude that $\Phi(\tau) = AL^{-1}T_N(l-\tau)$.

Proof Using the binomial expansion of $(l-\tau)^i$, we have

$$(l-\tau)^i = \sum_{j=0}^{i} (-1)^j \binom{i}{j} l^{i-j}\tau^j.$$

So, we get the following relation:

$$T_N(l-\tau) = LT_N(\tau), \tag{5.8}$$

where

$$L=\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l & -1 & 0 & \cdots & 0 \\ l^2 & -2l & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l^N & -\binom{N}{1}l^{N-1} & \binom{N}{2}l^{N-2} & \cdots & (-1)^N \end{bmatrix}. \tag{5.9}$$

Now, using Lemma 5.2 and relation (5.8), we obtain $\Phi(\tau)=AL^{-1}T_N(l-\tau)$, which completes the proof. □

Note: As a consequence of Lemma 5.3, we have

$${}_{\tau}^{C}D_{l}^{\alpha}\Phi(\tau)=AL^{-1}\,{}_{\tau}^{C}D_{l}^{\alpha}T_N(l-\tau)=AL^{-1}\big[{}_{\tau}^{C}D_{l}^{\alpha}1,{}_{\tau}^{C}D_{l}^{\alpha}(l-\tau),\ldots,{}_{\tau}^{C}D_{l}^{\alpha}(l-\tau)^N\big]^T. \tag{5.10}$$

Now, by taking the right Caputo fractional derivative operator of the vector $T_N(l-\tau)$ and using (2.12), we obtain

$$\begin{aligned} {}_{\tau}^{C}D_{l}^{\alpha}T_N(l-\tau)&=\begin{bmatrix} 0 \\ \frac{\Gamma(2)}{\Gamma(2-\alpha)}(l-\tau)^{1-\alpha} \\ \vdots \\ \frac{\Gamma(N+1)}{\Gamma(N+1-\alpha)}(l-\tau)^{N-\alpha} \end{bmatrix} \\ &=\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & \frac{\Gamma(2)}{\Gamma(2-\alpha)}(l-\tau)^{-\alpha} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\Gamma(N+1)}{\Gamma(N+1-\alpha)}(l-\tau)^{-\alpha} \end{bmatrix} T_N(l-\tau) \\ &=\boldsymbol{M}T_N(l-\tau). \end{aligned}$$

Furthermore, it can be written as follows:

$${}_{\tau}^{C}D_{l}^{\alpha}T_N(l-\tau)=\boldsymbol{M}T_N(l-\tau)=\boldsymbol{M}LT_N(\tau)=\boldsymbol{M}LA^{-1}\Phi(\tau). \tag{5.11}$$

Therefore, substituting (5.11) into (5.10) yields that

$${}_{\tau}^{C}D_{l}^{\alpha}\Phi(\tau)=AL^{-1}MLA^{-1}\Phi(\tau)=\mathcal{S}_{(\alpha)}\Phi(\tau). \tag{5.12}$$

Indeed equation (5.12) determines an easy way for calculating the right Caputo fractional derivative operational matrix of the GLPs vector $\Phi(\tau)$.

6 Description of the proposed method

Herein, we will concentrate on the numerical solution of the FOCP defined in (3.1) by applying the operational matrices for the GLPs and the spectral collocation technique. To do this, first, the necessary optimality conditions for the problem are attained from Theorem 3.1 as

$${}_{0}^{C}\mathfrak{D}_{\tau}^{\alpha}\mathfrak{V}(\tau)=\frac{\partial\mathcal{H}}{\partial\lambda}\big(\tau,\mathfrak{V}(\tau),\mathfrak{W}(\tau),\lambda(\tau)\big), \tag{6.1}$$

$$ {}_{\tau}\mathfrak{D}_{T}^{\alpha}\lambda(\tau) = \frac{\partial \mathcal{H}}{\partial \mathfrak{V}}\big(\tau, \mathfrak{V}(\tau), \mathfrak{W}(\tau), \lambda(\tau)\big), \tag{6.2} $$

$$ \frac{\partial \mathcal{H}}{\partial \mathfrak{U}}\big(\tau, \mathfrak{V}(\tau), \mathfrak{W}(\tau), \lambda(\tau)\big) = 0, \tag{6.3} $$

$$ \mathfrak{V}(0) = \mathfrak{V}_0, \qquad \lambda(T) = 0. \tag{6.4} $$

It should be mentioned that, in practice, we compute an expression for $\mathfrak{W}(\tau)$ in terms of $\mathfrak{V}(\tau)$ and $\lambda(\tau)$ from the condition given in (6.3) in a very straightforward manner. Also, we can replace ${}_{\tau}\mathfrak{D}_{T}^{\alpha}\lambda(\tau)$ with ${}_{\tau}^{C}\mathfrak{D}_{T}^{\alpha}\lambda(\tau)$ by using (2.10). Thus, we can rewrite the above-mentioned system in the following form:

$$ {}_{0}^{C}\mathfrak{D}_{\tau}^{\alpha}\mathfrak{V}(\tau) = \mathcal{M}\big(\tau, \mathfrak{V}(\tau), \lambda(\tau)\big), \tag{6.5} $$

$$ {}_{\tau}^{C}\mathfrak{D}_{T}^{\alpha}\lambda(\tau) = \mathcal{W}\big(\tau, \mathfrak{V}(\tau), \lambda(\tau)\big), \tag{6.6} $$

$$ \mathfrak{V}(0) = \mathfrak{V}_0, \qquad \lambda(T) = 0, $$

where $\mathcal{M}(\tau, \mathfrak{V}(\tau), \lambda(\tau))$ and $\mathcal{W}(\tau, \mathfrak{V}(\tau), \lambda(\tau))$ are known functions. Now, we can approximate $\mathfrak{V}(\tau)$ and $\lambda(\tau)$ as

$$ \begin{aligned} &\mathfrak{V}(\tau) \approx \mathfrak{V}_N(\tau) = \sum_{i=0}^{N} \mathfrak{V}_i \mu_i^{a,b}(\tau) = V^T \Phi(\tau), \\ &\lambda(\tau) \approx \lambda_N(\tau) = \sum_{i=0}^{N} \lambda_i \mu_i^{a,b}(\tau) = \Lambda^T \Phi(\tau), \end{aligned} \tag{6.7} $$

where

$$ V^T = [\mathfrak{V}_0, \mathfrak{V}_1, \ldots, \mathfrak{V}_N], \qquad \Lambda^T = [\lambda_0, \lambda_1, \ldots, \lambda_N] $$

are unknown vectors which should be determined. By virtue of Sect. 5, the functions ${}_{0}^{C}\mathfrak{D}_{\tau}^{\alpha}\mathfrak{V}(\tau)$ and ${}_{\tau}^{C}\mathfrak{D}_{T}^{\alpha}\lambda(\tau)$ can be approximated in the following manner:

$$ {}_{0}^{C}\mathfrak{D}_{\tau}^{\alpha}\mathfrak{V}(\tau) \approx \tau^{-\alpha} V^T \mathcal{S}^{(\alpha)} \Phi(\tau), \qquad {}_{\tau}^{C}\mathfrak{D}_{T}^{\alpha}\lambda(\tau) \approx \Lambda^T \mathcal{S}_{(\alpha)} \Phi(\tau). \tag{6.8} $$

In addition, the boundary conditions expressed in (6.4) yield

$$ \mathfrak{V}(0) = V^T \Phi(0) = \mathfrak{V}_0, \qquad \lambda(T) = \Lambda^T \Phi(T) = 0. \tag{6.9} $$

Substituting (6.7) and (6.8) into (6.5) and (6.6), the residuals of these equations can be computed as follows:

$$ \begin{aligned} &R(\tau) = \tau^{-\alpha} V^T \mathcal{S}^{(\alpha)} \Phi(\tau) - \mathcal{M}\big(\tau, V^T \Phi(\tau), \Lambda^T \Phi(\tau)\big), \\ &\tilde{R}(\tau) = \Lambda^T \mathcal{S}_{(\alpha)} \Phi(\tau) - \mathcal{W}\big(\tau, V^T \Phi(\tau), \Lambda^T \Phi(\tau)\big). \end{aligned} \tag{6.10} $$

The application of the spectral collocation technique is based on forcing the residuals to vanish at selected collocation nodes. There are even some other selections for choosing

these nodes; one can use the following collocation nodes as well:

$$t_j = \frac{T}{2} - \frac{T}{2}\cos\left(\frac{\pi}{N}j\right), \quad j = 0, 1, \ldots, N, \tag{6.11}$$

where t_j, $j = 0, 1, \ldots, N$, are the shifted Chebyshev–Gauss–Lobatto points in the interval $[0, T]$. We should construct the associated system of $(2N + 2)$ algebraic equations since $(2N + 2)$ unknown coefficients $\mathfrak{V}_j$ and λ_j $(j = 0, 1, \ldots, N)$ exist. For this purpose, the first equation of (6.10) is collocated at the nodes t_i, $i = 1, \ldots, N$, and the second equation of (6.10) is collocated at the nodes t_k, $k = 0, 1, \ldots, N - 1$, as follows:

$$\begin{aligned} &R(t_i) = 0, \quad i = 1, 2, \ldots, N, \\ &\tilde{R}(t_k) = 0, \quad k = 0, 1, \ldots, N - 1. \end{aligned} \tag{6.12}$$

Hence, the above-mentioned system contains $2N$ algebraic equations. Now, equations given in (6.12) together with the boundary conditions (6.9) form a nonlinear (or linear) system of algebraic equations in the unknown coefficients $\mathfrak{V}_j$ and λ_j $(j = 0, 1, \ldots, N)$ that has $(2N + 2)$ equations and $(2N + 2)$ unknowns. We can utilize the Newton iterative technique for solving this system; then, by determining $\mathfrak{V}_j$ and λ_j $(j = 0, 1, \ldots, N)$, the desired approximate solutions can be calculated from (6.7).

7 Numerical experiments

In this section, we introduce some examples to test the performance and efficiency of the proposed method. These examples are selected from the literature for their importance and repetition. They also cover a variety of FOCPs. In Examples 1, 2, 3, and 5, we consider a linear time-invariant system with a quadratic performance index. Example 5 is a practical example with engineering applications. Moreover, a nonlinear time-varying system with a quadratic performance index is presented in Example 4. For these examples, the exact solutions when $\alpha = 1$ are known, and we can compare them with the approximate solutions obtained by the proposed method. The numerical simulations are implemented by MAPLE 18 with Digits = 20. All computations are performed on a Core i5 PC Laptop with 6 GB of RAM and 1.80 GHz of CPU to show no limitation on memory usage. In the following examples, the parameter N denotes the number of the GLPs.

Example 1 ([14, 15, 18, 23, 43]) Consider the following FOCP:

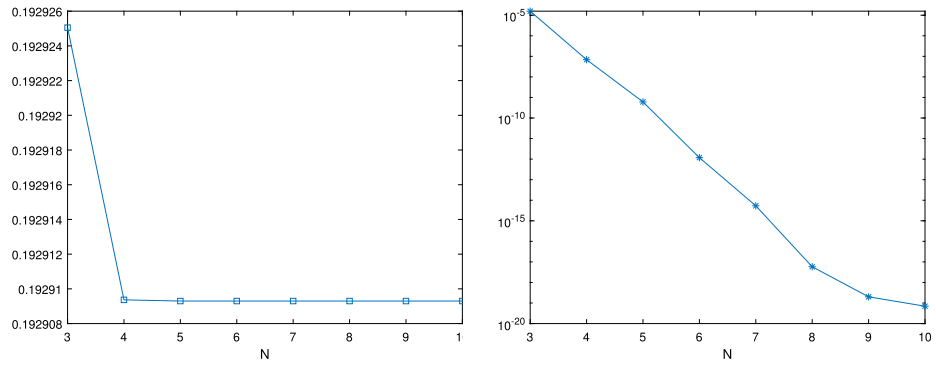
$$\operatorname{Min} \mathfrak{J}(\mathfrak{W}) = \frac{1}{2}\int_0^1 \left(\mathfrak{V}^2(\tau) + \mathfrak{W}^2(\tau)\right) d\tau$$

subject to

$$\begin{aligned} &{}_0^C\mathfrak{D}_\tau^\alpha \mathfrak{V}(\tau) = -\mathfrak{V}(\tau) + \mathfrak{W}(\tau), \\ &\mathfrak{V}(0) = 1. \end{aligned}$$

The exact solution to this problem when $\alpha = 1$ is as follows:

$$\mathfrak{V}^*(\tau) = \cosh(\sqrt{2}\tau) + \theta \sinh(\sqrt{2}\tau),$$



(a) The approximate values of $\mathfrak{J}$ (b) The absolute errors of $\mathfrak{J}$

Figure 1 Graphs of the approximate values and the absolute errors of $\mathfrak{J}$ when $\alpha = 1$ for some values of N in Example 1

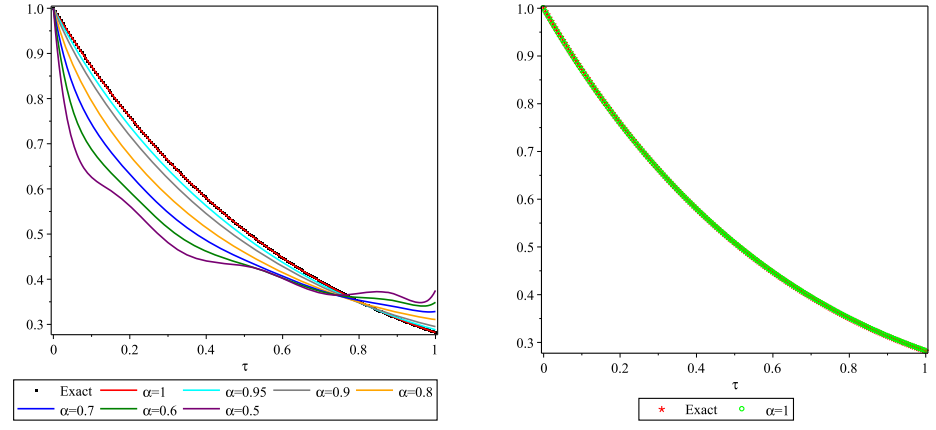


Figure 2 Graphs of the exact and numerical solutions of $\mathfrak{V}(\tau)$ for various values of α in Example 1

$$\mathfrak{W}^*(\tau) = (1 + \sqrt{2}\theta)\cosh(\sqrt{2}\tau) + (\sqrt{2} + \theta)\sinh(\sqrt{2}\tau),$$

where $\theta = -\frac{\cosh(\sqrt{2})+\sqrt{2}\sinh(\sqrt{2})}{\sqrt{2}\cosh(\sqrt{2})+\sinh(\sqrt{2})}$. The minimum value of the performance index $\mathfrak{J}$ when $\alpha = 1$ is $\mathfrak{J}^* = 0.1929092980932$. In Fig. 1, the approximate values and the absolute errors of $\mathfrak{J}$ for some values of N when $\alpha = 1$ are plotted. Figures 2 and 3 compare the exact solutions and the approximate solutions of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ for various values of α and $N = 8$, respectively. Tables 2 and 3 show a comparison between the approximate solutions obtained by our method for various values of α at some values of τ and the exact solutions. In addition, the CPU time for various values of α is included in Table 2. From these results, it is clear that the approximate solutions at $\alpha = 1$ are in very good agreement with the corresponding exact solutions. Furthermore, as α approaches 1, the approximate solutions of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ converge to the exact solutions. The approximate values of $\mathfrak{J}$ at $\alpha = 0.8$, 0.9, and 1 for the proposed and several numerical methods are included in Table 4. We compare the approximate values of $\mathfrak{J}$ and the CPU time obtained using the methods in [44, 45] and the proposed method in Table 5. From this table, it is clear that our method requires significantly less CPU time. In Table 5, M_1, M_2, and N_1 are the order of Bernoulli polynomials,

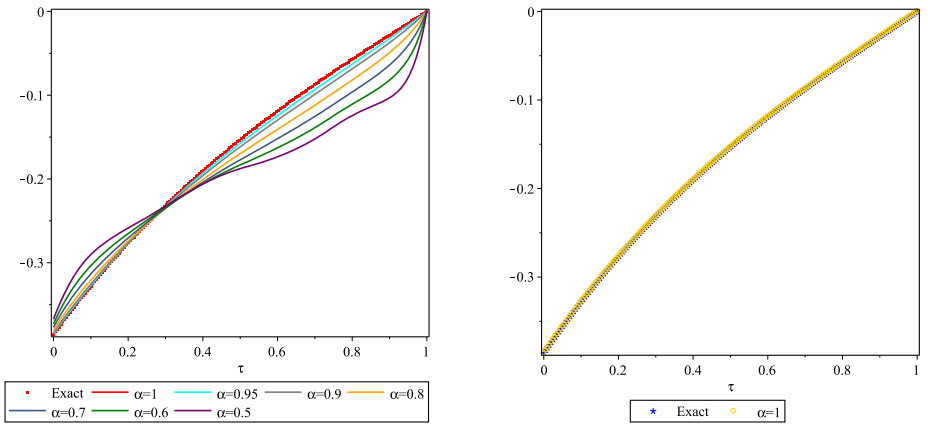


Figure 3 Graphs of the exact and numerical solutions of $\mathfrak{W}(\tau)$ for $\alpha = 1$ in Example 1

Taylor polynomials, and block-pulse functions, respectively. The absolute errors of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ when $\alpha = 1$ and $N = 4, 6, 8$, and 10 are shown in Figs. 4 and 5. These figures also illustrate the fast convergence rate of the proposed method since the errors decay rapidly by increasing the number of the GLPs. Moreover, Table 6 reports the maximum absolute errors of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ and the absolute errors of $\mathfrak{J}$ given by the proposed method in comparison to the methods in [18, 23] at $\alpha = 1$ and $N = 4, 6, 8$, and 10. The obtained results show that the errors, specially to control variable $\mathfrak{W}(\tau)$, are better for the proposed method than those obtained in [18, 23]. From these tables and figures, it can be seen that the state and the control variables are accurately approximated by our method.

Example 2 Consider the following FOCP:

$$\operatorname{Min} \mathfrak{J}(\mathfrak{W}) = \frac{1}{2}\int_0^1 \left[\left(\mathfrak{V}(\tau) - \tau^{\alpha+1}\right)^2 + \left(\mathfrak{W}(\tau) - \tau^{\alpha+1} - \tau\Gamma(\alpha+2)\right)^2\right] d\tau,$$

subject to

$$\begin{aligned} &{}_0^C\mathfrak{D}_\tau^\alpha \mathfrak{V}(\tau) = -\mathfrak{V}(\tau) + \mathfrak{W}(\tau), \\ &\mathfrak{V}(0) = 0. \end{aligned}$$

For any value of $\alpha > 0$, the exact solution to this problem is

$$\mathfrak{V}^*(\tau) = \tau^{\alpha+1}, \qquad \mathfrak{W}^*(\tau) = \tau^{\alpha+1} + \tau\Gamma(\alpha+2).$$

The minimum value of the performance index $\mathfrak{J}$ when $\alpha = 1$ is $\mathfrak{J}^* = 0$. Figure 6 compares the exact solutions and the approximate solutions of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ for $N = 8$ and $\alpha = 0.5$, 0.7, 0.9, and 1, respectively. In Fig. 7, we plot the state variable $\mathfrak{V}(\tau)$ and the control variable $\mathfrak{W}(\tau)$ for $\alpha = 0.5$ and some values of N along with the exact solutions. The absolute errors of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ for various values of α are listed in Tables 7 and 8. In addition, the CPU time and the absolute errors of $\mathfrak{J}$ for various values of α are included in these tables,

Table 2 Approximate solutions of $\mathfrak{V}(\tau)$ for various values of α where $N = 8$ along with CPU time in Example 1

τ	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 0.95$	$\alpha = 1$	Exact
0.1	0.62584500	0.68666710	0.74323669	0.79325450	0.83580800	0.85428224	0.87097241	0.87097241
0.2	0.56228260	0.59399381	0.63261220	0.67495369	0.71800733	0.73902711	0.75939333	0.75939333
0.3	0.48110286	0.51388507	0.54780242	0.58422853	0.62300270	0.64294652	0.66302744	0.66302744
0.4	0.44034576	0.46174082	0.48597000	0.51372055	0.54520062	0.56221791	0.57994422	0.57994422
0.5	0.42948855	0.43256175	0.44289130	0.45937629	0.48142481	0.49436936	0.50847923	0.50847923
0.6	0.40206008	0.40260088	0.40655351	0.41495118	0.42841422	0.43714154	0.44720078	0.44720078
0.7	0.36920422	0.37274202	0.37468952	0.37781684	0.38410943	0.38887743	0.39488126	0.39488126
0.8	0.36867554	0.35967523	0.35322486	0.34904107	0.34778268	0.34856867	0.35047254	0.35047254
0.9	0.36471034	0.35082831	0.33811766	0.32708465	0.31841614	0.31526500	0.31308497	0.31308496
1.0	0.37517682	0.34890244	0.32857936	0.31083431	0.29508251	0.28813145	0.28196953	0.28196953
CPU time	0.625 s	0.610 s	0.609 s	0.578 s	0.547 s	0.594 s	0.515 s	–

Table 3 Approximate solutions of $\mathfrak{W}(\tau)$ for various values of α where $N = 8$ in Example 1

τ	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 0.95$	$\alpha = 1$	Exact
0.1	−0.29024517	−0.30334744	−0.31328561	−0.32047676	−0.32530759	−0.32692973	−0.32806014	−0.32806014
0.2	−0.25869970	−0.26522810	−0.27043720	−0.27421650	−0.27639207	−0.27684564	−0.27687383	−0.27687383
0.3	−0.23231436	−0.23389564	−0.23494522	−0.23499330	−0.23378057	−0.23267242	−0.23123424	−0.23123424
0.4	−0.20601442	−0.20579033	−0.20392189	−0.20055293	−0.19590475	−0.19317829	−0.19022704	−0.19022704
0.5	−0.18767675	−0.18323648	−0.17711352	−0.16978015	−0.16162988	−0.15736423	−0.15303073	−0.15303073
0.6	−0.17370140	−0.16299753	−0.15204853	−0.14090284	−0.12975078	−0.12426686	−0.11890014	−0.11890014
0.7	−0.15198700	−0.13868339	−0.12529307	−0.11195858	−0.09909087	−0.09297956	−0.08715152	−0.08715152
0.8	−0.12438243	−0.11082380	−0.09631388	−0.08197536	−0.06872931	−0.06270956	−0.05714883	−0.05714883
0.9	−0.10332343	−0.08222159	−0.06406149	−0.04913442	−0.03733057	−0.03249359	−0.02829103	−0.02829103
1.0	2.13×10^{-16}	1.76×10^{-16}	-7.7×10^{-17}	4.4×10^{-18}	-1.9×10^{-18}	-2.5×10^{-19}	4.485×10^{-21}	5×10^{-20}

Table 4 The results obtained for $\mathfrak{J}$ with $\alpha = 0.8, 0.9$, and 1 via several numerical schemes for Example 1

Method	$\mathfrak{J}$		
	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1$
Legendre wavelets [20]	0.16707	0.17952	0.19290
Homotopy perturbation method [46]	0.16729	0.17952	0.19290
Laguerre polynomials [47]	0.16982	0.18155	0.19291
Hermite polynomials [23]	0.17999	0.18624	0.19290
Variational iteration method [14]	0.16711	0.17953	0.19290
Predictor-corrector method [15]	0.16711	0.17954	0.19290
Exact solution [43]	Not reported	Not reported	0.19290
The proposed method	0.16777	0.17994	0.19290

Table 5 The results obtained for $\mathfrak{J}$ and CPU time with $\alpha = 1$ via several numerical schemes for Example 1

Method	$\mathfrak{J}$	CPU Time
Hybrid of block-pulse and Bernoulli polynomials [44]		
$M_1 = 3$ and $N_1 = 1$	0.1929094450245	4.79688
$M_1 = 5$ and $N_1 = 1$	0.1929092980929	6.25000
Hybrid of block-pulse functions and Taylor polynomials [45]		
$M_2 = 3$ and $N_1 = 1$	0.1929094450240	2.42188
$M_2 = 5$ and $N_1 = 1$	0.1929092980930	3.73225
The proposed method		
$N = 3$	0.1929250524756	0.469
$N = 5$	0.1929092986997	0.484
$N = 7$	0.1929092980932	0.500
Exact	0.1929092980932	

respectively. From these results, it is worthwhile to note that the approximate solutions obtained by the proposed method completely coincide with the exact solutions.

Example 3 ([14, 21, 25]) Consider the following FOCP:

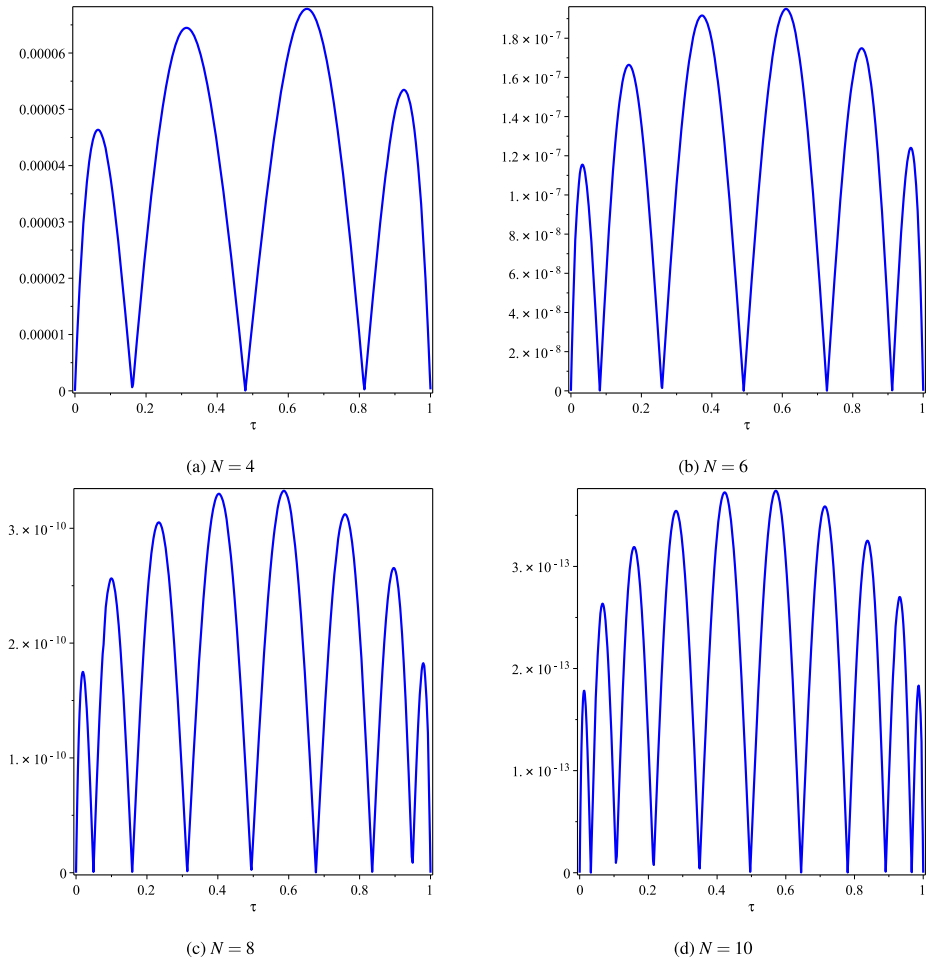
$$\operatorname{Min} \mathfrak{J}(\mathfrak{W}) = \frac{1}{2}\int_0^1 \left(\mathfrak{V}_1^2(\tau) + \mathfrak{V}_2^2(\tau) + \mathfrak{W}^2(\tau)\right) d\tau$$

subject to

$$\begin{aligned}
&{}_0^C\mathfrak{D}_\tau^\alpha \mathfrak{V}_1(\tau) = -\mathfrak{V}_1(\tau) + \mathfrak{V}_2(\tau) + \mathfrak{W}(\tau),\\
&{}_0^C\mathfrak{D}_\tau^\alpha \mathfrak{V}_2(\tau) = -2\mathfrak{V}_2(\tau),\\
&\mathfrak{V}_1(0) = 1, \qquad \mathfrak{V}_2(0) = 1.
\end{aligned}$$

We obtain the exact solution to this problem when $\alpha = 1$ as follows:

$$\begin{aligned}
&\mathfrak{V}_1^*(\tau) = -\frac{3}{2}e^{-2\tau} + (\sqrt{2}+1)\theta_1 e^{-\sqrt{2}\tau} + (-\sqrt{2}+1)\theta_2 e^{\sqrt{2}\tau},\\
&\mathfrak{V}_2^*(\tau) = e^{-2\tau},\\
&\mathfrak{W}^*(\tau) = \frac{1}{2}e^{-2\tau} - \theta_1 e^{-\sqrt{2}\tau} - \theta_2 e^{\sqrt{2}\tau},
\end{aligned}$$

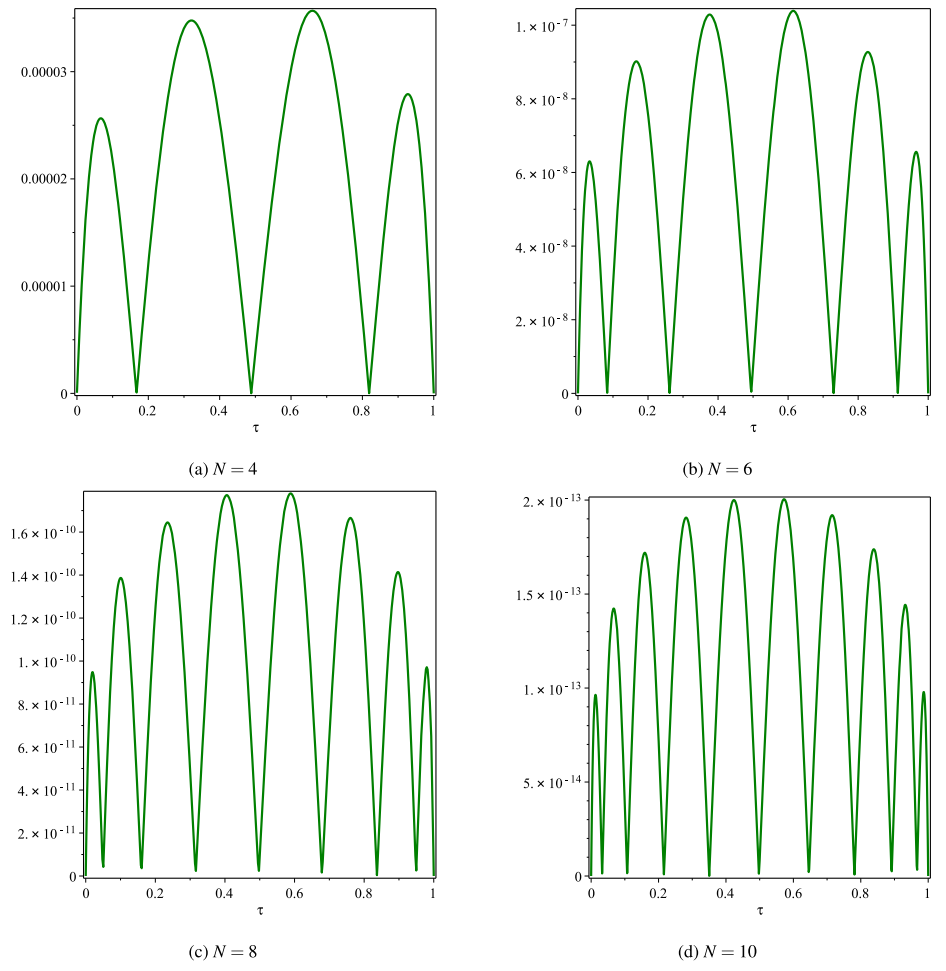


(a) $N = 4$ (b) $N = 6$

(c) $N = 8$ (d) $N = 10$

Figure 4 Graphs of the absolute errors of $\mathfrak{V}(\tau)$ when $\alpha = 1$ for some values of N in Example 1

where $\theta_1 = \frac{e^{-2}\sqrt{2}+5e^{\sqrt{2}}-e^{-2}}{2(e^{-\sqrt{2}}\sqrt{2}+e^{\sqrt{2}}\sqrt{2}-e^{-\sqrt{2}}+e^{\sqrt{2}})}$ and $\theta_2 = \frac{e^{-2}\sqrt{2}-5e^{-\sqrt{2}}+e^{-2}}{2(e^{-\sqrt{2}}\sqrt{2}+e^{\sqrt{2}}\sqrt{2}-e^{-\sqrt{2}}+e^{\sqrt{2}})}$. The minimum value of the performance index $\mathfrak{J}$ when $\alpha = 1$ is $\mathfrak{J}^* = 0.4319872403$. Figure 8 compares the exact solutions and the approximate solutions of $\mathfrak{V}_1(\tau)$, $\mathfrak{V}_2(\tau)$, and $\mathfrak{W}(\tau)$ for various values of α and $N = 8$, respectively. From this figure, it is clear that the approximate solutions when $\alpha = 1$ are in very good agreement with the corresponding exact solutions. Furthermore, as α approaches 1, the approximate solutions of $\mathfrak{V}_1(\tau)$, $\mathfrak{V}_2(\tau)$, and $\mathfrak{W}(\tau)$ converge to the exact solutions. Table 9 reports the absolute errors of $\mathfrak{V}_1(\tau)$, $\mathfrak{V}_2(\tau)$, and $\mathfrak{W}(\tau)$ obtained by the proposed method in comparison to the method in [25] at $\alpha = 1$ and $N = 5$ and 8. The yielded results show that the approximate solutions are more accurate for the proposed method than the method in [25]. Moreover, the absolute errors of $\mathfrak{V}_1(\tau)$, $\mathfrak{V}_2(\tau)$, and $\mathfrak{W}(\tau)$ for $N = 11$ and $\alpha = 1$ are shown in Fig. 9. These results also illustrate the fast convergence rate of the proposed method since the errors decay rapidly by increasing the number of the GLPs. The approximate values of $\mathfrak{J}$ at $\alpha = 0.5$, 0.8, 0.9, 0.99, and 1 for the proposed method and the methods in [25] are included in Table 10. In addition, the CPU time for various values of α is included in Table 10. From these tables and figures, it can be seen that the state and the control variables are accurately approximated by the proposed method.



(a) $N = 4$ (b) $N = 6$

(c) $N = 8$ (d) $N = 10$

Figure 5 Graphs of the absolute errors of $\mathfrak{W}(\tau)$ when $\alpha = 1$ for some values of N in Example 1

Example 4 ([48, 49]) Consider the following FOCP:

$$\operatorname{Min} \mathfrak{J}(\mathfrak{W}) = \int_0^1 \left[\left(\mathfrak{V}(\tau) - \tau^2 \right)^2 + \left(\mathfrak{W}(\tau) - \tau e^{-\tau} + \frac{1}{2} e^{\tau^2 - \tau} \right)^2 \right] d\tau$$

subject to

$$\begin{aligned} &{}_0^C\mathfrak{D}_\tau^\alpha \mathfrak{V}(\tau) = e^{\mathfrak{V}(\tau)} + 2e^\tau \mathfrak{W}(\tau), \\ &\mathfrak{V}(0) = 0. \end{aligned}$$

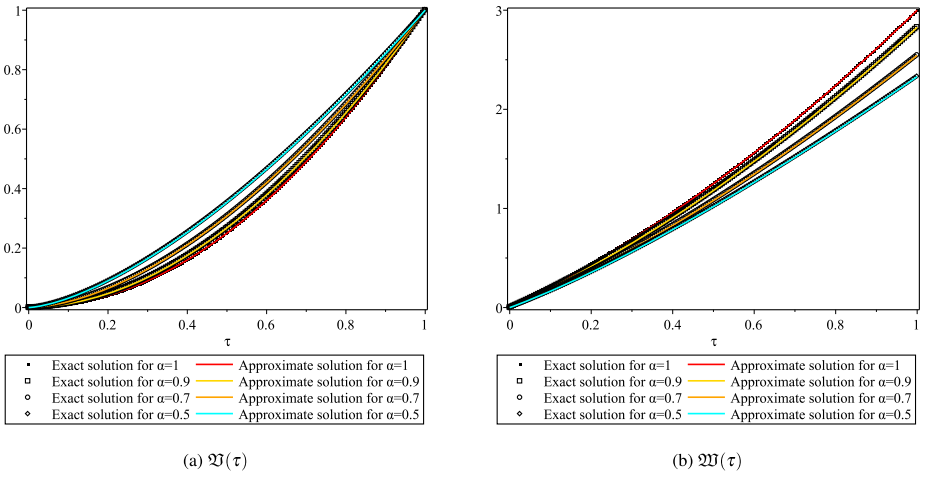
The exact solution to this problem when $\alpha = 1$ is as follows:

$$\mathfrak{V}^*(\tau) = \tau^2, \qquad \mathfrak{W}^*(\tau) = \tau e^{-\tau} - \frac{1}{2} e^{\tau^2 - \tau}.$$

The minimum value of the performance index $\mathfrak{J}$ when $\alpha = 1$ is $\mathfrak{J}^* = 0$. Figure 10 compares the exact solutions and the approximate solutions of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ for various values of α and $N = 6$, respectively. From this figure, it is clear that the approximate solutions

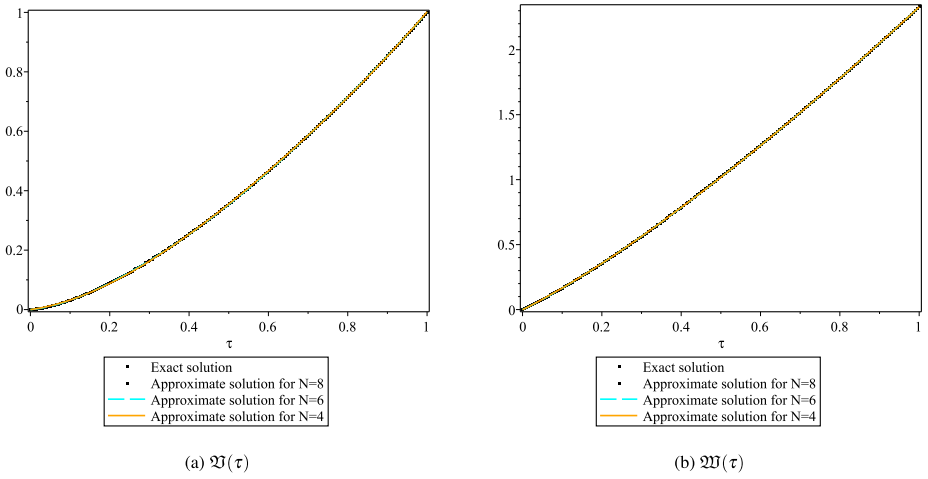
Table 6 A comparison between the results obtained by our method with those obtained in [18, 23] with various values of N for Example 1

N	$\lvert\mathfrak{J}-\mathfrak{J}^*\rvert$	$\lvert\mathfrak{V}-\mathfrak{V}^*\rvert$	$\lvert\mathfrak{W}-\mathfrak{W}^*\rvert$
The results obtained in [18]			
4	1.024×10^{-6}	1.782×10^{-4}	5.325×10^{-3}
6	6.795×10^{-11}	1.015×10^{-6}	4.923×10^{-5}
8	1.554×10^{-15}	3.433×10^{-9}	2.405×10^{-7}
10	6.661×10^{-16}	7.636×10^{-12}	7.215×10^{-10}
The results obtained in [23]			
4	1.469×10^{-7}	4.225×10^{-5}	5.404×10^{-4}
6	2.539×10^{-12}	1.232×10^{-7}	2.250×10^{-6}
8	4.670×10^{-18}	2.097×10^{-10}	5.021×10^{-9}
10	7.951×10^{-18}	2.354×10^{-13}	6.973×10^{-12}
The results obtained by our method			
4	6.879×10^{-8}	6.783×10^{-5}	3.567×10^{-5}
6	1.181×10^{-12}	1.949×10^{-7}	1.038×10^{-7}
8	5.820×10^{-18}	3.300×10^{-10}	1.771×10^{-10}
10	8.000×10^{-20}	3.725×10^{-13}	1.999×10^{-13}



(a) $\mathfrak{V}(\tau)$ (b) $\mathfrak{W}(\tau)$

Figure 6 Graphs of the exact and numerical solutions for various values of α in Example 2



(a) $\mathfrak{V}(\tau)$ (b) $\mathfrak{W}(\tau)$

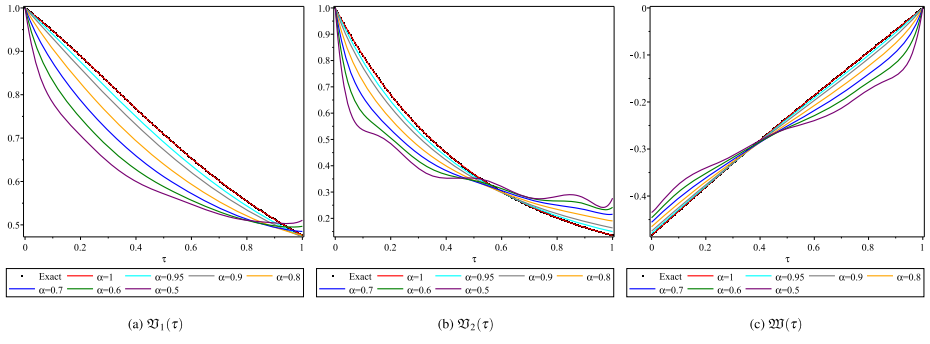
Figure 7 Graphs of the exact and numerical solutions when $\alpha = 0.5$ for some values of N in Example 2

Table 7 The results obtained for the absolute errors of $\mathfrak{V}(\tau)$ with various values of α along with CPU time where $N = 8$ for Example 2

τ	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 0.95$	$\alpha = 1$
0.1	1.470×10^{-4}	1.116×10^{-4}	7.083×10^{-5}	3.469×10^{-5}	9.941×10^{-6}	3.019×10^{-6}	4.564×10^{-20}
0.2	1.706×10^{-4}	1.055×10^{-4}	5.868×10^{-5}	2.929×10^{-5}	1.217×10^{-5}	5.946×10^{-6}	1.072×10^{-19}
0.3	8.719×10^{-5}	3.705×10^{-5}	9.747×10^{-6}	2.331×10^{-6}	4.867×10^{-6}	3.366×10^{-6}	5.396×10^{-20}
0.4	1.057×10^{-4}	7.278×10^{-5}	4.266×10^{-5}	1.946×10^{-5}	4.847×10^{-6}	1.098×10^{-6}	4.341×10^{-20}
0.5	1.086×10^{-4}	5.281×10^{-5}	2.186×10^{-5}	8.012×10^{-6}	3.183×10^{-6}	1.826×10^{-6}	1.950×10^{-19}
0.6	6.059×10^{-5}	4.533×10^{-5}	2.886×10^{-5}	1.591×10^{-5}	7.140×10^{-6}	3.630×10^{-6}	2.778×10^{-19}
0.7	1.228×10^{-4}	6.372×10^{-5}	2.814×10^{-5}	9.236×10^{-6}	8.868×10^{-7}	4.396×10^{-7}	1.091×10^{-19}
0.8	1.366×10^{-5}	5.103×10^{-6}	8.351×10^{-6}	5.165×10^{-6}	1.169×10^{-6}	2.235×10^{-8}	1.541×10^{-19}
0.9	7.047×10^{-5}	4.508×10^{-5}	2.470×10^{-5}	1.177×10^{-5}	4.707×10^{-6}	2.331×10^{-6}	4.786×10^{-21}
1.0	1.350×10^{-4}	4.863×10^{-5}	1.479×10^{-5}	4.073×10^{-6}	1.542×10^{-6}	9.608×10^{-7}	3.491×10^{-19}
CPU Time	0.594 s	0.578 s	0.593 s	0.578 s	0.593 s	0.562 s	0.547 s

Table 8 The results obtained for the absolute errors of $\mathfrak{W}(\tau)$ and $\mathfrak{J}$ with various values of α where $N = 8$ for Example 2

τ	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 0.95$	$\alpha = 1$
0.1	9.131×10^{-6}	4.345×10^{-6}	1.058×10^{-6}	9.630×10^{-7}	1.591×10^{-6}	1.157×10^{-6}	6.879×10^{-20}
0.2	2.088×10^{-5}	1.105×10^{-5}	5.484×10^{-6}	3.003×10^{-6}	1.845×10^{-6}	1.144×10^{-6}	8.012×10^{-20}
0.3	1.636×10^{-5}	9.146×10^{-6}	4.195×10^{-6}	1.027×10^{-6}	5.267×10^{-7}	5.820×10^{-7}	9.709×10^{-20}
0.4	1.353×10^{-5}	7.694×10^{-6}	3.480×10^{-6}	7.930×10^{-7}	4.785×10^{-7}	5.038×10^{-7}	1.156×10^{-19}
0.5	1.374×10^{-5}	7.210×10^{-6}	3.602×10^{-6}	1.932×10^{-6}	1.118×10^{-6}	6.768×10^{-7}	1.064×10^{-19}
0.6	8.078×10^{-6}	4.724×10^{-6}	2.528×10^{-6}	1.404×10^{-6}	8.307×10^{-7}	5.069×10^{-7}	6.959×10^{-20}
0.7	1.468×10^{-5}	7.651×10^{-6}	3.406×10^{-6}	1.061×10^{-6}	1.579×10^{-8}	1.510×10^{-7}	4.455×10^{-20}
0.8	2.270×10^{-6}	1.870×10^{-6}	9.835×10^{-7}	2.235×10^{-7}	1.541×10^{-7}	1.544×10^{-7}	5.468×10^{-20}
0.9	9.783×10^{-6}	5.434×10^{-6}	2.728×10^{-6}	1.234×10^{-6}	4.675×10^{-7}	2.197×10^{-7}	5.632×10^{-20}
1.0	3.57×10^{-18}	1.160×10^{-18}	8.20×10^{-19}	1.09×10^{-19}	6.5×10^{-20}	4.89×10^{-20}	5.769×10^{-20}
$\lvert J - J^* \rvert$	6.119×10^{-9}	2.198×10^{-9}	6.730×10^{-10}	1.643×10^{-10}	2.661×10^{-11}	6.432×10^{-12}	1.233×10^{-38}



(a) $\mathfrak{V}_1(\tau)$ (b) $\mathfrak{V}_2(\tau)$ (c) $\mathfrak{W}(\tau)$

Figure 8 Graphs of the exact and numerical solutions for various values of α in Example 3

for the case of $\alpha = 1$ are in very good agreement with the corresponding exact solutions. Furthermore, as α approaches 1, the approximate solutions of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ converge to the exact solutions. The absolute errors of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ at $\alpha = 1$ and $N = 5$ are shown in Fig. 11. Moreover, Table 11 reports the absolute errors of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ obtained by our method in comparison to the method in [48] at $\alpha = 1$ and $N = 5$. Table 12 lists the maximum absolute errors of $\mathfrak{V}(\tau)$ and $\mathfrak{W}(\tau)$ and the absolute errors of $\mathfrak{J}$ given by the proposed method in comparison to the method in [49] at $\alpha = 1$ and $N = 6$. The obtained results show that the errors, specially to $\mathfrak{W}(\tau)$, are better for the proposed method than

Table 9 The comparison of absolute errors with the method in [25] for $N = 5$ and 8 in Example 3

τ	$N = 5$			$N = 8$		
	$\mathfrak{V}_1(\tau)$	$\mathfrak{V}_2(\tau)$	$\mathfrak{W}(\tau)$	$\mathfrak{V}_1(\tau)$	$\mathfrak{V}_2(\tau)$	$\mathfrak{W}(\tau)$
The results obtained in [25]						
0.0	5.0873×10^{-3}	1.1759×10^{-4}	3.7671×10^{-2}	6.4701×10^{-4}	9.2045×10^{-8}	2.7317×10^{-3}
0.2	1.3401×10^{-3}	9.6759×10^{-6}	6.1908×10^{-3}	2.4746×10^{-5}	7.0205×10^{-10}	5.9560×10^{-4}
0.4	4.8751×10^{-4}	1.7241×10^{-6}	1.1315×10^{-2}	5.0303×10^{-5}	1.1698×10^{-9}	1.2007×10^{-4}
0.6	4.3069×10^{-4}	4.3754×10^{-7}	1.0657×10^{-2}	3.4906×10^{-5}	9.3245×10^{-10}	7.3763×10^{-5}
0.8	1.2658×10^{-3}	7.6358×10^{-6}	4.7850×10^{-3}	3.8725×10^{-5}	1.6391×10^{-9}	5.4601×10^{-4}
1.0	4.8997×10^{-3}	1.1139×10^{-4}	3.4480×10^{-2}	6.4545×10^{-4}	8.9541×10^{-8}	2.5760×10^{-3}
The results obtained by our method						
0.0	0.0	2×10^{-20}	3.6908×10^{-9}	0.0	8.0×10^{-20}	5.6978×10^{-16}
0.2	2.3436×10^{-5}	2.1666×10^{-5}	7.8041×10^{-6}	5.8730×10^{-9}	4.3139×10^{-9}	1.8615×10^{-9}
0.4	1.6825×10^{-5}	1.5141×10^{-5}	5.4356×10^{-6}	8.0689×10^{-9}	5.9364×10^{-9}	2.5491×10^{-9}
0.6	7.0832×10^{-6}	7.1373×10^{-6}	2.5745×10^{-6}	7.8782×10^{-9}	5.7997×10^{-9}	2.4855×10^{-9}
0.8	2.1970×10^{-5}	2.0483×10^{-5}	7.3610×10^{-6}	5.0522×10^{-9}	3.7254×10^{-9}	1.5878×10^{-9}
1.0	4.7942×10^{-8}	3.0254×10^{-8}	2.0×10^{-20}	6.9046×10^{-15}	4.0177×10^{-15}	6.0×10^{-21}

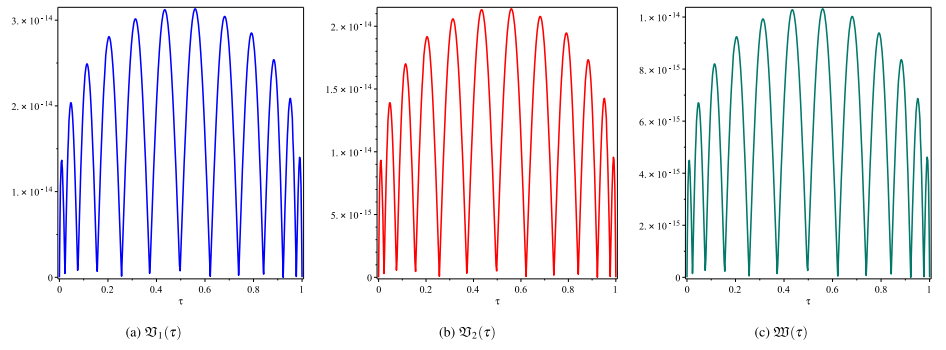


Figure 9 Graphs of the absolute errors when $\alpha = 1$ and $N = 12$ in Example 3

Table 10 A comparison between the results obtained by our method with those obtained in [14, 21] for $\mathfrak{J}$ and the CPU time with some values of α for Example 3

α	Method in [14]	Method in [21]	Our method	CPU Time
1.0	0.4319089	0.43198	0.43198724	0.656 s
0.99	0.4290055	0.42900	0.42909532	0.734 s
0.9	0.4030897	0.40308	0.40385286	0.750 s
0.8	0.3760124	0.37627	0.37758212	0.766 s
0.5	–	0.32909	0.31047699	0.797 s

those obtained in [48, 49]. From these tables and figures, it can be seen that the state and the control variables are accurately approximated by the proposed method.

Example 5 Consider the vibration of a mass-spring-damper system subjected to an external force. In particular, we aim to examine the step forcing functions, impulses, and response to harmonic excitations. Mostly, motors, rotating machinery, and so on lead to periodic motions of structures to induce vibrations into other mechanical devices and structures nearby [50]. Here, the action of an actuator force caused the control force $F(\tau) = b\mathfrak{W}(\tau)$, where b is a constant. On summing the forces, the equation for the forced

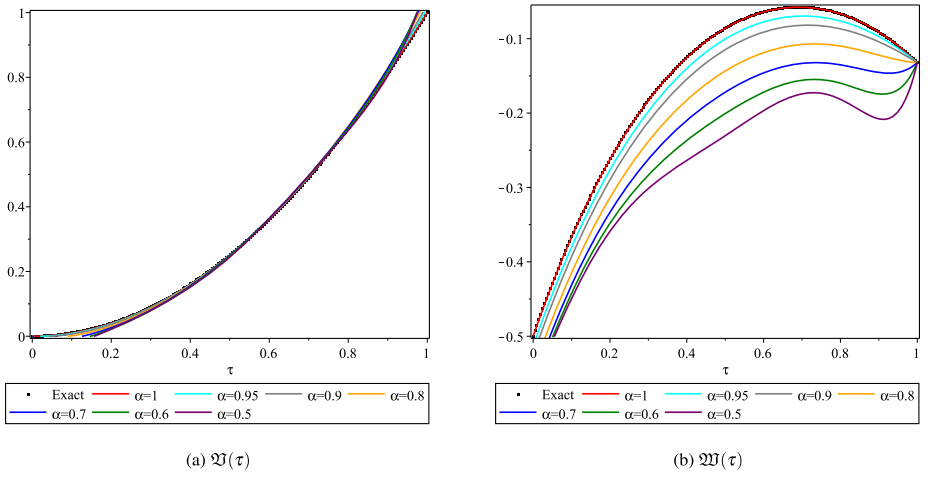


Figure 10 Graphs of exact and numerical solutions for various values of α in Example 4

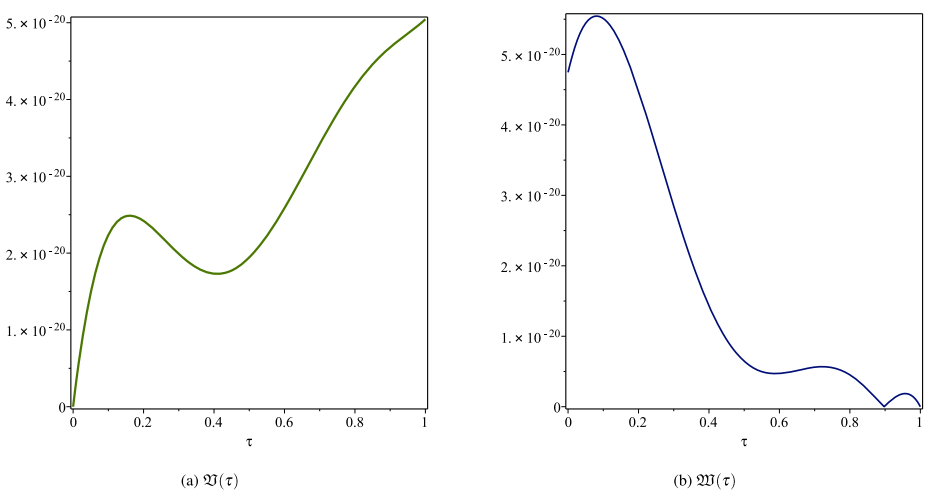


Figure 11 Graphs of the absolute errors when $\alpha = 1$ and $N = 5$ in Example 4

Table 11 A comparison between the absolute errors obtained by our method with those obtained in [48] with $N = 5$ for Example 4

τ	Our method		Method in [48]	
	$\mathfrak{V}(\tau)$	$\mathfrak{W}(\tau)$	$\mathfrak{V}(\tau)$	$\mathfrak{W}(\tau)$
0.0	0.0	4.7460×10^{-20}	5.1454×10^{-27}	2.7547×10^{-4}
0.1	2.2357×10^{-20}	5.5117×10^{-20}	5.9358×10^{-5}	3.3361×10^{-5}
0.2	2.4210×10^{-20}	4.4821×10^{-20}	4.9622×10^{-5}	4.2236×10^{-5}
0.3	1.9896×10^{-20}	2.8470×10^{-20}	2.3649×10^{-5}	9.7096×10^{-5}
0.4	1.7317×10^{-20}	1.4329×10^{-20}	7.1658×10^{-6}	5.7043×10^{-5}
0.5	1.9448×10^{-20}	6.4627×10^{-21}	6.1834×10^{-6}	1.3750×10^{-5}
0.6	2.5851×10^{-20}	4.7187×10^{-21}	1.4421×10^{-5}	3.8228×10^{-5}
0.7	3.4182×10^{-20}	5.6070×10^{-21}	2.0724×10^{-5}	3.7231×10^{-6}
0.8	4.1702×10^{-20}	4.5163×10^{-21}	1.6485×10^{-5}	5.7195×10^{-5}
0.9	4.6790×10^{-20}	1.2947×10^{-22}	3.0606×10^{-6}	2.9022×10^{-5}
1.0	5.0449×10^{-20}	2.1746×10^{-36}	8.0434×10^{-7}	1.0987×10^{-4}

Table 12 A comparison between the absolute errors obtained by our method with those obtained in [49] with $N = 6$ for Example 4

Method	$\lvert\mathfrak{V} - \mathfrak{V}^*\rvert$	$\lvert\mathfrak{W} - \mathfrak{W}^*\rvert$	$\lvert\mathfrak{J} - \mathfrak{J}^*\rvert$
Method in [49]	1.00×10^{-30}	1.68×10^{-5}	2.10×10^{-31}
Our method	5.46×10^{-30}	2.01×10^{-29}	1.64×10^{-58}

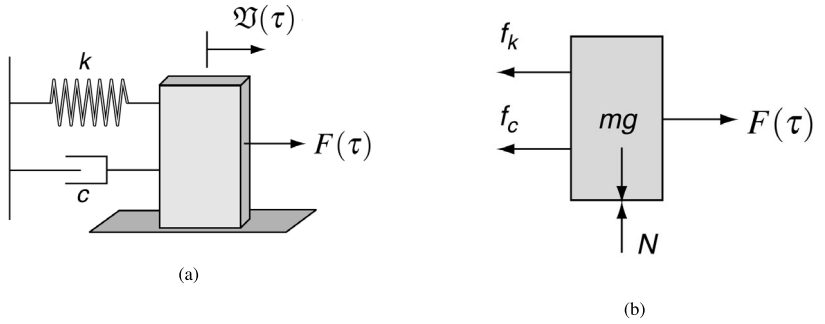


Figure 12 (**a**) Schematic of the forced mass-damper system assuming no friction on the surface and (**b**) free body diagram of the system of part (**a**) [50] in Example 5

vibration of the system in Fig. 12 becomes

$$m\ddot{\mathfrak{V}}(\tau) + c\dot{\mathfrak{V}}(\tau) + k\mathfrak{V}(\tau) = b\mathfrak{W}(\tau),$$

where m, c, and k are constants. We remind that the mass-spring-damper system can be used to model the response of most dynamic systems as well as study the elasticity and mechanical behavior of nonlinear and viscoelastic material. Based on the number and arrangement (parallel or series combination) of the elements of this system (i.e. mass, spring, or damper), the mass-spring-damper systems have various practical applications, including but not limited to suspension systems of vehicles, vibrations of building on viscoelastic-like foundations, simulation of the motion of tendons and muscle deformations, and computer animations. With the specific application of the linear regulator problem in vibration suppression, extracted from [23], we find the following FOCP:

$$\operatorname{Min} \mathfrak{J}(\mathfrak{W}) = \frac{1}{2}\int_0^1 \left(\mathfrak{V}_1^2(\tau) + a\mathfrak{W}^2(\tau)\right) d\tau$$

subject to

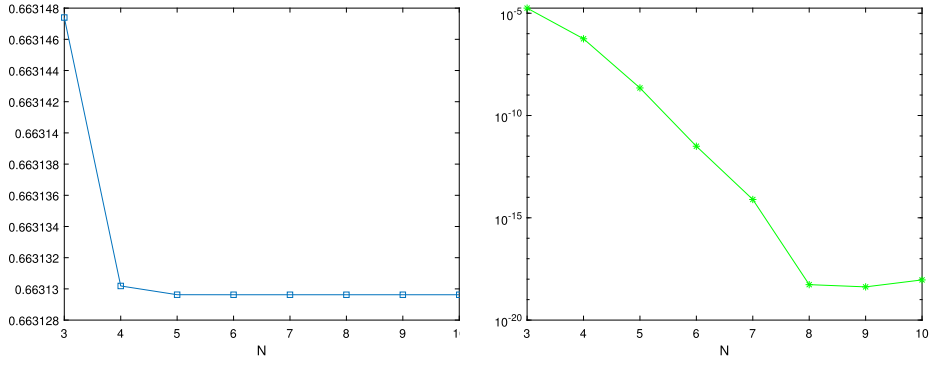
$$\begin{aligned}
&{}_0^C\mathfrak{D}_\tau^\alpha \mathfrak{V}_1(\tau) = \mathfrak{V}_2(\tau),\\
&{}_0^C\mathfrak{D}_\tau^\alpha \mathfrak{V}_2(\tau) = -\frac{k}{m}\mathfrak{V}_1(\tau) - \frac{c}{m}\mathfrak{V}_2(\tau) + \frac{b}{m}\mathfrak{U}(\tau),\\
&\mathfrak{V}_1(0) = 1, \qquad \mathfrak{V}_2(0) = 1.
\end{aligned}$$

Choosing $c = 2$ and $a = b = m = k = 1$, we obtain the exact solution when $\alpha = 1$ as follows:

$$\mathfrak{V}_1^*(\tau) = \left(\frac{2{,}448{,}542{,}446{,}934}{574{,}274{,}351{,}289} e^{-\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)} - \frac{1{,}056{,}415{,}030{,}945}{26{,}501{,}017{,}876{,}847} e^{\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}\right)$$

$$
\begin{aligned}
&\times \sin\left(\frac{1}{2}\sqrt{-2+2\sqrt{2}}\tau\right)\\
&+\left(\frac{182{,}130{,}319{,}402}{2{,}268{,}083{,}818{,}399}e^{\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}+\frac{717{,}441{,}983{,}179}{780{,}083{,}809{,}860}e^{-\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}\right)\\
&\times \cos\left(\frac{1}{2}\sqrt{-2+2\sqrt{2}}\tau\right),\\
\mathfrak{V}_2^*(\tau)=&\left(-\frac{4{,}619{,}908{,}248{,}187}{905{,}327{,}915{,}158}e^{-\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}-\frac{247{,}515{,}980{,}953}{3{,}080{,}802{,}211{,}846}e^{\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}\right)\\
&\times \sin\left(\frac{1}{2}\sqrt{-2+2\sqrt{2}}\tau\right)\\
&+\left(\frac{107{,}822{,}929{,}289}{1{,}538{,}469{,}380{,}682}e^{\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}+\frac{1{,}430{,}646{,}451{,}393}{1{,}538{,}469{,}380{,}682}e^{-\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}\right)\\
&\times \cos\left(\frac{1}{2}\sqrt{-2+2\sqrt{2}}\tau\right),\\
\mathfrak{W}^*(\tau)=&\left(-\frac{1{,}038{,}973{,}168{,}371}{1{,}369{,}025{,}535{,}412}e^{-\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}-\frac{476{,}880{,}084{,}019}{1{,}486{,}948{,}346{,}550}e^{\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}\right)\\
&\times \sin\left(\frac{1}{2}\sqrt{-2+2\sqrt{2}}\tau\right)\\
&+\left(\frac{4{,}059{,}677{,}169{,}262}{15{,}559{,}760{,}490{,}977}e^{\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}-\frac{584{,}626{,}462{,}642}{1{,}035{,}676{,}735{,}491}e^{-\frac{1}{2}\sqrt{2+2\sqrt{2}}\tau)}\right)\\
&\times \cos\left(\frac{1}{2}\sqrt{-2+2\sqrt{2}}\tau\right).
\end{aligned}
$$

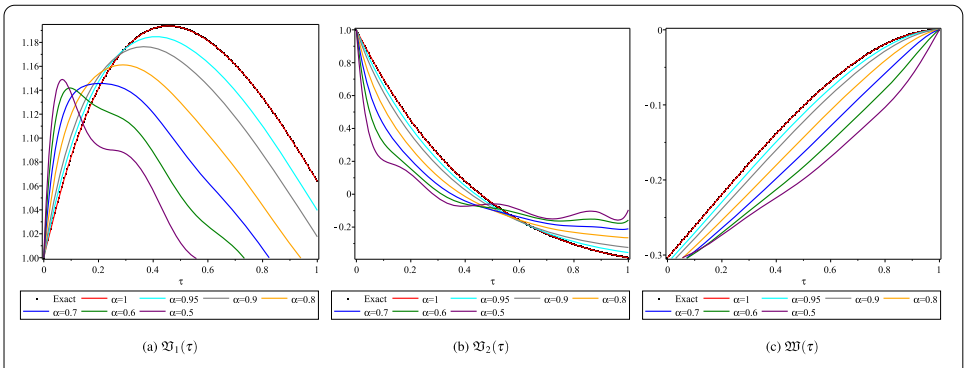
The minimum value of the performance index $\mathfrak{J}$ when $\alpha = 1$ is $\mathfrak{J}^* = 0.6631296243$. In Fig. 13, the approximate values and the absolute errors of $\mathfrak{J}$ for some values of N when $\alpha = 1$ are plotted. Figure 14 compares the exact solutions and the approximate solutions of $\mathfrak{V}_1(\tau)$, $\mathfrak{V}_2(\tau)$, and $\mathfrak{W}(\tau)$ for various values of α at $N = 8$, respectively. In Table 13, the absolute errors of $\mathfrak{V}_1(\tau)$, $\mathfrak{V}_2(\tau)$, and $\mathfrak{W}(\tau)$ for $N = 4$ and 10 at $\alpha = 1$ along with the CPU time are listed. Moreover, the absolute errors of $\mathfrak{V}_1(\tau)$, $\mathfrak{V}_2(\tau)$, and $\mathfrak{W}(\tau)$ when $\alpha = 1$ and



(a) The approximate values of $\mathfrak{J}$.

(b) The absolute errors of $\mathfrak{J}$.

Figure 13 Graphs of the approximate values and absolute errors of $\mathfrak{J}$ when $\alpha = 1$ for some values of N in Example 5

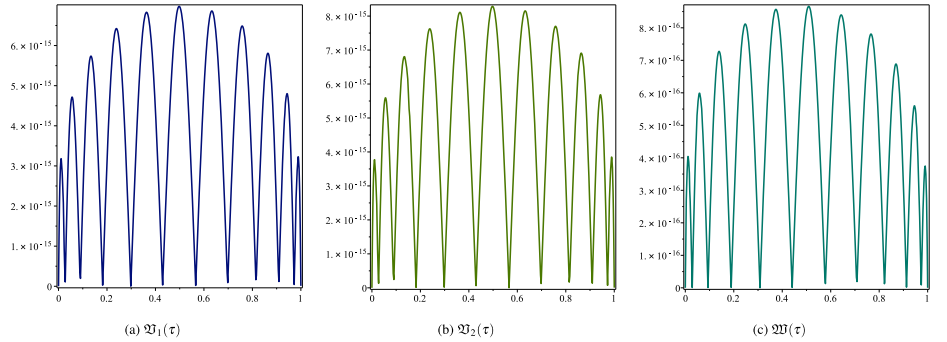


(a) $\mathfrak{V}_1(\tau)$ (b) $\mathfrak{V}_2(\tau)$ (c) $\mathfrak{W}(\tau)$

Figure 14 Graphs of the exact and numerical solutions for various values of α in Example 5

Table 13 The absolute errors obtained by our method with $N = 4, 10$ and $\alpha = 1$ along with the CPU time for Example 5

τ	$\mathfrak{V}_1(\tau)$		$\mathfrak{V}_2(\tau)$		$\mathfrak{W}(\tau)$	
	$N = 4$	$N = 10$	$N = 4$	$N = 10$	$N = 4$	$N = 10$
0.1	8.155×10^{-5}	3.247×10^{-14}	1.067×10^{-4}	5.942×10^{-14}	4.695×10^{-5}	1.659×10^{-14}
0.2	5.210×10^{-5}	6.518×10^{-14}	5.954×10^{-5}	1.171×10^{-13}	1.849×10^{-5}	3.106×10^{-14}
0.3	1.353×10^{-4}	1.663×10^{-13}	1.695×10^{-4}	2.877×10^{-13}	6.724×10^{-5}	6.810×10^{-14}
0.4	9.583×10^{-5}	1.746×10^{-13}	1.272×10^{-4}	2.946×10^{-13}	5.656×10^{-5}	6.411×10^{-14}
0.5	2.195×10^{-5}	1.625×10^{-14}	1.779×10^{-5}	1.930×10^{-14}	1.211×10^{-6}	1.975×10^{-15}
0.6	1.248×10^{-4}	1.593×10^{-13}	1.508×10^{-4}	2.764×10^{-13}	5.491×10^{-5}	6.597×10^{-14}
0.7	1.311×10^{-4}	1.792×10^{-13}	1.663×10^{-4}	3.030×10^{-13}	6.734×10^{-5}	6.653×10^{-14}
0.8	2.411×10^{-5}	8.725×10^{-14}	3.714×10^{-5}	1.433×10^{-13}	1.984×10^{-5}	2.837×10^{-14}
0.9	1.017×10^{-4}	4.781×10^{-14}	1.229×10^{-4}	7.761×10^{-14}	4.597×10^{-5}	1.473×10^{-14}
1.0	6.838×10^{-8}	6.470×10^{-18}	6.756×10^{-8}	4.836×10^{-17}	0.0	2×10^{-20}
CPU Time	0.641 s	0.765 s	–	–	–	–



(a) $\mathfrak{V}_1(\tau)$ (b) $\mathfrak{V}_2(\tau)$ (c) $\mathfrak{W}(\tau)$

Figure 15 Graphs of the absolute errors when $\alpha = 1$ and $N = 11$ in Example 5

$N = 11$ are shown in Fig. 15. These results also illustrate the fast convergence rate of the proposed method since the errors decay rapidly by increasing the number of the GLPs.

8 Conclusions and remarks

In this paper, we established an accurate and efficient new scheme to solve a class of FOCPs. By applying the GLPs, determining the operational matrices of fractional derivatives and the necessary optimality conditions, we reduced the main problem to the simple

problem of solving a system of algebraic equations. The proposed scheme is illustrated in some test examples, and the results demonstrate that our scheme is perfectly valid. It is also shown that the new scheme is quite reliable, simple, and reasonably accurate to solve FOCPs. As further research works, we will use the proposed scheme for delay FOCPs. Moreover, the new method has the ability to be applied for solving variable order FOCPs in new research.

Acknowledgements
Not available.

Funding
The financial assistance is not applicable.

Availability of data and materials
Data sharing is not applicable to this study.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author contribution
All authors contributed equally to the writing of this paper. All authors read and approved the final manuscript.

Publisher's Note

Received: 28 October 2021 Accepted: 8 November 2022 Published online: 24 November 2022

References

1. Bushnaq, S., Saeed, T., Torres, D.F.M., Zeb, A.: Control of COVID-19 dynamics through a fractional-order model. Alex. Eng. J. **60**(4), 3587–3592 (2021)
2. Dong, N.P., Long, H.V., Khastan, A.: Optimal control of a fractional order model for granular SEIR epidemic with uncertainty. Commun. Nonlinear Sci. Numer. Simul. **88**, 1–39 (2020)
3. Naik, P.A., Zu, J., Owolabi, K.M.: Global dynamics of a fractional order model for the transmission of HIV epidemic with optimal control. Chaos Solitons Fractals **138**, 1–24 (2020)
4. Shi, R., Li, Y., Wang, C.: Stability analysis and optimal control of a fractional-order model for African swine fever. Virus Res. **288**, 1–24 (2020)
5. Agrawal, O.P.: A general formulation and solution scheme for fractional optimal control problems. Nonlinear Dyn. **38**(1), 323–337 (2004)
6. Agrawal, O.P.: A formulation and numerical scheme for fractional optimal control problems. J. Vib. Control **14**(9–10), 1291–1299 (2008)
7. Sweilam, N.H., Al-Ajami, T.M., Hoppe, R.H.W.: Numerical solution of some types of fractional optimal control problems. Sci. World J. **2013**, Article ID 306237 (2013)
8. Pooseh, S., Almeida, R., Torres, D.F.M.: Fractional order optimal control problems with free terminal time. J. Ind. Manag. Optim. **10**(2), 363–381 (2014)
9. Sweilam, N.H., Al-Ajami, T.M.: Legendre spectral-collocation method for solving some types of fractional optimal control problems. J. Adv. Res. **6**(3), 393–403 (2015)
10. Tohidi, E., Saberi Nik, H.: A Bessel collocation method for solving fractional optimal control problems. Appl. Math. Model. **39**(2), 455–465 (2015)
11. Yang, Y., Zhang, J., Liu, H., Vasilev, A.O.: An indirect convergent Jacobi spectral collocation method for fractional optimal control problems. Math. Methods Appl. Sci. **44**(4), 2806–2824 (2021)
12. Habibli, M., Noori Skandari, M.H.: Fractional Chebyshev pseudospectral method for fractional optimal control problems. Optim. Control Appl. Methods **40**(3), 558–572 (2019)
13. Kumar, N., Mehra, M.: Legendre wavelet collocation method for fractional optimal control problems with fractional Bolza cost. Numer. Methods Partial Differ. Equ. **37**(2), 1693–1724 (2021)
14. Alizadeh, A., Effati, S.: An iterative approach for solving fractional optimal control problems. J. Vib. Control **24**(1), 18–36 (2018)
15. Jajarmi, A., Baleanu, D.: On the fractional optimal control problems with a general derivative operator. Asian J. Control **23**(2), 1062–1071 (2021)
16. Lotfi, A., Yousefi, S.A., Dehghan, M.: Numerical solution of a class of fractional optimal control problems via the Legendre orthonormal basis combined with the operational matrix and the Gauss quadrature rule. J. Comput. Appl. Math. **250**, 143–160 (2013)
17. Keshavarz, E., Ordokhani, Y., Razzaghi, M.: A numerical solution for fractional optimal control problems via Bernoulli polynomials. J. Vib. Control **22**(18), 3889–3903 (2016)

18. Ezz-Eldien, S.S., Doha, E.H., Baleanu, D., Bhrawy, A.H.: A numerical approach based on Legendre orthonormal polynomials for numerical solutions of fractional optimal control problems. J. Vib. Control **23**(1), 16–30 (2017)
19. Heydari, M.H., Hooshmandasl, M.R., Maalek Ghaini, F.M., Cattani, C.: Wavelets method for solving fractional optimal control problems. Appl. Math. Comput. **286**, 139–154 (2016)
20. Sahu, P.K., Saha Ray, S.: Comparison on wavelets techniques for solving fractional optimal control problems. J. Vib. Control **24**(6), 1185–1201 (2018)
21. Rabiei, K., Ordokhani, Y., Babolian, E.: The Boubaker polynomials and their application to solve fractional optimal control problems. Nonlinear Dyn. **88**(2), 1013–1026 (2017)
22. Abdelhakem, M., Moussa, H., Baleanu, D., El-Kady, M.: Shifted Chebyshev schemes for solving fractional optimal control problems. J. Vib. Control **25**(15), 2143–2150 (2019)
23. Yari, A.: Numerical solution for fractional optimal control problems by Hermite polynomials. J. Vib. Control **27**(5–6), 698–716 (2021)
24. Barikbin, Z., Keshavarz, E.: Solving fractional optimal control problems by new Bernoulli wavelets operational matrices. Optim. Control Appl. Methods **41**(4), 1188–1210 (2020)
25. Dehestani, H., Ordokhani, Y.: A spectral framework for the solution of fractional optimal control and variational problems involving Mittag-Leffler nonsingular kernel. J. Vib. Control **28**(3–4), 260–275 (2022)
26. Hassani, H., Tenreiro Machado, J.A., Mehrabi, S.: An optimization technique for solving a class of nonlinear fractional optimal control problems: application in cancer treatment. Appl. Math. Model. **93**, 868–884 (2021)
27. Hassani, H., Tenreiro Machado, J.A., Hosseini Asl, M.K., Dahaghin, M.S.: Numerical solution of nonlinear fractional optimal control problems using generalized Bernoulli polynomials. Optim. Control Appl. Methods **42**(4), 1045–1063 (2021)
28. Abd-Elhameed, W.M., Youssri, Y.H.: Spectral solutions for fractional differential equations via a novel Lucas operational matrix of fractional derivatives. Rom. J. Phys. **61**(5–6), 795–813 (2016)
29. Abd-Elhameed, W.M., Youssri, Y.H.: Generalized Lucas polynomial sequence approach for fractional differential equations. Nonlinear Dyn. **89**(2), 1341–1355 (2017)
30. Mokhtar, M.M., Mohamed, A.S.: Lucas polynomials semi-analytic solution for fractional multi-term initial value problems. Adv. Differ. Equ. **2019**(1), 1 (2019)
31. Oruç, Ö.: A new algorithm based on Lucas polynomials for approximate solution of 1D and 2D nonlinear generalized Benjamin–Bona–Mahony–Burgers equation. Comput. Math. Appl. **74**(12), 3042–3057 (2017)
32. Oruç, Ö.: A new numerical treatment based on Lucas polynomials for 1D and 2D sinh-Gordon equation. Commun. Nonlinear Sci. Numer. Simul. **57**, 14–25 (2018)
33. Dehestani, H., Ordokhani, Y., Razzaghi, M.: A novel direct method based on the Lucas multiwavelet functions for variable-order fractional reaction-diffusion and subdiffusion equations. Numer. Linear Algebra Appl. **28**(2), e2346 (2021)
34. Dehestani, H., Ordokhani, Y., Razzaghi, M.: Combination of Lucas wavelets with Legendre–Gauss quadrature for fractional Fredholm–Volterra integro-differential equations. J. Comput. Appl. Math. **382**, 113070 (2021)
35. Dehestani, H., Ordokhani, Y., Razzaghi, M.: Fractional Lucas optimization method for evaluating the approximate solution of the multi-dimensional fractional differential equations. Eng. Comput. **38**, 481–495 (2022)
36. Kumar, R., Koundal, R., Srivastava, K., Baleanu, D.: Normalized Lucas wavelets: an application to Lane-Emden and pantograph differential equations. Eur. Phys. J. Plus **135**(11), 1–24 (2020)
37. Ali, I., Haq, S., Nisar, K.S., Baleanu, D.: An efficient numerical scheme based on Lucas polynomials for the study of multidimensional Burgers-type equations. Adv. Differ. Equ. **2021**(1), 1 (2021)
38. Youssri, Y.H., Abd-Elhameed, W.M., Mohamed, A.S., Sayed, S.M.: Generalized Lucas polynomial sequence treatment of fractional pantograph differential equation. Int. J. Appl. Comput. Math. **7**(2), 1–16 (2021)
39. Kilbas, A.A., Srivastava, H.M., Trujillo, J.J.: Theory and Applications of Fractional Differential Equations, vol. 204. Elsevier, Amsterdam (2006)
40. Podlubny, I.: Fractional Differential Equations, vol. 198. Elsevier, Amsterdam (1999)
41. Agrawal, O.P.: Fractional variational calculus in terms of Riesz fractional derivatives. J. Phys. A, Math. Theor. **40**(24), 6287–6303 (2007)
42. Koshy, T.: Fibonacci and Lucas Numbers with Applications, vol. 2. Wiley, New York (2019)
43. Agrawal, O.P.: General formulation for the numerical solution of optimal control problems. Int. J. Control **50**(2), 627–638 (1989)
44. Mashayekhi, S., Razzaghi, M.: An approximate method for solving fractional optimal control problems by hybrid functions. J. Vib. Control **24**(9), 1621–1631 (2018)
45. Yonthanthum, W., Rattana, A., Razzaghi, M.: An approximate method for solving fractional optimal control problems by the hybrid of block-pulse functions and Taylor polynomials. Optim. Control Appl. Methods **39**(2), 873–887 (2018)
46. Akbarian, T., Keyanpour, M.: A new approach to the numerical solution of fractional order optimal control problems. Appl. Appl. Math. **8**(2), 523–534 (2013)
47. Singha, N., Nahak, C.: An efficient approximation technique for solving a class of fractional optimal control problems. J. Optim. Theory Appl. **174**(3), 785–802 (2017)
48. Dehestani, H., Ordokhani, Y., Razzaghi, M.: Fractional-order Bessel wavelet functions for solving variable order fractional optimal control problems with estimation error. Int. J. Syst. Sci. **51**(6), 1032–1052 (2020)
49. Heydari, M.H., Avazzadeh, Z.: A new wavelet method for variable-order fractional optimal control problems. Asian J. Control **20**(5), 1804–1817 (2018)
50. Inman, D.J.: Vibration with Control. Wiley, New York (2017)