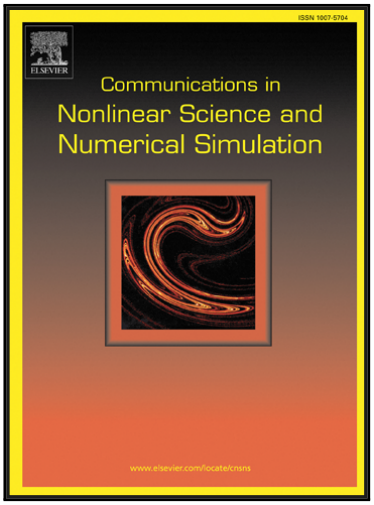

Direct transcription methods based on fractional integral approximation formulas for solving nonlinear fractional optimal control problems

Abubakar Bello Salati, Mostafa Shamsi, Delfim F. M. Torres

PII: S1007-5704(18)30155-2
DOI: 10.1016/j.cnsns.2018.05.011
Reference: CNSNS 4528

To appear in: *Communications in Nonlinear Science and Numerical Simulation*

Received date: 25 March 2018
Accepted date: 16 May 2018

Please cite this article as: Abubakar Bello Salati, Mostafa Shamsi, Delfim F. M. Torres, Direct transcription methods based on fractional integral approximation formulas for solving nonlinear fractional optimal control problems, *Communications in Nonlinear Science and Numerical Simulation* (2018), doi: 10.1016/j.cnsns.2018.05.011

Highlights

- A general class of fractional optimal control problems was considered, which contains the bang-bang, free final time and path constraints problems.
- Fractional integration matrices of Grunwald-Letnikov, trapezoidal and Simpson's formulas are derived.
- Using fractional integration matrix, the fractional optimal control problems are reduced to a finite-dimensional optimization problem.
- In order to improve the speed and accuracy, the gradient of objective function and the Jacobian of constraints are supplied for optimization solver.
- Numerical simulations are provided to illustrate the presented method on different types of fractional optimal control problems.

Direct transcription methods based on fractional integral approximation formulas for solving nonlinear fractional optimal control problems

Abubakar Bello Salati[a], Mostafa Shamsi[a,*], Delfim F. M. Torres[b]

[a] *Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, No. 424 Hafez Avenue, Tehran, Iran*
[b] *Department of Mathematics, Center for Research and Development in Mathematics and Applications (CIDMA), University of Aveiro, 3810-193 Aveiro, Portugal*

Abstract

This paper presents three direct methods based on Grünwald-Letnikov, trapezoidal and Simpson fractional integral formulas to solve fractional optimal control problems (FOCPs). At first, the fractional integral form of FOCP is considered, then the fractional integral is approximated by Grünwald-Letnikov, trapezoidal and Simpson formulas in a matrix approach. Thereafter, the performance index is approximated either by trapezoidal or Simpson quadrature. As a result, FOCP are reduced to nonlinear programming problems, which can be solved by many well-developed algorithms. To improve the efficiency of the presented method, the gradient of the objective function and the Jacobian of constraints are prepared in closed forms. It is pointed out that the implementation of the methods is simple and, due to the fact that there is no need to derive necessary conditions, the methods can be simply and quickly used to solve a wide class of FOCPs. The efficiency and reliability of the presented methods are assessed by ample numerical tests involving a free final time with path constraint FOCP, a bang-bang FOCP and an optimal control of a fractional-order HIV-immune system.

Keywords: Fractional optimal control, Direct numerical solution, Fractional integration matrix, Grünwald-Letnikov, trapezoidal and Simpson fractional integral formulas.
2010 MSC: 26A33, 49M25.

1. Introduction

Fractional calculus may be considered an old and yet an interesting topic [1]. It deals with the investigation of integrals and derivatives of an arbitrary order. At the initial stage, the fractional derivative was a mathematical tool without a tangible application. But presently, fractional calculus has wide applications in various disciplines like viscoelastic materials [2, 3], bioengineering applications [4, 5], signal processing [6, 7], mathematical finance [8, 9], control theory [10, 11] and many other areas [12, 13]. The nonlocal nature of the fractional calculus has given it a unique characteristic in modeling some complex systems with memory and hereditary properties that arise from physical, engineering and even economic processes [14, 15]. More elaborate details on the theory and

*Corresponding author.
Email addresses: a.salati@aut.ac.ir, absalati@gmail.com (Abubakar Bello Salati), m_shamsi@aut.ac.ir (Mostafa Shamsi), delfim@ua.pt (Delfim F. M. Torres)

applications of fractional calculus can be found in [2, 16, 17, 18].

Application of fractional calculus to optimal control problems has given birth to another new specialization, known as fractional optimal control [18, 19, 20, 21]. According to [22], a Fractional Optimal Control Problem (FOCP) is an optimal control problem in which the performance index and/or the dynamic equations contain at least one fractional derivative operator. In simple terms, FOCPs are a generalization of the integer-order optimal control problems, which are obtained by replacing integer-order derivatives with fractional ones. Recently, some interesting and real-life models of FOCPs have been presented by the researchers in [23, 24, 25, 26, 27].

Like integer-order optimal control problems, the numerical methods for FOCPs can be categorized into "direct" and "indirect" methods. With indirect methods, the solution is obtained by solving a fractional Hamiltonian boundary-value problem, which is derived from the optimality conditions. Accordingly, the first step in indirect methods is the derivation of first-order optimality conditions. On the other hand, direct methods do not rely on optimality conditions. In these approaches, FOCPs are solved by transcribing them into Nonlinear Programming problems (NLP). Thereafter, a NLP-solver is used to solve the resulting problem [28]. The indirect methods, in comparison with the direct, have some disadvantages, including, (i) difficulties in deriving the Hamiltonian boundary-value problem, especially for problems with path constraints, and (ii) sensitivity to initial guess for state functions as well as costate or adjoint functions. Due to the aforementioned reasons, direct methods are easier to employ than the indirect methods to solve both integer and fractional order optimal control problems.

As earlier works on the formulation, derivation of optimality conditions and direct/indirect solution schemes for FOCPs, we can refer to [29, 30, 31, 32, 33, 34, 35]. After these pioneer works, many other extensive researches have been done on the development of numerical methods for FOCPs. For instance, we can refer to Oustaloup recursive approximation [22], direct methods based on pseudo-state-space formulations of FOCP [36], spectral methods based on orthogonal polynomials and fractional operational matrices [37, 38, 39, 40, 41, 42, 43], Legendre multiwavelet collocation methods [44], direct methods based on Bernstein polynomials [45, 46, 47], nonstandard finite difference methods [48], linear programming approaches [49], integral fractional pseudospectral methods [50], direct methods based on Ritz's techniques [51, 52], the epsilon-Ritz method [53], direct methods based on hybrid block-pulse with other basis functions [54, 55], pseudospectral methods based on Legendre Müntz basis functions [56], dynamic Hamilton-Jacobi-Bellman methods [57], penalty and variational methods [58], control parameterization methods [59], differential and integral fractional pseudospectral methods [60], as well as other numerical techniques [61, 62, 63]. Efforts were also done to derive optimality conditions for special types of FOCPs, such as bang-bang FOCPs [64] and free final and terminal time problems [65, 66].

Direct transcription methods based on local methods, such as trapezoidal and Simpson methods, are widely used to solve integer-order optimal control problems and some softwares, such as SOCS [67] and ICLOCS [68], are developed based on these methods. However, it is somewhat surprising that these methods have not been yet fully explored to solve FOCPs. In this paper, we extend these methods to fractional order optimal control problems. For this purpose, the matrix form of Grünwald-Letnikov, trapezoidal and Simpson approximation

formulas for fractional integrals, which are called *fractional integral matrices*, are derived. These fractional integration matrices are used to reduce the FOCP to a NLP. Thereafter, the resulted NLP is solved by using a well-developed solver to obtain an approximate solution of the FOCP. Moreover, in order to increase the speed of the proposed direct methods, the exact gradient of the objective function and the Jacobian of constraints are derived and supplied to the NLP-solver. Finally, the reliability, efficiency and accuracy of the proposed direct method are demonstrated with four test problems, including a nonlinear and complex FOCP, a FOCP with path and terminal constraints, a bang-bang FOCP and an applied optimal control problem of a fractional order HIV-immune system with memory.

The paper is organized as follows. In Section 2, the notions of fractional derivative and integral are reviewed and three approximation methods for the fractional integral, in matrix form, are introduced. The considered formulation of fractional optimal control problems is stated in Section 3. The detailed implementation of direct methods is presented in Section 4. In Section 5, four numerical examples are provided to show the efficiency and reliability of the proposed methods. Finally, a conclusion is given in Section 6.

2. Definition and approximation of fractional integral and derivative

In this section, some definitions, together with three approximation formulas for fractional integrals, are reviewed. Moreover, for easy implementation, the matrix forms of these approximation formulas are derived.

Definition 2.1 (See, e.g., [16, 17]). *The left sided fractional integral with order $\alpha > 0$ of a given function $y(x)$, $x \in (a, b)$, is defined as*

$$ {}_0\mathcal{I}_x^{\alpha} y(x) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-t)^{\alpha-1} y(t)\, \mathrm{d}t, \tag{1} $$

where $\Gamma(\cdot)$ is Euler's gamma function.

Definition 2.2 (See, e.g., [16, 17]). *The left sided Caputo fractional derivative of order $\alpha \in (0, 1)$ of a function $y(x)$ is defined as*

$$ {}_0^C\mathcal{D}_x^{\alpha} y(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^x (x-t)^{-\alpha} \dot{y}(t)\, \mathrm{d}t. \tag{2} $$

The first step in developing our numerical methods is to approximate the fractional integral of a given function. For this purpose, we briefly review Grünwald-Letnikov (GL), trapezoidal (TR) and Simpson (SI) formulas for approximating the fractional integral.

2.1. The Grünwald-Letnikov formula for approximating a fractional integral

Let the fractional integral of function y be defined on the interval $[0, 1]$. To approximate the fractional integral of function y, the interval $[0, 1]$ is divided into n equal parts by the following mesh points:

$$ x_i := ih, \qquad i = 0, 1, \ldots, n, $$

where $h = 1/n$. The Grünwald-Letnikov (GL) formula for approximating the fractional integral of function y at $x = x_i$, can be expressed as

$$ {}_0\mathcal{I}_x^\alpha y(x)|_{x=x_i} \simeq \sum_{j=0}^{i} {}^{\mathrm{G}}\varpi_{ij}^{(\alpha)} y(x_j), \quad i = 0, \ldots, n, \tag{3} $$

where ${}^{\mathrm{G}}\varpi_{ij}^{(\alpha)}$ are the coefficients of the GL approximation formula, defined by

$$ {}^{\mathrm{G}}\varpi_{ij}^{(\alpha)} := \begin{cases} 0, & i = 0, \\ w_{i-j}^{(\alpha)}, & i > 0, \end{cases} \tag{4} $$

such that

$$ \omega_k^{(\alpha)} := (-1)^k \binom{-\alpha}{k} h^\alpha = \frac{(-1)^k \Gamma(1-\alpha)}{\Gamma(1+k)\Gamma(1-\alpha-k)} h^\alpha $$

(see [69, 70]). Let

$$ \mathbf{y} := [y(x_0), y(x_1), \ldots, y(x_n)]^T, \tag{5a} $$

$$ \hat{\mathbf{y}}^{(-\alpha)} := [{}_0\mathcal{I}_x^\alpha y(x_0), {}_0\mathcal{I}_x^\alpha y(x_1), \ldots, {}_0\mathcal{I}_x^\alpha y(x_n)]^T. \tag{5b} $$

Then, all the $n+1$ formulas in (3) can be written simultaneously in the following matrix form:

$$ \hat{\mathbf{y}}^{(-\alpha)} \simeq \mathbf{W}_{\mathrm{GL}}^{(\alpha)} \mathbf{y}, \tag{6} $$

where

$$ \mathbf{W}_{\mathrm{GL}}^{(\alpha)} := \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & 0 & \cdots & 0 \\ \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \vdots \\ \omega_n^{(\alpha)} & \omega_{n-1}^{(\alpha)} & \omega_{n-2}^{(\alpha)} & \cdots & \omega_0^{(\alpha)} \end{bmatrix}. \tag{7} $$

The matrix $\mathbf{W}_{\mathrm{GL}}^{(\alpha)}$ is called the *fractional GL integration matrix*.

2.2. The trapezoidal formula for approximating a fractional integral

Let $h = 1/n$ and $x_i := ih$, $i = 0, \ldots, n$. If the given function y is approximated by its piecewise linear interpolant based on the nodes x_i, $i = 0, \ldots, n$, and the fractional integral is applied to this approximation, then the trapezoidal (TR) formula is derived as

$$ {}_0\mathcal{I}_x^\alpha y(x)|_{x=x_i} \simeq \sum_{j=0}^{i} {}^{\mathrm{T}}\varpi_{ij}^{(\alpha)} y(x_j), \quad i = 0, 1, \ldots, n, \tag{8} $$

where

$$ {}^{\mathrm{T}}\varpi_{ij}^{(\alpha)} := \frac{h^\alpha}{\Gamma(\alpha+2)} \begin{cases} (i-1)^{\alpha+1} - (i-1-\alpha)i^\alpha, & j = 0, i > 0, \\ (i-j+1)^{\alpha+1} + (i-1-j)^{\alpha+1} - 2(i-j)^{\alpha+1}, & 1 \le j \le i-1, \\ 1, & j = i, \\ 0, & \text{otherwise} \end{cases} \tag{9} $$

(see [69]). If we set

$$\bar{a}_k^{(\alpha)} := \tfrac{h^\alpha}{\Gamma(\alpha+2)} \begin{cases} 0, & k=0, \\ (k-1)^{\alpha+1} - (k-1-\alpha)k^\alpha, & k>0, \end{cases}$$

$$\bar{b}_k^{(\alpha)} := \tfrac{h^\alpha}{\Gamma(\alpha+2)} \begin{cases} 1, & k=0, \\ (k+1)^{\alpha+1} + (k-1)^{\alpha+1} - 2k^{\alpha+1}, & k>0, \end{cases}$$

then, we can express the coefficients of the trapezoidal formula (8) as

$${}^{\mathrm{T}}\varpi_{ij}^{(\alpha)} := \begin{cases} \bar{a}_i^{(\alpha)}, & k=0, j>0, \\ \bar{b}_{i-j}^{(\alpha)}, & 1\le j\le i, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Considering (5), the trapezoidal formula (8) can be expressed in the following matrix form:

$$\hat{\mathbf{y}}^{(-\alpha)} \simeq \mathbf{W}_{\mathrm{TR}}^{(\alpha)}\,\mathbf{y}, \tag{11}$$

where $\mathbf{W}_{\mathrm{TR}}^{(\alpha)}$ is the *fractional TR integration matrix* defined by

$$\mathbf{W}_{\mathrm{TR}}^{(\alpha)} := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \bar{a}_1^{(\alpha)} & \bar{b}_0^{(\alpha)} & 0 & 0 & 0 & \cdots & 0 & 0 \\ \bar{a}_2^{(\alpha)} & \bar{b}_1^{(\alpha)} & \bar{b}_0^{(\alpha)} & 0 & 0 & \cdots & 0 & 0 \\ \bar{a}_3^{(\alpha)} & \bar{b}_2^{(\alpha)} & \bar{b}_1^{(\alpha)} & \bar{b}_0^{(\alpha)} & 0 & \cdots & 0 & 0 \\ \bar{a}_3^{(\alpha)} & \bar{b}_3^{(\alpha)} & \bar{b}_2^{(\alpha)} & \bar{b}_1^{(\alpha)} & \bar{b}_0^{(\alpha)} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \bar{a}_n^{(\alpha)} & \bar{b}_{n-2}^{(\alpha)} & \bar{b}_{n-3}^{(\alpha)} & \bar{b}_{n-4}^{(\alpha)} & \cdots & \bar{b}_1^{(\alpha)} & \bar{b}_0^{(\alpha)} & 0 \\ \bar{a}_n^{(\alpha)} & \bar{b}_{n-1}^{(\alpha)} & \bar{b}_{n-2}^{(\alpha)} & \bar{b}_{n-3}^{(\alpha)} & \cdots & \bar{b}_2^{(\alpha)} & \bar{b}_1^{(\alpha)} & \bar{b}_0^{(\alpha)} \end{bmatrix}. \tag{12}$$

2.3. The Simpson formula for approximating a fractional integral

Let n be even and the interval $[0,1]$ be divided into n parts by the mesh points $x_i := ih$, $i=0,\ldots,n$. Then the given function y on the intervals $[x_{2i}, x_{2i+2}]$, $i=0,\ldots,\frac{n}{2}-2$, can be approximated by its corresponding quadratic interpolation polynomial, i.e., y is approximated by a piecewise quadratic polynomial on the whole interval $[0,1]$. By replacing $y(t)$ with this piecewise quadratic polynomial approximation in (1), the Simpson formula for approximating the fractional integral of function y on the nodes x_i, $i=0,\ldots,n$, is derived as

$${}_0\mathcal{I}_x^\alpha y(x)\big|_{x=x_i} \simeq \sum_{j=0}^{i+1} {}^{\mathrm{S}}\varpi_{ij}^{(\alpha)} y(x_j), \quad i=0,1,\ldots,n, \tag{13}$$

where

$${}^{\mathrm{S}}\varpi_{ij}^{(\alpha)} := \begin{cases} \gamma_i^{(\alpha)}, & j=0, \\ \theta_{i-j+1}^{(\alpha)}, & j \text{ is odd}, \\ \mu_{i-j+2}^{(\alpha)}, & j \text{ is even}, \end{cases} \tag{14}$$

such that the parameters $\gamma_k^{(\alpha)}$, $\theta_k^{(\alpha)}$ and $\mu_k^{(\alpha)}$ are defined as

$$\gamma_k^{(\alpha)} := \frac{h^\alpha}{\Gamma(\alpha+3)} \begin{cases} 0, & k \le 0, \\ \frac{1}{2}(2\alpha+3)\alpha, & k = 1, \\ \lambda_{0,k}^{(\alpha)}, & 2 \le k, \end{cases}$$

$$\theta_k^{(\alpha)} := \frac{h^\alpha}{\Gamma(\alpha+3)} \begin{cases} 0, & k \le 0, \\ 2\alpha+2, & k = 1, \\ \lambda_{1,k}^{(\alpha)}, & 2 \le k, \end{cases}$$

$$\mu_k^{(\alpha)} := \frac{h^\alpha}{\Gamma(\alpha+3)} \begin{cases} 0, & k \le 0, \\ -\frac{1}{2}\alpha, & k = 1, \\ \lambda_{2,2}^{(\alpha)}, & k = 2, \\ \lambda_{2,3}^{(\alpha)} + \frac{1}{2}(2\alpha+3)\alpha, & k = 3, \\ \lambda_{2,k}^{(\alpha)} + \lambda_{0,k-2}^{(\alpha)}, & 4 \le k, \end{cases}$$

where

$$\lambda_{0,k}^{(\alpha)} := \frac{1}{2}\left[2\alpha^2 - (3k-6)\alpha + 2k^2 - 6k + 4\right]k^\alpha - \frac{1}{2}(2k+\alpha-2)(k-2)^{\alpha+1},$$
$$\lambda_{1,k}^{(\alpha)} := 2(k-2)^{\alpha+1}(k+\alpha) - 2k^{\alpha+1}(k-\alpha-2),$$
$$\lambda_{2,k}^{(\alpha)} := \frac{1}{2}k^{\alpha+1}(2k-\alpha-2) - \frac{1}{2}(k-2)^\alpha\left(2k^2 + (3\alpha-2)k + 2\alpha^2\right).$$

Considering (5), the Simpson formula (13) can be expressed in the following matrix form

$$\hat{\mathbf{y}}^{(-\alpha)} \simeq \mathbf{W}_{\text{SI}}^{(\alpha)}\,\mathbf{y}, \tag{15}$$

where $\mathbf{W}_{\text{SI}}^{(\alpha)}$ is the *fractional SI integration matrix*

$$\mathbf{W}_{\text{SI}}^{(\alpha)} := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \gamma_1^{(\alpha)} & \theta_1^{(\alpha)} & \mu_1^{(\alpha)} & 0 & 0 & \cdots & 0 & 0 \\ \gamma_2^{(\alpha)} & \theta_2^{(\alpha)} & \mu_2^{(\alpha)} & 0 & 0 & \cdots & 0 & 0 \\ \gamma_3^{(\alpha)} & \theta_3^{(\alpha)} & \mu_3^{(\alpha)} & \theta_1^{(\alpha)} & \mu_1^{(\alpha)} & \cdots & 0 & 0 \\ \gamma_4^{(\alpha)} & \theta_4^{(\alpha)} & \mu_4^{(\alpha)} & \theta_2^{(\alpha)} & \mu_2^{(\alpha)} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \gamma_{n-1}^{(\alpha)} & \theta_{n-1}^{(\alpha)} & \mu_{n-1}^{(\alpha)} & \theta_{n-3}^{(\alpha)} & \mu_{n-3}^{(\alpha)} & \cdots & \theta_1^{(\alpha)} & \mu_1^{(\alpha)} \\ \gamma_n^{(\alpha)} & \theta_n^{(\alpha)} & \mu_n^{(\alpha)} & \theta_{n-2}^{(\alpha)} & \mu_{n-2}^{(\alpha)} & \cdots & \theta_2^{(\alpha)} & \mu_2^{(\alpha)} \end{bmatrix}. \tag{16}$$

The Simpson and trapezoidal approximation formulas have been presented in [69]. Due to the fact that these matrix forms have some advantages in the implementation of our method, in this paper we derived the explicit forms of these matrices.

3. The fractional optimal control problem

In this work, we consider a general formulation for FOCPs, which is described as follows: find the optimal control $\mathbf{u}(t) = [u_1(t), \ldots, u_q(t)] \in \mathbb{R}^q$, the state $\mathbf{x}(t) = [x_1(t), \ldots, x_p(t)] \in \mathbb{R}^p$ and possibly the terminal time t_f that minimize the performance index

$$J[\mathbf{u}] = h\left(t_\mathrm{f}, \mathbf{x}(t_\mathrm{f})\right) + \int_0^{t_\mathrm{f}} g\left(\mathbf{x}(t), \mathbf{u}(t), t\right)\mathrm{d}t \tag{17a}$$

subject to the fractional dynamic system

$$ {}_0^{\mathrm{C}}\mathcal{D}_t^{\alpha}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad 0 < \alpha \leq 1, \tag{17b}$$

with the initial and boundary conditions

$$\mathbf{x}(0) = \mathbf{x}_0, \tag{17c}$$

$$\boldsymbol{\psi}(\mathbf{x}(t_\mathrm{f}), t_\mathrm{f}) = \mathbf{0}, \tag{17d}$$

and with the path constraint

$$\boldsymbol{\phi}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq \mathbf{0}. \tag{17e}$$

Here, functions h, g, $\mathbf{f}$, $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ are sufficiently continuously differentiable and defined by the following mappings:

$$\begin{aligned}
&h : \mathbb{R} \times \mathbb{R}^p \to \mathbb{R},\\
&g : \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R} \to \mathbb{R},\\
&\mathbf{f} : \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R} \to \mathbb{R}^p,\\
&\boldsymbol{\psi} : \mathbb{R}^p \times \mathbb{R} \to \mathbb{R}^{r_1}, \quad 0 \leq r_1 \leq q,\\
&\boldsymbol{\phi} : \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R} \to \mathbb{R}^{r_2}, \quad 0 \leq r_2.
\end{aligned}$$

Most FOCPs found in the literature, in various disciplines like engineering, control, signal processing and others, belong to the above general class of FOCPs [23, 24, 25].

For easy application of our numerical method, the time domain $[0, t_\mathrm{f}]$ is mapped to the canonical interval $[0, 1]$, by using the affine transformation $t \to \tau t_\mathrm{f}$. Applying this mapping and by noting that

$$ {}_0^{C}\mathcal{D}_t^{\alpha}\,\mathbf{x}(t) = (t_\mathrm{f})^{-\alpha}\ {}_0^{C}\mathcal{D}_\tau^{\alpha}\,\mathbf{x}(\tau),$$

the optimal control problem (17) is converted to the following form:

$$\begin{cases}
\min J[\mathbf{u}] = h\left(t_\mathrm{f}, \mathbf{x}(1)\right) + t_\mathrm{f}\displaystyle\int_0^1 g\left(\mathbf{x}(\tau), \mathbf{u}(\tau), t_\mathrm{f}\tau\right)\mathrm{d}\tau & \text{(18a)}\\
\quad \text{s.t.} \ \ {}_0^{\mathrm{C}}\mathcal{D}_t^{\alpha}\mathbf{x}(\tau) = (t_\mathrm{f})^{\alpha}\mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), t_\mathrm{f}\tau), & \text{(18b)}\\
\quad\quad \mathbf{x}(0) = \mathbf{x}_0, & \text{(18c)}\\
\quad\quad \boldsymbol{\psi}(\mathbf{x}(1), t_\mathrm{f}) = \mathbf{0}, & \text{(18d)}\\
\quad\quad \boldsymbol{\phi}(\mathbf{x}(\tau), \mathbf{u}(\tau), t_\mathrm{f}\tau) \leq \mathbf{0}. & \text{(18e)}
\end{cases}$$

It should to be noted that, after applying the mentioned transformation, the symbols of variables in (18) should be changed to new symbols. However, for the sake of simplicity, we shall retain the symbols already used.

4. The proposed method

We present a direct numerical approach to solve the FOCPs stated in (18). Let the interval $[0,1]$ be divided into n equal parts by the following mesh points:

$$\tau_k := kh,\ k = 0, \ldots, n,$$

where $h = 1/n$. Moreover, for the sake of simplicity, we consider the notations

$$\mathbf{x}_i := \mathbf{x}(\tau_i), \quad \mathbf{u}_i := \mathbf{u}(\tau_i), \quad \mathbf{f}_i := \mathbf{f}(\mathbf{x}(\tau_i), \mathbf{u}(\tau_i), t_{\mathrm{f}}\tau_i). \tag{19}$$

4.1. Discretization of the performance index

The performance index (18a) is discretized by a quadrature formula, such as the trapezoidal, Simpson or other rules. In this way, we have

$$J_n = h(t_{\mathrm{f}}, \mathbf{x}_n) + t_{\mathrm{f}} \sum_{i=0}^{n} w_i g(\mathbf{x}_i, \mathbf{u}_i, t_{\mathrm{f}}\tau_i), \tag{20}$$

where, in the case of using the trapezoidal rule,

$$w_0 := \frac{h}{2}, \quad w_i := h,\ i = 1, \ldots, n-1, \quad w_n := \frac{h}{2}$$

and, in the case of using Simpson rule,

$$w_0 := \frac{h}{3}, \quad w_{2i-1} := \frac{4h}{3},\ w_{2i} := \frac{2h}{3},\ i = 1, \ldots, \frac{n}{2} - 1, \quad w_n := \frac{h}{3}.$$

4.2. Discretization of the fractional dynamic equation

By applying the fractional integral operator ${}_0\mathcal{I}_\tau^\alpha$ to both sides of the fractional dynamic equation (18b), and by noting that ${}_0\mathcal{I}_\tau^\alpha \left({}_0^{\mathrm{C}}\mathcal{D}_\tau^\alpha \mathbf{x}(\tau)\right) = \mathbf{x}(\tau) - \mathbf{x}(0)$, the above fractional dynamic equations can be converted into the following fractional integral form:

$$\mathbf{x}(\tau) = \mathbf{x}(0) + (t_{\mathrm{f}})^\alpha \, {}_0\mathcal{I}_\tau^\alpha \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), t_{\mathrm{f}}\tau). \tag{21}$$

By collocating the above equation at $\tau = \tau_i$, $i = 1, \ldots, n$, we have

$$\mathbf{x}(\tau_i) = \mathbf{x}(0) + (t_{\mathrm{f}})^\alpha \left[{}_0\mathcal{I}_t^\alpha \left(\mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), t_{\mathrm{f}}\tau)\right)\right]_{\tau=\tau_i}. \tag{22}$$

Considering the initial condition (18c) and notations (19), and by applying GL/TR/SI approximation formulas (3)/(8)/(13), we can approximate the equation (22) as follows:

$$\mathbf{x}_i = \mathbf{x}_0 + (t_{\mathrm{f}})^\alpha \sum_{j=0}^{i} \varpi_{ij}^{(\alpha)} \mathbf{f}(\mathbf{x}_j, \mathbf{u}_j, t_{\mathrm{f}}\tau_j), \quad i = 1, 2, \ldots, n, \tag{23}$$

where $\varpi_{ij}^{(\alpha)}$ are the coefficients of GL/TR/SI approximation formula defined in (4)/(9)/(14). Similarly, we can discretize the path constraint (18e) as follows:

$$\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{u}_i, t_\mathrm{f}\tau_i) \leq \mathbf{0}, \quad i = 0, \ldots, n. \tag{24}$$

Moreover, the terminal condition (18d) can be discretized as

$$\boldsymbol{\psi}(\mathbf{x}_n, t_\mathrm{f}) = \mathbf{0}. \tag{25}$$

In summary, the FOCP (18) is transcribed into the following nonlinear programming problem (NLP):

$$\begin{cases} \min J_n = h(t_\mathrm{f}, \mathbf{x}_n) + t_\mathrm{f} \sum_{i=0}^{n} w_i g(\mathbf{x}_i, \mathbf{u}_i, t_\mathrm{f}\tau_i) & \text{(26a)} \\ \text{s.t. } \mathbf{x}_i = \mathbf{x}_0 + (t_\mathrm{f})^\alpha \sum_{j=0}^{i} \varpi_{ij}^{(\alpha)} \mathbf{f}\left(\mathbf{x}_j, \mathbf{u}_j, t_\mathrm{f}\tau_j\right), \quad i = 0, 1, \ldots, n, & \text{(26b)} \\ \boldsymbol{\psi}(\mathbf{x}_n, t_\mathrm{f}) = \mathbf{0}, & \text{(26c)} \\ \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{u}_i, t_\mathrm{f}\tau_i) \leq \mathbf{0}, \quad i = 0, \ldots, n. & \text{(26d)} \end{cases}$$

It should be remembered that the decision variables of the above optimization problem are: $\mathbf{x}_i$, $i = 0, \ldots, n$, $\mathbf{u}_i$, $i = 0, \ldots, n$, and maybe t_f. By utilizing an optimization solver, we can solve the optimization problem (26) and find the optimal value of the decision variables, which are the approximations of state and control functions at the points τ_i, $i = 0, \ldots, n$.

4.2.1. Reformat of the resulted optimization problem to the classical form

Traditionally, in an NLP, it is preferred that the decision variables are placed in a vector. This helps to analyze and utilize a solver for solving the problem. Here, we reformulate the NLP (26) into the standard form, where the decision variables are collected in a vector $\mathbf{z}$. For fixed final time problems, the vector of all decision variables is

$$\mathbf{z} := \left[\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \ldots, \mathbf{x}_n, \mathbf{u}_n\right]^T$$

and for free final time problems

$$\mathbf{z} := \left[\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1, \ldots, \mathbf{x}_n, \mathbf{u}_n, t_\mathrm{f}\right]^T.$$

If we define new functions $\widehat{h}$ and $\widehat{g}$ as

$$\widehat{h}(\mathbf{z}) := h(t_\mathrm{f}, \mathbf{x}_n), \quad \widehat{\mathbf{g}}(\mathbf{z}) := \begin{bmatrix} g(\mathbf{x}_0, \mathbf{u}_0, t_\mathrm{f}\tau_0) \\ g(\mathbf{x}_1, \mathbf{u}_1, t_\mathrm{f}\tau_1) \\ \vdots \\ g(\mathbf{x}_n, \mathbf{u}_n, t_\mathrm{f}\tau_n) \end{bmatrix}, \tag{27}$$

then we can express the objective function (26a) as

$$J_n(\mathbf{z}) := \widehat{h}(\mathbf{z}) + t_\mathrm{f} \mathbf{w}^T \widehat{\mathbf{g}}(\mathbf{z}),$$

where $\mathbf{w}$ is the vector of quadrature weights in (20), i.e.,

$$\mathbf{w} := [w_0, \ldots, w_n]^T. \tag{28}$$

It should be noted that, in case of using trapezoidal quadrature, we have

$$\mathbf{w} = \mathbf{w}_{\text{TR}} := \left[\tfrac{h}{2}, h, \ldots, h, \tfrac{h}{2}\right] \tag{29}$$

and, for Simpson quadrature, we have

$$\mathbf{w} = \mathbf{w}_{\text{SI}} := \left[\tfrac{h}{3}, \tfrac{4h}{3}, \tfrac{2h}{3}, \tfrac{4h}{3}, \tfrac{2h}{3}, \ldots, \tfrac{4h}{3}, \tfrac{2h}{3}, \tfrac{h}{3}\right]. \tag{30}$$

The constraints (26b), for $i = 0, 1, \ldots, n$, can be represented in the following matrix form:

$$\mathbf{X} = \mathbf{X}_0 + (t_{\text{f}})^{\alpha} \mathbf{F} \left[\mathbf{W}^{(\alpha)}\right]^T, \tag{31}$$

where $\mathbf{X}$, $\mathbf{X}_0$ and $\mathbf{F}$ are $p \times (n+1)$ matrices defined as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 & \ldots & \mathbf{x}_n \end{bmatrix}, \quad \mathbf{X}_0 = \begin{bmatrix} \mathbf{x}_0 & \ldots & \mathbf{x}_0 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0, t_{\text{f}}\tau_0) & \ldots & \mathbf{f}(\mathbf{x}_n, \mathbf{u}_n, t_{\text{f}}\tau_n) \end{bmatrix}, \tag{32}$$

and $\mathbf{W}^{(\alpha)}$ is the fractional GL/TR/SI integration matrix defined in (7)/(12)/(16). The matrix equation (31) can be reformulated as follows:

$$\mathbf{vec}\left(\mathbf{X}\right) = \mathbf{vec}\left(\mathbf{X}_0\right) + {t_{\text{f}}}^{\alpha} \mathbf{vec}\left(\mathbf{F}\left[\mathbf{W}^{(\alpha)}\right]^T\right), \tag{33}$$

where **vec** is the vectorization operator, which converts a matrix into a column vector by stacking the columns of the matrix on the top of each other.

Let $\mathbf{I}_k$ be the identity matrix of order k and the matrix $\mathbf{R}$ be defined as

$$\mathbf{R} := \begin{cases} [\mathbf{I}_p \mid \mathbf{0}_{p\times q}], & \text{if } {t_{\text{f}}}^{\alpha} \text{ be fixed,} \\ [\mathbf{I}_p \mid \mathbf{0}_{p\times q} \mid 0], & \text{if } {t_{\text{f}}}^{\alpha} \text{ be free.} \end{cases}$$

Then, we can express $\mathbf{vec}\left(\mathbf{X}\right)$ based on $\mathbf{z}$ as

$$\mathbf{vec}\left(\mathbf{X}\right) = \left(\mathbf{I}_n \otimes \mathbf{R}\right)\mathbf{z}, \tag{34}$$

where $\otimes$ denotes the Kronecker product (or tensor product). Moreover, if $\mathbf{1}_n$ is a row n-vector whose entries are all equal to 1, then we can express $\text{vec}(\mathbf{X}_0)$ based on $\mathbf{z}$ as

$$\mathbf{vec}(\mathbf{X}_0) = \mathbf{x}_0 \otimes \mathbf{1}_n. \tag{35}$$

To express $\mathbf{vec}\left(\mathbf{F}\left[\mathbf{W}^{(\alpha)}\right]^T\right)$ based on $\mathbf{z}$, we use Theorem 13.26 in [71], which states that for any three matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{E}$ for which the matrix product $\mathbf{ABE}$ is defined, one has

$$\mathbf{vec}(\mathbf{ABE}) = (\mathbf{E}^T \otimes \mathbf{A})\mathbf{vec}(\mathbf{B}).$$

Now, by using the above equation and by noting that $\mathbf{F}\left[\mathbf{W}^{(\alpha)}\right]^T = \mathbf{I}_p\mathbf{F}\left[\mathbf{W}^{(\alpha)}\right]^T$, we conclude that

$$\mathbf{vec}\left(\mathbf{F}\left[\mathbf{W}^{(\alpha)}\right]^T\right) = \left(\mathbf{W}^{(\alpha)} \otimes \mathbf{I}_p\right)\mathbf{vec}\left(\mathbf{F}\right). \tag{36}$$

In view of (32), we can write

$$\mathbf{vec}\left(\mathbf{F}\right) = \widehat{\mathbf{f}}(\mathbf{z}) := \begin{bmatrix} \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0, t_{\mathrm{f}}\tau_0) \\ \mathbf{f}(\mathbf{x}_1, \mathbf{u}_1, t_{\mathrm{f}}\tau_1) \\ \vdots \\ \mathbf{f}(\mathbf{x}_n, \mathbf{u}_n, t_{\mathrm{f}}\tau_n) \end{bmatrix}. \tag{37}$$

Using the above notation and equations (34)–(36), we can write (33) as

$$\mathbf{c}(\mathbf{z}) := \left(\mathbf{I}_n \otimes \mathbf{R}\right)\mathbf{z} - \mathbf{x}_0 \otimes \mathbf{1}_n - (t_{\mathrm{f}})^{\alpha}\left(\mathbf{W}^{(\alpha)} \otimes \mathbf{I}_p\right)\widehat{\mathbf{f}}(\mathbf{z}) = \mathbf{0}.$$

To replace the constraints (26c) and (26d) by other constraints based on $\mathbf{z}$, we define functions $\widehat{\boldsymbol{\phi}}$ and $\widehat{\boldsymbol{\psi}}$ as

$$\widehat{\boldsymbol{\phi}}(\mathbf{z}) = \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}_0, \mathbf{u}_0, t_{\mathrm{f}}\tau_0) \\ \boldsymbol{\phi}(\mathbf{x}_1, \mathbf{u}_1, t_{\mathrm{f}}\tau_1) \\ \vdots \\ \boldsymbol{\phi}(\mathbf{x}_n, \mathbf{u}_n, t_{\mathrm{f}}\tau_n) \end{bmatrix}, \qquad \widehat{\boldsymbol{\psi}}(\mathbf{z}) = \boldsymbol{\psi}(\mathbf{x}_n, t_{\mathrm{f}}).$$

Then the constraints (26c) and (26d) can be expressed as

$$\widehat{\boldsymbol{\psi}}(\mathbf{z}) = \mathbf{0}, \qquad \widehat{\boldsymbol{\phi}}(\mathbf{z}) \leq \mathbf{0}.$$

The above results are summarized in the following theorem.

Theorem 4.1. *The optimization problem* (26) *can be expressed in the standard form*

$$\begin{cases} \min \widehat{J}_n(\mathbf{z}) = \widehat{h}(\mathbf{z}) + t_{\mathrm{f}}\mathbf{w}^T\widehat{\mathbf{g}}(\mathbf{z}) & \text{(38a)} \\ \quad s.t. \quad \mathbf{c}(\mathbf{z}) := \left(\mathbf{I}_n \otimes \mathbf{R}\right)\mathbf{z} - \mathbf{x}_0 \otimes \mathbf{1}_n - (t_{\mathrm{f}})^{\alpha}\left(\mathbf{W}^{(\alpha)} \otimes \mathbf{I}_p\right)\widehat{\mathbf{f}}(\mathbf{z}) = \mathbf{0}, & \text{(38b)} \\ \qquad\quad \widehat{\boldsymbol{\psi}}(\mathbf{z}) = \mathbf{0}, & \text{(38c)} \\ \qquad\quad \widehat{\boldsymbol{\phi}}(\mathbf{z}) \leq \mathbf{0}. & \text{(38d)} \end{cases}$$

In summary, the solution of the FOCP (17) is reduced to the solution of the NLP (38). In this NLP, $\mathbf{w}$ can be $\mathbf{w}_{\mathrm{TR}}$ or $\mathbf{w}_{\mathrm{SI}}$ defined in (29) and (30), respectively. Moreover, the fractional integration matrix $\mathbf{W}^{(\alpha)}$ can be $\mathbf{W}^{(\alpha)}_{\mathrm{GL}}$, $\mathbf{W}^{(\alpha)}_{\mathrm{TR}}$ or $\mathbf{W}^{(\alpha)}_{\mathrm{SI}}$, defined in (7), (12) and (16), respectively. We categorize our methods as follows:

- If $\mathbf{w} = \mathbf{w}_{\mathrm{GL}}$ and $\mathbf{W}^{(\alpha)} = \mathbf{W}^{(\alpha)}_{\mathrm{GL}}$, then the method is called the "direct GL method".
- If $\mathbf{w} = \mathbf{w}_{\mathrm{TR}}$ and $\mathbf{W}^{(\alpha)} = \mathbf{W}^{(\alpha)}_{\mathrm{TR}}$, then the method is called the "direct TR method".
- If $\mathbf{w} = \mathbf{w}_{\mathrm{SI}}$ and $\mathbf{W}^{(\alpha)} = \mathbf{W}^{(\alpha)}_{\mathrm{SI}}$, then the method is called the "direct SI method".

4.2.2. Gradient of the objective function and Jacobian of the constraints

The speed of the proposed method depends on the speed one solves the NLP (38). On the other hand, a crucial task in solving NLPs is computing the first derivatives of the objective and constraint functions (gradients of the objective function and Jacobians of the constraints). Many NLP-solvers allow users to supply the exact first derivatives. If the derivatives are not provided, then the solvers approximate them numerically, by finite difference formulas. However, the use of first derivative approximations may seriously degrade the performance and convergence of the NLP-solver. Thus, it is highly recommended that at least the exact gradient of the objective function and the Jacobian of the constraints are provided by the user. These can make a great improvement in the convergence, accuracy and computation time of the NLP-solver. In what follows, we detail the closed forms for the gradient of the objective function and the Jacobian of constraints.

For free time problems, the gradient of the objective function can be obtained as

$$\nabla \widehat{J}_n(\mathbf{z}) = \begin{bmatrix} \partial \widehat{J}_n(\mathbf{z})/\partial \mathbf{x}_0 \\ \partial \widehat{J}_n(\mathbf{z})/\partial \mathbf{u}_0 \\ \partial \widehat{J}_n(\mathbf{z})/\partial \mathbf{x}_1 \\ \partial \widehat{J}_n(\mathbf{z})/\partial \mathbf{u}_1 \\ \vdots \\ \partial \widehat{J}_n(\mathbf{z})/\partial \mathbf{x}_n \\ \partial \widehat{J}_n(\mathbf{z})/\partial \mathbf{u}_n \\ \partial \widehat{J}_n(\mathbf{z})/\partial t_{\mathrm{f}} \end{bmatrix} = \begin{bmatrix} t_{\mathrm{f}}\, w_0\, g_{\mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0, t_{\mathrm{f}}\tau_0) \\ t_{\mathrm{f}}\, w_0\, g_{\mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0, t_{\mathrm{f}}\tau_0) \\ t_{\mathrm{f}}\, w_1\, g_{\mathbf{x}}(\mathbf{x}_1, \mathbf{u}_1, t_{\mathrm{f}}\tau_1) \\ t_{\mathrm{f}}\, w_1\, g_{\mathbf{u}}(\mathbf{x}_1, \mathbf{u}_1, t_{\mathrm{f}}\tau_1) \\ \vdots \\ h_{\mathbf{x}}(t_{\mathrm{f}}, \mathbf{x}_n) + t_{\mathrm{f}} w_n g_{\mathbf{x}}(\mathbf{x}_n, \mathbf{u}_n, t_{\mathrm{f}}\tau_n) \\ t_{\mathrm{f}} w_n g_{\mathbf{u}}(\mathbf{x}_n, \mathbf{u}_n, t_{\mathrm{f}}\tau_n) \\ h_t(t_{\mathrm{f}}, \mathbf{x}_n) + \sum_{i=0}^{n} w_i \big[g(\mathbf{x}_i, \mathbf{u}_i, t_{\mathrm{f}}\tau_i) + t_{\mathrm{f}}\tau_i\, g_t(\mathbf{x}_i, \mathbf{u}_i, t_{\mathrm{f}}\tau_i) \big] \end{bmatrix}. \tag{39}$$

For fixed final time problems, the gradient is the same as above, except that the last row is removed.

The Jacobian of constraints (38b), for free final time problems, can be derived as

$$\nabla \mathbf{c}(\mathbf{z}) = \Big[\; (\mathbf{I}_n \otimes \mathbf{R}) - (t_{\mathrm{f}})^{\alpha} \left(\mathbf{W}^T \otimes \mathbf{I}_p\right) \nabla \widehat{\mathbf{f}}\,(\mathbf{z}) \;\Big|\; (t_{\mathrm{f}})^{\alpha-1} \left(\mathbf{W}^T \otimes \mathbf{I}_p\right) \left(-\alpha \widehat{\mathbf{f}}\,(\mathbf{z}) - t_{\mathrm{f}} \widehat{\mathbf{f}}_{t_{\mathrm{f}}}\,(\mathbf{z})\right) \; \Big], \tag{40}$$

such that

$$\nabla \widehat{\mathbf{f}}\,(\mathbf{z}) := \begin{bmatrix} [\mathbf{f}_{\mathbf{x}}]_0 & [\mathbf{f}_{\mathbf{u}}]_0 & & & & \\ & & [\mathbf{f}_{\mathbf{x}}]_1 & [\mathbf{f}_{\mathbf{u}}]_1 & & \\ & & & & \ddots & \\ & & & & [\mathbf{f}_{\mathbf{x}}]_n & [\mathbf{f}_{\mathbf{u}}]_n \end{bmatrix}, \quad \widehat{\mathbf{f}}_{t_{\mathrm{f}}}\,(\mathbf{z}) := \begin{bmatrix} \tau_0 \mathbf{f}_t(\mathbf{x}_0, \mathbf{u}_0, t_{\mathrm{f}}\tau_0) \\ \tau_1 \mathbf{f}_t(\mathbf{x}_1, \mathbf{u}_1, t_{\mathrm{f}}\tau_1) \\ \vdots \\ \tau_n \mathbf{f}_t(\mathbf{x}_n, \mathbf{u}_n, t_{\mathrm{f}}\tau_n) \end{bmatrix},$$

where $[\mathbf{f}_{\mathbf{x}}]_i$ and $[\mathbf{f}_{\mathbf{u}}]_i$ are the p-vectors defined as

$$[\mathbf{f}_{\mathbf{x}}]_i := \mathbf{f}_{\mathbf{x}}(\mathbf{x}_i, \mathbf{u}_i, t_{\mathrm{f}}\tau_i), \quad [\mathbf{f}_{\mathbf{u}}]_i := \mathbf{f}_{\mathbf{u}}(\mathbf{x}_i, \mathbf{u}_i, t_{\mathrm{f}}\tau_i).$$

In the case of fixed final time problems, the Jacobian of constraints (38b) is obtained as

$$\nabla \mathbf{c}(\mathbf{z}) = \Big[\; (\mathbf{I}_n \otimes \mathbf{R}) - (t_{\mathrm{f}})^{\alpha} \left(\mathbf{W}^T \otimes \mathbf{I}_p\right) \nabla \widehat{\mathbf{f}}\,(\mathbf{z}) \; \Big]. \tag{41}$$

The Jacobian of constraint (38c), for fixed final time FOCPs, is derived as

$$\nabla\widehat{\boldsymbol{\psi}}(\mathbf{z}) = \left[\begin{array}{c|c|c} \mathbf{0}_{r_1\times(n-1)(p+q)} & \boldsymbol{\psi}_{\mathbf{x}}(\mathbf{x}_n, t_{\mathrm{f}}) & \mathbf{0}_{r_1\times q} \end{array}\right] \tag{42}$$

while for free final time FOCPs is given by

$$\nabla\widehat{\boldsymbol{\psi}}(\mathbf{z}) = \left[\begin{array}{c|c|c|c} \mathbf{0}_{r_1\times(n-1)(p+q)} & \boldsymbol{\psi}_{\mathbf{x}}(\mathbf{x}_n, t_{\mathrm{f}}) & \mathbf{0}_{r_1\times q} & \boldsymbol{\psi}_t(\mathbf{x}_n, t_{\mathrm{f}}) \end{array}\right]. \tag{43}$$

In case of free final time FOCPs, the Jacobian of the inequality constraints (38d) is obtained as

$$\nabla\widehat{\boldsymbol{\phi}}(\mathbf{z}) = \begin{bmatrix} [\boldsymbol{\phi}_{\mathbf{x}}]_0 & [\boldsymbol{\phi}_{\mathbf{u}}]_0 & & & & \tau_0\boldsymbol{\phi}_t(\mathbf{x}_0, \mathbf{u}_0, t_{\mathrm{f}}\tau_0) \\ & & [\boldsymbol{\phi}_{\mathbf{x}}]_1 & [\boldsymbol{\phi}_{\mathbf{u}}]_1 & & \tau_1\boldsymbol{\phi}_t(\mathbf{x}_1, \mathbf{u}_1, t_{\mathrm{f}}\tau_1) \\ & & & & \ddots & \\ & & & & [\boldsymbol{\phi}_{\mathbf{x}}]_n & [\boldsymbol{\phi}_{\mathbf{u}}]_n \quad \tau_n\boldsymbol{\phi}_t(\mathbf{x}_n, \mathbf{u}_n, t_{\mathrm{f}}\tau_n) \end{bmatrix}, \tag{44}$$

where $[\boldsymbol{\phi}_{\mathbf{x}}]_i$ and $[\boldsymbol{\phi}_{\mathbf{u}}]_i$ are the r_2-vectors defined by

$$[\boldsymbol{\phi}_{\mathbf{x}}]_i := \boldsymbol{\phi}_{\mathbf{x}}(\mathbf{x}_i, \mathbf{u}_i, t_{\mathrm{f}}\tau_i), \quad [\boldsymbol{\phi}_{\mathbf{u}}]_i := \boldsymbol{\phi}_{\mathbf{u}}(\mathbf{x}_i, \mathbf{u}_i, t_{\mathrm{f}}\tau_i), \quad i = 0, \ldots, n.$$

In case of fixed final time problems, the last column of the Jacobian matrix (44) is removed.

As we can see, the partial derivatives of g, $\mathbf{f}$, $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ with respect to $\mathbf{x}$ and $\mathbf{u}$ are needed to provide the gradient of the objective function and the Jacobian of the constraints. We can use symbolic computation to supply these partial derivatives.

5. Numerical examples

We now illustrate the direct methods presented in Section 4 through numerical experiments. We have implemented the direct GL, TR and SI methods using MATLAB on a 3.5 GHz Core i7 personal computer with 8 GB of RAM. Moreover, for solving the nonlinear programming problem (38), the solver IPOPT [72] was used, which is based on an interior-point algorithm. In IPOPT, we can adjust the accuracy of solution by the input parameter `tolrfun`. In our numerical experiments, we set `tolrfun`=10^{-12}.

We consider four nontrivial examples. The first two examples are new while the last two examples have been investigated before in the literature. At each example, the capability of the proposed direct methods is highlighted. For the first example, which has an exact solution, the accuracy of our methods, and the required CPU time, are assessed. In the second example, the ability of the method in solving a free final time FOCP with path constraints is investigated. In the third example, we show that the proposed methods are also suitable to deal with bang-bang optimal control problems. In the last example, we assess the ability of our direct methods in solving a practical and challenging problem of HIV.

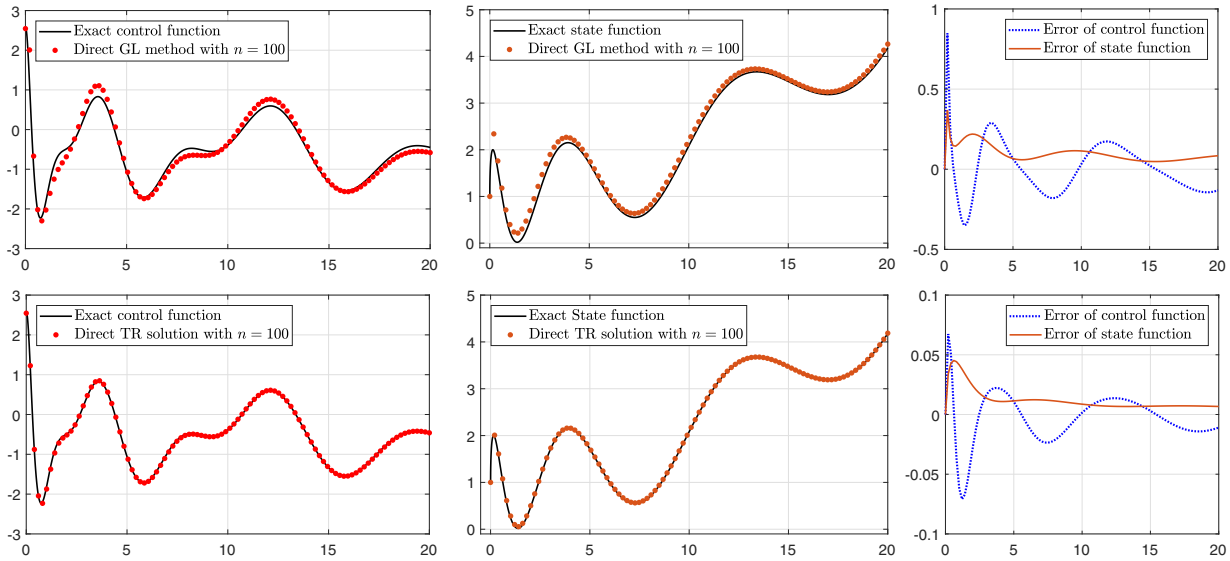

Figure 1: (Example 1, with $\alpha = 0.5$) Comparison of the exact and obtained solutions and error functions. Above: Direct GL method with $n = 100$. Below: Direct TR method with $n = 100$.

5.1. Example 1: A FOCP with exact solution

In this example, we consider the following nonlinear FOCP:

$$\begin{aligned} \min J[\mathbf{u}] &= \int_0^{20} \left[1 - \left(x - 0.01\,t^2 - 1\right)^2 + u - 2\sqrt{\pi} J_0\left(4\sqrt{t}\right)\right]^2 \mathrm{d}t \\ \text{s.t. } & {}_0^{\mathrm{C}}\mathcal{D}_t^{\alpha} x(t) = -\left(x(t) - 0.01\,t^2 - 1\right)^2 + u(t) + 1 + \frac{2\,t^{3/2}}{75\sqrt{\pi}}, \\ & x(0) = 1, \\ & x(20) = 5 + \sin(8\sqrt{5}). \end{aligned} \tag{45}$$

The exact solution for $\alpha = 1/2$ is

$$\begin{aligned} u_{\mathrm{ex}}(t) &= -\cos^2\left(4\sqrt{t}\right) + 2\sqrt{\pi} J_0\left(4\sqrt{t}\right), \\ x_{\mathrm{ex}}(t) &= \sin\left(4\sqrt{t}\right) + 0.01\,t^2 + 1, \end{aligned}$$

where J_0 is the first kind Bessel function of order zero [73]. The domain of this problem is large and the control and state functions have algebraic singularities and an oscillating behavior. Consequently, this example is a suitable test problem for assessing the accuracy and efficiency of the presented methods.

By applying the direct GL and TR methods with $n = 100$, the obtained control, state and error functions are plotted in Figure 1. We see that, with $n = 100$, the accuracy of the direct GL method is not satisfactory. However, the direct TR method generates a solution with reasonable accuracy. The results of the direct GL, TR and SI methods, with various values of n, are reported in Table 1. In this table, to show the impact of using the gradient of the objective function and the Jacobian of the constraints, we compare the CPU times in the two cases "NPD" and "PD". By "NPD" we refer to the CPU time of the methods without providing derivatives

Table 1: (Example 1, with $\alpha = 0.5$) Obtained CPU times and the ℓ^2 norm of the errors by direct GL, TR and SI methods for various values of n. All CPU times are in seconds.

	Direct GL method				Direct TR method				Direct SI method			
n	CPU time (s)		ERROR		CPU time (s)		ERROR		CPU time (s)		ERROR	
	NPD	PD	$E_n(u)$	$E_n(x)$	NPD	PD	$E_n(u)$	$E_n(x)$	NPD	PD	$E_n(u)$	$E_n(x)$
100	8.9	0.5	1.68e-1	1.11e-1	6.3	0.6	2.07e-2	1.48e-2	6.4	0.5	8.99e-4	5.60e-4
200	22.6	0.7	9.19e-2	5.71e-2	17.9	0.8	5.21e-3	3.71e-3	12.7	0.8	7.66e-5	4.91e-5
300	23.2	1.3	6.37e-2	3.94e-2	22.7	1.5	2.32e-3	1.65e-3	25.2	1.1	1.80e-5	1.18e-5
400	33.6	2.1	4.88e-2	3.04e-2	34.4	2.2	1.31e-3	9.31e-4	35.6	2.1	6.48e-6	4.30e-6
500	53.1	3.0	3.95e-2	2.48e-2	47.6	3.6	8.39e-4	5.96e-4	50.8	3.2	2.94e-6	1.97e-6
600	76.5	4.4	3.32e-2	2.11e-2	63.6	5.0	5.84e-4	4.15e-4	76.5	5.1	1.54e-6	1.04e-6
700	84.9	6.5	2.87e-2	1.84e-2	94.3	6.9	4.29e-4	3.05e-4	100.6	7.7	8.98e-7	6.04e-7
800	130.1	8.7	2.52e-2	1.63e-2	110.4	8.7	3.29e-4	2.34e-4	107.5	9.2	5.62e-7	3.80e-7
900	163.2	12.3	2.25e-2	1.47e-2	150.9	11.9	2.60e-4	1.85e-4	188.2	13.4	3.70e-7	2.50e-7
1000	228.1	16.1	2.03e-2	1.34e-2	210.8	14.7	2.11e-4	1.50e-4	228.9	17.0	2.56e-7	1.73e-7
1100	287.9	23.9	1.85e-2	1.23e-2	304.3	20.0	1.74e-4	1.24e-4	364.6	19.7	1.83e-7	1.24e-7
1200	397.0	30.8	1.70e-2	1.14e-2	377.3	29.6	1.46e-4	1.04e-4	471.7	22.6	1.35e-7	9.15e-8
1300	485.5	39.4	1.57e-2	1.06e-2	454.4	32.6	1.25e-4	8.87e-5	516.7	34.6	1.02e-7	6.91e-8
1400	543.9	42.0	1.46e-2	9.89e-3	561.0	37.4	1.08e-4	7.65e-5	602.4	42.3	7.84e-8	5.34e-8
1500	654.4	46.7	1.36e-2	9.29e-3	676.9	46.2	9.38e-5	6.67e-5	652.8	49.1	6.15e-8	4.20e-8
1600	707.8	57.3	1.28e-2	8.77e-3	710.2	51.0	8.30e-5	5.90e-5	725.6	55.6	4.92e-8	3.36e-7
1700	>800	67.2	1.20e-2	8.31e-3	>800	54.5	7.30e-5	5.19e-5	>800	70.2	4.00e-8	2.73e-8
1800	>800	72.7	1.14e-2	7.89e-3	>800	69.7	6.52e-5	4.63e-5	>800	76.0	3.31e-8	2.26e-8
1900	>800	90.5	1.08e-2	7.52e-3	>800	74.5	5.84e-5	4.15e-5	>800	81.3	2.79e-8	1.90e-6
2000	>800	96.4	1.03e-2	7.18e-3	>800	97.6	5.26e-5	3.74e-5	>800	96.6	2.37e-8	1.61e-8

of the objective and constraint functions. In contrast, "PD" refers to the CPU time when the first derivatives of the objective and constraint functions are supplied to the direct methods. In addition, in Table 1, $E_n(u)$ and $E_n(x)$ are the ℓ_2 norm of the error in control and state, i.e.,

$$E_n(u) := \left(\frac{1}{n}\sum_{i=1}^{n}(u_i - u_{\text{ex}}(\tau_i))^2\right)^{\frac{1}{2}}, \qquad E_n(x) := \left(\frac{1}{n}\sum_{i=1}^{n}(x_i - x_{\text{ex}}(\tau_i))^2\right)^{\frac{1}{2}},$$

where u_i and x_i are the obtained control and state functions in $\tau = \tau_i$ by the presented methods with n nodes. Figure 2 visualizes the results in Table 1. In Figure 2 (left), the `loglog` plot of $E_n(u)$ and $E_n(x)$ versus n are plotted. Moreover, the regression lines and their slopes are reported too. In Figure 2 (right), the CPU times of the direct GL, TR and SI methods are plotted for the two cases NPD and PD. From Table 1 and Figure 2, we can see that by supplying the gradient of the objective function and the Jacobian of the constraints, the CPU time is significantly reduced. Moreover, we find out that the computational times for direct GL, TR and SI methods are almost the same. However, the direct SI method is more accurate than the direct TR and GL methods. In view of Figure 2 (left), the accuracy order of the direct GL, TR and SI methods, in solving the fractional optimal control problem (45), is $\mathcal{O}(h)$, $\mathcal{O}(h^2)$ and $\mathcal{O}(h^{2.5})$, respectively.

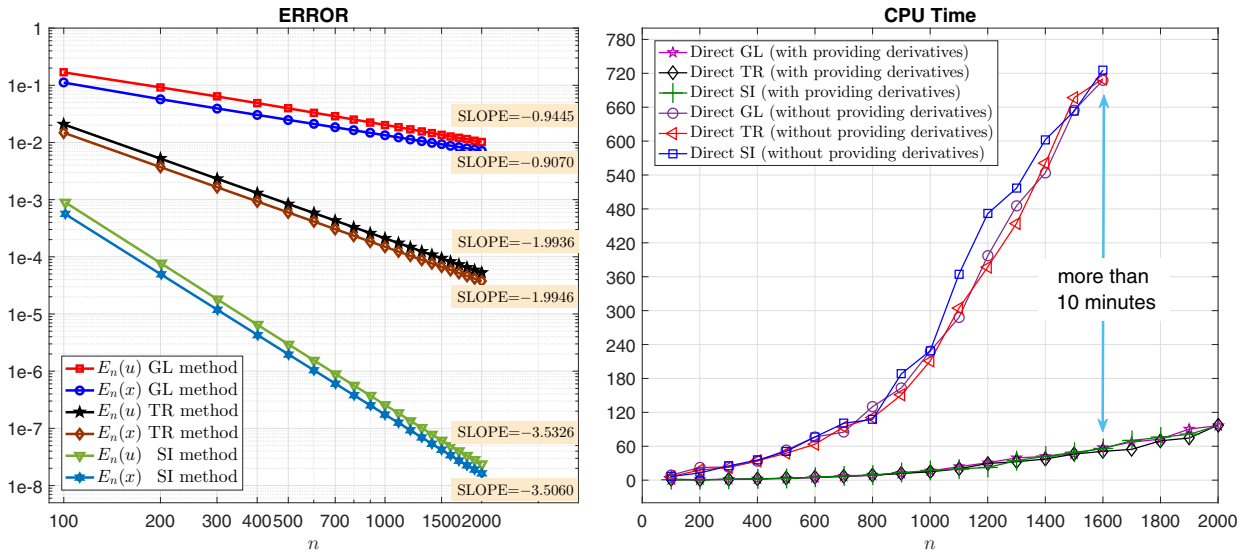

Figure 2: (Example 1, with $\alpha = 0.5$) Left: The measured $E_n(u)$ and $E_n(x)$ in the direct GL/TR/SL methods against various n in `loglog` scale. Right: CPU time of direct GL/TR/SI methods, with and without providing derivatives, versus n.

5.2. Example 2: A free final time FOCP with path and terminal constraints

In this example, a free final time FOCP with path constraint is considered. Moreover, the state function is forced to lie on a circle at the final time:

$$\begin{aligned} \min J[u] &= \frac{1}{2}\int_0^{t_{\rm f}} \left[x^2(t) + u^2(t)\right] \mathrm{d}t \\ \text{s.t. } & {}_0^{\rm C}\mathcal{D}_t^{\alpha} x(t) = -x(t) + u(t), \\ & x(0) = 1, \\ & u(t) \geq 0.2, \quad 0 \leq t \leq t_{\rm f}, \\ & \left[x(t) - 0.2\right]^2 + \left[t - 0.5\right]^2 \geq 0.25, \\ & \left[x(t_{\rm f}) - 0.2\right]^2 + \left[t_{\rm f} - 2\right]^2 = 0.04. \end{aligned} \tag{46}$$

Because of existence of path and terminal constraints, derivation of the optimality condition for this example is difficult. Consequently, solving this problem by indirect methods is cumbersome. We remind that in this paper we use a direct approach, which does not rely on the existence of optimality conditions.

By applying the direct GL, TR and SI methods on this example, the obtained final time $t_{\rm f}$, the optimal objective value and CPU times for various values of α and n are reported in Table 2. It is worthwhile to note that problem (46) is a free final time problem and has a nonsmooth solution. As a result, obtaining an accurate solution for this problem is not easily accessible. However, from Table 2, the precision and accuracy of the methods, especially the direct SI method, are concluded.

Table 2: (Example 2, with $\alpha = 0.2, 0.4, \ldots, 1.0$) CPU time, final time and value of the performance index, obtained through the direct GL, TR and SI methods with various values of n. All CPU times are in seconds.

		Direct GL method			Direct TR method			Direct SI method		
α	n	CPU	t_{f}	J_n	CPU	t_{f}	J_n	CPU	t_{f}	J_n
0.2	31	0.2	1.859043	0.309055	0.1	1.859490	0.318655	0.1	1.859530	0.315002
0.2	61	0.3	1.859345	0.309217	0.2	1.859575	0.313881	0.3	1.859601	0.312309
0.2	91	0.6	1.859440	0.309338	0.6	1.859595	0.312419	0.9	1.859614	0.311426
0.2	501	73.1	1.859599	0.309759	54.2	1.859628	0.310313	83.8	1.859632	0.310177
0.4	31	0.3	1.820755	0.313741	0.2	1.820827	0.320906	0.2	1.821028	0.317973
0.4	61	0.4	1.820724	0.314198	0.4	1.820796	0.3178	0.4	1.820789	0.316745
0.4	91	0.7	1.820728	0.314534	0.9	1.820776	0.316984	0.8	1.820761	0.31639
0.4	501	75.5	1.820722	0.315516	85.3	1.820731	0.316007	67.8	1.820728	0.315953
0.6	31	0.2	1.806128	0.323821	0.1	1.806192	0.329454	0.1	1.806075	0.327235
0.6	61	0.3	1.806159	0.324495	0.3	1.805935	0.327472	0.4	1.805796	0.326846
0.6	91	0.6	1.806034	0.324961	1.1	1.805920	0.327014	0.7	1.805890	0.326683
0.6	501	91.1	1.805853	0.326182	60.5	1.805841	0.326606	77.4	1.805833	0.326589
0.8	31	0.1	1.801137	0.335882	0.1	1.801207	0.339415	0.1	1.801154	0.337831
0.8	61	0.3	1.801029	0.336271	0.4	1.801109	0.338177	0.4	1.801077	0.337766
0.8	91	0.5	1.801000	0.336595	0.5	1.801076	0.337928	0.9	1.801053	0.337733
0.8	501	63.2	1.800979	0.337453	55.5	1.801017	0.337723	93.1	1.801012	0.337716
1.0	31	0.1	1.800442	0.346938	0.1	2.088769	0.363078	0.2	1.800840	0.347474
1.0	61	0.3	1.800650	0.34678	0.3	1.800884	0.347631	0.3	1.800904	0.34732
1.0	91	0.5	1.800743	0.346865	0.6	1.800901	0.347456	0.8	1.800917	0.347311
1.0	501	44.8	1.800907	0.347191	71.8	1.800939	0.347304	112.4	1.800942	0.347298

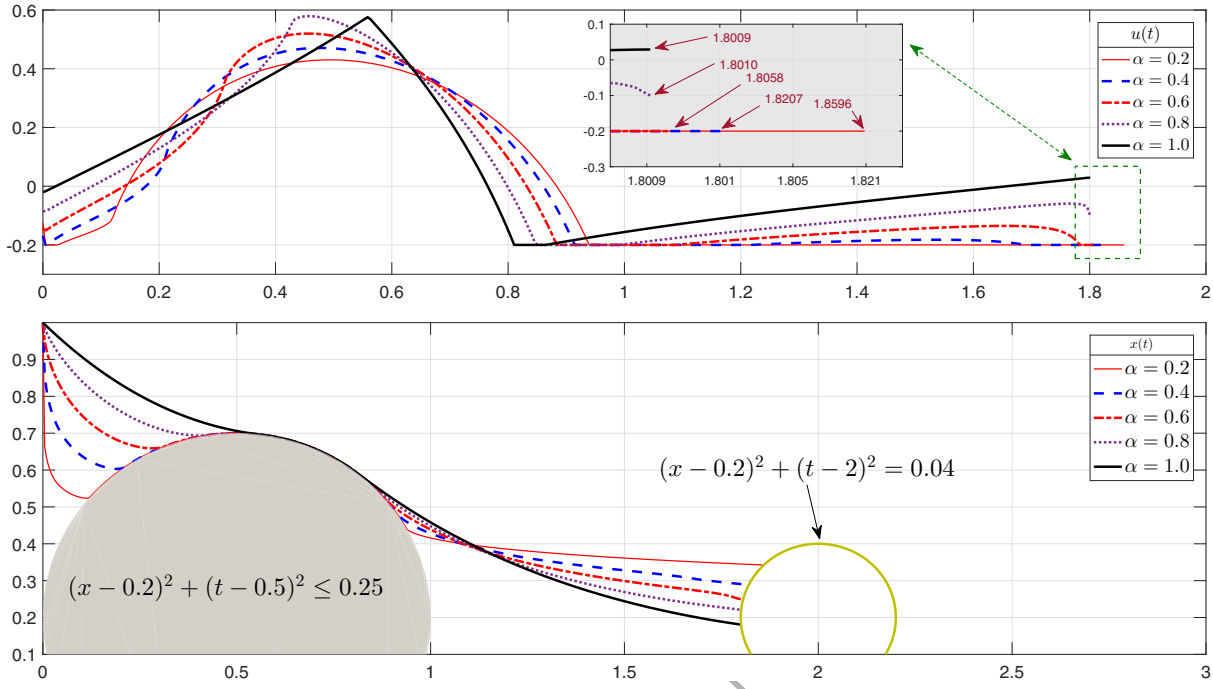

Figure 3: (Example 2, with $\alpha = 0.2, 04, \ldots, 1.0$) The obtained control (above) and state (below) functions by the direct TR method.

5.3. Example 3: A Bang-bang problem

In this example, we apply our new numerical technique to the following bang-bang problem, which is treated in [64]: determine the state $x(t)$ and control $u(t)$ on the interval $t \in [0, 2]$ that minimize the performance index

$$J(x, u) = \int_0^2 [x_1(t) - x_2(t) + u(t)]\mathrm{d}t$$

subject to the dynamic constraints

$$\begin{aligned} {}_0^{\mathrm{C}}\mathcal{D}_t^{\alpha} x_1(t) &= x_2(t) - u(t), \\ {}_0^{\mathrm{C}}\mathcal{D}_t^{\alpha} x_2(t) &= -u(t), \end{aligned}$$

and the initial conditions

$$x_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The exact solution to this problem is given by

$$u_*(t) = \begin{cases} 1 & t \in [0, 1], \\ 0 & t \in [1, 2], \end{cases} \qquad x_*(t) = \begin{bmatrix} x_*^1(t) \\ x_*^2(t) \end{bmatrix} = \begin{cases} \begin{bmatrix} -t \\ 1 - \frac{\sqrt{t}}{\Gamma(1.5)} \end{bmatrix}, & t \in [0, 1], \\ \begin{bmatrix} \frac{\sqrt{t-1}}{\Gamma(1.5)} - 1 \\ 1 - \frac{\sqrt{t}-\sqrt{t-1}}{\Gamma(1.5)} \end{bmatrix}, & t \in [1, 2]. \end{cases}$$

It is stressed that, in optimal control theory, the field of bang-bang optimal control problems is a classical topic. For bang-bang problems, the control function switches abruptly between its bounds. Bang-bang optimal control

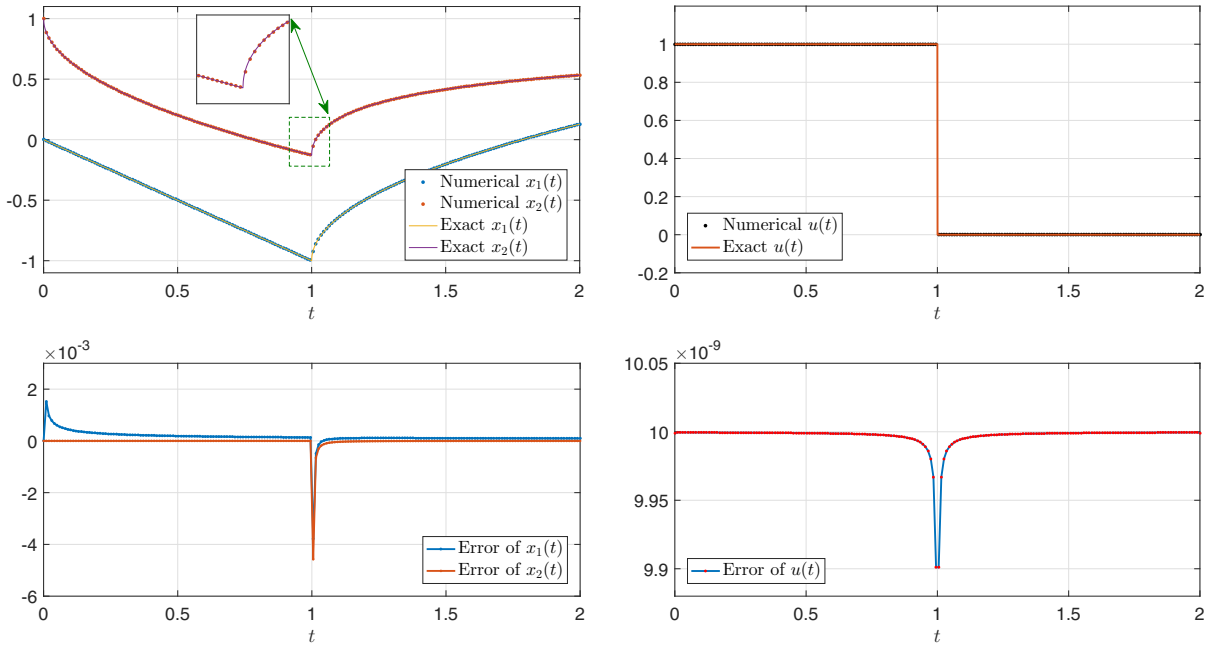

Figure 4: (Example 3) Obtained solution via the direct TR method with $n = 100$. Above Left: The exact and obtained states. Above Right: The exact and obtained control. Below Left: Error of the obtained states. Below Right: Error of the obtained control.

problems have received considerable attention, due to the difficulties arising in their numerical solutions. See [74] and references therein. Here, we assess our methods with this bang-bang problem.

The obtained control and state functions, by the direct TR method with $n = 100$, are plotted in Figure 4. In addition, the errors of the obtained state and control are plotted in this figure too. It is seen that the control and state functions are accurately approximated by the direct TR method. Moreover, we apply the direct TR method with $n = 100$ on this example for $\alpha = 0.1, 0.2, \ldots, 0.9, 1.0$. The obtained state functions are plotted in Figure 5. We can see that in the all cases the control functions are bang-bang. In addition, the measured CPU times, optimal value of the objective function and switching times are reported in Table 3. To show the precision of the method, the results with $n = 400$ are also reported in this table. We note that this problem is bang-bang and the control function is discontinuous and the state functions are nonsmooth. Naturally, the accuracy of a numerical method for such problems is lower in comparison with smooth problems. Nevertheless, according to Table 3, the direct TR method provides solutions with reasonable accuracy for this bang-bang problem.

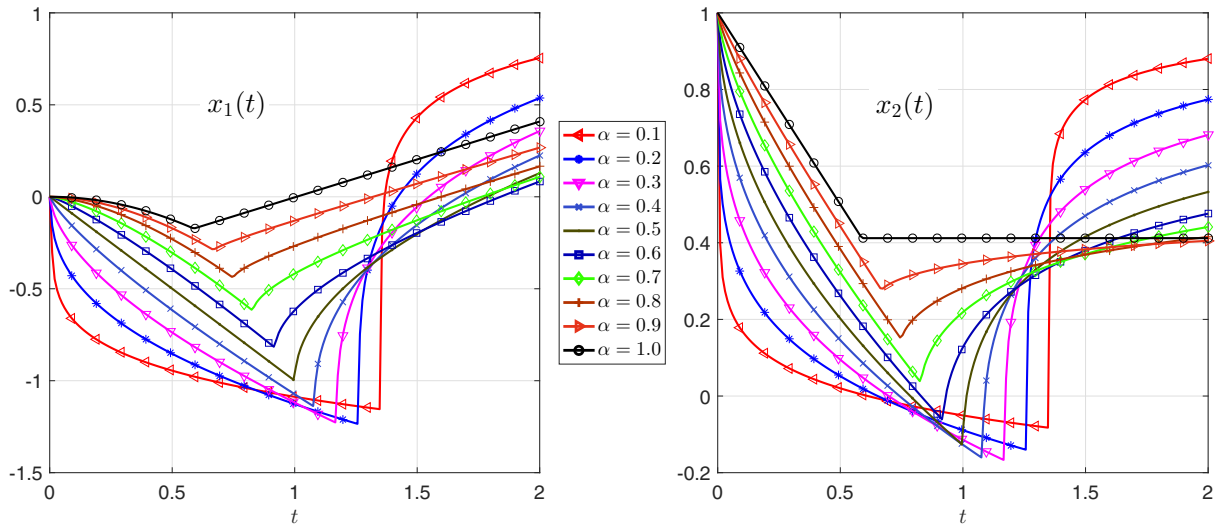

Figure 5: (Example 3, with $\alpha = 0.1, 0.2, \ldots, 1.0$) The obtained state and control functions, by the direct TR method with $n = 100$.

Table 3: (Example 3, with $\alpha = 0.1, 0.2, \ldots, 1.0$) The obtained CPU time, in seconds, value of the performance index and switching time, by the direct TR method with $n = 100$ and $n = 400$.

	$n = 100$			$n = 400$		
α	CPU	J_n	Switch	CPU	J_n	Switch
0.10	1.1	-0.14900	1.34343	56.2	-0.14621	1.34586
0.20	0.5	-0.25034	1.26262	31.3	-0.25109	1.26065
0.30	1.5	-0.32036	1.17449	58.5	-0.32070	1.17042
0.40	0.5	-0.35859	1.08080	36.6	-0.35912	1.08521
0.50	0.4	-0.37187	1.00002	22.0	-0.37225	1.00000
0.60	0.5	-0.36618	0.91910	42.6	-0.36644	0.91478
0.70	0.5	-0.34794	0.83838	36.3	-0.34813	0.83458
0.80	1.4	-0.32337	0.74496	34.1	-0.32343	0.74937
0.90	0.9	-0.29773	0.67676	88.5	-0.29785	0.66917
1.00	1.4	-0.27611	0.58389	51.2	-0.27613	0.58395

5.4. Example 4: Optimal control of a fractional-order HIV-immune system with memory

A HIV optimal control problem is now considered. The problem is stated in [23] as follows:

$$
\begin{aligned}
\min\ & J[u] = 500[x_1^2(t_{\rm f}) + x_3^2(t_{\rm f}) + x_4^2(t_{\rm f})] + \int_0^{t_{\rm f}} \left(500[x_1^2(t) + x_3^2(t) + x_4^2(t)] + 0.005\, u_1^2(t)\right) {\rm d}t,\\
& {}_0^{\rm C}\mathcal{D}_t^{\alpha} x_1(t) = -\,a_1\, x_1(t) - a_2\, x_1(t)\, x_2(t) + a_3\, a_4\, x_4(t)(1 - u_1(t)),\\
& {}_0^{\rm C}\mathcal{D}_t^{\alpha} x_2(t) = \frac{a_5}{1 + x_1(t)} - a_2\, x_1(t)\, x_2(t) - a_6\, x_2(t) + a_7 \left(1 - \frac{x_2(t) + x_3(t) + x_4(t)}{a_8}\right) x_2(t),\\
& {}_0^{\rm C}\mathcal{D}_t^{\alpha} x_3(t) = a_2\, x_1(t)\, x_2(t) - a_9\, x_3(t) - a_6\, x_3(t),\\
& {}_0^{\rm C}\mathcal{D}_t^{\alpha} x_4(t) = a_9\, x_3(t) - a_4\, x_4(t)\\
& x_1\,(0) = 0.049, \quad x_2\,(0) = 904,\\
& x_3\,(0) = 0.034, \quad x_4\,(0) = 0.0042,
\end{aligned}
$$

where the values of parameters a_i, $i = 1, \ldots, 9$, are as shown in Table 4. In [23], the authors used an indirect

Table 4: Parameter values used in the optimal control of the fractional HIV-immune system.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
2.4	2.4e-5	1200	0.24	10	0.02	0.03	1500	3e-3

method to solve this FOCP. There, necessary optimality conditions for the problem are firstly derived, then the problem is solved using an iterative algorithm. Here, we solve the problem by the proposed direct methods.

In Figure 6, we plot the obtained control and state functions, by applying the direct TR method with $n = 1000$ to the problem for $\alpha = 0.90, 0.95, 1.00$. Our results are in good agreement with those of [23]. In addition, to report the precision of the methods, the values of the performance index obtained by the direct TR and SI methods are given in Table 5. Based on these results, we observe that, without deriving the optimality conditions, the problem can be solved with a reasonable accuracy.

Table 5: (Example 4, with $\alpha = 0.90, 0.95, 1.00$) Optimal value of the performance index obtained using the direct TR and SI methods with $n = 500$ and 1000.

	Direct TR method		Direct SI method	
	$n = 500$	$n = 1000$	$n = 500$	$n = 1000$
$\alpha = 0.90$	22.70	22.65	22.63	22.61
$\alpha = 0.90$	18.07	18.03	18.01	17.99
$\alpha = 1.00$	14.72	14.39	14.32	14.31

6. Conclusion

In this paper, three direct methods based on Grünwald-Letnikov, trapezoidal and Simpson approximation formulas are presented for the numerical solution of a general class of fractional optimal control problems.

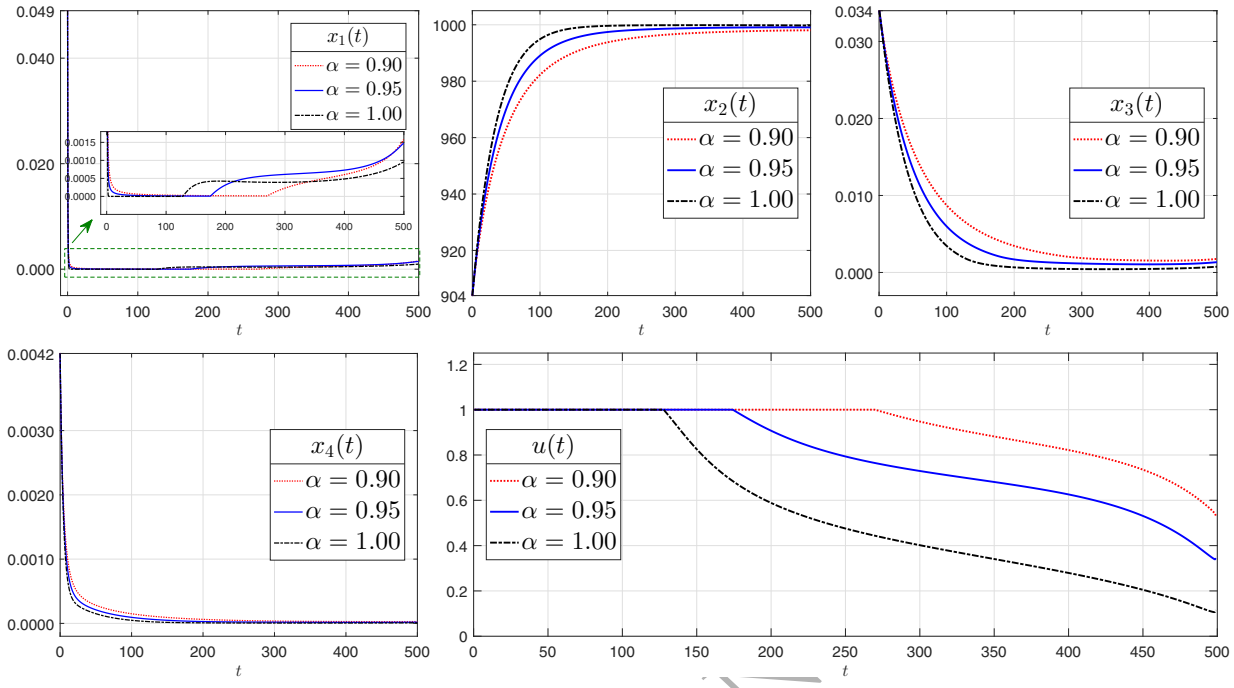

Figure 6: (Example 4, with $\alpha = 0.90, 0.95, 1.00$) Obtained states and control functions by the direct TR methods with $n = 1000$.

Optimality conditions are not needed in these methods. Thus, they can be applied to any type of fractional optimal control problem.

The proposed direct methods are illustrated in four academic and practical test problems and the results confirm that our methods are reliable. According to our numerical results, for problems with a smooth solution, the accuracy of the direct Simpson method is superior and for problems with discontinuous or a nonsmooth solution, the accuracy is satisfactory. Moreover, by providing the gradient of the objective function and the Jacobian of the constraints, the CPU time is significantly reduced. We conclude that the direct methods here proposed are simple, reliable, reasonably accurate and fast for solving fractional optimal control problems.

As further research works, we can refer to costate estimation and combining mesh generation techniques with the presented method to improve the accuracy and speed of the methods in solving problems with nonsmooth solutions.

Acknowledgements

Torres has been partially supported by FCT through the R&D Unit CIDMA (UID/MAT/04106/2013) and TOCCATA project PTDC/EEI-AUT/2933/2014 funded by FEDER and COMPETE 2020.

References

[1] Machado, J.T., Kiryakova, V., Mainardi, F.. Recent history of fractional calculus. Communications in Nonlinear Science and Numerical Simulation 2011;16(3):1140–1153. doi:10.1016/j.cnsns.2010.05.027.

[2] Mainardi, F.. Fractional calculus and waves in linear viscoelasticity. Imperial College Press, London; 2010.

[3] Ahmadian, A., Ismail, F., Salahshour, S., Baleanu, D., Ghaemi, F.. Uncertain viscoelastic models with fractional order: A new spectral tau method to study the numerical simulations of the solution. Communications in Nonlinear Science and Numerical Simulation 2017;53:44 – 64. doi:10.1016/j.cnsns.2017.03.012.

[4] Magin, R.L.. Fractional calculus in bioengineering. Begell House Redding; 2006.

[5] Ionescu, C., Lopes, A., Copot, D., Machado, J., Bates, J.. The role of fractional calculus in modeling biological phenomena: A review. Communications in Nonlinear Science and Numerical Simulation 2017;51:141–159. doi:10.1016/j.cnsns.2017.04.001.

[6] Marks, R., Hall, M.. Differintegral interpolation from a bandlimited signal's samples. IEEE Transactions on Acoustics, Speech, and Signal Processing 1981;29(4):872–877.

[7] Ortigueira, M.D.. Fractional calculus for scientists and engineers; vol. 84 of *Lecture Notes in Electrical Engineering*. Springer, Dordrecht; 2011. doi:10.1007/978-94-007-0747-4.

[8] Gorenflo, R., Mainardi, F., Scalas, E., Raberto, M.. Fractional calculus and continuous-time finance. III. The diffusion limit. In: Mathematical finance (Konstanz, 2000). Trends Math.; Birkhäuser, Basel; 2001, p. 171–180.

[9] Scalas, E., Gorenflo, R., Mainardi, F.. Fractional calculus and continuous-time finance. Physica A: Statistical Mechanics and its Applications 2000;284(1):376–384.

[10] Mozyrska, D., Torres, D.F.M.. Minimal modified energy control for fractional linear control systems with the Caputo derivative. Carpathian Journal of Mathematics 2010;26(2):210–221.

[11] Podlubny, I.. Fractional-order systems and fractional-order controllers. Institute of Experimental Physics, Slovak Academy of Sciences, Kosice 1994;12(3):1–18.

[12] Agarwal, R.P., Baleanu, D., Nieto, J.J., Torres, D.F.M., Zhou, Y.. A survey on fuzzy fractional differential and optimal control nonlocal evolution equations. Journal of Computational and Applied Mathematics in press;doi:10.1016/j.cam.2017.09.039.

[13] Machado, J.T., Galhano, A.M.. A fractional calculus perspective of distributed propeller design. Communications in Nonlinear Science and Numerical Simulation 2018;55:174–182. doi:10.1016/j.cnsns.2017.07.009.

[14] Moreles, M.A., Lainez, R.. Mathematical modelling of fractional order circuit elements and bioimpedance applications. Communications in Nonlinear Science and Numerical Simulation 2017;46:81 – 88. doi:10.1016/j.cnsns.2016.10.020.

[15] Tarasova, V.V., Tarasov, V.E.. Concept of dynamic memory in economics. Communications in Nonlinear Science and Numerical Simulation 2018;55:127 – 145. doi:10.1016/j.cnsns.2017.06.032.

[16] Podlubny, I.. Fractional differential equations; vol. 198 of *Mathematics in Science and Engineering*. Academic Press, Inc., San Diego, CA; 1999.

[17] Diethelm, K.. The analysis of fractional differential equations; vol. 2004 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin; 2010. doi:10.1007/978-3-642-14574-2.

[18] Malinowska, A.B., Torres, D.F.M.. Introduction to the fractional calculus of variations. Imperial College Press, London; 2012.

[19] Almeida, R., Pooseh, S., Torres, D.F.M.. Computational methods in the fractional calculus of variations. Imperial College Press, London; 2015. doi:10.1142/p991.

[20] Malinowska, A.B., Odzijewicz, T., Torres, D.F.M.. Advanced methods in the fractional calculus of variations. SpringerBriefs in Applied Sciences and Technology; Springer, Cham; 2015.

[21] Fard, O.S., Soolaki, J., Torres, D.F.M.. A necessary condition of Pontryagin type for fuzzy fractional optimal control problems. Discrete and Continuous Dynamical Systems Series S 2018;11(1):59–76. doi:10.3934/dcdss.2018004.

[22] Tricaud, C., Chen, Y.. An approximate method for numerically solving fractional order optimal control problems of general form. Computers & Mathematics with Applications 2010;59(5):1644–1655. doi:10.1016/j.camwa.2009.08.006.

[23] Ding, Y., Wang, Z., Ye, H.. Optimal control of a fractional-order HIV-immune system with memory. Control Systems Technology, IEEE Transactions on 2012;20(3):763–769.

[24] Sweilam, N., AL-Mekhlafi, S.. On the optimal control for fractional multistrain TB model. Optimal Control Applications and Methods 2016;37(6):1355–1374. doi:10.1002/oca.2247.

[25] Sweilam, N.H., Al-Mekhlafi, S.M.. Legendre spectral-collocation method for solving fractional optimal control of HIV infection of CD4+T cells mathematical model. The Journal of Defense Modeling and Simulation 2017;14(3):273–284. doi:10.1177/1548512916677582.

[26] Zaky, M., Machado, J.T.. On the formulation and numerical simulation of distributed-order fractional optimal control problems. Communications in Nonlinear Science and Numerical Simulation 2017;52:177 – 189. doi:10.1016/j.cnsns.2017.04.026.

[27] Wojtak, W., Silva, C.J., Torres, D.F.M.. Uniform asymptotic stability of a fractional tuberculosis model. Math Model Nat Phenom 2018;13(1):in press. doi:10.1051/mmnp/2018015.

[28] Pooseh, S., Almeida, R., Torres, D.F.M.. Discrete direct methods in the fractional calculus of variations. Computers & Mathematics with Applications An International Journal 2013;66(5):668–676. doi:10.1016/j.camwa.2013.01.045.

[29] Agrawal, O.P.. A general formulation and solution scheme for fractional optimal control problems. Nonlinear Dynamics 2004;38(1-4):323–337. doi:10.1023/B:NODY.0000045544.96418.bf.

[30] Agrawal, O.P., Baleanu, D.. A Hamiltonian formulation and a direct numerical scheme for fractional optimal control problems. Journal of Vibration and Control 2007;13(9-10):1269–1281. doi:10.1177/1077546307077467.

[31] Agrawal, O.P.. A formulation and numerical scheme for fractional optimal control problems. Journal of Vibration and Control 2008;14(9-10):1291–1299. doi:10.1177/1077546307087451.

[32] Frederico, G.a.S.F., Torres, D.F.M.. Fractional conservation laws in optimal control theory. Nonlinear Dynamics 2008;53(3):215–222. doi:10.1007/s11071-007-9309-z.

[33] Frederico, G.a.S.F., Torres, D.F.M.. Fractional optimal control in the sense of Caputo and the fractional Noether's theorem. International Mathematical Forum Journal for Theory and Applications 2008;3(9-12):479–493.

[34] Baleanu, D., Defterli, O., Agrawal, O.P.. A central difference numerical scheme for fractional optimal control problems. Journal of Vibration and Control 2009;15(4):583–597. doi:10.1177/1077546308088565.

[35] Agrawal, O.P., Defterli, O., Baleanu, D.. Fractional optimal control problems with several state and control variables. Journal of Vibration and Control 2010;16(13):1967–1976. doi:10.1177/1077546309353361.

[36] Biswas, R.K., Sen, S.. Fractional optimal control problems: a pseudo-state-space approach. Journal of Vibration and Control 2011;17(7):1034–1041. doi:10.1177/1077546310373618.

[37] Lotfi, A., Dehghan, M., Yousefi, S.A.. A numerical technique for solving fractional optimal control problems. Comput Math Appl 2011;62(3):1055–1067. doi:10.1016/j.camwa.2011.03.044.

[38] Lotfi, A., Yousefi, S., 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. Journal of Computational and Applied Mathematics 2013;250:143–160. doi:10.1016/j.cam.2013.03.003.

[39] Bhrawy, A.H., Ezz-Eldien, S.S.. A new legendre operational technique for delay fractional optimal control problems. Calcolo 2016;53(4):521–543. doi:10.1007/s10092-015-0160-1.

[40] Bhrawy, A., Doha, E., Tenreiro Machado, J., Ezz-Eldien, S.S.. An efficient numerical scheme for solving multi-dimensional fractional optimal control problems with a quadratic performance index. Asian Journal of Control 2015;17(6):2389–2402.

[41] Doha, E.H., Bhrawy, A.H., Baleanu, D., Ezz-Eldien, S.S., Hafez, R.M.. An efficient numerical scheme based on the shifted orthonormal jacobi polynomials for solving fractional optimal control problems. Advances in Difference Equations 2015;2015(1):15. doi:10.1186/s13662-014-0344-z.

[42] Ezz-Eldien, S., Doha, E., Baleanu, D., Bhrawy, A.. A numerical approach based on Legendre orthonormal polynomials for numerical solutions of fractional optimal control problems. Journal of Vibration and Control 2017;23(1):16–30.

[43] Ezz-Eldien, S.S., El-Kalaawy, A.A.. Numerical simulation and convergence analysis of fractional optimization problems with right-sided caputo fractional derivative. Journal of Computational and Nonlinear Dynamics 2017;13(1):011010–011010–8. doi:10.1115/1.4037597.

[44] Yousefi, S.A., Lotfi, A., Dehghan, M.. The use of a Legendre multiwavelet collocation method for solving the fractional optimal control problems. Journal of Vibration and Control 2011;17(13):2059–2065. doi:10.1177/1077546311399950.

[45] Alipour, M., Rostamy, D., Baleanu, D.. Solving multi-dimensional fractional optimal control problems with inequality constraint by Bernstein polynomials operational matrices. Journal of Vibration and Control 2013;19(16):2523–2540. doi:10.1177/1077546312458308.

[46] Keshavarz, E., Ordokhani, Y., Razzaghi, M.. A numerical solution for fractional optimal control problems via bernoulli polynomials. Journal of Vibration and Control 2016;22(18):3889–3903. doi:10.1177/1077546314567181.

[47] Rabiei, K., Ordokhani, Y., Babolian, E.. Numerical solution of 1D and 2D fractional optimal control of system via Bernoulli polynomials. International Journal of Applied and Computational Mathematics 2018;4(1):7.

[48] Zahra, W., Hikal, M.. Non standard finite difference method for solving variable order fractional optimal control problems. Journal of Vibration and Control 2017;23(6):948–958. doi:10.1177/1077546315586646.

[49] Rakhshan, S.A., Kamyad, A.V., Effati, S.. An efficient method to solve a fractional differential equation by using linear programming and its application to an optimal control problem. J Vib Control 2016;22(8):2120–2134. doi:10.1177/1077546315584471.

[50] Tang, X., Liu, Z., Wang, X.. Integral fractional pseudospectral methods for solving fractional optimal control problems. Automatica 2015;62:304–311.

[51] Jahanshahi, S., Torres, D.F.M.. A simple accurate method for solving fractional variational and optimal control problems. Journal of Optimization Theory and Applications 2017;174(1):156–175. doi:10.1007/s10957-016-0884-3.

[52] Nemati, A., Yousefi, S., Soltanian, F., Ardabili, J.S.. An efficient numerical solution of fractional optimal control problems by using the Ritz method and Bernstein operational matrix. Asian Journal of Control 2016;.

[53] Lotfi, A., Yousefi, S.A.. Epsilon-ritz method for solving a class of fractional constrained optimization problems. Journal of Optimization Theory and Applications 2014;163(3):884–899. doi:10.1007/s10957-013-0511-5.

[54] Mashayekhi, S., Razzaghi, M.. An approximate method for solving fractional optimal control problems by hybrid functions. Journal of Vibration and Control 2018;In press. doi:10.1177/1077546316665956.

[55] 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. Optimal Control Applications and Methods 2018;39(2):873–887. doi:10.1002/oca.2383.

[56] Ejlali, N., Hosseini, S.M.. A pseudospectral method for fractional optimal control problems. Journal of Optimization Theory and Applications 2017;174(1):83–107.

[57] Rakhshan, S.A., Effati, S., Kamyad, A.V.. Solving a class of fractional optimal control problems by the Hamilton-Jacobi-Bellman equation. Journal of Vibration and Control in press;doi:10.1177/1077546316668467.

[58] Lotfi, A.. A combination of variational and penalty methods for solving a class of fractional optimal control problems. Journal of Optimization Theory and Applications 2017;174(1):65–82. doi:10.1007/s10957-017-1106-3.

[59] Mu, P., Wang, L., Liu, C.. A control parameterization method to solve the fractional-order optimal control problem. Journal of Optimization Theory and Applications 2018;In press:1–14. doi:10.1007/s10957-017-1163-7.

[60] Tang, X., Shi, Y., Wang, L.L.. A new framework for solving fractional optimal control problems using fractional pseudospectral methods. Automatica 2017;78:333–340.

[61] Singha, N., Nahak, C.. An efficient approximation technique for solving a class of fractional optimal control problems. Journal of Optimization Theory and Applications 2017;174(3):785–802.

[62] Almeida, R., Torres, D.F.. A discrete method to solve fractional optimal control problems. Nonlinear Dynamics 2015;80(4):1811–1816. doi:10.1007/s11071-014-1378-1.

[63] Baleanu, D., Jajarmi, A., Hajipour, M.. A new formulation of the fractional optimal control problems involving Mittag-Leffler nonsingular kernel. Journal of Optimization Theory and Applications 2017;175(3):718–737.

[64] Kamocki, R., Majewski, M.. Fractional linear control systems with Caputo derivative and their optimization. Optimal Control Applications and Methods 2015;36(6):953–967.

[65] Biswas, R.K., Sen, S.. Free final time fractional optimal control problems. Journal of the Franklin Institute 2014;351(2):941–951. doi:10.1016/j.jfranklin.2013.09.024.

[66] Pooseh, S., Almeida, R., Torres, D.F.M.. Fractional order optimal control problems with free terminal time. Journal of Industrial and Management Optimization 2014;10(2):363–381. doi:10.3934/jimo.2014.10.363.

[67] Betts, J.T., Huffman, W.P.. Sparse optimal control software SOCS. Mathematics and Engineering Analysis Technical Document MEA-LR-085, Boeing Information and Support Services, The Boeing Company, PO Box 1997;3707:98124–2207.

[68] Falugi, P., Kerrigan, E., Wyk, E.V.. Imperial College London Optimal Control Software User Guide (ICLOCS). London, UK: Department of Electrical Engineering, Imperial College London. http://www.ee.ic.ac.uk/ICLOCS.; 2010.

[69] Li, C., Zeng, F.. Numerical methods for fractional calculus; vol. 24. CRC Press; 2015.

[70] Brzeziski, D.W., Ostalczyk, P.. About accuracy increase of fractional order derivative and integral computations by applying the Grünwald-Letnikov formula. Communications in Nonlinear Science and Numerical Simulation 2016;40:151–162. doi:10.1016/j.cnsns.2016.03.020.

[71] Laub, A.. Matrix Analysis for Scientists and Engineers. Society for Industrial and Applied Mathematics; 2005.

[72] Wächter, A., Biegler, L.T.. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 2006;106(1, Ser. A):25–57.

[73] Abramowitz, M., Stegun, I.A.. Handbook of mathematical functions: with formulas, graphs, and mathematical tables; vol. 55. Courier Corporation; 1964.

[74] Shamsi, M.. A modified pseudospectral scheme for accurate solution of Bang-bang optimal control problems. Optimal Control Applications and Methods 2011;32(6):668–680.