A space-time pseudospectral discretization method for solving diffusion optimal control problems with two-sided fractional derivatives

Mushtaq Salh Ali¹, Mostafa Shamsi¹, Hassan Khosravian-Arab¹, Delfim F. M. Torres² and Farid Bozorgnia³

Abstract
We propose a direct numerical method for the solution of an optimal control problem governed by a two-side space-fractional diffusion equation. The presented method contains two main steps. In the first step, the space variable is discretized by using the Jacobi–Gauss pseudospectral discretization and, in this way, the original problem is transformed into a classical integer-order optimal control problem. The main challenge, which we faced in this step, is to derive the left and right fractional differentiation matrices. In this respect, novel techniques for derivation of these matrices are presented. In the second step, the Legendre–Gauss–Radau pseudospectral method is employed. With these two steps, the original problem is converted into a convex quadratic optimization problem, which can be solved efficiently by available methods. Our approach can be easily implemented and extended to cover fractional optimal control problems with state constraints. Five test examples are provided to demonstrate the efficiency and validity of the presented method. The results show that our method reaches the solutions with good accuracy and a low CPU time.

Keywords
Optimal control of partial differential equations, two sided space-time fractional diffusion equations, pseudospectral methods, Jacobi polynomials, left and right differentiation matrices.

Introduction
Fractional (non-integer order) order diffusion equations, in comparison with classical integer order counterparts, can describe more accurately irregular diffusion processes, such as gas diffusion in fractal porous media and heat conduction (Wu et al. 2015, 2017; Yang et al. 2017b; Yang and Machado 2017). Accordingly, the numerical solution of fractional diffusion equations has gained considerable attention (Meerschaert and Tadjeran 2006; Doha et al. 2014; Zaky et al. 2016; Yang et al. 2017a, 2016; Feng et al. 2018; Chen et al. 2018; Yang et al. 2017c).

Optimal control of fractional diffusion equations has recently received special attention, due to their applications in various fields, such as control of temperature in a thermo conduction process or mass diffusive transport in a porous media (Du et al. 2016).

Mophou (2011) considered the optimal control of a time-fractional diffusion equation in a bounded domain and investigated the questions of existence and uniqueness of solution. Moreover, the first-order optimality conditions are derived in (Mophou 2011).

The optimal control of time-fractional diffusion equations with state constraints and non-homogeneous Dirichlet boundary conditions is considered in Mophou and N’Guérékata (2011) and Dorville et al. (2011), respectively. In these papers, existence and uniqueness of solution and first-order optimality conditions are also studied. Regarding numerical methods for solving optimal control problems of time-fractional diffusion equations, we refer to Li and Zhou (2018); Yamamoto (2018). The reader interested on the optimal control of fractional partial differential equations is referred to Agarwal et al. (2018); Zaky and Machado (2017); Zaky (2018); Bai et al. (2018); Darehmiraki et al. (2018) and references therein.

Recently, an optimal control problem governed by a two-sided space-fractional diffusion equation was considered in Du et al. (2016) and Wu and Huang (2018). More precisely, a constrained time-dependent optimal control problem defined on the space interval (0, 1) and time period [0, T] was considered. The aim is to find the control u(x, t) ∈

¹Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran
²Department of Mathematics, Center for Research and Development in Mathematics and Applications (CIDMA), University of Aveiro, 3810-193 Aveiro, Portugal
³Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisbon, Portugal

Corresponding author:
Mostafa Shamsi, Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, Hafez Avenue no. 424, Tehran, Iran
Email: m_shamsi@aut.ac.ir
where \( u \) and \( y \) is a known function, which represents the observed or \( \min \) is presented for problem (1). In Wu and Huang (2018), to get faithful and fast gradient projection method is developed to optimality conditions for problem (1) are derived and a more reduction of computational times, a parallel-in-time solve them numerically. In Wu and Huang (2018), to get useful properties of the Jacobi polynomials, we present role in the pseudospectral method and, therefore, need these differentiation matrices, which are approximations of fractional derivatives in the considered problem, we need to because of the existence of both left and right space-time optimal control problem. It is worthwhile to note that, as a result, the problem is reduced to a classical Gauss pseudospectral method is used for space discretization and inequality constraints on state and control are handled simpler, in comparison with indirect methods (Pooshe et al. 2013; Salati et al. 2019; Mohammadzadeh et al. 2018).

Pseudospectral methods approximate the unknown function(s) using interpolating polynomials with specific collocation points such as Legendre–Gauss (LG) (Benson et al. 2006), Legendre–Gauss–Lobatto (LGL) (Elıııar et al. 1995) and Legendre–Gauss–Radau (LGR) points (Garg et al. 2010). These selections lead to the three most common types of pseudospectral methods, which are referred as LG pseudospectral (Benson et al. 2006), LGL pseudospectral (Fahroo and Ross 2001; Elıııar et al. 1995), and LGR pseudospectral (Garg 2011; Garg et al. 2010) methods.

It is a well-known fact that, to solve ordinary or partial differential equations with a simple domain and a smooth solution, pseudospectral methods can usually achieve an accurate solution even with a small number of nodes. On the other hand, for problems with a non-smooth solution, the accuracy of pseudospectral methods is reduced. However, for problems with a non-smooth solution, they lead to a reasonably accurate approximation with less demand on computational time and computer memory.

Due to the mentioned computational efficiency, pseudospectral methods have been popular for the numerical solution of optimal control problems governed by integer order differential equations (Garg et al. 2010; Foroozandeh et al. 2017; Ezz-Eldien et al. 2017; Foroozandeh et al. 2018). More recently, pseudospectral methods have been extended for solving optimal control problems governed by fractional ordinary and partial differential equations (Tang et al. 2017; Li and Zhou 2018; Khaksar-eOshagh and Shamsi 2017).

Here we present a direct method for solving problem (1) that consists in two steps. In the first step, the Jacobi–Gauss pseudospectral method is used for space discretization and, as a result, the problem is reduced to a classical optimal control problem. It is worthwhile to note that, because of the existence of both left and right space-fractional derivatives in the considered problem, we need to compute the left and right fractional differentiation matrices. These differentiation matrices, which are approximations of the left and right fractional operators, play an important role in the pseudospectral method and, therefore, need to be obtained accurately. In this paper, by using some useful properties of the Jacobi polynomials, we present efficient strategies for computing the left and right fractional differentiation matrices. In the second step of our method, we apply the Legendre–Gauss–Radau pseudospectral method. The resulting ordinary optimal control problem is then transcribed into a quadratic optimization problem. In

\[
L^2(0,T; (0,1)) \mbox{ and state } y(x,t) \in L^2(0,T; (0,1)) \mbox{ such that the space-fractional diffusion equation}
\]

\[
\frac{\partial y}{\partial t}(x,t) = c(x,t) \left[ r \mathcal{D}_x^{2-\beta} y(x,t) + (1-r) \mathcal{D}_t^{2-\beta} y(x,t) \right]
\]

\[
+ f(x,t) + u(x,t),
\]

(1a)

with initial condition

\[
y(x,0) = g(x), \quad x \in (0,1),
\]

(1b)

and the boundary conditions

\[
y(0,t) = y(1,t) = 0, \quad t \in (0,T),
\]

(1c)

are satisfied and the following performance index is minimized:

\[
J[u] := \frac{1}{2} \int_0^T \int_0^1 [y(x,t) - z(x,t)]^2 \, dx \, dt + \frac{1}{2} \int_0^T \int_0^1 u^2(x,t) \, dx \, dt.
\]

(1d)

Moreover, the following constraint on the control function is considered:

\[
u(x,t) \geq u_{\min}(x,t), \quad x \in (0,1), \quad t \in (0,T),
\]

(1e)

where \( u_{\min} \) is a known function. In problem (1), \( z(x,t) \) is a known function, which represents the observed or desired values of state function \( y \), \( f(x,t) \) is the source or sink term, \( d(x,t) \) is the diffusivity coefficient and \( 2 - \beta \) is the order of diffusion, where \( 0 < \beta < 1 \). Moreover, \( \mathcal{D}_x^{2-\beta} \) and \( \mathcal{D}_t^{2-\beta} \) denote the left and right Riemann–Liouville fractional derivatives, defined (see, e.g., Malinowska and Torres (2012)) as

\[
\mathcal{D}_x^{-\alpha} g(x) = \frac{1}{\Gamma(2 - \alpha)} \frac{d^2}{dx^2} \int_0^x (x-s)^{1-\alpha} g(s) \, ds.
\]

\[
\mathcal{D}_t^{-\alpha} g(x) = \frac{1}{\Gamma(2 - \alpha)} \frac{d^2}{dx^2} \int_s^1 (s-x)^{1-\alpha} g(s) \, ds.
\]

Generally speaking, the two main challenges in solving problem (1) numerically are: (i) the problem contains both left and right fractional derivatives, (ii) the fractional derivatives are nonlinear and singular operators. Moreover, when using finite-difference-based methods, the stiffness of the approximation matrix of the fractional derivative is added to the two aforementioned drawbacks (Wu and Huang 2018). Thus, finite difference schemes, for solving problem (1), require expensive computation time and storage cost. In this regard, the main concern in the finite difference based methods for solving the problem (1) is to reduce the demand for time and memory computation (Du et al. 2016; Wu and Huang 2018). In Du et al. (2016), necessary optimality conditions for problem (1) are derived and a faithful and fast gradient projection method is developed to solve them numerically. In Wu and Huang (2018), to get more reduction of computational times, a parallel-in-time algorithm for implementing the gradient projection method is presented for problem (1).

In this paper, we present another method for solving problem (1) numerically. There are two major differences between the method we propose here and the ones of Du et al. (2016) and Wu and Huang (2018). First, our approach is based on high-order and global pseudospectral methods rather than finite difference schemes. Second, our method is direct, that is, does not rely on necessary optimality conditions. In a direct method, the optimal control problem is solved by transcribing it into a Non-Linear Programming (NLP) problem; thereafter, an NLP-solver is used to solve the resulting NLP problem (Pooshe et al. 2013; Salati et al. 2019; Behroozifar and Habibi 2018; Mashayehki and Razzaghi 2018). The direct methods are easily implemented and inequality constraints on state and control are handled simpler, in comparison with indirect methods (Pooshe et al. 2013; Salati et al. 2019; Mohammadzadeh et al. 2018).
addition, for easy implementation and analysis, the standard form of the quadratic optimization problem is derived.

The rest of the paper is organized as follows. In section “Preliminaries and notations”, we recall some necessary definitions and relevant properties of the Jacobi polynomials, quadrature rules, and the Kronecker product. Next section is devoted to the first step of our method, where the Jacobi–Gauss pseudospectral method is applied for space discretization of the problem. Moreover, the computation of the differentiation matrices is presented in this section. Then the second step of our method is presented, where the Legendre–Gauss–Radau pseudospectral method is utilized to reduce the problem to a quadratic optimization one. In section “Numerical experiments”, five illustrative examples are investigated to assess the accuracy and efficiency of the proposed numerical method. We end with a section of conclusions.

Preliminaries and notations

Jacobi polynomials have great flexibility to develop efficient numerical methods for solving a wide range of fractional differential models. Consequently, in the last decade, Jacobi polynomials have been widely used to solve fractional problems (Easmaeel and Shamsi 2011; Easmaeel et al. 2011; Bhrawy and Zaky 2015; Dehghan et al. 2016; Bhrawy and Zaky 2016; Zaky et al. 2018). Our method uses Jacobi polynomials too. Thus, in this section we briefly review the Jacobi polynomials, Jacobi quadrature rules, and some relevant theorems on the derivatives of Jacobi polynomials.

Jacobi Polynomials

The Jacobi polynomials \( P_n^{(a,b)}(x) \), where \( r \in [-1, 1] \) and \( n = 0, 1, \ldots \), are given explicitly by Gautschi (1996):

\[
P_n^{(a,b)}(x) = \frac{1}{n!} \sum_{k=0}^{n} \binom{n + a}_{n-k} \binom{n + b}_{k} (x-a)^k (x+b)^{n-k} (1-x)^{(n-k)/2} (1+x)^{(n-k)/2}.
\]

However, in practice, one can use the so-called recurrence Bonnet’s relation to generate the Jacobi polynomials in a stable and accurate manner (Gautschi 1996). If \( a, b > -1 \), Jacobi polynomials are called classical Jacobi polynomials. The well-known Legendre polynomials are a special case of the Jacobi polynomials when \( a = b = 0 \). In the following, some useful properties of the classical Jacobi polynomials are reviewed.

The classical Jacobi polynomials are orthogonal on the canonical interval \([−1, 1]\) with respect to the weight function \((1−τ)alfa(1+τ)beta\), i.e.,

\[
\int_{-1}^{1} (1-τ)alfa(1+τ)beta P_n^{(a,b)}(τ)P_m^{(a,b)}(τ)dτ = \begin{cases} 0, & m \neq n, \\ \frac{2^alpha+1\Gamma(n+a+1)\Gamma(n+b+1)}{n!(2n+a+b+1)(n+a+b+1)}, & m = n. \end{cases}
\] (2)

A useful formula that relates the Jacobi polynomials and their derivatives is

\[
\frac{d^k}{dτ^k} P_n^{(a,b)}(τ) = \frac{\Gamma(k + n + a + b + 1)}{2^k \Gamma(n + a + b + 1)} P_{n-k}^{(a+k,b+k)}(τ),
\]

\( n \geq k \).

The Jacobi polynomials also satisfy the following properties:

\[
P_k^{(a,b-1)}(2x-1) = \frac{k+a+b}{2k+a+b} P_k^{(a,b)}(2x-1) + \frac{k+a}{2k+a+b} P_{k-1}^{(a,b)}(2x-1), \quad (3)
\]

\[
P_k^{(a-1,b)}(2x-1) = \frac{k+a+b}{2k+a+b} P_k^{(a,b)}(2x-1) - \frac{k+a}{2k+a+b} P_{k-1}^{(a,b)}(2x-1). \quad (4)
\]

In the following, we recall two important theorems, which are used later to establish our method.

**Theorem 1.** See Zayernouri and Karniadakis (2013). Let \( r - a > -1 \) and \( q + α > -1 \). Then, for \( x \in [0, 1] \), we have:

\[
\int_0^1 x^α P_k^{(q,r)}(2x-1) dx = \frac{\Gamma(k+r+1)}{\Gamma(k+r-α+1)} x^α P_k^{(q+a,r-α)}(2x-1).
\]

**Theorem 2.** See Zayernouri and Karniadakis (2013). Let \( q - a > -1 \) and \( r + α > -1 \). Then, for \( x \in [0, 1] \), we have:

\[
\int_0^1 (1-x)^α P_k^{(q,r)}(2x-1) dx = \frac{\Gamma(k+q+1)}{\Gamma(k+q-α+1)} (1-x)^α P_k^{(q-a,r+α)}(2x-1).
\]

Jacobi nodes and quadratures

In the pseudospectral methods, the Jacobi–Gauss and Jacobi–Gauss–Radau nodes are successfully used as discretization points (Garg et al. 2010). Here we recall the definitions of these nodes and their corresponding quadrature rules (Gautschi 1996; Garg 2011).

Since the Jacobi polynomials are orthogonal in \([-1, 1]\), all the zeros of \( P_n^{(a,b)}(τ) \) are simple and belong to the interval \([-1, 1]\) (Gautschi 1996). These zeros are called the Jacobi–Gauss nodes with parameters \( a \) and \( b \), which we denote by \( \{x_i^{(a,b)}\}_{i=1}^n \). The Jacobi–Gauss quadrature rule with parameters \( a \) and \( b \) is based on the Jacobi–Gauss nodes \( \{x_i^{(a,b)}\}_{i=1}^n \) and can be used for approximating the integral of a function over the interval \([-1, 1]\) with weight \( (1-x)^α(1+x)^β \) as

\[
\int_{-1}^{1} (1-x)^α(1+x)^β f(x)dx \approx \sum_{i=1}^{n} \omega_i^{(a,b)} f(x_i^{(a,b)}),
\] (5)

where \( \omega_i^{(a,b)}, i = 1, \ldots, n \), are the Jacobi–Gauss quadrature weights. The above quadrature is exact whenever \( f(x) \) is a polynomial of degree equal or less than \( 2n-1 \), i.e., in the Jacobi–Gauss quadrature rule, the degree of exactness is \( 2n-1 \). In the Jacobi–Gauss–Radau nodes \( \{x_i^{(a,b)}\}_{i=1}^m \), the first node is \( x_1 = -1 \) and the last \( m-1 \) nodes are the zeros of \( P_{m-2}^{(a,b)}(τ) + P_{m-1}^{(a,b)}(τ) \). We note that the zeros of \( P_{m-2}^{(a,b)}(τ) + P_{m-1}^{(a,b)}(τ) \) belong to \([-1, 1]\). Thus, the Jacobi–Gauss–Radau points lie in the interval \([-1, 1]\). The Jacobi–Gauss–Radau quadrature rule with parameters \( a \) and \( b \) is given by

\[
\int_{-1}^{1} (1-x)^α(1+x)^β f(x)dx \approx \sum_{i=1}^{m} \omega_i^{(a,b)} f(x_i^{(a,b)}),
\] (6)

where \( x_i^{(a,b)}, i = 1, \ldots, m \), are the Jacobi–Gauss–Lobatto quadrature weights. The degree of exactness of the Jacobi–Gauss–Radau quadrature is \( 2m-2 \).
Kronecker products

Let $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$. The Kronecker product of matrices $A$ and $B$ is defined as

$$a_{ij}B \cdots a_{in}B \cdots a_{mj}B \cdots a_{mn}B \in \mathbb{R}^{mp \times nq},$$

Theorem 3. See Laub (2005). For any three matrices $A$, $B$ and $E$, for which the matrix product $ABE$ is defined, one has

$$\text{vec}(ABE) = (R^T \otimes A)\text{vec}(B),$$

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 one another.

Step I: The Jacobi–Gauss pseudospectral method for spatial discretization

The first step of our method consists to discretize the spatial variable $x$. For that, we use the pseudospectral method based on the Jacobi–Gauss nodes with parameters $(a, b) = (1, 1)$. Let $n$ be a positive integer number. Since the spatial domain of the problem is $[0, 1]$, we consider the Jacobi–Gauss nodes correspondant to the interval $[0, 1]$ as

$$\xi_i := \frac{1}{2}\left(\xi_i^{1n} + 1\right), \quad i = 1, \ldots, n. \quad (7)$$

The control and state functions are approximated as

$$u(x, t) \simeq \tilde{u}_n(x, t) := \sum_{j=1}^{n} u_j(t) \ell_j(x), \quad (8)$$

$$y(x, t) \simeq \tilde{y}_n(x, t) := \sum_{j=1}^{n} y_j(t) \hat{\ell}_j(x), \quad (9)$$

where $\ell_j(x), j = 1, \ldots, n$, are the Lagrange basis polynomials based on the Jacobi–Gauss nodes $\left\{\xi_i \right\}_{i=1}^{n}$, i.e.,

$$\ell_j(x) := \prod_{k=1 \atop k \neq j}^{n} \frac{x - \xi_k}{\xi_j - \xi_k}, \quad (10)$$

and

$$\hat{\ell}_j(x) := \frac{x(1 - x)}{\xi_j \left(1 - \xi_j\right)} \ell_j(x).$$

Note that $\ell_j(x)$ and $\hat{\ell}_j(x)$ satisfy the following Kronecker properties in the Jacobi–Gauss nodes:

$$\hat{\ell}_j(\xi_i) = \ell_j(\xi_i) = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j, \end{cases} \quad j = 1, \ldots, n. \quad (11)$$

Moreover, $\hat{\ell}_j(0) = \hat{\ell}_j(1) = 0, j = 1, \ldots, n$. Therefore, for each value of $y_j(t), j = 1, \ldots, n$, the approximation $\tilde{y}_n(x, t)$ satisfies the boundary conditions $(1c)$.

By substituting the approximations $(8)$ and $(9)$ in equation $(1a)$, and by simple algebraic manipulation, we get

$$\sum_{j=1}^{n} y_j(t) \hat{\ell}_j(x) = c(x, t) \left[ \sum_{j=1}^{n} y_j(t) D^2_x \hat{\ell}_j(x) + (1 - r) \sum_{j=1}^{n} u_j(t) \ell_j(x) \right] + f(x, t) + \sum_{j=1}^{n} u_j(t) \ell_j(x).$$

Now, by collocating the above equation in $\tilde{\ell}_j, i = 1, \ldots, n$, and using the Kronecker property $(11)$, we get

$$\hat{y}_i(t) = c(\xi_i, t) \left[ \sum_{j=1}^{n} y_j(t) D^2_x \hat{\ell}_j(\xi_i) + (1 - r) \sum_{j=1}^{n} u_j(t) \ell_j(\xi_i) \right] + f(\xi_i, t) + u_i(t). \quad (12)$$

Let

$$d_{ij}^+ := D^2_x \hat{\ell}_j(\xi_i), \quad d_{ij}^- := D^2_x \hat{\ell}_j(\xi_i). \quad (13)$$

Then, equation $(12)$ can be written as

$$\hat{y}_i(t) = c(\xi_i, t) \left[ \sum_{j=1}^{n} d_{ij}^+ y_j(t) + (1 - r) \sum_{j=1}^{n} d_{ij}^- y_j(t) \right] + f(\xi_i, t) + u_i(t), \quad i = 1, \ldots, n. \quad (14)$$

The above equations form a linear system of time-varying ordinary differential equations. To derive the vector form of the above system, let

$$y(t) := [y_1(t), \ldots, y_n(t)]^T, \quad u(t) := [u_1(t), \ldots, u_n(t)]^T, \quad f(t) := [f(\xi_1, t), \ldots, f(\xi_n, t)]^T, \quad C(t) := \text{diag}\left(c(\xi_1, t), \ldots, c(\xi_n, t)\right).$$

Using the above notations, the system of differential equations $(14)$ can be expressed as

$$\dot{y}(t) = C(t) \left[ r D^2_+ - (1 - r) D^2_- \right] y(t) + f(t) + u(t),$$

where $D^2_- \text{ and } D^2_-$ are called the left and right fractional differentiation matrices, respectively, and are defined as

$$D^2_- := \left[ d_{ij}^- \right]_{i=1, \ldots, n \atop j=1, \ldots, n}, \quad D^2_+ := \left[ d_{ij}^+ \right]_{i=1, \ldots, n \atop j=1, \ldots, n}.$$

The initial condition $(1b)$ is discretized to

$$y(0) = g := [g(\xi_1), \ldots, g(\xi_n)]^T. \quad (15)$$

Moreover, the inequality condition $(1c)$ on the control function is discretized into the following inequality constraints:

$$u(\xi_i, t) \geq u_{\min}(\xi_i, t), \quad i = 1, \ldots, n.$$

The above inequalities can be rewritten in vector form as

$$u(t) \geq u_{\min}(t), \quad (16)$$
where
\[ u_{\text{min}}(t) := \left[ u_{\text{min}}(\xi_1, t), \ldots, u_{\text{min}}(\xi_n, t) \right]^T. \]

Now, we turn to discretize the performance index. By using the Jacobi–Gauss quadrature rule (5), we approximate the inner integrals in the performance index (1d) as
\[
J_n[u] = \frac{1}{2} \int_0^T \left[ y(t) - z(t) \right]^T W \left[ y(t) - z(t) \right] dt + \frac{1}{2} \int_0^T u^T(t) W u(t) dt,
\]
where \( \omega^{(1, n)}_j \), \( j = 1, \ldots, n \), are the Jacobi–Gauss weights. We rewrite \( J_n \) as
\[
J_n[u] = \frac{1}{2} \int_0^T \left[ y(t) - z(t) \right]^T W \left[ y(t) - z(t) \right] dt + \frac{1}{2} \int_0^T u^T(t) W u(t) dt,
\]
where
\[
z(t) := \left[ \frac{z(\xi_1, t)}{\xi_1(1-\xi_1)} \cdots \frac{z(\xi_n, t)}{\xi_n(1-\xi_n)} \right]^T,
\]
\[ W := \frac{1}{2} \text{diag} \left( \omega^{(1, n)}_1, \ldots, \omega^{(1, n)}_n \right). \]

In summary, the Jacobi–Gauss pseudospectral spatial discretization, problem (1) is transcribed into the following classical optimal control problem:
\[
\begin{align*}
\text{min} J_n &= \frac{1}{2} \int_0^T \left[ y(t) - z(t) \right]^T W \left[ y(t) - z(t) \right] dt + \frac{1}{2} \int_0^T u^T(t) W u(t) dt, \\
\text{s.t.} & \quad \dot{y}(t) = C(t) \mathcal{D}_-^{2-\beta} y(t) + f(t) + u(t), \\
& \quad y(0) = g, \\
& \quad u(t) \geq u_{\text{min}}(t),
\end{align*}
\]
where \( \mathcal{D}_-^{2-\beta} := r \mathcal{D}^{2-\beta} + (1 - r) \mathcal{D}^{\beta} \).

**On the derivation of the left and right fractional differentiation matrices**

As we have seen, the fractional differentiation matrices \( \mathcal{D}_+^{2-\beta} \) and \( \mathcal{D}_-^{2-\beta} \) have a crucial role in the proposed Jacobi–Gauss pseudospectral method. These matrices simplify the discretization process by replacing left and right fractional differentiations with matrix-vector products. At first glance, it seems that by using (13), the \((i, j)\)-th component of the differentiation matrix \( \mathcal{D}_+^{2-\beta} / \mathcal{D}_-^{2-\beta} \) can be easily computed by taking the analytical left/right fractional derivative of \( \ell_j(x) \) and evaluating it at collocation points \( \xi_i \). However, taking the analytical fractional derivative, especially for large \( n \), is not practical and accessible. Accordingly, we need stable and accurate methods for generating these matrices. In this respect, in what follows we present some lemmas and theorems, which play key roles in computing the left and right fractional differentiation matrices in an accurate and stable method.

In the sequel, we use \( \hat{P}_k^{(a,b)}(x) \), \( k = 0, 1, \ldots \), to denote the shifted Jacobi polynomials, which are defined as
\[
\hat{P}_k^{(a,b)}(x) = P_k^{(a,b)}(2x - 1), \quad k = 0, 1, \ldots.
\]

**Lemma 1.** For \( j = 1, \ldots, n \), we have
\[
\ell_j(x) = \sum_{k=1}^n \lambda_{jk} x(1-x) \hat{P}_k^{(1,1)}(x),
\]
where
\[
\lambda_{jk} := \frac{(2k+1)(k+2)}{(k+1)(1-\xi_j)} \hat{P}_k^{(1,1)}(\xi_j).
\]

**Proof.** We first note that \( \ell_j(x) \), defined in (10), is a polynomial of degree \( n-1 \). Thus, we can expand it in terms of the shifted Jacobi polynomials \( \hat{P}_k^{(1,1)}(x) \), \( k = 1, \ldots, n \), as follows:
\[
\ell_j(x) = \sum_{k=1}^n \lambda_{jk} \hat{P}_k^{(1,1)}(x).
\]

Multiplying both sides of the above equation by \( x(1-x) \hat{P}_k^{(1,1)}(x) \), then integrating both sides of the resulted equation from 0 to 1 and using the orthogonality property (2), we finally get that
\[
\lambda_{jk} := \frac{(2k+1)(k+2)}{(k+1)} \int_0^1 x(1-x) \ell_j(x) \hat{P}_k^{(1,1)}(x) dx.
\]

Now, by noting that \( \ell_j(x) \hat{P}_k^{(1,1)}(x) \) is a polynomial of degree at most \( 2n-2 \), we can compute exactly the integral in the right-hand side of (21) by using the Jacobi–Gauss quadrature rule (5) with \( a = b = 1 \). In this way, we have
\[
\lambda_{jk} := \frac{(2k+1)(k+2)}{(k+1)} \sum_{i=1}^n \omega_i \ell_j(\xi_i) \hat{P}_k^{(1,1)}(\xi_i).
\]

Finally, using the Kronecker property (11), \( \lambda_{jk} \) is obtained as
\[
\lambda_{jk} := \frac{(2k+1)(k+2)}{(k+1)} \omega_j \hat{P}_k^{(1,1)}(\xi_j).
\]

The proof is complete by multiplying both sides of equation (20) by \( \frac{x(1-x)}{\xi_j(1-\xi_j)} \).

**Theorem 4.** The \((i, j)\)-th element of the left fractional differentiation matrix \( \mathcal{D}_+^{2-\beta} \) can be obtained explicitly as
\[
d_+^{i,j} \lambda_{jk} \xi_k \hat{\ell}_j(\xi_k) = \sum_{k=1}^n \lambda_{jk} \xi_k \hat{\ell}_j(\xi_k),
\]
where \( \hat{\ell}_j(x) \) are the shifted Jacobi–Gauss nodes defined in (7) and
\[
\hat{\ell}_j(x) := \frac{\Gamma(k+1)}{\Gamma(k-1+\beta)} x^{\beta-1} \hat{P}_k^{(3-\beta,\beta-1)}(x)
- \frac{\Gamma(k+2)}{\Gamma(k+\beta)} x^{\beta} \hat{P}_k^{(3-\beta,\beta)}(x) - \frac{\Gamma(k+1)}{\Gamma(k+1+\beta)} x^{\beta} \hat{P}_k^{(3-\beta,\beta)}(x).
\]

**Proof.** Using Lemma 1, we have
\[
d_+^{i,j} \hat{\ell}_j(x) = \sum_{k=1}^n \lambda_{jk} d_+^{i,k} \hat{\ell}_j(x) x(1-x) \hat{P}_k^{(1,1)}(x).
\]
With \( a = 1 \) and \( b = 2 \), the property (3) gives
\[
x(1-x)P_{k-1}^{(1,1)}(x) = xP_{k-1}^{(1,1)}(x) - x^2 P_{k-1}^{(2,1)}(x)
\]
\[
= xP_{k-1}^{(1,1)}(x) - \frac{k+2}{2k+1} x^2 P_{k-1}^{(2,1)}(x) - \frac{k}{2k+1} x^2 P_{k-2}^{(1,2)}(x).
\]

Using the above equation and Theorem 1, we get
\[
\frac{d}{dx} x^{2-}\beta \left[x(1-x)\tilde{P}_{k-1}^{(1,1)}(x)\right]
\]
\[
= \frac{\Gamma(k+1)}{\Gamma(k-1+\beta)} \frac{\Gamma(3-\beta,1-\beta)}{\Gamma(k-1+\beta)} + \frac{\Gamma(k+2)}{\Gamma(k-1+\beta)} \frac{\Gamma(3-\beta,1-\beta)}{\Gamma(k-1+\beta)} + \frac{k}{2k+1} x^2 P_{k-2}^{(1,2)}(x).
\]

We complete the proof by substituting the above equation in (22).

**Theorem 5.** The \((i,j)\)-th element of the right element of the right

differentiation matrix \(D^{2-}\beta\) is obtained explicitly as
\[
d_{ij}^{-} := \frac{d}{dx} x^{2-}\beta \tilde{\ell}_i(\xi_i) = \sum_{k=1}^{n} \lambda_{jk} \zeta_k(1-\xi_i),
\]
where
\[
\zeta_k(x) := \frac{\Gamma(k+1)}{\Gamma(k+1+\beta)} x^{2-}\beta P_{k-1}^{(3-\beta,1-\beta)}(x)
\]
\[
- \frac{\Gamma(k+2)}{\Gamma(k+1+\beta)} x^{2-}\beta P_{k-1}^{(3-\beta,1-\beta)}(x) - \frac{k}{2k+1} \frac{\Gamma(3-\beta,1-\beta)}{\Gamma(k-1+\beta)} x^2 P_{k-2}^{(1,2)}(x).
\]

**Proof.** The proof is fairly similar to the proof of Theorem 4. At first, by using property (4) for \( a = 2 \) and \( b = 1 \), we get
\[
x(1-x)P_{k-1}^{(1,1)}(2x-1)
\]
\[
= (1-x)P_{k-1}^{(1,1)}(2x-1) - (1-x)^2 P_{k-1}^{(2,1)}(2x-1)
\]
\[
= (1-x)^2 P_{k-1}^{(2,1)}(2x-1) - \frac{k+2}{2k+1} (1-x)^2 P_{k-2}^{(2,1)}(2x-1) - \frac{k}{2k+1} (1-x)^2 P_{k-2}^{(2,1)}(2x-1).
\]

Using the above equation and Theorem 2, we conclude that
\[
\frac{d}{dx} x^{2-}\beta \left[x(1-x)\tilde{P}_{k-1}^{(1,1)}(2x-1)\right]
\]
\[
= \frac{\Gamma(k+1)}{\Gamma(k+1+\beta)} (1-x)^{2-}\beta P_{k-1}^{(3-\beta,1-\beta)}(2x-1)
\]
\[
- \frac{\Gamma(k+2)}{\Gamma(k+1+\beta)} (1-x)^{2-}\beta P_{k-1}^{(3-\beta,1-\beta)}(2x-1)
\]
\[
- \frac{k}{2k+1} (1-x)^2 P_{k-2}^{(2,1)}(2x-1)
\]
\[
= \zeta_k(1-x).
\]

The proof is immediately completed by collocating the above equation at points \( \xi_i, i = 1, \ldots, n \).

**Step II: The Legendre–Gauss–Radau pseudospectral method for time discretization**

In the second step of our method, the optimal control problem (17) is solved numerically. This problem can be solved by any well-developed method, such as direct and indirect shooting methods. However, in this paper, we use the Legendre–Gauss–Radau pseudospectral method (Garg et al. 2010). Let \( m \) be a positive integer number and \( \tilde{\tau}_j, j = 1, \ldots, m \), be the Legendre–Gauss–Radau points in the interval \([0, T]\), i.e., \( \tilde{\tau}_j := \frac{T}{2} \left( \tau_j^{(0,0)} + 1 \right) \). Note that \( \tilde{\tau}_1 = 0 \) and \( \tilde{\tau}_j < T, j = 2, \ldots, m \). We define the extra node \( \tilde{\tau}_{m+1} = T \). Now, to solve the optimal control (17), we approximate the state function \( y(t) \) as
\[
y(t) \approx \sum_{j=1}^{m+1} y_j \tilde{\ell}_j(t),
\]
where \( \tilde{\ell}_j(t), j = 1, \ldots, m, m + 1, \) are the Lagrange polynomials based on \( \{\tilde{\tau}_j\}_{j=1}^{m+1} \) and \( y_1, \ldots, y_{m+1} \) are unknown \( n \)-vectors. From the Kronecker property, and using the initial condition (17a), the vector \( y_1 \) can be obtained explicitly as
\[
y_1 = g.
\]

By substituting the approximation (24) in the dynamic equation (17b), we get
\[
\sum_{j=1}^{m+1} y_j \frac{d}{dt} \tilde{\ell}_j(t) = C(t) D^{2-\beta} \sum_{j=1}^{m+1} y_j \tilde{\ell}_j(t) + f(t) + u(t).
\]

Now, by collocating the above equation in \( \tilde{\tau}_i \) for \( i = 1, \ldots, m \), we get
\[
\sum_{j=1}^{m+1} y_j \frac{d}{dt} \tilde{\ell}_j(\tilde{\tau}_i) = C(\tilde{\tau}_i) D^{2-\beta} \sum_{j=1}^{m+1} y_j \tilde{\ell}_j(\tilde{\tau}_i) + f(\tilde{\tau}_i) + u_i, \quad i = 1, \ldots, m,
\]

where \( u_i := u(\tilde{\tau}_i) \). It is worthwhile to note that \( \tilde{x}_{m+1} \) is used in approximating \( y(t) \), however, this point is not used as a collocation point. Using the Kronecker property, from the above equation, we get that
\[
\sum_{j=1}^{m+1} y_j \tilde{d}_{ij} = C(\tilde{\tau}_i) D^{2-\beta} y_i + f(\tilde{\tau}_i) + u_i, \quad i = 1, \ldots, m,
\]

where \( \tilde{d}_{ij} = \frac{d}{dt} \tilde{\ell}_j(\tilde{\tau}_i) \). According to Baltensperger and Trummer (2003), for stable and accurate computation of \( \tilde{d}_{ij} \), \( i = 1, \ldots, m, j = 1, \ldots, m+1 \), we can use the following formula:
\[
\tilde{d}_{ij} = \begin{cases} \lambda_j, & i \neq j, \\ \sum_{k=1, k \neq i}^{m} \tilde{d}_{ik}, & i = j, \end{cases}
\]
where \( \lambda_i = \prod_{k=1, k \neq i}^{m} \tilde{\tau}_i - \tilde{\tau}_k \).

Next, the path constraint (17d) is collocated at the \( m \) collocation points \( \tilde{\tau}_1, \ldots, \tilde{\tau}_m \) and, finally, the following inequality constraints are obtained:
\[
u_i \geq u_{\min}(\tilde{\tau}_i), \quad i = 1, \ldots, m.
\]

By using the Legendre–Gauss–Radau quadrature (6) with \( a = b = 0 \), the performance index (17a) is approximated by
\[
J_{n,m} = \frac{1}{2} \sum_{j=1}^{m} \left[ \sum_{j=1}^{m} e_{\tilde{\tau}_j}^{(n,m)} [y_j - z(\tilde{\tau}_j)]^{T} W_j [y_j - z(\tilde{\tau}_j)] \right]
\]
\[
+ \frac{1}{2} \sum_{j=1}^{m} e_{\tilde{\tau}_j}^{(n,m)} u_j^{T} W_j u_j,
\]
where \( e_{\tilde{\tau}_j}^{(n,m)}, j = 1, \ldots, m \), are the Legendre–Gauss–Radau weights.

In summary, by applying the Legendre–Gauss–Radau method, the optimal control problem (17) is transcribed into...
the following optimization problem:

\[
\min J_{n,m} = \frac{1}{2} \sum_{j=1}^{m} T \omega_{jn}^m [y_j - z(\tilde{r}_j)]^T W [y_j - z(\tilde{r}_j)] + \frac{1}{2} \sum_{j=1}^{m} T \omega_{jn}^m u_j^T W u_j,
\]

(28a)

\[
\text{s.t. } \sum_{j=1}^{m+1} y_j d_{ij} = C(\tilde{r}_i) D_{2-\beta} y_i + f(\tilde{r}_i) + u_i, \quad i = 1, \ldots, m,
\]

(28b)

\[
y_1 = g,
\]

(28c)

\[
u_1 \geq u_{\min}(\tilde{r}_i), \quad i = 1, \ldots, m,
\]

(28d)

where the decision variables are the \(n\)-vectors \(u_i\), \(i = 1, \ldots, m\), and \(y_j\), \(i = 1, \ldots, m+1\). We note that the objective function is quadratic and the constraints are affine, that is, problem (28) is a quadratic programming problem (QP).

**Converting the QP into standard form**

As we have seen, with our method the numerical solution of problem (1) is reduced to solving the QP (28). However, this QP is not in standard form. In the following, we derive the standard form of (28), which helps us to easily analyze and utilize a solver on it.

At first, we consider the objective function (28a) and expand it as

\[
J_{n,m} = \frac{1}{2} \sum_{j=1}^{m} T \omega_{jn}^m y_j^T W y_j + \frac{1}{2} \sum_{j=1}^{m} T \omega_{jn}^m u_j^T W u_j - \sum_{j=1}^{m} T \omega_{jn}^m z(\tilde{r}_j)^T W y_j + \frac{1}{2} \sum_{j=1}^{m} T \omega_{jn}^m z(\tilde{r}_j)^T W z(\tilde{r}_j). \tag{29}
\]

If we define the matrices \(Y \in \mathbb{R}^{n \times (m+1)}\), \(U \in \mathbb{R}^{n \times m}\), \(Z \in \mathbb{R}^{m \times m}\), the \(n\)-vector \(\omega\) and the \(m\)-vector \(\varpi\) as

\[
Y := \begin{bmatrix} y_1 & \ldots & y_m & y_{m+1} \end{bmatrix}, \quad U := \begin{bmatrix} u_1 & \ldots & u_m \end{bmatrix},
\]

\[
Z := \begin{bmatrix} z(\tilde{r}_1, \tilde{r}_2) & \ldots & z(\tilde{r}_{m-1}, \tilde{r}_m) \end{bmatrix}, \quad \omega := \begin{bmatrix} \omega_{11} & \ldots & \omega_{n1} \\ \omega_{12} & \ldots & \omega_{n2} \\ \vdots & \ddots & \vdots \\ \omega_{1m} & \ldots & \omega_{nm} \end{bmatrix}^T.
\]

then the objective function (29) can be rewritten as

\[
J_{n,m} = \frac{1}{2} \text{vec}(Y)^T \bar{S} \text{vec}(Y) + \frac{1}{2} \text{vec}(U)^T \text{vec}(U) - \text{vec}(Z)^T \bar{S} \text{vec}(Y) + \frac{1}{2} \text{vec}(Z)^T \text{vec}(Z),
\]

where \(S := \text{diag} (\varpi \otimes \omega)\), \(\bar{S} := \begin{bmatrix} S & 0_{n \times m} \\ 0_{m \times n} & 0_{n \times n} \end{bmatrix}\).

If we collect all decision variables in a vector \(v\) as

\[
v := \begin{bmatrix} \text{vec}(Y) \\ \text{vec}(U) \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_{m+1} \\ u_1 \\ \vdots \\ u_m \end{bmatrix}, \tag{30}
\]

then the performance index (28a) can be expressed as the following function of \(v\):

\[
J_{n,m}(v) = \frac{1}{2} v^T H v + c^T v + c_0,
\]

where

\[
H := \begin{bmatrix} \text{diag} (\varpi \otimes \omega) & 0_{nm \times n} \\ 0_{m \times n} & 0_{nm \times nm} \end{bmatrix},
\]

\[
c := -\begin{bmatrix} \bar{S} \text{vec}(Z) \\ 0_{m \times 1} \end{bmatrix}, \quad c_0 := \frac{1}{2} \text{vec}(Z)^T S \text{vec}(Z).
\]

In order to reformulate the constraints of the QP (28) as standard linear constraints, we define \(D \in \mathbb{R}^{m \times (m+1)}\), \(C \in \mathbb{R}^{m \times m}\) and \(F \in \mathbb{R}^{m \times (m+1)}\) as follows:

\[
D := \begin{bmatrix} d_{jk} \end{bmatrix}_{k=1,\ldots,m} \quad \text{for } j = 1, \ldots, m,
\]

\[
C := \begin{bmatrix} c(\xi, \tilde{r}_j) \end{bmatrix}_{j=1,\ldots, n} \quad \text{for } i = 1, \ldots, m,
\]

\[
F := \begin{bmatrix} f(\xi, \tilde{r}_j) \end{bmatrix}_{j=1,\ldots, n}.
\]

Considering the above notations, we can write all \(m\) equations in (26) in the following matrix equation:

\[
Y D^T = C \odot (D_{2-\beta} [Y]_{1:m}) + F + U, \tag{31}
\]

where \(\odot\) refers to element-wise or Hadamard product and \([Y]_{1:m}\) denotes the matrix obtained by removing the last column of \(Y\). Let \(I_n\) denote the identity matrix of dimension \(n\) and \(I_m\) denote the matrix obtained by removing the last column of the identity matrix \(I_{m+1}\). By noting that \(Y D^T = I_n Y D^T\) and \([Y]_{1:m} = Y I_m\), the matrix equation (31) can be rewritten as

\[
I_n Y D^T = C \odot (D_{2-\beta} Y I_m) + F + U.
\]

Now, by using Theorem 3, the above matrix equation is converted to the following linear system of equations:

\[
(D \otimes I_n) \text{vec}(Y) = I_C (I_m \odot D_{2-\beta}) \text{vec}(Y) + \text{vec}(U) + \text{vec}(F),
\]

where \(I_C := \text{diag} (\text{vec}(C))\). By utilizing (30), we express the above system of equations as

\[
[D \otimes I_n] \text{vec}(Y) = I_m \odot D_{2-\beta} \text{vec}(Y) \quad \text{and} \quad I_{nm} v = \text{vec}(F).
\]

The constraints (28c) and (28d) are expressed as

\[
\begin{bmatrix} I_{m} & 0_{n \times n(m-1)} & 0_{n \times nm} \\ 0_{nm \times n} & I_{nm} \end{bmatrix} v \geq \text{vec}(U_{\min}),
\]

where

\[
U_{\min} := \begin{bmatrix} u_{\min}(\xi, \tilde{r}_j) \end{bmatrix}_{j=1,\ldots, n}.
\]

In summary, the standard form of the QP (28) is obtained as

\[
\begin{align*}
\min J_{n,m} &= \frac{1}{2} v^T H v + c^T v + c_0, \\
\text{s.t.} & \quad A v = b, \\
& \quad B v \leq h,
\end{align*}\tag{32}
\]
where
\[
A := \begin{bmatrix}
I_n & 0_{n \times n(m-1)} \\
(D \otimes I_n) - I_C & 0_{n \times nm}
\end{bmatrix},
\]
\[
b := \left[\frac{g}{\text{vec}(F)}\right], \quad B := -\left[0_{n(m+1) \times n(m+1)} \mid I_{nm}\right],
\]
\[
h := -\text{vec}(U_{\min}).
\]

Note that the dimension of \(A\) is \(n(m+1) \times n(2m+1)\), where the number of non-zero elements is \(nm(n+m+1)+n\). We conclude that the percentage of non-zero elements decreases dramatically when the size of the matrix \(A\) is increased. For instance, the percentage of non-zero elements of \(A\) obtained with \(n = m = 10\) and \(n = m = 100\) is 9.1% and 1%, respectively. Thus, matrix \(A\) is a sparse matrix. The sparsity pattern of the matrix \(A\), which is obtained with \(n = m = 10\), is plotted in Figure 1.

The sparsity of the QP (32) is a suitable property from the optimization point of view. Another important and desirable property in optimization is convexity. We emphasize that sparsity, and especially convexity, help to solve the obtained problem efficiently by well-developed algorithms, such as interior-point methods. In the next theorem, we show that the obtained problem (32) is convex.

**Theorem 6.** The optimization problem (32) is a convex problem.

**Proof.** Since the quadrature weights \(\omega_i^{j,n}, i = 1, \ldots, n,\) and \(\bar{\omega}_j^{m,n}, j = 1, \ldots, m,\) are positive, it follows that \(H\) is a diagonal matrix with non-negative diagonal elements (Gautschi 1996). Consequently, \(H\) is a positive semi-definite matrix. As a result, the objective function is a convex function. Moreover, the equality and inequality constraints of problem (32) are affine. As a result, problem (32) is a convex quadratic program.

**Numerical experiments**

This section is devoted to illustrate the presented pseudospectral method using numerical experiments. We have implemented our method using MATLAB on a 3.5 GHz Core i7 personal computer with 8 GB of RAM. Moreover, for solving the QP (32), the solver IPOPT (Wächter and Biegler 2006) was used. In IPOPT, we can adjust the accuracy of the solution through the input parameter `tolrfun`. In our numerical experiments, we set `tolrfun=1e-12`.

We consider five different examples. The first example was treated in Du et al. (2016) and has an exact solution. We use this example to assess the accuracy of our method. The second example is considered in Wu and Huang (2018) and has a high variation in its solution. With this example we check the efficiency of our method. The third example is a new problem with state constraints and shows that, with little changes in the presented method, we can easily solve hard problems. With the last two examples, we investigate the effect of parameter \(\beta\) on the behavior of the solution and the performance of our method.

**Example 1**

In this example, which is taken from Du et al. (2016), the problem (1) with the following data is considered:

\[
\beta = 0.2, \quad r = 0.8, \quad T = 1, \quad g(x) = 0, \quad u_{\min}(x,t) = 1,
\]
\[
c(x,t) = \frac{1+x(1-x)}{10}, \quad f(x,t) = 1, \quad z(x,t) = 1 + \frac{x}{1.17},
\]
\[
u_{\min}(x,t) = \max \left\{ x e^{-2(x-0.5)} \sin(2\pi(1-x) t^{0.6}) \right\}.
\]

The resulted control and state functions by the presented method, with \(n = m = 100\), are plotted in Figure 2. The obtained control and state functions are in good agreement with the results of Wu and Huang (2018). To report the efficiency and precision of the method, we give in Table 2 the CPU time and the obtained values of the performance index.

**Example 2**

This example, with the following data, is treated in Wu and Huang (2018):

\[
\beta = 0.5, \quad r = 0.25, \quad T = 30, \quad g(x,0) = 1,
\]
\[
c(x,t) = \frac{1+x(1-x)}{10}, \quad f(x,t) = 1, \quad z(x,t) = 1 + \frac{x}{1.17},
\]
\[
u_{\min}(x,t) = \max \left\{ x e^{-2(x-0.5)} \sin(2\pi(1-x) t^{0.6}) \right\}.
\]

The resulted control and state functions by the presented method, with \(n = m = 100\), are plotted in Figure 3. The obtained control and state functions are in good agreement with the results of Wu and Huang (2018). To report the efficiency and precision of the method, we give in Table 2 the CPU time and the obtained values of the performance index.
Figure 1. The sparsity pattern of the matrix \( A \) with \( n = m = 10 \).

Figure 2. Control, state and error functions obtained by the proposed method with \( n = m = 50 \) (Example 1).

Table 1. Consumed CPU time and obtained norms of errors for various values of \( n \) and \( m \) together with the results of Du et al. (2016) (Example 1).

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
\( n \) & \( m \) & CPU & \( E_{n,m}(J) \) & \( E_{n,m}^\infty(u) \) & \( E_{n,m}^\infty(y) \) & \( N \) & \( CPU \) & \( \|u_{ex} - u_h\|_2 \) & \( \|y_{ex} - y_h\|_2 \) \\
\hline
10  & 10  & 0.7 & 1.24e-3 & 3.67e-5 & 1.38e-4 & 7.01e-6 & 2.18e-5 & 32 & 0.25 & 1.8638e-4 & 9.5396e-5 \\
20  & 20  & 0.6 & 5.71e-4 & 9.36e-7 & 5.23e-6 & 7.65e-8 & 3.27e-7 & 64 & 0.52 & 4.7116e-5 & 2.4422e-5 \\
30  & 30  & 1.0 & 1.48e-5 & 9.24e-8 & 3.91e-7 & 5.56e-9 & 2.50e-8 & 128 & 0.92 & 1.4296e-5 & 6.1531e-6 \\
40  & 40  & 2.5 & 2.24e-6 & 1.98e-8 & 8.79e-8 & 2.33e-9 & 6.82e-9 & 256 & 1.85 & 2.9647e-5 & 1.5343e-6 \\
50  & 50  & 7.1 & 6.07e-7 & 6.28e-9 & 4.84e-8 & 2.30e-9 & 6.62e-9 & 512 & 3.77 & 7.5488e-7 & 3.7709e-7 \\
60  & 60  & 14.9 & 1.00e-7 & 2.50e-9 & 2.34e-8 & 2.37e-9 & 6.51e-9 & 1024 & 7.57 & 2.9799e-7 & 9.3049e-8 \\
\hline
\end{tabular}
\end{table}

\section*{Example 3}

Now we consider the problem of Example 1 subject to the extra state constraint \( y(x, t) \geq y_{\min}(x, t) \), where

\[ y_{\min}(x, t) := \sqrt{\max \{0, 0.1 - (x - 0.5)^2 - (t - 0.5)^2\}}. \]

The numerical solution of this problem with indirect methods is troublesome because the derivation of necessary optimality conditions for such optimal control problems
with state constraints is difficult. Moreover, the numerical solution of the resulted necessary optimality conditions is complicated. However, extending our direct method to solve such problems is straightforward. We just need to add the following bound constraints to the QP (28):

$$y_j \geq y_{\min}(\bar{\tau}_j), \quad j = 1, \ldots, m + 1,$$

where $y_{\min}(t) := [y_{\min}^{(\xi_1)}(t), \ldots, y_{\min}^{(\xi_m)}(t)]^T$. Imposing the above constraints is simply done by changing the definition of $\mathbf{B}$ and $\mathbf{h}$ in the QP (32) as follows:

$$\mathbf{B} := -[\mathbf{I}_{n(m+1)} \mid \mathbf{I}_{nm}], \quad \mathbf{h} := -[\text{vec}(\mathbf{Y}_{\min})]_{\text{vec}(\mathbf{U}_{\min})},$$

where

$$\mathbf{Y}_{\min} := [y_{\min}^{(\xi_i)}(\bar{\tau}_j)]_{i=1,\ldots,n, j=1,\ldots,n+1}.$$

By applying our direct method with $n = m = 100$, the obtained control and state functions are plotted in Figure 4. Moreover, function $y_{\min}(x, t)$ is plotted beside the state function $y_{100}(x, t)$. It is clear that in this example, contrary to Example 1, the state function is not zero and lies above function $y_{\min}(x, t)$.

### Example 4

Consider now problem (1) with

$$r = 0.5, \quad T = 3, \quad y(x, 0) = \sin(\pi x), \quad c(x, t) = 1, \quad f(x, t) = 0, \quad z(x, t) = 0.5, \quad u_{\min}(x, t) = 0.$$

In Figure 5, the obtained solutions with $n = m = 60$ on this problem, for $\beta = 0.1, 0.5, 0.9$, are plotted. Moreover, the values of the first and second terms in the performance index (1d) are reported as well. It is seen, for smaller values of $\beta$, that the obtained state $\tilde{y}_{100}$ is not close to the target function $z(x, t) = \frac{1}{2}$. Indeed, when the value of $\beta$ is small, then a large magnitude control function is needed to generate a state function close to $z(x, t)$. In other words, much energy is needed to force $y(x, t)$ to become close to $z(x, t)$. As a result, in these cases, the optimal control is close to its lower bound and the corresponding state function is not close to the target function.

### Example 5

As a final example, let

$$r = 0.5, \quad T = 3, \quad y(x, 0) = x^4(1 - x)^4, \quad u_{\min}(x, t) = 1, \quad c(x, t) = 1, \quad z(x, t) = e^x x^4(1 - x)^4,$$

$$f(x, t) = e^x [x^4(1 - x)^4 - 12 (\tilde{f}(x) + \tilde{f}(1 - x))],$$

where

$$\tilde{f}(x) = \frac{1680 (x^6 + \beta)}{7(7 + \beta)} - \frac{840 x^5 + \beta}{(6 + \beta)} + \frac{180 x^4 + \beta}{5(3 + \beta)} - \frac{20 x^3 + \beta}{4(3 + \beta)} + \frac{x^2 + \beta}{3(3 + \beta)}.$$

The exact solution is given by

$$u_{\text{ex}} := 0, \quad y_{\text{ex}} := e^{x^4(1 - x)^4}, \quad J_{\text{ex}} = 1.5.$$

We applied the presented method on this problem with different values of $\beta$. The obtained errors $E_n^2(y)$ and $E_n^2(J)$, for various values of $n = m$, are reported in Table 3. As we see, when $\beta$ is close to 1, the accuracy is decreased slightly. This fact is predictable. Indeed, if $\beta$ is close to 1, then equation (1a) tends to the advection equation, which is a difficult problem from the numerical point of view. However, we see that our method can obtain a solution with 8 digits of accuracy, even for large values of $\beta$. 

### Table 2. Consumed CPU time and obtained values of the performance index for various values of $n$ and $m$ (Example 2)

| $n$ | $m$ | CPU Time | $J_{n,m}$ |
|-----|-----|----------|-----------|
| 20  | 20  | 0.6 s    | 17.3176450|
| 40  | 40  | 3.2 s    | 17.3055445|
| 60  | 60  | 22.7 s   | 17.3028343|
| 80  | 80  | 100.6 s  | 17.3028684|
| 100 | 100 | 311.6 s  | 17.3028411|
the left and right fractional differentiation matrices were polynomials, accurate and stable procedures for deriving the optimal control of a two-sided space-fractional diffusion equation. Thanks to useful properties of the Jacobi polynomials, accurate and stable procedures for deriving the left and right fractional differentiation matrices were presented. In our method, the solution of the problem reduces to the solution of a convex quadratic programming problem. Five examples were solved and results reported. These results show that the fully direct pseudospectral method is efficient and provides accurate results, whereas a small number of collocation points is used and a low CPU time is consumed. Moreover, our third example shows that the method here introduced can be also applied with success to

Conclusions

We proposed a fully direct pseudospectral method to solve the optimal control of a two-sided space-fractional diffusion equation. Thanks to useful properties of the Jacobi polynomials, accurate and stable procedures for deriving the left and right fractional differentiation matrices were presented. In our method, the solution of the problem reduces to the solution of a convex quadratic programming problem. Five examples were solved and results reported. These results show that the fully direct pseudospectral method is efficient and provides accurate results, whereas a small number of collocation points is used and a low CPU time is consumed. Moreover, our third example shows that the method here introduced can be also applied with success to

\begin{table}[h]
\centering
\begin{tabular}{ccccccc}
\hline
\(n = m\) & \(\beta = 0.1\) & \(\beta = 0.3\) & \(\beta = 0.5\) & \(\beta = 0.7\) & \(\beta = 0.9\) \\
\hline
3 & 1.7e-02 & 1.4e-02 & 1.0e-02 & 6.6e-03 & 2.7e-03 \\
4 & 4.9e-03 & 3.8e-03 & 2.7e-03 & 1.7e-03 & 6.7e-04 \\
5 & 5.3e-04 & 4.5e-04 & 3.7e-04 & 2.8e-04 & 1.5e-04 \\
6 & 1.2e-04 & 1.2e-04 & 1.1e-04 & 9.1e-05 & 6.0e-05 \\
7 & 8.5e-08 & 8.9e-08 & 8.9e-08 & 8.7e-08 & 8.2e-08 \\
8 & 7.2e-09 & 7.4e-09 & 8.0e-09 & 8.1e-09 & \\
9 & 9.6e-10 & 1.3e-09 & 2.2e-09 & 4.1e-09 & 9.4e-09 \\
10 & 8.5e-10 & 1.3e-09 & 2.1e-09 & 4.1e-09 & 9.5e-09 \\
\hline
\end{tabular}
\caption{Obtained norms of errors for various values of \(\beta\) and \(n = m\). (Example 5).}
\end{table}
### Acknowledgements

Torres has been partially supported by FCT (The Portuguese National Science Foundation) through the R&D unit CIDMA, project UID/MAT/04106/2013. Bozorgnia was supported by FCT fellowship SFRH/BPD/33962/2009.

The authors are very grateful to three anonymous referees for carefully reading their manuscript and for several comments and suggestions which helped them to improve the paper.

### References

- Agarwal RP, Baleanu D, Nieto JJ, Torres DFM and Zhou Y (2018) A survey on fuzzy fractional differential and optimal control nonlocal evolution equations. *Journal of Computational and Applied Mathematics* 339: 3–29. arXiv:1709.07766
- Bai Y, Baleanu D and Wu GC (2018) Existence and discrete approximation for optimization problems governed by fractional differential equations. *Communications in Nonlinear Science and Numerical Simulation* 59: 338–348.
- Baltensperger R and Trummer MR (2003) Spectral differencing with a twist. *SIAM Journal on Scientific Computing* 24(5): 1465–1487.
- Behroozifar M and Habibi N (2018) A numerical approach for solving a class of fractional optimal control problems via operational matrix Bernoulli polynomials. *JVC/Journal of Vibration and Control* 24(12): 2494–2511.
- Benson DA, Huntington GT, Thorvaldsen TP and Rao AV (2006) Direct trajectory optimization and costate estimation via an orthogonal collocation method. *Journal of Guidance, Control, and Dynamics* 29(6): 1435–1440.
- Bhray A and Zaky M (2015) A method based on the Jacobi tau approximation for solving multi-term timespace fractional partial differential equations. *Journal of Computational Physics* 281: 876–895.
- Bhray A and Zaky M (2016) Shifted fractional-order Jacobi orthogonal functions: Application to a system of fractional differential equations. *Applied Mathematical Modelling* 40(2): 832–845.
- Chen S, Liu F, Turner I and Anh V (2018) A fast numerical method for two-dimensional Riesz space fractional diffusion equations on a convex bounded region. *Applied Numerical Mathematics* 134: 66–80. DOI:10.1016/j.apnum.2018.07.007.
- Darchmiraki M, Farahi M and Effati S (2018) Solution for fractional distributed optimal control problem by hybrid meshless method. *JVC/Journal of Vibration and Control* 24(11): 2149–2164.
- Dehghan M, Hamedi EA and Khosravian-Arab H (2016) A numerical scheme for the solution of a class of fractional variational and optimal control problems using the modified Jacobi polynomials. *JVC/Journal of Vibration and Control* 22(6): 1547–1559.
- Doha E, Bhrawy A, Baleanu D and Ezz-Eldien S (2014) The operational matrix formulation of the Jacobi tau approximation for space fractional diffusion equation. *Advances in Difference Equations* 2014(1): 1–14.
- Dorville R, Mophou GM and Valmorin VS (2011) Optimal control of a nonhomogeneous Dirichlet boundary fractional diffusion equation. *Computers & Mathematics with Applications* 62(3): 1472–1481.
- Du N, Wang H and Liu W (2016) A fast gradient projection method for a constrained fractional optimal control. *Journal of Scientific Computing* 68(1): 1–20.
- Elmaghraby M, Karam M and Razzaghi M (1995) The pseudospectral Legendre method for discretizing optimal control problems. *IEEE Transactions on Automatic Control* 40(10): 1793–1796.
- Esmaeili S and Shamsi M (2011) A pseudo-spectral scheme for the approximate solution of a family of fractional differential equations. *Communications in Nonlinear Science and Numerical Simulation* 16(9): 3646–3654.
- Esmaeili S, Shamsi M and Luchko Y (2011) Numerical solution of fractional differential equations with a collocation method based on Müntz polynomials. *Computers & Mathematics with Applications* 62(3): 918–929.
- Ezz-Eldien S, Doha E, Baleanu D and Bhrawy A (2017) A numerical approach based on Legendre orthonormal polynomials for numerical solutions of fractional optimal control problems. *JVC/Journal of Vibration and Control* 23(1): 16–30.
- Fahroo F and Ross IM (2001) Costate estimation by a Legendre pseudospectral method. *Journal of Guidance, Control, and Dynamics* 24(2): 270–277.
- Feng L, Liu F, Turner I, Yang Q and Zhuang P (2018) Unstructured mesh finite difference/finite element method for the 2D timespace Riesz fractional diffusion equation on irregular convex domains. *Applied Mathematical Modelling* 59: 441–463. DOI:10.1016/j.apm.2018.01.044.
- Foroozandeh Z, Shamsi M, Azhmyakov V and Shafiee M (2017) A modified pseudospectral method for solving trajectory optimization problems with singular arc. *Mathematical Methods in the Applied Sciences* 40(5): 1783–1793.
- Foroozandeh Z, Shamsi M and de R de Pinho M (2018) A mixed-binary non-linear programming approach for the numerical solution of a family of singular optimal control problems. *International Journal of Control* DOI:10.1080/00207179.2017.1399216.
- Garg D (2011) *Advances in global pseudospectral methods for optimal control*. PhD Thesis, University of Florida.
- Garg D, Patterson M, Hager WW, Rao AV, Benson DA and Huntington GT (2010) A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica* 46(11): 1843–1851.
- Gautschi W (1996) *Orthogonal Polynomials: Applications and Computation*. in A. Iserles (Ed.), *Acta Numerica*, Cambridge Univ. Press.
- Khakzar-Esghagh M and Shamsi M (2017) Direct pseudo-spectral method for optimal control of obstacle problem – an optimal control problem governed by elliptic variational inequality. *Mathematical Methods in the Applied Sciences* 40(13): 4993–5004.
- Laub A (2005) *Matrix Analysis for Scientists and Engineers*. Society for Industrial and Applied Mathematics.
- Li S and Zhou Z (2018) Legendre pseudo-spectral method for optimal control problem governed by a time-fractional
Mophou GM (2011) Optimal control of fractional diffusion equation. *International Journal of Computer Mathematics* 95(6-7): 1308–1325. DOI:10.1080/00207160.2017.1417591.

Malinowska AB and Torres DFM (2012) Introduction to the fractional calculus of variations. Imperial College Press, London.

Mashayekhi S and Razzaghi M (2018) An approximate method for solving fractional optimal control problems by hybrid functions. *JVC/Journal of Vibration and Control* 24(9): 1621–1631.

Meerschaert MM and Tadjeran C (2006) Finite difference approximations for two-sided space-fractional partial differential equations. *Applied Numerical Mathematics* 56(1): 80–90.

Mohammadzadeh E, Pariz N, Hosseini Sani S and Jajarmi A (2018) An efficient numerical method for the optimal control of fractional-order dynamic systems. *JVC/Journal of Vibration and Control* DOI:10.1177/1077546317751755.

Mophou G and N’Guerekata GM (2011) Optimal control of a fractional diffusion equation. *Computers & Mathematics with Applications* 62(3): 1413–1426.

Mophou GM (2011) Optimal control of fractional diffusion equation with state constraints. *Computers & Mathematics with Applications* 62(5): 668–676. arXiv:1205.4843

Pooleh S, Almeida R and Torres DFM (2013) Discrete direct methods in the fractional calculus of variations. *Computers & Mathematics with Applications* 66(5): 668–676. arXiv:1805.06537

Salati AB, Shamsi M and Torres DFM (2019) Direct transcription methods based on fractional integral approximation formulas for solving nonlinear fractional optimal control problems. *Communications in Nonlinear Science and Numerical Simulation* 67: 334–350. DOI:10.1016/j.cnsns.2018.05.011. arXiv:1805.06537

Tang X, Shi Y and Wang LL (2017) A new framework for solving fractional optimal control problems using fractional pseudospectral methods. *Automatica* 78: 333–340.

Wächter A and Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* 106(1, Ser. A): 25–57.

Wu GC, Baleanu D, Deng ZG and Zeng SD (2015) Lattice fractional diffusion equation in terms of a Riesz-Caputo difference. *Physica A: Statistical Mechanics and its Applications* 438: 335–339. DOI:10.1016/j.physa.2015.06.024.

Wu GC, Baleanu D, Xie HP and Zeng SD (2017) Lattice fractional diffusion equation of random order. *Mathematical Methods in the Applied Sciences* 40(17): 6054–6060. DOI:10.1002/mma.3644.

Wu SL and Huang TZ (2018) A fast second-order parareal solver for fractional optimal control problems. *Journal of Vibration and Control* 24(15): 3418–3433. DOI:10.1177/1077546317705557.

Yamamoto M (2018) Weak solutions to non-homogeneous boundary value problems for time-fractional diffusion equations. *Journal of Mathematical Analysis and Applications* 460(1): 365–381.

Yang XJ and Machado J (2017a) A new fractional operator of variable order: Application in the description of anomalous diffusion. *Physica A: Statistical Mechanics and its Applications* 481: 276–283.

Yang XJ, Machado J and Baleanu D (2017b) Anomalous diffusion models with general fractional derivatives within the kernels of the extended Mittag-Leffler type functions. *Romanian Reports in Physics* 69(4).

Yang XJ, Machado J, Baleanu D and Gao F (2016) A new numerical technique for local fractional diffusion equation in fractal heat transfer. *Journal of Nonlinear Science and Applications* 9(10): 5621–5628.

Yang XJ, Srivastava H, Torres D and Debbouche A (2017b) General fractional-order anomalous diffusion with non-singular power-law kernel. *Thermal Science* 21: S1–S9.

Yang Z, Yuan Z, Nie Y, Wang J, Zhu X and Liu F (2017c) Finite element method for nonlinear Riesz space fractional diffusion equations on irregular domains. *Journal of Computational Physics* 330: 863–883. DOI:10.1016/j.jcp.2016.10.053.

Zaky M (2018) A Legendre collocation method for distributed-order fractional optimal control problems. *Nonlinear Dynamics* 91(4): 2667–2681.

Zaky M, Doha E and Machado JT (2018) A spectral framework for fractional variational problems based on fractional Jacobi functions. *Applied Numerical Mathematics* 132: 51–72. DOI:10.1016/j.apnum.2018.05.009.

Zaky M, Ezz-Eldien S, Doha E, Machado J and Bhrawy A (2016) An efficient operational matrix technique for multidimensional variable-order time fractional diffusion equations. *Journal of Computational and Nonlinear Dynamics* 11(6).

Zaky MA and Machado JAT (2017) On the formulation and numerical simulation of distributed-order fractional optimal control problems. *Communications in Nonlinear Science and Numerical Simulation* 52: 177–189.

Zayernouri M and Karniadakis GE (2013) Fractional Sturm-Liouville eigen-problems: Theory and numerical approximation. *Journal of Computational Physics* 252: 495–517.