



Generalized Lagrange–Jacobi–Gauss–Radau collocation method for solving a nonlinear optimal control problem with the classical diffusion equation

Sobhan Latifi^{1,a}, Kourosh Parand^{2,3,b}, Mehdi Delkhosh^{4,c}

¹ Department of Computer Sciences, Amirkabir University of Technology, Tehran, Iran

² Department of Computer Sciences, Shahid Beheshti University, G.C. Tehran, Iran

³ Department of Cognitive Modeling, Institute for Cognitive and Brain Sciences, Shahid Beheshti University, G.C. Tehran, Iran

⁴ Department of Mathematics and Computer Science, Islamic Azad University, Bardaskan Branch, Bardaskan, Iran

Received: 4 May 2020 / Accepted: 7 October 2020 / Published online: 15 October 2020

© Società Italiana di Fisica and Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract In this paper, a well-known nonlinear optimal control problem (OCP) arisen in a variety of biological, chemical, and physical applications is considered. The quadratic form of the nonlinear cost function is endowed with the state and control functions. In this problem, the dynamic constraint of the system is given by a classical diffusion equation. This article is concerned with a generalization of Lagrange functions. Besides, a generalized Lagrange–Jacobi–Gauss–Radau (GLJGR) collocation method is introduced and applied to solve the aforementioned OCP. Based on initial and boundary conditions, the time and space variables, t and x , are clustered with Jacobi–Gauss–Radau points. Then, to solve the OCP, Lagrange multipliers are used and the optimal control problem is reduced to a parameter optimization problem. Numerical results demonstrate its accuracy, efficiency, and versatility of the presented method.

1 Introduction

In order to present OCP solved in this manuscript, firstly, we give an introduction to the OCP and provide an explanation of the functions and parameters defined in this problem. A brief review and history of these equations and spectral and pseudospectral methods are in the following subsections.

1.1 The governing equations

Optimum control problems arise in the minimization of a functional over a set of admissible control functions subject to dynamic constraints on the state and control functions [1, 2]. Regarding that the equations of dynamics in the system are reformed by a partial differential

^a e-mail: s_latifi@aut.ac.ir

^b e-mail: k_parand@sbu.ac.ir

^c e-mail: mehdidelkhosh@yahoo.com (corresponding author)

equation—with time and space variables—this OCP is known as an optimal control of a distributed system [1]. The formulation of this optimal control problem is [3]:

$$\min J = \frac{1}{2} \int_0^1 \int_0^R x^r (c_1 z^2(x, t) + c_2 y^2(x, t)) dx dt, \quad (1)$$

subject to

$$\frac{\partial z}{\partial t} = k \left(\frac{\partial^2 z}{\partial x^2} + \frac{r}{x} \frac{\partial z}{\partial t} \right) + y(x, t), \quad (2)$$

with initial and boundary conditions

$$\begin{aligned} z(x, 0) &= z_0(x), \quad 0 < x < R, \\ z(R, t) &= 0, \quad t > 0. \end{aligned} \quad (3)$$

In fact, this is a nonlinear quadratic optimal control problem with the dynamic system of the classical diffusion equation.

In Eqs. (1)–(3), $y(x, t)$ is the optimization function that must be calculated so that it and the function of $z(x, t)$ are held in Eqs. (1)–(3) and the value of J is minimized. $z(x, t)$ and $y(x, t)$ are called the state and control functions and usually calculated in such a way that they are smooth and the derivatives and integral in Eqs. (1) and (2) can be calculated analytically or numerically. $z_0(x)$ is a known real function, c_1 and c_2 are two arbitrary constants, k is a numerical constant that is specified in the problem, and r and R are positive real constants. Here, $t > 0$ and its upper bound is 1. The parameter r is specified in numerical examples as $r = 1$ or $r = 2$.

The purpose of solving this problem is to approximate the control and state functions that minimize the J .

This OCP arises in a variety of biological, chemical, and physical applications; for more details, see Refs. [4–6].

1.2 The literature on optimal control problems

Two-dimensional (2D) systems and their beneficial applications in many different industrial fields draw the attention of scientists presently. These applications rise in heat transfer, image processing, seismological and geophysical data processing, distributed systems, restoration of noisy images, earthquake signal processing, water stream heating gas absorption, smart materials, and transmission lines [7–10]. The miscellaneous chemical, biological, and physical problems are modeled by diffusion processes involving control function mentioned in Eqs. (1)–(3). By the aid of Roesser's model [10], Attasi's model [11, 12], Fornasini–Marchesini's models [13, 14], the state-space models of 2D systems are organized [7]. Remarkable studies have been done in the area of optimal controls, and excellent articles are written hereby [15–19]. Among these studies, numerical techniques have been used to solve optimal control problems [17, 20]. Moreover, Agrawal [1] presented a general formulation and a numerical scheme for fractional optimal control for a class of distributed systems. He used eigenfunctions to develop his method. In other works, Manabe [21], Bode et al. [22] and Rabiee et al. [23] studied fractional-order optimal control problems. Additionally, Mamehrashia and Yousefi [3, 7] and Lotfi et al. [24] employed the Ritz method to solve optimal control problems. With the variational method, Yousefi et al. [25] found the solution of the optimal control of linear systems, approximately. Li et al. [26] considered a continuous-time 2D system and converted it to the discrete-time 2D model. In other works, Wei et al. [27] and Zhang et al. [28] investigated an optimal control problem in continuous-time 2D Roesser's model. They

employed iterative adaptive critic design algorithm and the adaptive dynamic programming method to approximate the solution. Sabeh et al. [29] introduced a pseudo-spectral method for the solution of distributed optimal control problem with the viscous Burgers equation.

1.3 The literature of spectral and pseudospectral methods

The main feature of spectral methods is to use different orthogonal polynomials/functions as trial functions. These polynomials/functions are global and infinitely differentiable. These methods are applied to four types of problems: periodic problems, nonperiodic problems, whole-line problems, and half-line problems. These can be categorized as: Trigonometric polynomials for periodic problems; classical Jacobi, ultraspherical, Chebyshev and Legendre polynomials for non-periodic problems; Hermite polynomials for problems on the whole line; and Laguerre polynomials for problems on the half line [30]. With the truncated series of smooth global functions, spectral methods are giving the solution of a particular problem [31,32]. These methods, with a relatively limited number of degrees of freedom, provide such an accurate approximation for a smooth solution. Spectral coefficients tend to zero faster than any algebraic power of their index n ; see page 2 in [33].

Spectral methods can fall into three categories: Tau, Collocation and Galerkin methods [34].

- The Tau spectral method is used to approximate numerical solutions of various differential equations. This method considers the solution as an expansion of orthogonal polynomials/functions. Such coefficients, in this expansion, are set to approximate the solution correctly [35].
- Collocation method helps obtain highly accurate solutions to nonlinear/linear differential equations; see page 2 in [36–39]. Two common steps in collocation methods are: First, suitable nodes (Gauss/Gauss–Radau/Gauss–Lobatto) are selected to restate a finite or discrete form of the differential equations. Second, a system of algebraic equations from the discretization of the original equation is achieved [40,41].
- In the Galerkin spectral method, trial and test functions are chosen the same [42]; This method can result in a highly accurate solution.

It is said that spectral Galerkin methods are similar to Tau methods where in approximation by Tau method, the differential equation is enforced [33].

Furthermore, some other numerical methods like finite difference method (FDM) and finite element method (FEM) need network construction of data and they perform locally. Although spectral methods are continuous and globally performing, they do not require network construction of data [43].

In addition to spectral methods, pseudospectral methods have also attracted researchers recently [44–47]. As mentioned previously, Sabeh et al. [29] investigated a pseudospectral method to solve the optimal control problem. Pseudospectral methods are also utilized in the solution of other optimal control problems as well [48–51]. These methods become popular because of their computational feasibility and efficiency. In fact, in standard pseudospectral methods, interpolation operators are used to reduce the cost of computation of the inner product we encounter in some of the spectral methods. For this purpose, a set of distinct interpolation points $\{x_i\}_{i=0}^n$ is considered by which the corresponding Lagrange interpolants are achieved. Besides, when applying collocation points, $\{x_i\}_{i=0}^n$, the residual function is set to vanish on the same set of points. Notwithstanding, the collocation points do not need to be chosen the same as the interpolation nodes. Indeed, just for having the Kronecker delta property, they are considered to be the same: as a consequence, this property helps reduce computational cost noticeably as well. There are such authors who utilized pseudospec-

tral methods for the solution of optimal control as well. William [52] introduced a Jacobi pseudospectral method for solving an optimal control problem. He reported that significant differences in computation time can be seen for different parameters of the Jacobi polynomial. Jacobi pseudospectral method endowed with Lagrange basis is also in [53] to solve a problem defined in semi-infinite domain problem. In another work in [54], a set of nonlinear PDEs was solved by applying this Jacobi pseudospectral method. Garge et al. [50] presented a unified framework for the numerical solution of optimal control problems using collocation at Legendre–Gauss (LG), Legendre–Gauss–Radau (LGR), and Legendre–Gauss–Lobatto (LGL) points and discussed the advantages of each for solving optimal control problems. Chebyshev pseudospectral method was utilized by Fahroo et al. [49] to provide an optimal solution for the optimal control problem.

1.4 The main aim of this paper

To the best of our knowledge, the use of pseudospectral methods for solving optimal control problems has been limited in the literature to either Chebyshev or Legendre methods. Noteworthy, the pseudospectral method based on Jacobi can encompass a wide range of other pseudospectral methods since the Legendre and Chebyshev nodes can be obtained as particular cases of the general Jacobi. Therefore, examining different choices of the parameters in the Jacobi polynomials and a proper selection of the Jacobi parameters will allow us to obtain more accurate real-time solutions to nonlinear optimal control problems. Meanwhile, an arbitrary and not precise selection of nodes may result in poor interpolation characteristics such as the Runge phenomenon; therefore, the nodes in pseudospectral methods are selected as the Gauss–Radau points [52].

In this paper, we present a general formulation and a suitable numerical method called the GLJGR collocation method to solve OCP for a class of distributed systems. Using GLJGR collocation method helps lessen the computational and CPU time. The developed method is exponentially accurate and obtained by a generalization of classical Lagrange polynomials [55]. Additionally, the equation of the dynamics of optimal control problem is reformed as a partial differential equation.

This paper is arranged as follows: In Sect. 2, we present some preliminaries and drive some tools for introducing GL function, GLJGR collocation method, and their relevant derivative matrices. In Sect. 3, we apply the GLJGR collocation method to the solution of the OCP. Section 4 shows numerical examples to demonstrate the effectiveness of the proposed method. Also, a conclusion is given in Sect. 5.

2 Preliminaries, conventions, and notations

In this section, we review some necessary definitions and relevant properties of Jacobi polynomials. In the next step, we introduce generalized Lagrange (GL) functions. Then, we state some theorems on GL functions and develop GLJGR collocation method. Finally, in terms of GLJGR collocation method, we give a formula that expresses the derivative matrix of the mentioned functions.

2.1 Jacobi polynomials

A basic property of the Jacobi polynomials is that they are the eigenfunctions to a singular Sturm–Liouville problem. Jacobi polynomials are defined on $[-1, 1]$ and are of high interest recently [56–59]. The following recurrence relation generates the Jacobi polynomials [60]:

$$P_{k+1}^{\alpha,\beta}(x) = (a_k^{\alpha,\beta}x - b_k^{\alpha,\beta})P_k^{\alpha,\beta} - c_k^{\alpha,\beta}P_{k-1}^{\alpha,\beta}(x), \quad k \geq 1$$

$$P_0^{\alpha,\beta}(x) = 1, \quad P_1^{\alpha,\beta}(x) = \frac{1}{2}(\alpha + \beta + 2)x + \frac{1}{2}(\alpha - \beta),$$

where

$$a_k^{\alpha,\beta} = \frac{(2k + \alpha + \beta + 1)(2k + \alpha + \beta + 2)}{2(k + 1)(k + \alpha + \beta + 1)},$$

$$b_k^{\alpha,\beta} = \frac{(\beta^2 - \alpha^2)(2k + \alpha + \beta + 1)}{2(k + 1)(k + \alpha + \beta + 1)(2k + \alpha + \beta)},$$

$$c_k^{\alpha,\beta} = \frac{(k + \beta)(k + \alpha)(2k + \alpha + \beta + 2)}{(k + 1)(k + \alpha + \beta + 1)(2k + \alpha + \beta)},$$

The Jacobi polynomials satisfy the following identities:

$$P_n^{\alpha,\beta}(-x) = (-1)^n P_n^{\beta,\alpha}(x),$$

$$P_n^{\alpha,\beta}(-1) = \frac{(-1)^n \Gamma(n + \beta + 1)}{n! \Gamma(\beta + 1)},$$

$$P_n^{\alpha,\beta}(1) = \frac{\Gamma(n + \alpha + 1)}{n! \Gamma(\alpha + 1)},$$

$$\left(P_n^{\alpha,\beta}(x) \right)^{(m)} = 2^{-m} \frac{\Gamma(m + n + \alpha + \beta + 1)}{\Gamma(n + \alpha + \beta + 1)} P_{n-m}^{\alpha+m,\beta+m}(x),$$

and its weight function is $w^{\alpha,\beta}(x) = (1 - x)^\alpha (1 + x)^\beta$.

Moreover, the Jacobi polynomials are orthogonal on $[-1, 1]$:

$$\int_{-1}^1 P_n^{\alpha,\beta}(x) P_m^{\alpha,\beta}(x) w^{\alpha,\beta}(x) dx = \delta_{m,n} \gamma_n^{\alpha,\beta},$$

$$\gamma_n^{\alpha,\beta} = \frac{2^{\alpha+\beta+1} \Gamma(n + \alpha + 1) \Gamma(n + \beta + 1)}{(2n + \alpha + \beta + 1) \Gamma(n + 1) \Gamma(n + \alpha + \beta + 1)},$$

where $\delta_{m,n}$ is the Kronecker delta function and $L^2_{w^{\alpha,\beta}}[-1, 1]$ is the weighted space. The inner product and the norm of $L^2_{w^{\alpha,\beta}}[-1, 1]$ with respect to the weight function are defined as:

$$(g, h)_{w^{\alpha,\beta}} = \int_{-1}^1 g(x)h(x)w^{\alpha,\beta}(x)dx, \quad \|g\|_{w^{\alpha,\beta}} = (g, g)_{w^{\alpha,\beta}}^{\frac{1}{2}}$$

It is noted that the set of Jacobi polynomials forms a complete $L^2_{w^{\alpha,\beta}}[-1, 1]$ system [36].

It is known that the $(n + 1)$ points of Gauss–Radau are the n roots of $P_n^{\alpha,\beta+1}(x)$ over $(-1, 1)$ and one particular root as -1 or 1 . Similarly, for the interval $[a,b]$ with the shifting parameter $u(x)$, these points are n roots of $P_n^{\alpha,\beta+1}(u(x))$ over (a, b) and one particular root as a or b on the boundaries, where more details on $u(x)$ are discussed in the following sections.

2.2 Generalized Lagrange (GL) functions

In this section, generally, the GL functions are introduced and the suitable formulas for the first- and second-order derivative matrices of these functions are presented.

Definition 1 Let $w(x) = \prod_{i=0}^N (u(x) - u(x_i))$, then the generalized Lagrange (GL) formula is shown as [55]

$$L_j^u(x) = \frac{w(x)}{(u - u_j)\partial_u w(x_j)} = \frac{u'_j w(x)}{(u - u_j)\partial_x w(x_j)} = \kappa_j \frac{w(x)}{(u - u_j)}, \tag{4}$$

where $\partial_u w(x) = \frac{1}{u'} \partial_x w(x)$, $\kappa_j = \frac{u'_j}{\partial_x w(x_j)}$, $u(x)$ is a continuous, arbitrary, and sufficiently differentiable function, x_i s are arbitrary distinct points on domain of $u(x)$ such that $u'_i \neq 0$ for any $i = 0, \dots, N$ and $u_i \neq u_j$ for all $i \neq j$.

For simplicity, $u = u(x)$ and $u_i = u(x_i)$ are considered.

Theorem 1 Considering the GL functions $L_j^u(x)$ in Eq. (4), one can exhibit the first-order derivative matrices of GL functions as

$$D^{(1)} = [d_{kj}] \in \mathfrak{R}^{(n+1) \times (n+1)}, \quad 0 \leq j, k \leq n,$$

where

$$d_{kj} = \begin{cases} \kappa_j \frac{\partial_x w(x_k)}{u_k - u_j}, & j \neq k, \\ \kappa_j \frac{u'_j \partial_x^2 w(x_j) - u''_j \partial_x w(x_j)}{2u'_j{}^2}, & j = k. \end{cases}$$

Proof As the GL functions defined in Eq. (4), the first-order derivative formula for the case $k \neq j$ can be achieved as follows:

$$d_{kj} = \partial_x L_j^u(x_k) \stackrel{Eq.(4)}{=} \kappa_j \frac{\partial_x w(x_k)(u_k - u_j) - u'_k w(x_k)}{(u_k - u_j)^2} = \kappa_j \frac{\partial_x w(x_k)}{u_k - u_j}. \tag{5}$$

But, when $k = j$, with L'Hopital's rule:

$$\begin{aligned} d_{jj} &= \partial_x L_j^u(x_j) = \lim_{x \rightarrow x_j} \kappa_j \frac{(u - u_j)\partial_x w(x) - u'w(x)}{(u - u_j)^2} \stackrel{\text{Hop}}{=} \\ &\lim_{x \rightarrow x_j} \kappa_j \frac{(u - u_j)\partial_x^2 w(x) - u''w(x)}{2u'(u - u_j)} \stackrel{\text{Hop}}{=} \kappa_j \frac{u'_j \partial_x^2 w(x_j) - u''_j \partial_x w(x_j)}{2u'_j{}^2}. \end{aligned}$$

This completes the proof. □

Theorem 2 Let $D^{(1)}$ be the above matrix (first-order derivative matrix of GL functions) and define matrix Q such that $Q = \text{Diag}(u'_0, u'_1, \dots, u'_N)$, $Q^{(1)} = \text{Diag}(u''_0, u''_1, \dots, u''_N)$; then, the second-order derivative matrix of GL functions can be formulated as:

$$D^{(2)} = (Q^{(1)} + QD^{(1)})Q^{-1}D^{(1)}. \tag{6}$$

Proof See Ref. [55]. □

For simplicity, from now on, $D^{(1)}$ is considered D .

2.3 Generalized Lagrange–Jacobi–Gauss–Radau (GLJGR) collocation method

It is a well-established fact that a proper choice of collocation points is crucial in terms of accuracy and computational stability of the approximation by Lagrange basis [61]. As an appropriate choice of such collocation points, we can refer to the well-known Gauss–Radau points in which points lie inside (a,b) and one point is clustered near the endpoints [29,61]. In this sequel, we use Jacobi–Gauss–Radau nodes.

In the case of GLJGR collocation method, $w(x)$ in Eq. (4) can be considered as two approaches: in the first approach, one point is clustered near the end of the interval:

$$w(x) = \lambda P_n^{\alpha,\beta+1}(u)(u - u_n), \tag{7}$$

and in the second approach, one point at the beginning of the interval is clustered:

$$w(x) = \lambda(u - u_0)P_n^{\alpha,\beta+1}(u), \tag{8}$$

where λ is a real constant. As a matter of simplification, we write

$$G(u) = P_n^{\alpha,\beta+1}(u), \tag{9}$$

with the following important properties:

$$\partial_x G(u) = u' \frac{\Gamma(\alpha + \beta + n + 3)}{2\Gamma(\alpha + \beta + n + 2)} P_{n-1}^{\alpha+1,\beta+2}(u), \tag{10}$$

$$\partial_x^2 G(u) = \frac{\Gamma(\alpha + \beta + n + 3)}{4\Gamma(\alpha + \beta + n + 2)} \left(2u'' P_{n-1}^{\alpha+1,\beta+2}(u) + (u')^2(\alpha + \beta + n + 3) P_{n-2}^{\alpha+2,\beta+3}(u) \right), \tag{11}$$

Let us speak of the first approach as in Eq. (7). Assuming

$$w(x) = \lambda P_n^{\alpha,\beta+1}(u)(u - u_n),$$

then we have:

$$\partial_x w(x) = \lambda [u' G(u) + (u - u_n) \partial_x G(u)], \tag{12}$$

$$\partial_x^2 w(x) = \lambda [u'' G(u) + 2u' \partial_x G(u) + (u - u_n) \partial_x^2 G(u)]. \tag{13}$$

$$G(u_n) = \frac{\Gamma(\alpha + 1 + n)}{(n)! \Gamma(\alpha + 1)}, \quad \partial_x G(u_n) = u'_n \frac{\Gamma(\alpha + \beta + n + 3)}{2\Gamma(\alpha + \beta + n + 2)} \frac{\Gamma(\alpha + 1 + n)}{(n - 1)! \Gamma(\alpha + 2)}. \tag{14}$$

We can rewrite the formula in Theorem 1 for a Jacobi–Gauss–Radau specific case: It is easy to do this by setting $w(x) = \lambda(u - u_0)P_n^{\alpha,\beta+1}(u)$ and recalling that $\{P_n^{\alpha,\beta+1}(u_j) = 0\}_{j=0}^{n-1}$. So, by this, the formula in Theorem 1 can be obtained as a Jacobi–Gauss–Radau case, and the entry of the first-order derivative matrix of GL functions will be:

$$d_{kj} = \begin{cases} \frac{u'_k P_{n-1}^{\alpha+1, \beta+2}(u_k)(n!(\alpha+\beta+n+2)\Gamma(\alpha+1))}{2\Gamma(\alpha+1+n)}, & j = n, 0 \leq k \leq n-1, \\ \frac{2u'_k \Gamma(\alpha+n+1)}{\Gamma(\alpha+1)(n!(\alpha+\beta+n+2)P_{n-1}^{\alpha+1, \beta+2}(u_j)(u_k-u_j)(u_j-u_k))}, & k = n, 0 \leq j \leq n-1, \\ \frac{u'_k(u_k-u_n)}{(u_j-u_n)(u_k-u_j)} \frac{P_{n-1}^{\alpha+1, \beta+2}(u_k)}{P_{n-1}^{\alpha+1, \beta+2}(u_j)}, & 0 \leq j \neq k \leq n-1, \\ \frac{u'_j}{u_j-u_n} + u'_j(\alpha+\beta+n+3) \frac{P_{n-3}^{\alpha+3, \beta+3}(u_j)}{4P_{n-2}^{\alpha+2, \beta+2}(u_j)}, & 0 \leq k = j \leq n-1, \\ \frac{u'_k(n-1)(\alpha+\beta+n+2)}{2(\alpha+1)}, & k = j = n, \end{cases}$$

Similar to this fashion, for the second approach shown in Eq. (8), one can write:

$$\begin{aligned} w(x) &= \lambda(u-u_0)P_n^{\alpha, \beta+1}(u), \\ G(u_0) &= \frac{\Gamma(\beta+2+n)(-1)^n}{(n)!\Gamma(\beta+2)}, \quad \partial_x G(u_0) \\ &= u'_0 \frac{\Gamma(\alpha+\beta+n+3)}{2\Gamma(\alpha+\beta+n+2)} \frac{(-1)^{n-1}\Gamma(\beta+2+n)}{(n-1)!\Gamma(\beta+3)}. \end{aligned} \tag{15}$$

Now this is obvious that $\{P_n^{\alpha, \beta+1}(u_j) = 0\}_{j=1}^n$. Therefore, by the second approach, the entries of defined matrix D can be filled as:

$$d_{kj} = \begin{cases} \frac{u'_k P_{n-1}^{\alpha+1, \beta+2}(u_k)(n!(\alpha+\beta+n+2)\Gamma(\beta+2))}{2\Gamma(\beta+2+n)(-1)^n}, & j = 0, 1 \leq k \leq n, \\ \frac{2u'_k \Gamma(\beta+n+2)(-1)^n}{\Gamma(\beta+2)(n!(\alpha+\beta+n+2)P_{n-1}^{\alpha+1, \beta+2}(u_j)(u_k-u_j)(u_j-u_k))}, & k = 0, 1 \leq j \leq n, \\ \frac{u'_k(u_k-u_0)}{(u_j-u_0)(u_k-u_j)} \frac{P_{n-1}^{\alpha+1, \beta+2}(u_k)}{P_{n-1}^{\alpha+1, \beta+2}(u_j)}, & 1 \leq j \neq k \leq n, \\ \frac{u'_j}{u_j-u_0} + u'_j(\alpha+\beta+n+3) \frac{P_{n-3}^{\alpha+3, \beta+3}(u_j)}{4P_{n-2}^{\alpha+2, \beta+2}(u_j)}, & 1 \leq k = j \leq n, \\ \frac{-u'_k(n)(\alpha+\beta+n+2)}{2(\beta+2)}, & k = j = 0, \end{cases}$$

More specifically, Legendre, Chebyshev, and ultraspherical polynomials can be obtained as special cases from the proposed method. These cases are summarized in the following remark:

Remark 1 We have all the mentioned formulas of GL functions, i.e., $D, D^{(2)}$ for Gegenbauer (ultraspherical), Legendre, Chebyshev (the first, second, third, and fourth kinds) polynomials by simply setting $\alpha = \beta, \alpha = \beta = 0, \alpha = \beta = -0.5, \alpha = \beta = 0.5, \alpha = -0.5, \beta = 0.5$, and $\alpha = 0.5, \beta = -0.5$, respectively. These specifications can be named generalized Lagrange–[Gegenbauer, Legendre, Chebyshev (first, second, third, fourth)]–Gauss–Radau for that manner.

2.4 Operational matrix of GL functions

In this subsection, we intend to clarify that when it comes to the collocation method, the operational matrix of derivative for GL functions is equal to transpose of the derivative matrix mentioned in 2.3.

Defining the one-column vectors

$$\hat{L}_n^u(x) = \begin{bmatrix} L_0^u(x) \\ L_1^u(x) \\ \vdots \\ L_n^u(x) \end{bmatrix}, \hat{H}_n^u(x) = \begin{bmatrix} \partial_x L_0^u(x) \\ \partial_x L_1^u(x) \\ \vdots \\ \partial_x L_n^u(x) \end{bmatrix}, C = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix},$$

for the approximation of a function $\xi(x)$, one can write

$$\xi(x) \approx \sum_{i=0}^n c_i L_i^u(x) = C^T \hat{L}_n^u(x),$$

and similarly for the derivative of this function, one can rewrite it as

$$\partial_x \xi(x) = C^T K \hat{L}_n^u(x), \tag{16}$$

where $K \in \mathfrak{R}^{(n+1) \times (n+1)}$ is the operational matrix of derivative where

$$K \hat{L}_n^u(x) = \hat{H}_n^u(x),$$

and in other words,

$$\begin{bmatrix} K_{00} & K_{01} & \dots & K_{0n} \\ K_{10} & K_{11} & \dots & K_{1n} \\ \vdots & & & \\ K_{n0} & K_{n1} & \dots & K_{nn} \end{bmatrix} \begin{bmatrix} L_0^u(x) \\ L_1^u(x) \\ \vdots \\ L_n^u(x) \end{bmatrix} = \begin{bmatrix} \partial_x L_0^u(x) \\ \partial_x L_1^u(x) \\ \vdots \\ \partial_x L_n^u(x) \end{bmatrix}.$$

To prove it, take the j th row

$$[K_{j0} \ K_{j1} \ \dots \ K_{jn}] \begin{bmatrix} L_0^u(x) \\ L_1^u(x) \\ \vdots \\ L_n^u(x) \end{bmatrix} = \partial_x L_j^u(x),$$

and by collocating the Jacobi–Gauss–Radau nodes $(\{x_i\}_{i=0}^n)$ in this equation, we obtain:

$$\begin{aligned} K_{ji} &= \partial_x L_j^u(x_i), \\ K_{ji} &= d_{ij}. \end{aligned}$$

This means that the operational matrix for these functions is

$$K = D^T, \tag{17}$$

where D is defined in Sect. 2.3. Similarly, for $\partial_{xx} \xi(x)$, one can also say

$$\partial_{xx} \xi(x) = C^T (D^{(2)})^T \hat{L}_n^u(x), \tag{18}$$

where $D^{(2)}$ is defined in Eq. (6).

3 Numerical method

The main objective of this section is to develop the GLJGR collocation method to solve OCP. In this section, firstly, a promising function approximation method has been presented. Then, GLJGR collocation is implemented so as to accomplish the introduction of the presented numerical method for OCP of interest.

3.1 Function approximation

If $H = L^2(\eta)$, $\eta = 0 \cup (0, 1) \times (0, 1) \cup 1$, where

$$s = \{L_i^u(x)L_j^u(t) \mid 0 \leq i \leq n, 0 \leq j \leq m\},$$

$s \subset H$, s is the set of GL functions product and

$$V_{nm} = Span\{L_i^u(x)L_j^u(t) \mid 0 \leq i \leq n, 0 \leq j \leq m\},$$

where V_{nm} is a finite-dimensional vector space. For any $q \in S$, one can find the best approximation of q in space V_{nm} as $P_{nm}(x, t)$ such that

$$\exists P_{nm}(x, t) \in V_{nm}, \forall l_{nm}(x, t) \in V_{nm}, \|q - P_{nm}(x, t)\|_2 \leq \|q - l_{nm}(x, t)\|_2,$$

where $\|f\|_2 = \sqrt{\langle f, f \rangle}$ and $\langle \cdot, \cdot \rangle$ is the inner product. Therefore, for any $P_{nm}(x, t) \in V_{nm}$, one can write

$$P_{nm}(x, t) = \sum_{i=0}^n \sum_{j=0}^m c_{ij} L_i^u(x)L_j^u(t) = (\hat{L}_n^u(x))^T C \hat{L}_m^u(t), \tag{19}$$

in which C is a matrix of $\mathfrak{R}^{(m+1) \times (n+1)}$, and c_{ij} are the relevant coefficients and calculable. $L_i^u(x)$ is considered by the first approach mentioned in Sect. 2.3 in which $u(x) = \frac{2}{R}x - 1$, and $L_j^u(t)$ is based on the second approach and considered as $u(t) = 2t - 1$. This can be proved that as n and m increase, the approximate solution tends to the exact solution [3].

3.2 Implementation of GLJGR collocation method for solving the OCP

Now, for the approximation of state and control functions

$$y(x, t) \approx y_{nm}(x, t) = \sum_{i=0}^n \sum_{j=0}^m a_{ij} L_i^u(x)L_j^u(t) = (\hat{L}_n^u(x))^T A \hat{L}_m^u(t), \tag{20}$$

$$z(x, t) \approx z_{nm}(x, t) = \sum_{i=0}^n \sum_{j=0}^m b_{ij} L_i^u(x)L_j^u(t) = (\hat{L}_n^u(x))^T B \hat{L}_m^u(t), \tag{21}$$

where

$$A = \begin{bmatrix} a_{00} & a_{01} & \dots & a_{0m} \\ a_{10} & a_{11} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n0} & a_{n1} & \dots & a_{nm} \end{bmatrix}, B = \begin{bmatrix} b_{00} & b_{01} & \dots & b_{0m} \\ b_{10} & b_{11} & \dots & b_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n0} & b_{n1} & \dots & b_{nm} \end{bmatrix}, \hat{L}_m^u(t) = \begin{bmatrix} L_0^u(t) \\ L_1^u(t) \\ \vdots \\ L_m^u(t) \end{bmatrix}, \hat{L}_n^u(x) = \begin{bmatrix} L_0^u(x) \\ L_1^u(x) \\ \vdots \\ L_n^u(x) \end{bmatrix},$$

We define residual functions $res(x, t)$ by substituting Eqs. (20) and (21) in Eq. (2)

$$res(x, t) = -\frac{x \partial z_{nm}(x, t)}{\partial t} + k \left(\frac{x \partial^2 z_{nm}(x, t)}{\partial x^2} + r \frac{\partial z_{nm}(x, t)}{\partial t} \right) + x y_{nm}(x, t); \tag{22}$$

in terms of Eqs. (17) and (18), one can write

$$\frac{\partial z_{nm}(x, t)}{\partial x} = \left(D^T \hat{L}_n^u(x) \right)^T B \hat{L}_m^u(t) = \hat{L}_n^u(x)^T D B \hat{L}_m^u(t), \tag{23}$$

$$\frac{\partial^2 z_{nm}(x, t)}{\partial x^2} = \left((D^{(2)})^T \hat{L}_n^u(x) \right)^T B \hat{L}_m^u(t) = \hat{L}_n^u(x)^T D^{(2)} B \hat{L}_m^u(t), \tag{24}$$

$$\frac{\partial z_{nm}(x, t)}{\partial t} = \hat{L}_n^u(x)^T B \hat{D}^T \hat{L}_m^u(t), \tag{25}$$

$A, B \in \mathfrak{R}^{(n+1) \times (m+1)}$, $D \in \mathfrak{R}^{(n+1) \times (n+1)}$ and $\hat{D} \in \mathfrak{R}^{(m+1) \times (m+1)}$ so Eq. (22) can be restated as

$$\begin{aligned} res(x, t) = & -x \hat{L}_n^u(x)^T B \hat{D}^T \hat{L}_m^u(t) + k \left(x \hat{L}_n^u(x)^T D^{(2)} B \hat{L}_m^u(t) \right. \\ & \left. + r \hat{L}_n^u(x)^T B \hat{D}^T \hat{L}_m^u(t) \right) + x \hat{L}_n^u(x)^T A \hat{L}_m^u(t), \end{aligned} \tag{26}$$

and the initial and boundary conditions of the problem are obtained as

$$z_{nm}(x, 0) \approx z_0(x), \quad 0 < x < R, \quad z_{nm}(R, t) \approx 0, \quad t > 0, \tag{27}$$

and within the assumption of Eqs. (21) and (20)

$$\hat{L}_n^u(x)^T B \hat{L}_m^u(0) \approx z_0(x), \quad \hat{L}_n^u(R)^T B \hat{L}_m^u(t) \approx 0. \tag{28}$$

As 0 and R are the t_0 and x_n , with the aid of the characteristic of Lagrange polynomials, we write

$$\hat{L}_n^u(x)^T B \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \hat{L}_n^u(x)^T B[:, 0], \quad [0 \ 0 \ \dots \ 1] B \hat{L}_m^u(t) = B[n, :] \hat{L}_m^u(t), \tag{29}$$

where $B[:, 0]$ and $B[n, :]$ are the first column and $(n + 1)$ th row of Matrix B .

Then, with $(n + 1)$ collocation nodes in x space and $(m + 1)$ collocation nodes in t space, a set of algebraic equations is constructed by using Eq. (26) together with the conditions in Eq. (28). (Simplification of Eq. (29) is also used.)

$$\begin{cases} F[i] = \hat{L}_n^u(x_i)^T B[:, 0] - z_0(x_i) = B_{i0} - z_0(x_i), \quad i = 0, \dots, n, \\ F[n + j] = B[n, :] \hat{L}_m^u(t_j) = B_{nj}, \quad j = 1, \dots, m, \\ F[n + m + (i)(m + 1) + j + 1] = res(x_i, t_j), \quad i = 0, \dots, n, \quad j = 0, \dots, m, \end{cases} \tag{30}$$

in which $res(x_i, t_j)$ can be considered as

$$res(x_i, t_j) = -x_i B[i, :] \hat{D}^T[:, j] + k \left(x_i D^{(2)}[i, :] B[:, j] + r B[i, :] \hat{D}^T[:, j] \right) + x_i a_{ij}.$$

or

$$res(x_i, t_j) = (kr - x_i) B[i, :] \hat{D}^T[:, j] + kx_i D^{(2)}[i, :] B[:, j] + x_i a_{ij}. \tag{31}$$

The reason why j starts from 1 (in the second case of Eq. (30)) is that: in both examples we will consider later, for the first case of Eq. (30), we have $z_0(x_n = R) = 0$ and this makes a redundancy with the second case of Eq. (30) at $j = 0$. To avoid this, j starts from 1.

We now describe the method of the selection of points in this paper. In what follows, we have used x_i and t_j which are considered as the following statements. Firstly,

$$\{x_i\}_{i=0}^n = \{x | (P_n^{\alpha, \beta+1}(u(x))(u(x) - u(x_n))) = 0\}, \tag{32}$$

which should $x \in (0, R]$. Therefore, we set $u(x) = \frac{2x}{R} - 1$, and we have:

$$\{x_i\}_{i=0}^{n-1} = \{x | P_n^{\alpha, \beta+1} \left(\frac{2x}{R} - 1 \right) = 0\}, \text{ and } x_n = R. \tag{33}$$

One can see that this assignment is based on the first approach in Sect. 2.3. With the same fashion, for variable t we have

$$\{t_j\}_{j=0}^m = \{t | (u(t) - u(t_0))P_n^{\alpha, \beta+1}(u(t)) = 0\} \tag{34}$$

that should $t \in [0, 1)$. Therefore, we set $u(t) = 2t - 1$, and we have:

$$t_0 = 0, \text{ and } \{t_j\}_{j=1}^m = \{t | P_n^{\alpha, \beta+1}(2t - 1) = 0\} \tag{35}$$

Again, one can consider this assignment based on the second approach shown by Eq. (8) in Sect. 2.3.

At the next step, we approximate the integral existing in the OCP. For this, we exploit Gauss–Jacobi quadrature.

For estimating an integral by Gauss–Jacobi quadrature, we do as follows:

$$\begin{aligned} \int_a^b f(v)w^{\alpha, \beta} \left(\frac{2(v-a)}{b-a} - 1 \right) dv &\approx \left(\frac{b-a}{2} \right) \int_{-1}^1 f \left(\frac{b-a}{2}s + \frac{b+a}{2} \right) w^{\alpha, \beta}(s) ds, \\ &= \frac{b-a}{2} \sum_{i=0}^N f \left(\frac{b-a}{2}s_i + \frac{b+a}{2} \right) \varpi_i \\ &= \frac{b-a}{2} \sum_{i=0}^N f(v_i) \varpi_i, \end{aligned} \tag{36}$$

where $v \in [a, b]$ and $s \in [-1, 1]$, and Eq. (36) is exact when $degree(f(v)) \leq (2N + 1)$ for Gauss–Jacobi quadrature.

$\{s_i\}_{i=0}^N$ are Gauss–Jacobi nodes, and their relevant weights $\{\varpi\}_{i=0}^N$ are [64]

$$\varpi_i = \frac{\Gamma(N + \alpha + 2)\Gamma(N + \beta + 2)}{\left(P_{N+1}^{\alpha, \beta}(s_i) \right)' (1 - s_i^2)} \frac{2^{\alpha+\beta+1}}{\Gamma(N + 2 + \alpha + \beta)\Gamma(N + 2)}, \tag{37}$$

The cost functional J is estimated by a numerical integration method. For this, we applied Gauss–Jacobi quadratures in Eq. (36) for both variables t and x .

$$\begin{aligned} \min J &= \frac{1}{2} \int_0^1 \int_0^R x^r (c_1 z_{nm}^2(x, t) + c_2 y_{nm}^2(x, t)) dx dt, \\ &\approx \frac{R}{8} \left(c_1 \sum_{i=0}^N \sum_{j=0}^M \frac{\varpi_x^i(\hat{x}_{i+1})^r \varpi_t^j z_{nm}^2(\frac{\hat{x}_{i+1}}{2}, \frac{\hat{t}_{j+1}}{2})}{w^{\alpha, \beta}(\hat{x}_i) w^{\alpha, \beta}(\hat{t}_i)} \right. \\ &\quad \left. + c_2 \sum_{i=0}^N \sum_{j=0}^M \frac{\varpi_x^i(\frac{\hat{x}_{i+1}}{2})^r \varpi_t^j y_{nm}^2(\frac{\hat{x}_{i+1}}{2}, \frac{\hat{t}_{j+1}}{2})}{w^{\alpha, \beta}(\hat{x}_i) w^{\alpha, \beta}(\hat{t}_i)} \right), \end{aligned} \tag{38}$$

Table 1 Numerical results of Example 1

<i>n</i>	<i>m</i>	α	β	<i>J</i>	<i>n</i>	<i>m</i>	α	β	<i>J</i>	<i>n</i>	<i>m</i>	α	β	<i>J</i>
2	2	0	0	0.0248814	5	5	0	0	0.01391215	5	7	0	0	0.0089648
		-0.5	-0.5	0.0284523			-0.5	-0.5	0.01567798			-0.5	-0.5	0.0095673
		0.5	0.5	0.0230089			0.5	0.5	0.01248992			0.5	0.5	0.0084049
		-0.5	0.5	0.0278755			-0.5	0.5	0.01794318			-0.5	0.5	0.0105544
		0.5	-0.5	0.0224106			0.5	-0.5	0.01128294			0.5	-0.5	0.0079716
		0	1	0.0255469			0	1	0.01516258			0	1	0.0092312
		1	0	0.0209952			1	0	0.01069205			1	0	0.0078231
		0	2	0.0265474			0	2	0.01506972			0	2	0.0084796
		2	0	0.0205804			2	0	0.00913921			2	0	0.0074379
		3	1	0.0193088			3	1	0.00859934			3	1	0.0071598
7	5	0	0	0.0186896	7	10	0	0	0.00813737	10	10	0	0	0.0089628
		-0.5	-0.5	0.0207502			-0.5	-0.5	0.00862000			-0.5	-0.5	0.0095609
		0.5	0.5	0.0168244			0.5	0.5	0.00765917			0.5	0.5	0.0083101
		-0.5	0.5	0.0223249			-0.5	0.5	0.00777906			-0.5	0.5	0.0091915
		0.5	-0.5	0.0163430			0.5	-0.5	0.00850356			0.5	-0.5	0.0087234
		0	1	0.0195875			0	1	0.00765102			0	1	0.0082872
		1	0	0.0151638			1	0	0.00757913			1	0	0.0083260
		0	2	0.0189639			0	2	0.00652118			0	2	0.0067052
		2	0	0.0128480			2	0	0.00733578			2	0	0.0079058
		3	1	0.0114265			3	1	0.00699366			3	1	0.0072780

where $\varpi_x^i, \varpi_t^j, \hat{x}_i, \hat{t}_j$ are as follows

$$\{\hat{x}_i\}_{i=0}^N = \{x | P_{N+1}^{\alpha,\beta}(x) = 0\}, \tag{39}$$

$$\{\hat{t}_j\}_{j=0}^M = \{t | P_{M+1}^{\alpha,\beta}(t) = 0\}, \tag{40}$$

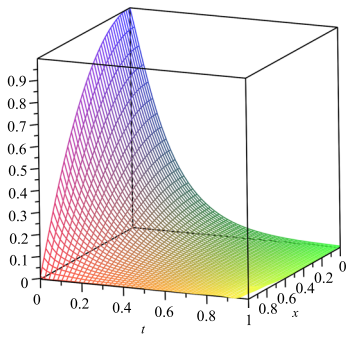
$$\varpi_x^i = \frac{\Gamma(N + \alpha + 2)\Gamma(N + \beta + 2)}{\left(P_{N+1}^{\alpha,\beta}(\hat{x}_i)\right)'(1 - \hat{x}_i^2)} \frac{2^{\alpha+\beta+1}}{\Gamma(N + 2 + \alpha + \beta)\Gamma(N + 2)}, \tag{41}$$

$$\varpi_t^j = \frac{\Gamma(M + \alpha + 2)\Gamma(M + \beta + 2)}{\left(P_{M+1}^{\alpha,\beta}(\hat{t}_j)\right)'(1 - \hat{t}_j^2)} \frac{2^{\alpha+\beta+1}}{\Gamma(M + 2 + \alpha + \beta)\Gamma(M + 2)}. \tag{42}$$

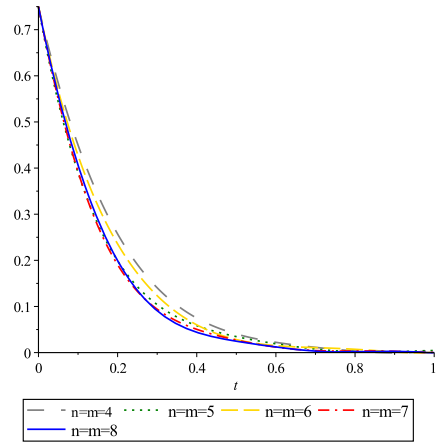
From Eq. (38) to Eq. (42) for both variables t and x , we consider 15 Gauss–Jacobi nodes ($M = N = 14$).

Thus, on the basis of what we have just discussed, the optimal control problem is reduced to a parameter optimization problem. This can be stated as follows:

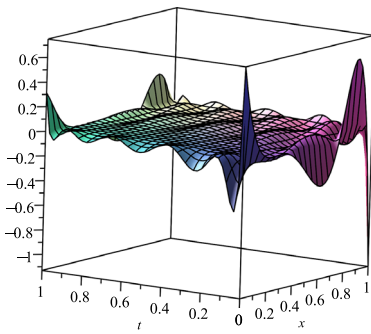
$$L(A, B, \hat{\lambda}) = J + \sum_{i=0}^{2(n+m)+nm+1} \lambda_i F[i], \tag{43}$$



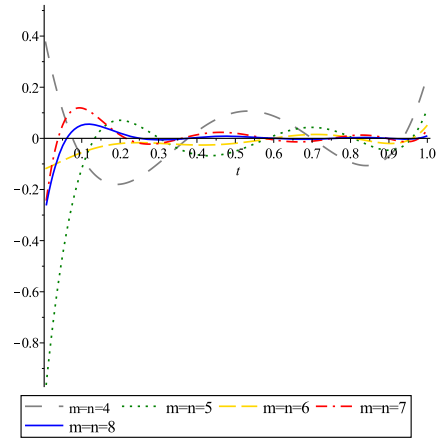
(a) Approximation of state function $z(x, t)$



(b) Approximation of $z(x, t)$ for $x = 0.5$ with various choices of m, n



(c) Approximation of control variable $y(x, t)$



(d) Approximation of $y(x, t)$ for $x = 0.5$ with various choices of m, n

Fig. 1 Plots of approximate state and control functions for Example 1.

$\hat{\lambda} = \{\lambda_i\}_{i=0}^{2(n+m)+nm+1}$ are Lagrange multipliers. Therefore, the minimization problem can, under these new conditions

$$\begin{cases} \frac{\partial L}{\partial a_{ij}} = 0, & i = 0, \dots, n, j = 0, \dots, m, \\ \frac{\partial L}{\partial b_{ij}} = 0, & i = 0, \dots, n, j = 0, \dots, m, \\ \frac{\partial L}{\partial \lambda_i} = 0, & i = 0, \dots, 2(n+m) + nm + 1, \end{cases}$$

produce a system of $4(n+m+1) + 3nm$ algebraic equations which can be solved by a mathematical software for achieving the unknowns. This system is solved by Newton's method via "fsolve" command in Maple. The solution of this system is given by using Maple software.

Table 2 Numerical results for J in Example 1 with the presented method and the Ritz method [3] ($\alpha = 0, \beta = 2$)

n	1	2	2	2	3	3	3
m	4	4	6	7	7	9	10
Ritz method [3]	0.081044	0.028790	0.018283	0.016484	0.013027	0.010405	0.007569
presented method	0.01586417	0.01408943	0.01396815	0.01401548	0.00473073	0.00470629	0.00468442

4 Numerical examples

In this section, the GLJGR collocation method is used to solve two cases of aforementioned OCP in Eq. (1). The aim is to find the state and control functions $z(x, t), y(x, t)$ that minimize the cost function J .

Example 1 Consider the aforementioned OCP with $c_1 = c_2 = r = R = k = 1$ and initial condition [3,62]

$$z(x, 0) = z_0(x) = 1 - \left(\frac{x}{R}\right)^2, \quad 0 < x < R.$$

so the optimal control problem of Eq. (1) would be

$$\min J = \frac{1}{2} \int_0^1 \int_0^1 x(z^2(x, t) + y^2(x, t)) dx dt, \tag{44}$$

subject to

$$x \frac{\partial z}{\partial t} = x \left(\frac{\partial^2 z}{\partial x^2} + \frac{1}{x} \frac{\partial z}{\partial t} \right) + xy(x, t), \tag{45}$$

with boundary and initial conditions

$$z(x, 0) = z_0(x) = 1 - x^2, \quad 0 < x < 1, \quad z(1, t) = 0, \quad t > 0. \tag{46}$$

Consider the assumptions mentioned in Example 1. With the methodology presented in Sects. 2 and 3, we approximate the function $z(x, t)$ and $y(x, t)$. In Table 1, the presented method is used to solve the OCP of Example 1. This table, by presenting the value of cost functional, simply shows the accuracy of the presented method for the different choices of n and m . The effect of α and β is shown as well to provide Chebyshev (all four kinds), Legendre cases, and other different cases. As the aim of this paper is to find $z(x, t), y(x, t)$ in order to minimize J , we plot these state and control functions in Fig. 1a, c. Also, the graph of these functions for the different number of bases is illustrated in Fig. 1b, d; they are the surface plots of the state and control functions. The comparison with the Ritz method by Mamehrashi and Yousefi [3] is made in Table 2. These results show that the presented method is accurate and provides such solutions rather faster than the Ritz method. In other words, with $n = 1, m = 4$, we obtained such a result that the Ritz method, somehow, provides with $n = 2, m = 7$. Similar to what has been concluded in [3], we obtained that the control and state functions initially have distinct values over the x -axis and as time goes by they tend to reach the same value: This phenomenon is representative of a diffusion process.

Example 2 In this example, the $r = 2, c_1 = c_2 = R = k = 1$ and the initial condition is [3,63]

$$z(x, 0) = z_0(x) = \sin(2\pi x), \quad 0 < x < R.$$

Table 3 Numerical results of Example 2

<i>n</i>	<i>m</i>	α	β	<i>J</i>	<i>n</i>	<i>m</i>	α	β	<i>J</i>	<i>n</i>	<i>m</i>	α	β	<i>J</i>
2	2	0	0	0.981774266	5	5	0	0	2.36052795	5	7	0	0	0.7709361
		-0.5	-0.5	0.598751916			-0.5	-0.5	2.56788894			-0.5	-0.5	0.8706653
		0.5	0.5	1.226311717			0.5	0.5	2.15726157			0.5	0.5	0.8083347
		-0.5	0.5	0.135957972			-0.5	0.5	2.26726831			-0.5	0.5	0.6988001
		0.5	-0.5	2.079513828			0.5	-0.5	2.38904227			0.5	-0.5	0.8083347
		0	1	0.383652016			0	1	2.04317188			0	1	0.5890023
		1	0	2.022399132			1	0	2.22743042			1	0	0.7332970
		0	2	0.101961113			0	2	1.68004117			0	2	0.4074085
		2	0	2.012714897			2	0	2.08193333			2	0	0.6923438
		3	1	1.877493818			3	1	1.77464161			3	1	0.5390153
7	5	0	0	2.498801296	7	10	0	0	0.16882398	10	10	0	0	0.1877156
		-0.5	-0.5	2.650340836			-0.5	-0.5	0.20321134			-0.5	-0.5	0.2208658
		0.5	0.5	2.305646950			0.5	0.5	0.13032446			0.5	0.5	0.1469488
		-0.5	0.5	2.415879813			-0.5	0.5	0.14219532			-0.5	0.5	0.1510822
		0.5	-0.5	2.499141221			0.5	-0.5	0.18137586			0.5	-0.5	0.2050793
		0	1	2.201622390			0	1	0.10295491			0	1	0.1096786
		1	0	2.349064681			1	0	0.15196838			1	0	0.1759308
		0	2	1.837451911			0	2	0.05137739			0	2	0.0481434
		2	0	2.162014225			2	0	0.13443163			2	0	0.1631296
		3	1	1.831793809			3	1	0.08179826			3	1	0.1001137

Therefore, we have

$$\min J = \frac{1}{2} \int_0^1 \int_0^1 x^2(z^2(x, t) + y^2(x, t)) dx dt, \tag{47}$$

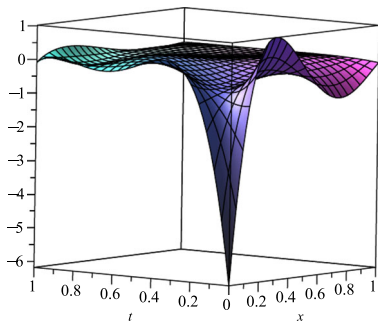
subject to

$$x \frac{\partial z}{\partial t} = x \left(\frac{\partial^2 z}{\partial x^2} + \frac{2}{x} \frac{\partial z}{\partial t} \right) + xy(x, t), \tag{48}$$

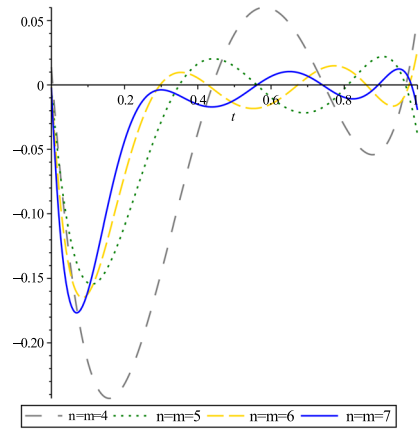
with boundary and initial conditions

$$z(x, 0) = z_0(x) = \sin(2\pi x), 0 < x < 1, \quad z(1, t) = 0, t > 0. \tag{49}$$

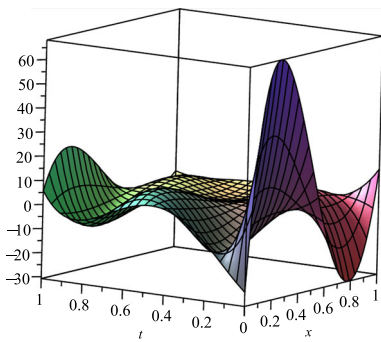
Like Example 1, the method discussed in Sects. 2, 3 is utilized to approximate the solution of Example 2. The cost functional *J* for the different selection of *n*, *m*, α , β is calculated in Table 3. Figure 2a, c shows numerical results for state and control functions, i.e., *z*(*x*, *t*) and *y*(*x*, *t*), respectively. These results are plotted in Fig. 2b, d at *x* = 0.5 in a surface plot. Note that initially the state and control at two different locations differ, but as time progresses, the two values become very close. As said in Example 1, this is because of diffusion. A comparison with the Ritz method [3] is made and reported in Table 4. The results in this table demonstrate that the presented method is accurate and reliable, and these results seem to be reached faster than the Ritz method.



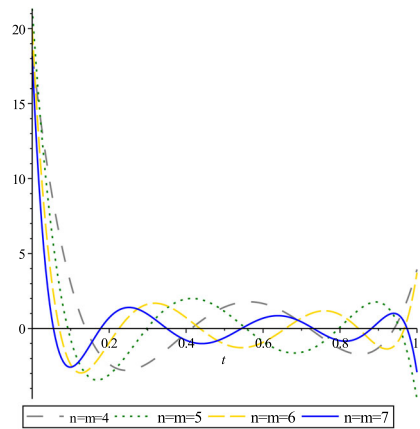
(a) Approximation of state function $z(x, t)$



(b) Approximation of $z(x, t)$ for $x = 0.5$ with various choices of m, n



(c) Approximation of control function $y(x, t)$



(d) Approximation of $y(x, t)$ for $x = 0.5$ with various choices of m, n

Fig. 2 Plots of approximate state and control functions for Example 2

Table 4 Numerical results for J in Example. 2 with the presented method and the Ritz method [3] ($\alpha = 0, \beta = 2$)

n	4	5	5	6	6	7	7
m	5	5	6	6	7	7	8
Ritz method [3]	2.72722	1.92027	1.27424	0.91850	0.55287	0.54935	0.36868
presented method	1.834166023	1.680041170	0.834822493	0.865965461	0.426441006	0.467294422	0.229834210

5 Conclusion

In this study, an OCP is investigated. This problem has beneficial applications in many chemical, biological, and physical fields of studies. The goal of this article is to develop an efficient and accurate method to solve this nonlinear OCP. The method is based upon GLJGR collocation method. Firstly, GL functions are introduced so as to satisfy in delta Kronecker function and GLJGR collocation method is described. The advantage of employing this method is that it reduces the computational and CPU time remarkably. As expressed, these functions are a generalization of the classical Lagrange polynomials. The corresponding differentiation matrices of $D^{(1)}$ and $D^{(2)}$ can be obtained by simple formulas. The main benefit of the proposed formulas is that these formulas are derivative-free. Additionally, the accuracy of the presented method by GL function has an exponential convergence rate. Secondly, the obtained results compared with those provided by Mamehrashi and Yousefi [3] show the accuracy and reliability of the presented method. This numerical approach is applicable and effective for such kind of nonlinear OCPs and other problems that can be approximated by Gauss–Radau nodes.

References

1. O.P. Agrawal, J. Comput. Nonlin. Dyn. 24:3(2) (2008)
2. O.P. Agrawal, IFAC Proc. 39(11), 68–72 (2006)
3. K. Mamehrashi, S.A. Yousefi, Int. J. Control 90(2), 298–306 (2017)
4. F. Andreu, V. Caselles, J.M. Mazon, S. Moll, Arch. Ration. Mech. Anal. 182, 269–297 (2006)
5. F. Andreu, V. Caselles, J.M. Mazon, S.J. Moll, Evol. Equat. 7, 113–143 (2007)
6. N.F. Britton, *Reaction-Diffusion Equations and Their Applications to Biology* (Academic Press, Cambridge, 1986)
7. K. Mamehrashi, S.A. Yousefi, Optim. Contr. Appl. Method. 37(4), 765–781 (2016)
8. F.L. Lewis, Automatica. 28(2), 345–354 (1992)
9. W. Marszalek, Appl. Math. Model. 8, 11–14 (1984)
10. R.P.I.E.E.E.T. Roesser, Automat. Contr. 20(1), 1–10 (1975)
11. S. Attasi, IRIA Rapport Laboria. 30 (1973)
12. S. Attasi, IRIA Rapport Laboria. 56 (1975)
13. E. Fornasini, IEEE T. Automat. Contr. 21(4), 484–491 (1976)
14. E. Fornasini, G. Marchesini, Math. Sys. Theory. 12(1), 59–72 (1978)
15. A.E. Bryson, *Applied Optimal Control: Optimization, Estimation and Control* (CRC Press, Boca Raton, 1975)
16. A.P. Sage, C.C. White, *Optimum Systems Control* (Prentice Hall, New York, 1977)
17. O.P. Agrawal, Int. J. Control 50(2), 627–638 (1989)
18. A. Lotfi, M. Dehghan, S.A. Yousefi, Comput. Math. Appl. 62(3), 1055–1067 (2011)
19. A.J. Samimi, S.A. Yousefi, A.M. Tehranchian, Appl. Math. Comput. 172(1), 198–209 (2006)
20. J. Gregory, C. Lin, *Constrained optimization in the calculus of variations and optimal control theory*. Springer Publishing Company Incorporated (2007)
21. S. Manabe, Proc. DETC. 3, 609–616 (2003)
22. H.W. Bode, *Network analysis and feedback amplifier design* (1945)
23. K. Rabiei, Y. Ordokhani, E. Babolian, Nonlin. Dyn. 88(2), 1013–1026 (2017)
24. A. Lotfi, S.A. Yousefi, J Optimiz. Theory App. 174(1), 238–255 (2017)
25. S.A. Yousefi, M. Dehghan, A. Lotfi, Int. J. Comput. Math. 87(5), 1042–1050 (2010)
26. J. Li, J.S. Tsai, L.S. Shieh, Asian J. Contr. 5(1), 78–87 (2003)
27. Q.L. Wei, H.G. Zhang, L.L. Cui, Acta Automatica Sinica. 35(6), 682–692 (2009)
28. H. Zhang, D. Liu, Y. Luo, D. Wang, *Adaptive Dynamic Programming for Control: Algorithms and Stability* (Springer, London, 2013)
29. Z. Sabeh, M. Shamsi, M. Dehghan, Math. Methods Appl. Sci. 39(12), 3350–3360 (2016)
30. E.H. Doha, W.M. Abd-Elhameed, A.H. Bhrawy, Collectanea Mathematica. 64(3), 373–394 (2013)
31. K. Parand, M. Delkhosh, J. Comput. Appl. Math. 317, 624–642 (2017)
32. K. Parand, M. Delkhosh, Boletim da Sociedade Paranaense de Matemática. 36(4), 33–54 (2018)

33. A.H. Bhrawy, M.M. Al-Shomrani, *Adv. Differ. E.* **2012**(1), 8 (2012)
34. E.H. Doha, A.H. Bhrawy, *Appl. Numer. Math.* **58**(8), 1224–1244 (2008)
35. E.H. Doha, A.H. Bhrawy, D. Baleanu, S. Ezz-Eldien, *Adv. Differ. E.* **2014**(1), 231 (2014)
36. A.H. Bhrawy, M.A. Alghamdi, *Boundary Value Problem.* **2012**, 1 (2012)
37. H.J. Tal-Ezer, *Numer. Anal.* **23**(1), 11–26 (1986)
38. H.J. Tal-Ezer, *Numer. Anal.* **26**(1), 1–11 (1989)
39. A.H. Bhrawy, Al-Shomrani, MM. *Abstract and Applied Analysis*, Hindawi Publishing Corporation. **947230**, 2012 (2012)
40. A.H. Bhrawy, E.H. Doha, M.A. Abdelkawy, R.A. Van Gorder, *Appl. Math. Model.* **40**(3), 1703–1716 (2016)
41. K. Parand, M. Delkhosh, M. Nikarya, *Tbilisi Math. J.* **10**(1), 31–55 (2017)
42. J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd edn. (Dover, New York, 2000)
43. K. Parand, M. Hemami, *Int. J. Appl. Comput. Math* 3(2):1053–1075 (2016)
44. M.A. Saker, S. Ezz-Eldien, A.H. Bhrawy, *Romanian. J. Phys.* **2017**(62), 105 (2017)
45. A.H. Bhrawy, M.A. Abdelkawy, F. Mallawi, *Boundary Value Problem.* **2015**(1), 1–20 (2015)
46. E.H. Doha, A.H. Bhrawy, M.A. Abdelkawy, *J. Comput. Nonlin. Dyn.* **2015**(1), 1–20 (2015)
47. K. Parand, S. Latifi, M.M. Moayeri, *SeMA J.* (2017). <https://doi.org/10.1007/s40324-017-0138-9>
48. G.E. Mohammad, A. Kazemi, M. Razzaghi, *IEEE T. Autom. Control* **40**(10), 1793–1796 (1995)
49. F. Fahroo, Michael Ross I. J. Guid. *Control D.* 25(1):160-166 (2002)
50. D. Garg, M. Patterson, W.W. Hager, A.V. Rao, D.A. Benson, G.T. Huntington, *Automatica.* **46**(11), 1843–1851 (2010)
51. M. Shamsi, *Oprim. Contr. Appl. Met.* **32**, 668–680 (2011)
52. P. J. Williams, *Guid. Control D.* 27(2):293–296 (2004)
53. K. Parand, S. Latifi, M. Delkhosh, M.M. Moayeri, *Eur. Physical J. Plus.* **133**(1), 28 (2018)
54. K. Parand, S. Latifi, M.M. Moayeri, M. Delkhosh, *Commun. Theor. Phys.* **69**(5), 519 (2018)
55. M. Delkhosh, K. Parand, *J. Comput. Sci.* **34**, 11–32 (2019)
56. A.H. Zaky, *M. Math. Method Appl. Sci.* 39(7): 1765-1779 (2015)
57. A.H. Bhrawy, J.F. Alzaidy, M.A. Abdelkawy, A. Biswas, *Nonlin. Dyn.* **84**(3), 1553–1567 (2016)
58. A.H. Bhrawy, E.H. Doha, S.S. Ezz-Eldien, M.A. Abdelkawy, *Comput. Model. Eng. Sci.* **104**(3), 185–209 (2015)
59. A.H. Bhrawy, E.H. Doha, D. Baleanu, R.M. Hafez, *Math. Method Appl. Sci.* 38(14): 3022-3032 (2015)
60. E.H. Doha, A.H. Bhrawy, M.A. Abdelkawy, R.A. Van Gorder, *J. Comput. Phys.* **261**, 244–255 (2014)
61. F. Daniele, *Polynomial approximation of differential equations*, Lecture Notes in Physics. New Series m: Monographs. 8. Springer- Verlag: Berlin (1992)
62. N. Ozdemir, O.P. Agrawal, D. Karadeniz, B.B. Iskender, *Phys. Scr.* **2009**(T136), 014024 (2009)
63. M.M. Hasan, X.W. Tangpong, O.P. Agrawal, *J. Vib. Control* **18**(10), 1506–1525 (2012)
64. J. Shen, T. Tang, L.L. Wang, *Spectral methods: algorithms, analysis and applications*. Springer Science & Business Media (2011)