Research Article

SOHAG JOURNAL OF SCIENCES

DOI: 10.21608/sjsci.2025.392634.1282

Jacobi Integral Pseudospectral Method for Solving Infinite-Horizon Optimal Control Problems

Kareem T. Elgindy[1], *Hareth M. Refat*[*,2], *and Ibrahim A. Abbas*[2]

[1] Department of Mathematics and Sciences, College of Humanities and Sciences, Ajman University, P.O. Box: 346 Ajman, United Arab Emirates
Nonlinear Dynamics Research Center (NDRC), Ajman University, Ajman P.O.Box 346, United Arab Emirates.
[2] Mathematics Department, Faculty of Science, Sohag University, Sohag 82524, Egypt.
[*]**Corresponding Author:** harith_refaat@science.sohag.edu.eg; hareth_mohamed2000@yahoo.com.

Received: 29th June 2025, Revised: 23th August 2025, Accepted: 25th August 2025.
Published online: 30th August 2025.

Abstract:

We introduce a novel direct integral pseudospectral (IPS) method for addressing a class of infinite-horizon optimal control problems (IHOCs) with continuous time. This approach transforms IHOCs into finite-horizon optimal control problems (FHOCs) in integral form through specific parametric mappings, which are then discretized into finite-dimensional nonlinear programming problems (NLPs) using rational collocations based on Jacobi polynomials and Jacobi-Gauss-Radau (JGR) nodes. Our method extends previous work that utilized Gegenbauer polynomials by employing the more general and flexible Jacobi polynomial family, which offers additional degrees of freedom through its two parameters α and β. We provide a comprehensive analysis of the interplay between parametric mappings, barycentric rational collocations based on Jacobi polynomials and JGR points, and the convergence properties of the collocated solutions. The paper presents a rigorous examination of the method's error bounds and convergence characteristics, along with a stability analysis based on the Lebesgue constant for JGR-based rational interpolation. We validate our theoretical findings through two illustrative examples, including a practical application to spacecraft attitude maneuvers. Our results demonstrate that the proposed collocation method, when combined with an efficient NLP solver (MATLAB's fmincon solver), converges exponentially to near-optimal approximations for coarse collocation mesh grid sizes. Furthermore, we show that certain parameter combinations with $\alpha \neq \beta$ yield more accurate solutions than those achievable with Gegenbauer, Legendre, or Chebyshev polynomials. The study also reveals that standard direct spectral/pseudospectral (PS) and IPS methods based on classical polynomials of Jacobi type and specific parametric mappings typically diverge as the number of collocation points increases when computations are performed using floating-point arithmetic.
Keywords: Integration matrix, Jacobi Polynomial, Jacobi-Gauss-Radau, Optimal Control, Pseudospectral.

1 Introduction

Direct PS methods have become a highly influential numerical approach for solving CTOCPs in recent decades, transforming these problems into standard optimization problems that can be efficiently solved using conventional optimization techniques. By approximating solutions with global polynomials, such as Jacobi or Legendre polynomials, evaluated at specific collocation points, PS methods use the properties of orthogonal polynomials to discretize continuous dynamics, achieving high accuracy and efficiency for smooth problems. A key strength of these methods is their ability to attain exponential convergence rates for sufficiently smooth solutions, even when using relatively coarse mesh grids.

The development of PS methods can be traced back to the pioneering works of Orszag [1] and Patterson and Orszag [2] in the early 1970s, establishing them as a cornerstone technology for solving partial differential equations. Over subsequent decades, these methods have been continuously refined and extended to address increasingly complex problems across various scientific domains. The evolution of PS methods has been particularly notable in aerospace applications, where they have demonstrated remarkable practical utility. A landmark achievement occurred in 2006-2007 when the International Space Station executed two large-angle maneuvers without propellant consumption by following an attitude trajectory developed using PS OC theory, resulting in substantial cost savings for NASA [3].

Acronym	Meaning	Acronym	Meaning
AE_J	Absolute Objective Function Value Error	CTOCP	Continuous-Time Optimal Control Problem
FHOC	Finite-Horizon Optimal Control Problem	FHOCI	Finite-Horizon Optimal Control Problem in Integral Form
FOCP	Fractional Optimal Control Problem	*fval*	Approximate Cost Function Value
GGR	Gegenbauer-Gauss-Radau	IHOC	Infinite-Horizon Optimal Control Problem
IPS	Integral Pseudospectral	JGR	Jacobi-Gauss-Radau
JGR-IPS	Jacobi-Gauss-Radau Integral Pseudospectral	JGR-IPS1	JGR-IPS with parametric mapping $T_{1,L}^{(\alpha,\beta)}$
JGR-IPS12	JGR-IPS with either parametric mapping $T_{1,L}^{(\alpha,\beta)}$ or $T_{2,L}^{(\alpha,\beta)}$	JGR-IPS2	JGR-IPS with parametric mapping $T_{2,L}^{(\alpha,\beta)}$
LG	Legendre Gauss	LGR	Legendre Gauss Radau
LQR	Linear Quadratic Regulator	$MAE_{x,u}$	Maximum Absolute Error of the State and Control Variables
MRE	Maximum Absolute Residual Computed at the Collocation Points	NLP	Nonlinear Programming Problem
OC	Optimal Control	PS	Pseudospectral
SQP	Sequential Quadratic Programming	SR	Switched Rational
TPBVP	Two-Point Boundary Value Problem		

Table 1: List of acronyms and their meanings.

The trend since 2007 shows a continued maturation of PS methods, moving from theoretical development to wider practical implementation across various complex problems, often in combination with other advanced optimization and computational techniques. The ability to achieve highly accurate solutions with relatively sparse discretization points and the development of robust mesh refinement and error estimation techniques contribute to their ongoing success [4,5,6,7].

PS methods share a close relationship with spectral methods but differ in their approach to solution representation. While spectral methods typically employ global orthogonal basis polynomials, PS methods utilize interpolation to extend solutions based on values at grid points. This nodal representation offers significant advantages, as solution values become immediately available at collocation points once discretization is implemented. Several comprehensive resources provide detailed expositions of spectral and PS methods, including the works by Fornberg [8], Hesthaven et al. [9], and Canuto et al. [10,11].

A closely related class of methods, known as IPS methods or PS integration methods, offers an effective alternative to traditional PS approaches. These methods require reformulation of the dynamical system equations into their integral form before the collocation phase, thereby avoiding precision degradation typically associated with numerical differentiation procedures. The spectral approach for approximating integral forms of ordinary differential equations was initially proposed in the 1960s by Clenshaw [12] and El-Gendi [13] in the spectral and physical spaces, respectively. This approach has been further developed in subsequent research [14,15,16,17,18,19].

Among the various classes of CTOCPs, IHOCs and OC problems defined over sufficiently long time domains have attracted significant attention due to their wide-ranging applications in finance, engineering, management science, medicine, aerospace, and other fields [20,21,22,23,24,25,26,27]. IHOCs, which seek to minimize a cost functional over an unbounded time domain, present unique challenges due to the need to capture long-term behavior and ensure numerical stability in the presence of diverging time scales. These difficulties necessitate specialized techniques, such as the parametric mappings and JGR collocation schemes developed in this work, to transform the infinite horizon into a finite domain suitable for PS approximation. Despite the abundance of direct PS approaches for solving FHOCs in the literature, relatively few publications have addressed IHOCs using this class of methods. Notable contributions include our previous work [28] using Gegenbauer polynomials, as well as approaches employing LG and LGR PS techniques [37,38,39], collocation at flipped LGR points [40], and the transformed Legendre spectral method [41].

1.1 Relationship to Our Previous Work Using Gegenbauer Polynomials

The present study builds upon and extends the framework introduced in our previous work [28], where we developed a direct IPS method for solving IHOCs using Gegenbauer polynomials. While we adopt a similar overall approach to problem transformation and discretization, our work introduces several significant advancements and differs in multiple important aspects.

In [28], we demonstrated that direct IPS methods based on Gegenbauer polynomials can effectively solve certain classes of IHOCs. We established theoretical foundations for the convergence properties of these methods and identified limitations related to the divergence of approximations for large mesh sizes. That work focused exclusively on Gegenbauer polynomials with parameter $\lambda > -0.5$, which includes Chebyshev polynomials of the first kind ($\lambda = 0$) and Legendre polynomials ($\lambda = 0.5$) as special cases. Our current research extends this foundation in several key directions:

1. We generalize the polynomial basis from Gegenbauer to Jacobi polynomials, which introduces an additional degree of freedom through the two-parameter family

(α and β). This generalization encompasses Gegenbauer polynomials as the special case where $\alpha = \beta = \lambda - 0.5$, but also enables exploration of asymmetric weight functions when $\alpha \neq \beta$. This broader parameter space allows for more flexible adaptation to specific problem characteristics.

2. We expand the parameter range considerably, allowing both α and β to vary within the recommended range $(-1, 2]$. This interval is wider than the recommended range of $[0.5, 1]$ for λ in our previous work [28] when the SR interpolation algorithm is adopted—although the feasible range for recommended λ values falls within $(-1/2, 2]$ for a standard barycentric rational interpolation algorithm. This expanded parameter space enables more refined tuning of the approximation properties to match specific problem characteristics.
3. Our investigation reveals that certain parameter combinations with $\alpha \neq \beta$ can yield more accurate approximations than those achievable with Gegenbauer polynomials (where $\alpha = \beta$). This finding represents a novel contribution that was not explored in the previous work and significantly expands the toolkit available for solving IHOCs with high precision.
4. We implement a simplified computational approach that eliminates the need for the switching technique used in [28] for computing barycentric weights. Our analysis demonstrates that for many parameter combinations within the Jacobi family, the standard formula performs adequately without requiring the more complex switching approach.
5. We investigate a practical application concerning attitude maneuvers of an asymmetric rigid-body spacecraft, demonstrating the real-world utility of our approach. This application example provides valuable insights into the practical performance of Jacobi-based IPS methods in aerospace control problems.

Jacobi polynomials have been widely utilized in solving various OC problems. To mention a few, [29] introduced a generalized PS method for OC problems, employing roots of derivatives of Jacobi polynomials for collocation. [42] developed differential and integral fractional PS methods for FOCPs using Jacobi polynomials. [54] proposed a shifted Jacobi PS method for solving nonlinear IHOCs. [43] applied a Jacobi–Gauss PS discretization to an OC problem governed by a two-sided space-fractional diffusion equation. [44] presented a multiple-interval PS scheme using Jacobi polynomials for collocation at shifted flipped JGR points to address nonlinear OC problems with time-varying delays. [45] employed a Jacobi spectral collocation method for FOCPs.

We investigate in our work whether superior precision and convergence rates can be achieved using Jacobi polynomials, even though PS approaches for solving IHOCs typically employ Legendre or Gegenbauer polynomials. The Jacobi polynomial family offers several compelling advantages for IHOC discretizations:

1. Jacobi polynomials offer a generalized framework in the sense that they encompass Gegenbauer polynomials (which include Chebyshev and Legendre polynomials) as special cases within their larger family. Consequently, all theoretical and experimental findings regarding Jacobi polynomials naturally extend to these more specialized polynomial types.
2. Jacobi polynomials offer more parameter flexibility. Since Gegenbauer, Chebyshev, and Legendre polynomials are subsets of Jacobi polynomials, we can implement any of these polynomial types by simply selecting appropriate values for the Jacobi parameters α and β. For instance, setting $\alpha = \beta = -1/2$ yields Chebyshev polynomials, $\alpha = \beta = 0$ produces Legendre polynomials, and $\alpha = \beta = \lambda - 0.5$ generates Gegenbauer polynomials with parameter λ. This provides considerable flexibility within a unified computational framework.
3. Jacobi polynomials offer broad applicability as they have demonstrated effectiveness in solving various mathematical problems, including ordinary differential equations, partial differential equations, integral equations, fractional differential equations, and OC problems [29,30,31,32,33,34,35,36].
4. Jacobi polynomials can provide improved approximation capabilities because the two-parameter nature of Jacobi polynomials allows for more refined tuning of the approximation properties to match specific problem characteristics, potentially yielding improved accuracy compared to single-parameter polynomial families.

Our contributions in this paper can be summarized as follows:

1. We demonstrate that Jacobi PS methods often converge exponentially to near-exact solutions using relatively small mesh grids, but typically diverge for fine meshes under certain parametric maps, extending the findings in [28] to the more general Jacobi polynomial family.
2. We show that certain Jacobi polynomials with $\alpha \neq \beta$ yield more accurate solutions than Gegenbauer, Legendre, and Chebyshev polynomials, providing a broader spectrum of high-accuracy approximation options.
3. The use of Jacobi polynomials provides greater flexibility in parameter selection, allowing α and β to lie within the interval $(-1, 2]$, compared to the more restricted parameter range for Gegenbauer polynomials where λ is limited to the recommended range $[0.5, 1]$ for optimal performance of the SR interpolation algorithm.
4. We investigate a practical application concerning attitude maneuvers of an asymmetric rigid-body

spacecraft, demonstrating the real-world utility of our approach.

An intriguing question we explore is how to optimally select the Jacobi polynomial parameters α and β for collocation purposes. For sufficiently smooth functions, appropriate choices of α and β can yield excellent approximations with higher accuracy than those provided by Chebyshev, Legendre, and Gegenbauer polynomials using relatively coarse mesh grids, while suboptimal choices may significantly degrade the accuracy of the numerical method.

Remark 11. *We acknowledge that the SR interpolation algorithm of our previous work [28] often produces slightly higher accuracies compared to the current work under similar parameter settings because: (i) it introduces stable formulas for GGR-based rational interpolation, reducing cancellation errors near the right boundary of the transformed finite time domain, and (ii) it uses* $\varepsilon \approx 0.1$ *to switch between weight formulas, improving numerical stability for fine meshes. Integrating the current work with the SR interpolation algorithm would surely outperform our former work [28] due to the added degree of freedom provided by the Jacobi polynomial indices compared with Gegenbauer polynomials, allowing for the selection of optimal parameters from a much larger parameter space. However, the selection of* $\varepsilon \approx 0.1$ *is based on extensive numerical experiments conducted for candidate* ε *values between 0 and 1, with no theoretical justification provided. In fact, this choice was based on comprehensive experiments that compared errors in computing barycentric weights for mesh sizes* $n = 10(10)100$ *and Gegenbauer index range* $\lambda = -0.49, -0.4(0.1)2$*, using MATLAB in double-precision floating-point arithmetic. Extending the SR interpolation algorithm to the current work based on a similar set of experiments would require comparing errors in computing barycentric weights for* $n = 10(10)100$ *and* $\alpha, \beta = -0.49, -0.4(0.1)2$*, which is computationally expensive and exhaustive. Therefore, extending the SR interpolation algorithm to the current work is not feasible at the moment, as it requires substantial computational processing capabilities to test all possible parameter combinations, which we currently do not possess. We hope to overcome this challenge in the near future.*

The remainder of this paper is organized as follows: Section 2 establishes the notation used throughout the paper. Section 3 formulates the problem under investigation. In Section 4, we describe the two parametric maps used to transform the IHOC into a FHOCI. Section 5 outlines the discretization scheme for the FHOCI, including the construction of the integration matrices and the NLP formulation. Section 6 presents a numerical analysis of the proposed method, examining the barycentric rational interpolation based on Jacobi polynomials and JGR points, and investigating the convergence properties of the collocated solutions. Section 7 provides numerical examples to validate the theoretical findings. Finally, Section 8 offers concluding remarks and suggestions for future research directions.

2 Notation

Before proceeding with the mathematical formulation of the problem, we establish the notation that will be used consistently throughout this paper. This standardized notation framework facilitates clear communication of the mathematical concepts, algorithms, and theoretical results presented in subsequent sections. We adopt conventional mathematical notation where possible, with specific adaptations for the Jacobi polynomial framework and IHOCs.

In this paper, we denote vectors by bold italicized lowercase letters (e.g., $\boldsymbol{x}$, $\boldsymbol{u}$) and matrices by bold uppercase letters (e.g., $\mathbf{A}$). The set of real numbers is represented by $\mathbb{R}$. For a function f, we use $f^{(n)}$ to denote its nth derivative. The Jacobi polynomial of degree n with parameters α and β is denoted by $J_n^{(\alpha,\beta)}$. The norm of a function f over an interval $[a,b]$ is represented by $\|f\|_{[a,b]}$. We use the notation $i = 0(1)n$ to indicate that the index i ranges from 0 to n with increments of 1.

With this notation established, we now turn to the formal definition of the IHOC under investigation.

3 Problem Statement

Having established our notation, we now present the formal definition of the IHOC under investigation. This class of problems arises in various applications including economics, engineering, and aerospace systems where control actions must be determined over an unbounded time horizon. The mathematical formulation provided here serves as the foundation for the development of our Jacobi-based IPS method.

We consider the following IHOC:

$$\min_{\boldsymbol{x},\boldsymbol{u}} J = \int_0^\infty g(\boldsymbol{x}(t), \boldsymbol{u}(t))\, dt, \tag{3.1a}$$

$$\text{subject to} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad t \in [0, \infty), \tag{3.1b}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{3.1c}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ represents the state vector, $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ is the control vector, $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is the running cost function, $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ defines the system dynamics, and $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$ is the initial state vector. The objective is to determine the optimal state and control trajectories that minimize the cost functional J while satisfying the system dynamics and constraints.

This formulation presents significant computational challenges due to the infinite integration domain in the cost functional and the unbounded time interval for the

constraints. To address these challenges, we will transform the IHOC into a FHOCI using appropriate parametric mappings, as described in the following section.

4 Transformation of the IHOC

A fundamental challenge in solving IHOCs is the unbounded domain of integration. To overcome this difficulty, we employ parametric mappings that transform the original IHOC into a FHOCI. This transformation is a critical step that enables the application of our Jacobi-based IPS method. While similar transformations have been used with Gegenbauer polynomials in previous work [28], our approach extends these techniques to the more general Jacobi polynomial framework.

We shall use the two specific parametric mappings, $T_{1,L}^{(\alpha,\beta)}$ and $T_{2,L}^{(\alpha,\beta)}$, defined as follows:

$$T_{1,L}^{(\alpha,\beta)}(\tau) = \frac{L(1+\tau)}{1-\tau}, \quad \tau \in [-1,1), \tag{4.1a}$$

$$T_{2,L}^{(\alpha,\beta)}(\tau) = L\ln\frac{2}{1-\tau}, \quad \tau \in [-1,1), \tag{4.1b}$$

where $L>0$ is a scaling parameter that can be adjusted to optimize the performance of the numerical method. These mappings transform the infinite interval $[0,\infty)$ to the finite interval $[-1,1)$, allowing us to reformulate the IHOC as a problem over a bounded domain. The notations $T_{1,L}^{(\alpha,\beta)}$ and $T_{2,L}^{(\alpha,\beta)}$ used for the parametric mappings in Eqs. (4.1a) and (4.1b) indicate their association with the Jacobi polynomial parameters α and β employed in the collocation scheme. Specifically, these mappings are evaluated at the JGR points, which are defined based on α and β. The mappings themselves do not directly depend on these parameters, and the superscript is used to reflect their role within the JGR-based collocation framework.

Figures 1 and 2 display the mesh-like surfaces of both parametric mappings for multiple values of L, n, α, and β. These figures show that the parametric maps:

1. increase as α decreases while holding n, L, and β fixed;
2. increase as L increases while holding other parameters fixed.

Figures 3 and 4 demonstrate that the parametric maps increase as β increases while holding n, L, and α fixed. Additionally, near $\tau=1$, as L and β increase (with other variables held constant), $T_{1,L}^{(\alpha,\beta)}$ grows significantly faster than $T_{2,L}^{(\alpha,\beta)}$, which increases at a slower rate. Also, in general, as α increases (with other variables held constant), $T_{1,L}^{(\alpha,\beta)}$ decreases significantly faster than $T_{2,L}^{(\alpha,\beta)}$, which decreases at a slower rate.

Using the change of variable $t=T(\tau)$, where T represents either $T_{1,L}^{(\alpha,\beta)}$ or $T_{2,L}^{(\alpha,\beta)}$, we define the transformed state and control variables as:

$$\tilde{\boldsymbol{x}}(\tau) = \boldsymbol{x}(T(\tau)), \quad \tau \in [-1,1), \tag{4.2a}$$

$$\tilde{\boldsymbol{u}}(\tau) = \boldsymbol{u}(T(\tau)), \quad \tau \in [-1,1). \tag{4.2b}$$

With these transformations, the IHOC described by Eqs. (3.1a)-(3.1c) can be reformulated as:

$$\min_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{u}}} \tilde{J} = \int_{-1}^{1} T'(\tau) g(\tilde{\boldsymbol{x}}(\tau),\tilde{\boldsymbol{u}}(\tau))\, d\tau, \tag{4.3a}$$

$$\text{subject to} \quad \frac{d\tilde{\boldsymbol{x}}(\tau)}{d\tau} = T'(\tau)\boldsymbol{f}(\tilde{\boldsymbol{x}}(\tau),\tilde{\boldsymbol{u}}(\tau)), \quad \tau \in [-1,1), \tag{4.3b}$$

$$\tilde{\boldsymbol{x}}(-1) = \boldsymbol{x}_0, \tag{4.3c}$$

where $T'(\tau)$ denotes the derivative of the mapping function with respect to τ. The derivatives of our parametric mappings are given by:

$$\left(T_{1,L}^{(\alpha,\beta)}\right)'(\tau) = \frac{2L}{(1-\tau)^2}, \quad \tau \in [-1,1), \tag{4.4a}$$

$$\left(T_{2,L}^{(\alpha,\beta)}\right)'(\tau) = \frac{L}{1-\tau}, \quad \tau \in [-1,1). \tag{4.4b}$$

To facilitate the application of our Jacobi-based IPS method, we further reformulate the problem in integral form. By integrating Eq. (4.3b) from -1 to τ, we obtain:

$$\tilde{\boldsymbol{x}}(\tau) = \boldsymbol{x}_0 + \int_{-1}^{\tau} T'(s)\boldsymbol{f}(\tilde{\boldsymbol{x}}(s),\tilde{\boldsymbol{u}}(s))\, ds, \quad \tau \in [-1,1). \tag{4.5}$$

This integral formulation offers several advantages over the differential form, particularly in avoiding the precision degradation typically associated with numerical differentiation procedures. The resulting FHOCI, described by Eqs. (4.3a), (4.5), and (4.3c), serves as the basis for our discretization scheme, which will be presented in the following section. The choice of Jacobi polynomials for the discretization provides greater flexibility compared to the Gegenbauer polynomials used in [28], as the two parameters α and β can be independently adjusted to optimize the numerical performance for specific problem characteristics.

5 Numerical Solution of the IHOC

Building upon the transformation established in the previous section, we now develop a discretization scheme for the resulting FHOCI. This approach employs the properties of Jacobi polynomials and JGR nodes to construct an efficient numerical solution method. The discretization process involves several key components, including the construction of integration matrices and the formulation of a NLP.

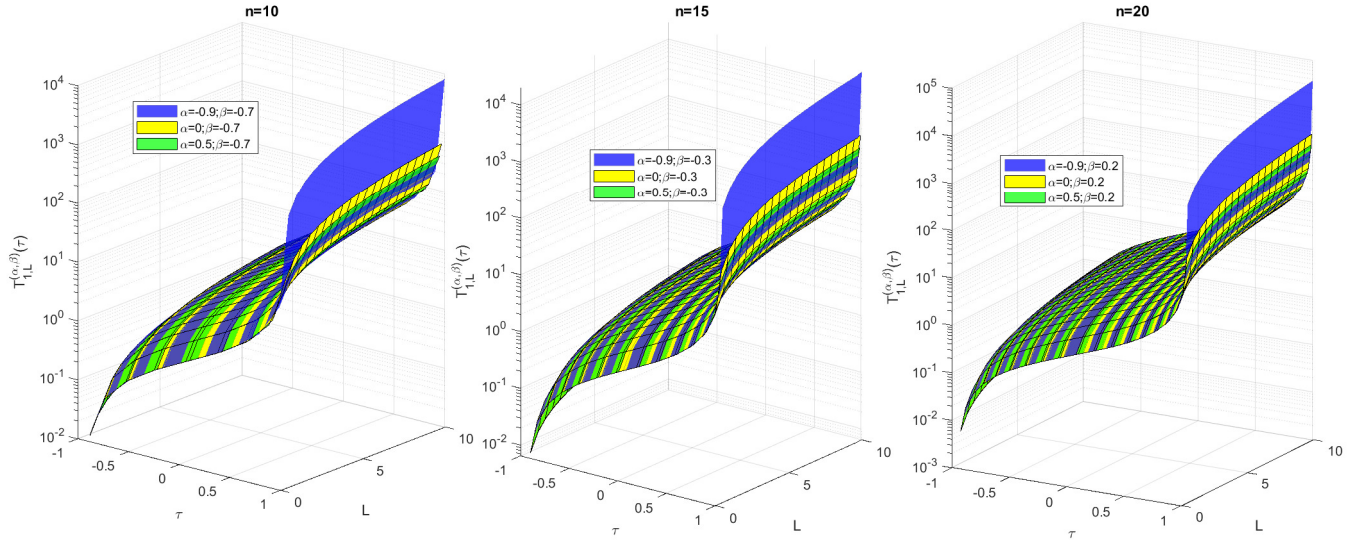

Figure 1: Grid-like surfaces of the parametric map $T_{1,L}^{(\alpha,\beta)}$ plotted above the planar region specified by the discrete set $\Omega_n = \{(\tau_i, L) : L = 0.5(0.5)10, i = 0(1)n\}$ using the parameters' ranges $n \in \{10, 15, 20\}$, $\alpha \in \{-0.9, 0, 0.5\}$, and $\beta \in \{-0.7, -0.3, 0.2\}$, with a logarithmic scale on the z-axis.

5.1 Collocation Points and Barycentric Interpolation

We begin by selecting a set of collocation points based on the JGR quadrature nodes. These points are particularly well-suited for our problem as they include the left endpoint of the interval $[-1, 1)$ while clustering points near the right endpoint, which corresponds to $t \to \infty$ in the original problem. The nodes of the JGR nodes set $\mathbb{S}_n = \{\tau_0, \tau_1, \ldots, \tau_n\}$ are defined as $-1 = \tau_0 < \tau_1 < \tau_2 < \ldots < \tau_n < 1$, where $\tau_i, i = 1(1)n$ are the roots of the scaled Jacobi polynomial $\mathscr{J}_{n+1}^{(\alpha,\beta)}(\tau) = (1+\tau)J_n^{(\alpha,\beta+1)}(\tau)$. Jacobi polynomials can be generated by the following recurrence relation [46]:

$$J_{n+1}^{(\alpha,\beta)}(\tau) = \left(a_n^{(\alpha,\beta)}\tau - b_n^{(\alpha,\beta)}\right) J_n^{(\alpha,\beta)}(\tau) - c_n^{(\alpha,\beta)} J_{n-1}^{(\alpha,\beta)}(\tau) \tag{5.1}$$

for $n \geq 1$, starting with $J_0^{(\alpha,\beta)}(\tau) = 1$ and $J_1^{(\alpha,\beta)}(\tau) = 0.5(\alpha+\beta+2)\tau + 0.5(\alpha-\beta)$, where

$$a_n^{(\alpha,\beta)} = \frac{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}{2(n+1)(n+\alpha+\beta+1)}, \tag{5.2}$$

$$b_n^{(\alpha,\beta)} = \frac{(\beta^2-\alpha^2)(2n+\alpha+\beta+1)}{2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)}, \tag{5.3}$$

$$c_n^{(\alpha,\beta)} = \frac{(n+\alpha)(n+\beta)(2n+\alpha+\beta+2)}{(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)}. \tag{5.4}$$

The orthonormal Jacobi basis polynomials are given by $\phi_j^{(\alpha,\beta)}(\tau) = J_j^{(\alpha,\beta)}(\tau)/\sqrt{\lambda_j}$, where

$$\lambda_j = \frac{2^{\alpha+\beta+1}\Gamma(j+\alpha+1)\Gamma(\beta+j+1)}{(2j+\alpha+\beta+1)j!\Gamma(j+\beta+\alpha+1)}, \quad j = 0(1)n. \tag{5.5}$$

Their discrete orthonormality relationship is given by:

$$\sum_{j=0}^{n} \varpi_j \phi_s^{(\alpha,\beta)}(\tau_j)\phi_k^{(\alpha,\beta)}(\tau_j) = \delta_{sk}, \quad s,k = 0(1)n, \tag{5.6}$$

where $\varpi_j, j = 0(1)n$, are the corresponding Christoffel numbers of the JGR quadrature formula on the domain $[-1, 1]$ defined by:

$$\varpi_0 = \frac{2^{\alpha+\beta+1}(\beta+1)\Gamma^2(\beta+1)n!\Gamma(n+\alpha+1)}{\Gamma(n+\beta+2)\Gamma(n+\alpha+\beta+2)}, \tag{5.7a}$$

$$\varpi_j = \frac{\vartheta_{n-1}^{\alpha,\beta+1}}{(1-\tau_j)(1+\tau_j)^2\left[\frac{d}{dx}\left(J_n^{(\alpha,\beta+1)}(\tau_j)\right)\right]^2}, \; j = 1(1)n, \tag{5.7b}$$

with

$$\vartheta_n^{\alpha,\beta} = \frac{2^{\alpha+\beta+1}\Gamma(n+\alpha+2)\Gamma(n+\beta+2)}{(n+1)!\Gamma(n+\alpha+\beta+2)}. \tag{5.7c}$$

For the interpolation of functions at the JGR nodes, we employ barycentric rational interpolation, which offers superior numerical stability compared to standard

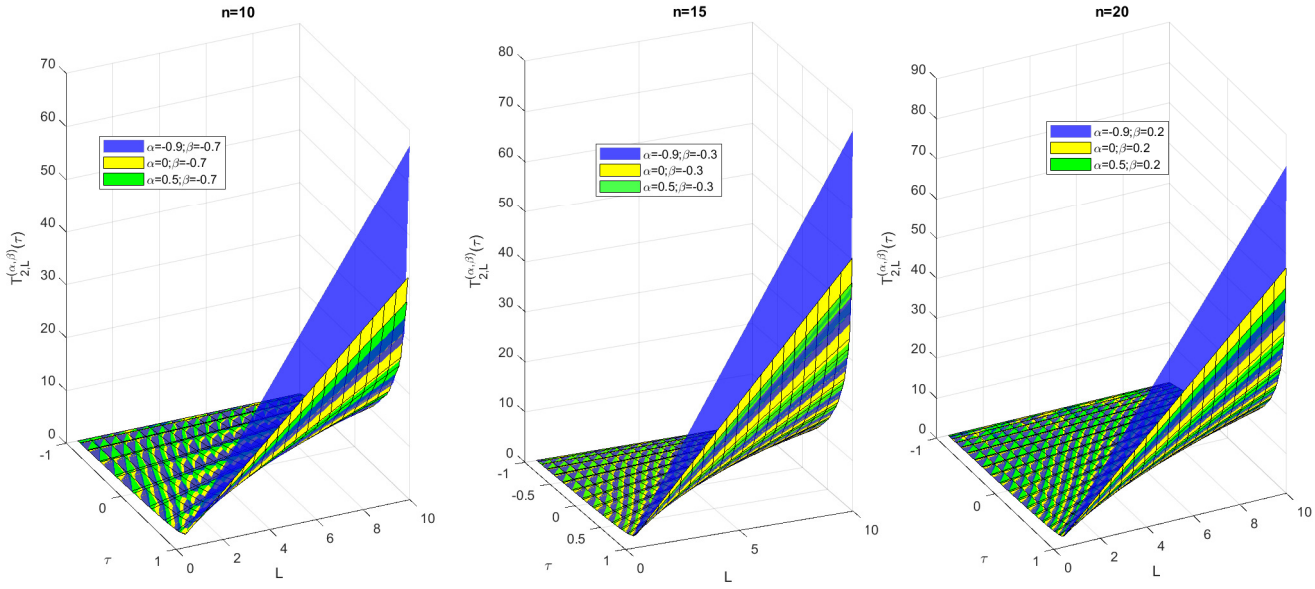

Figure 2: Grid-like surfaces of the parametric map $T_{2,L}^{(\alpha,\beta)}$ plotted above the planar region specified by the discrete set $\Omega_n = \{(\tau_i, L) : L = 0.5(0.5)10, i = 0(1)n\}$ using the parameters' ranges $n \in \{10, 15, 20\}, \alpha = -0.9, 0, 0.5$, and $\beta = -0.7, -0.3, 0.2$.

polynomial interpolation. The Lagrange interpolation polynomials in barycentric form are given by:

$$\mathscr{L}_{n,i}(\tau) = \frac{\xi_i/(\tau - \tau_i)}{\sum_{j=0}^{n} (\xi_j/(\tau - \tau_j))}, \quad i = 0(1)n, \tag{5.8}$$

where the barycentric weights $\xi_i, i = 0, \ldots, n$, are given by:

$$\xi_i = \frac{1}{\prod_{i \neq j}^{n} (\tau_j - \tau_i)}, \quad i = 0(1)n; \tag{5.9}$$

cf. [47]. For JGR nodes, the barycentric weights can be computed more efficiently using the following explicit formulas:

$$\xi_0 = -\sqrt{2(\beta+1)\varpi_0}, \tag{5.10a}$$

$$\xi_i = (-1)^{i-1}\sqrt{(1-\tau_i)\,\varpi_i}, \quad i = 1(1)n, \tag{5.10b}$$

where ϖ_i are the corresponding Christoffel numbers; cf. [48, Theorem 3.6].

Unlike the approach in [28], which employed a switching technique for computing barycentric weights across various ranges of Gegenbauer parameters, our analysis reveals that for many parameter combinations within the Jacobi family, the standard formula performs adequately without requiring the more complex switching approach. This simplification represents one of the practical advantages of our Jacobi-based method.

5.2 Construction of Integration Matrices

To discretize the integral constraints in Eq. (4.5), we construct integration matrices that approximate the definite integrals using the JGR nodes. Let $f \in C^{n+1}[-1, 1)$ be approximated by its interpolant $P_n f$ at the JGR nodes:

$$f(\tau) \approx P_n f(\tau) = \sum_{k=0}^{n} f_k \mathscr{L}_{n,k}(\tau), \quad \tau \in [-1, 1), \tag{5.11}$$

where $f_k = f(\tau_k)$ for $k = 0, 1, \ldots, n$. The integral of f from -1 to τ_j can then be approximated as:

$$\int_{-1}^{\tau_j} f(\tau)\, d\tau \approx \sum_{k=0}^{n} \mathscr{Q}_{j,k} f_k, \quad j = 0, 1, \ldots, n, \tag{5.12}$$

where $\mathscr{Q}_{j,k}$ are the elements of the integration matrix $\mathscr{Q}$, defined by:

$$\mathscr{Q}_{j,k} = \int_{-1}^{\tau_j} \mathscr{L}_{n,k}(\tau)\, d\tau, \quad j,k = 0, 1, \ldots, n$$

$$= \begin{cases} 0, & j = 0, k = 0(1)n, \\ \dfrac{(\tau_j + 1)}{2} \displaystyle\sum_{k=0}^{N} \bar{\varpi}_k \mathscr{L}_{n,k}(\bar{\tau}_k; -1, \tau_j), & j = 1(1)n, k = 0(1)n, \end{cases} \tag{5.13}$$

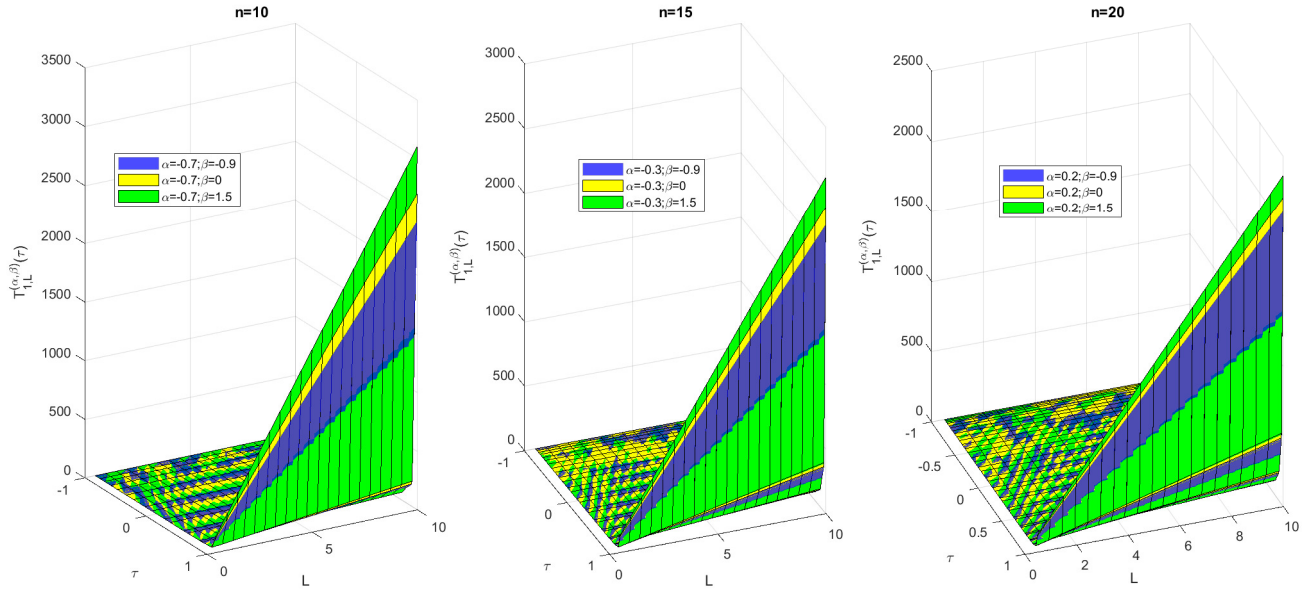

Figure 3: Grid-like surfaces of the parametric map $T_{1,L}^{(\alpha,\beta)}$ plotted above the planar region specified by the discrete set $\Omega_n = \{(\tau_i, L) : L = 0.5(0.5)10, i = 0(1)n\}$ using the parameters' ranges $n \in \{10, 15, 20\}$, $\alpha = -0.7, -0.3, 0.2$, and $\beta = -0.9, 0, 1.5$.

where $\{\bar{\tau}_k, \bar{\varpi}_k\}_{k=0}^{N}$ represent the set of weights and nodes in the LG quadrature, respectively:

$$\bar{\varpi}_k = \frac{2}{\left(1-\bar{\tau}_k^2\right)\left(L'_{N+1}(\bar{\tau}_k)\right)^2}, \quad k = 0(1)N, \tag{5.14}$$

and L'_{N+1} indicates the derivative of the $(N+1)$st-degree Legendre polynomial L_{N+1} [28]. The efficient computation of the integration matrix using the properties of Jacobi polynomials often results in a well-conditioned matrix that accurately approximates the integral operations required for our discretization through matrix-vector multiplications:

$$\int_{-1}^{\tau_j} f(\tau)\, d\tau \approx \mathscr{Q}\boldsymbol{f}, \quad j = 0(1)n, \tag{5.15}$$

where $\boldsymbol{f} = [f_0, f_1, \ldots, f_n]^t$.

5.3 Discretization of the FHOCI

Using the integration matrix $\mathscr{Q}$ and collocation at the JGR nodes, we discretize the FHOCI described by Eqs. (4.3a), (4.5), and (4.3c). Let $\tilde{\boldsymbol{x}}_j = \tilde{\boldsymbol{x}}(\tau_j)$ and $\tilde{\boldsymbol{u}}_j = \tilde{\boldsymbol{u}}(\tau_j)$ for $j = 0, 1, \ldots, n$ represent the discretized state and control variables at the collocation points. The integral constraint in Eq. (4.5) is discretized as:

$$\tilde{\boldsymbol{x}}_j = \boldsymbol{x}_0 + \sum_{k=0}^{n} \mathscr{Q}_{j,k} T'(\tau_k) \boldsymbol{f}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{u}}_k), \quad j = 0, 1, \ldots, n. \tag{5.16}$$

Similarly, the cost functional in Eq. (4.3a) is approximated by:

$$\tilde{J} \approx \sum_{k=0}^{n} \mathscr{Q}_{n+1,k} T'(\tau_k) g(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{u}}_k), \tag{5.17}$$

where $\mathscr{Q}_{n+1,k} = \sum_{i=0}^{N} \bar{\varpi}_i \mathscr{L}_{n,k}(\bar{\tau}_i)\ \forall k$.

5.4 Resulting NLP

The discretization process transforms the FHOCI into a finite-dimensional NLP of the form:

$$\min_{\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n, \tilde{\boldsymbol{u}}_0, \tilde{\boldsymbol{u}}_1, \ldots, \tilde{\boldsymbol{u}}_n} \sum_{k=0}^{n} \mathscr{Q}_{n+1,k} T'(\tau_k) g(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{u}}_k), \tag{5.18a}$$

$$\text{subject to} \quad \tilde{\boldsymbol{x}}_j = \boldsymbol{x}_0 + \sum_{k=0}^{n} \mathscr{Q}_{j,k} T'(\tau_k) \boldsymbol{f}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{u}}_k),\ j = 0(1)n, \tag{5.18b}$$

$$\tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_0. \tag{5.18c}$$

This NLP can be solved using standard optimization techniques such as SQP or interior point methods. The

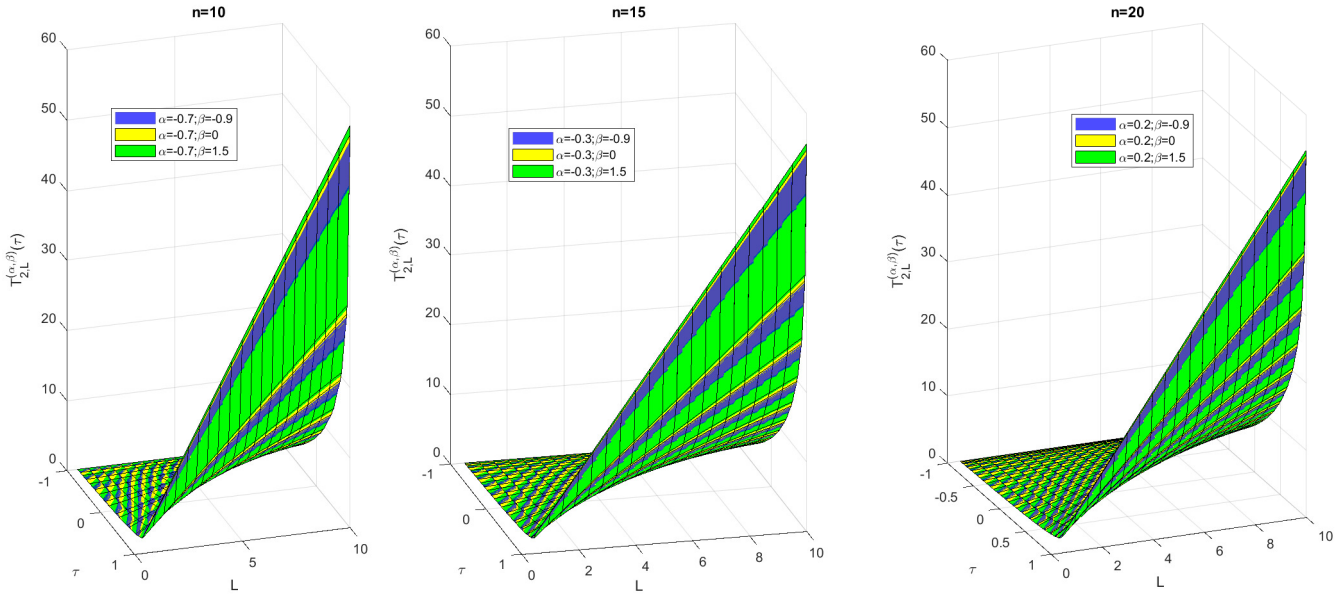

Figure 4: Grid-like surfaces of the parametric map $T_{2,L}^{(\alpha,\beta)}$ plotted above the planar region specified by the discrete set $\Omega_n = \{(\tau_i, L) : L = 0.5(0.5)10, i = 0(1)n\}$ using the parameters' ranges $n \in \{10, 15, 20\}$, $\alpha = -0.7, -0.3, 0.2$, and $\beta = -0.9, 0, 1.5$.

solution provides the optimal state and control trajectories at the collocation points, which can then be interpolated to obtain continuous approximations of the optimal trajectories.

The use of Jacobi polynomials in this discretization scheme offers greater flexibility compared to the Gegenbauer polynomials used in [28], as the two parameters α and β can be independently adjusted to optimize the numerical performance for specific problem characteristics. We refer to the present Jacobi-based IPS method by the *"JGR-IPS"* method. The acronyms *"JGR-IPS1"* and *"JGR-IPS2"* stand for the JGR-IPS method performed using the parametric mappings $T_{1,L}^{(\alpha,\beta)}$ and $T_{2,L}^{(\alpha,\beta)}$, respectively, while *"JGR-IPS12"* stands for the JGR-IPS method performed using either maps $T_{1,L}^{(\alpha,\beta)}$ and $T_{2,L}^{(\alpha,\beta)}$.

In the following section, we analyze the mathematical properties of this discretization scheme, including its convergence behavior and error bounds.

6 Numerical Analysis

Having developed our JGR-IPS, we now analyze its mathematical properties and performance characteristics. This analysis provides theoretical insights into the method's convergence behavior, stability properties, and error bounds, establishing a solid foundation for understanding its practical performance. While some aspects of this analysis build upon the framework established in [28] for Gegenbauer polynomials, we extend these results to the more general Jacobi polynomial family and derive new insights specific to this broader framework.

6.1 Analysis of JGR-Based Interpolation/Collocation: Stability and Sensitivity

The Lebesgue constant is a practical tool for assessing the accuracy and numerical stability of polynomial interpolations, as it indicates how closely a function's interpolant matches the function's best polynomial approximant. For the JGR-based interpolation scheme, the Lebesgue constant Λ_n is defined as:

$$\Lambda_n = \max_{\tau\in[-1,1]} \sum_{i=0}^{n} |\mathscr{L}_{n,i}(\tau)|, \tag{6.1}$$

where $\mathscr{L}_{n,i}(\tau)$ are the Lagrange interpolation polynomials in barycentric form as defined in Eq. (5.8).

The growth rate of the Lebesgue constant with respect to n provides valuable insights into the stability of the interpolation scheme. For JGR nodes with Jacobi

parameters α and β, we can establish bounds on the Lebesgue constant that depend on these parameters. This analysis extends the results presented in [28] for Gegenbauer polynomials to the more general Jacobi polynomial family.

The Lebesgue constant's surface for JGR points is shown in Figures 5 and 6. The Lebesgue constant is estimated by maximizing $\sum_{i=0}^{n}|\mathscr{L}_{n,i}(\tau)|$ using the MATLAB `fminbnd` solver. The surface is constructed using least-squares approximation and displayed alongside its cross-sections with the vertical planes specified in each subfigure. We infer the following observations:

(i) The Lebesgue constant generally remains stable as $\alpha \to -1$ or $\alpha \to 2$ when β is constant, and similarly, for constant α, it remains stable as $\beta \to -1$ or $\beta \to 2$ (see Figures 5 and 6).
(ii) The Lebesgue constant grows logarithmically with the number of collocation nodes, as evident in Figures 5 and 6.

Figures 7 and 8 depict the minimum Lebesgue constant computed for $\beta \in \{-0.999, -0.99, -0.9(0.1)2\}$ versus α and for $\alpha \in \{-0.999, -0.99, -0.9(0.1)2\}$ versus β, respectively, for various values of n. The following observations are noted:

(iii) For most values of n, the minimum Lebesgue constant decreases as α increases, with the smallest values often occurring near $\alpha = 3$ for larger n (see Figure 7).
(iv) Similarly, for most values of n, the minimum Lebesgue constant decreases as β increases, with the smallest values often occurring near $\beta = 3$ for larger n (see Figure 8).

Figures 9 and 10 illustrate the Jacobi weight function for various α and β values. Figure 9 shows that increasing α causes the weight function to diminish on the right of the interval, while Figure 10 shows that increasing β causes it to diminish on the left. Consequently, JGR quadrature becomes extrapolatory for large α and β values, relying heavily on nodes away from either the left or right of the interval, as established in [49, Theorem 2.5] for Gegenbauer polynomials. Thus, Jacobi polynomials with $\alpha > 2$ and $\beta > 2$ are generally unsuitable for interpolation or collocation.

The above analysis highlights the behavior of the parametric mappings in achieving accurate approximations with the JGR-IPS method. The following rule of thumb provides guidance on selecting optimal α and β values for interpolation and collocation based on Jacobi polynomials:

Rule of Thumb. We suggest choosing the Jacobi parameters α and β for interpolation and collocation as follows:

$$(\alpha,\beta) \in [-1+\varepsilon, 2]^2, \quad 0 < \varepsilon \ll 1. \tag{6.2}$$

The analysis of the Lebesgue constant for JGR-based interpolation using Jacobi polynomials reveals that it grows logarithmically with n for specific ranges of the parameters α and β, indicating good numerical stability. However, the specific growth rate is highly dependent on the chosen parameter values, with certain combinations yielding superior stability. Key observations include:

- For $\beta = -0.9$, 0, and 0.7, the Lebesgue constant achieves its minimum when $\alpha = -0.999$, -0.1, and -0.1, respectively.
- Conversely, for $\alpha = -0.9$, 0, and 0.7, the Lebesgue constant is minimized when $\beta = 0.5$, 1, and 1, respectively.

These observations underscore the advantage of the Jacobi polynomial approach, as they demonstrate that careful tuning of α and β can optimize numerical stability for specific problem characteristics. The dependence of the Lebesgue constant on α and β suggests that Jacobi polynomial-based interpolation can be tailored to minimize approximation errors by selecting optimal parameter pairs. Specifically, the identified values of α and β that minimize the Lebesgue constant indicate configurations where the interpolation process is particularly stable. This tunability is a significant advantage, enabling practitioners to adapt the interpolation scheme to the specific requirements of their problem, potentially leading to more accurate and robust numerical results.

Remark 61. *When $\alpha \neq \beta$, the Jacobi weight function $w^{(\alpha,\beta)}(x) = (1-x)^{\alpha}(1+x)^{\beta}$ becomes asymmetric, allowing the collocation scheme to adapt more flexibly to solution features that are unevenly distributed across the domain. This asymmetry can improve resolution near critical regions such as boundary layers or steep gradients, where symmetric weight functions may fail to concentrate sufficient collocation density. Consequently, selecting $\alpha \neq \beta$ can yield improved approximation quality for problems with localized features or exhibiting asymmetric solution profiles.*

Remark 62. *For the Lebesgue constant analysis conducted in this section, we assume the problem we are tackling is well-conditioned. Otherwise, the recommended interval $[-1+\varepsilon, 2]$ may not be valid, and its choice generally depends on the sources of sensitivity.*

6.2 Error Analysis and Convergence Properties

This section provides the bounds of the truncation error and convergence rates for the discretized integral constraints and cost functional. We begin with a theorem characterizing the truncation error of the Jacobi interpolant.

Theorem 63. *Let $f \in C^{n+1}[-1,1)$ be approximated by a Jacobi interpolant $P_n f$ based on the JGR nodes set $\mathbb{S}_n$.*

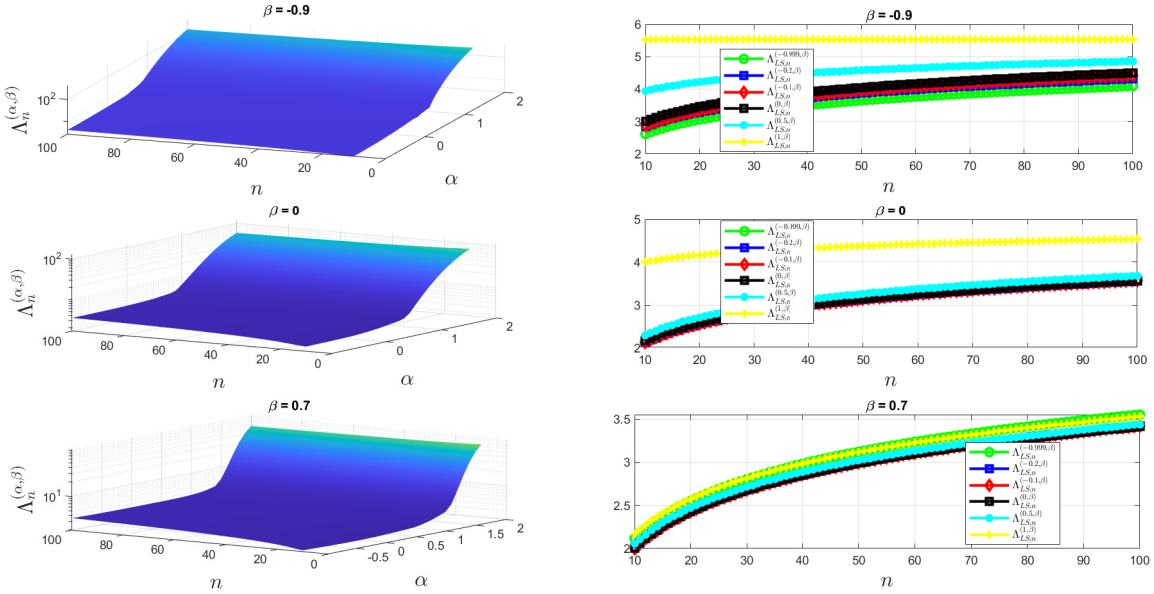

Figure 5: The left plots display the surfaces of the Lebesgue constant for JGR-based interpolation, sketched over the planar region defined by the discrete set $\{(n,\alpha) : n = 10(10)100, \alpha \in [-0.999, -0.99, -0.9(0.1)2]\}$, with $\beta \in \{-0.9, 0, 0.7\}$. These surfaces are constructed using a least-squares approximation with curves of the form $c_1 + c_2 \ln n$, for real parameters c_1 and c_2, and the z-axis is set to a logarithmic scale. The right plots illustrate the Lebesgue constant versus $n = 10(10)100$ for $\alpha \in \{-0.999, -0.2, -0.1, 0, 0.5, 1\}$ and the same values of β.

Then there exist $(n+1)$ *numbers* $\xi_i \in (-1,1), i = 0(1)n$, *such that the truncation error of the approximation is given by:*

$$ {}_fE_n(\tau_i,\xi_i) = \frac{f^{(n+1)}(\xi_i)}{(n+1)!K_n^{(\alpha,\beta+1)}} \int_{-1}^{\tau_i} \mathscr{J}_{n+1}^{(\alpha,\beta)}(\tau)\,d\tau \quad \forall i, \tag{6.3} $$

where

$$ K_n^{(\alpha,\beta)} = \frac{\Gamma(2n+\alpha+\beta+1)}{2^n n!\Gamma(n+\alpha+\beta+1)}, \forall n. \tag{6.4} $$

Proof. By the error formula for Lagrange interpolation, we can write:

$$ f(\tau) = \sum_{k=0}^{n} f_k \mathscr{L}_{n,k}(\tau) + {}_fE_n(\tau,\xi), \quad \forall -1 \le \tau < 1, \tag{6.5} $$

for specified $-1 < \xi < 1$, where ${}_fE_n$ is the interpolation truncation error at the JGR nodes given by:

$$ {}_fE_n(\tau,\xi) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (\tau - \tau_k). \tag{6.6} $$

By acknowledging that $\mathscr{J}_{n+1}^{(\alpha,\beta)}(\tau) = K_n^{(\alpha,\beta+1)} \prod_{k=0}^{n}(\tau-\tau_k)$ and integrating Eq. (6.5) over $[-1,\tau_i)\ \forall i$, the proof is established.

Building on this result, we can establish error bounds for the discretized integral constraints and cost functional. The following corollary applies this error analysis to our specific problem formulation.

Corollary 64. *Let* $\eta : [-1,1) \to \mathbb{R}$ *and* $\psi_k : [-1,1) \to \mathbb{R}$ *be defined as* $\eta(\tau) = T'(\tau) g(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau))$ *and* $\psi_k(\tau) = T'(\tau) f_k(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau))$, *for each* $k = 1(1)n_x$. *Then the truncation errors of the discretized cost functional and integral constraints at each collocation point* $\tau_j \in \mathbb{S}_n$ *are given by:*

$$ {}_\eta E_n(\zeta) = \frac{\eta^{(n+1)}(\zeta)}{(n+1)!K_n^{(\alpha,\beta+1)}} \int_{-1}^{1} \mathscr{J}_{n+1}^{(\alpha,\beta)}(\tau)\,d\tau, \tag{6.7} $$

and

$$ {}_{\psi_k}E_n(\tau_j,\xi_j) = \frac{\psi_k^{(n+1)}(\xi_j)}{(n+1)!K_n^{(\alpha,\beta+1)}} \int_{-1}^{\tau_j} \mathscr{J}_{n+1}^{(\alpha,\beta)}(\tau)\,d\tau, \quad k = 1(1)n_x, \quad j = 0(1)n, \tag{6.8} $$

respectively, where $\zeta, \xi_j \in (-1,1)\,\forall j$.

These error expressions can be further bounded to provide more practical estimates of the truncation error. The following theorem establishes such bounds.

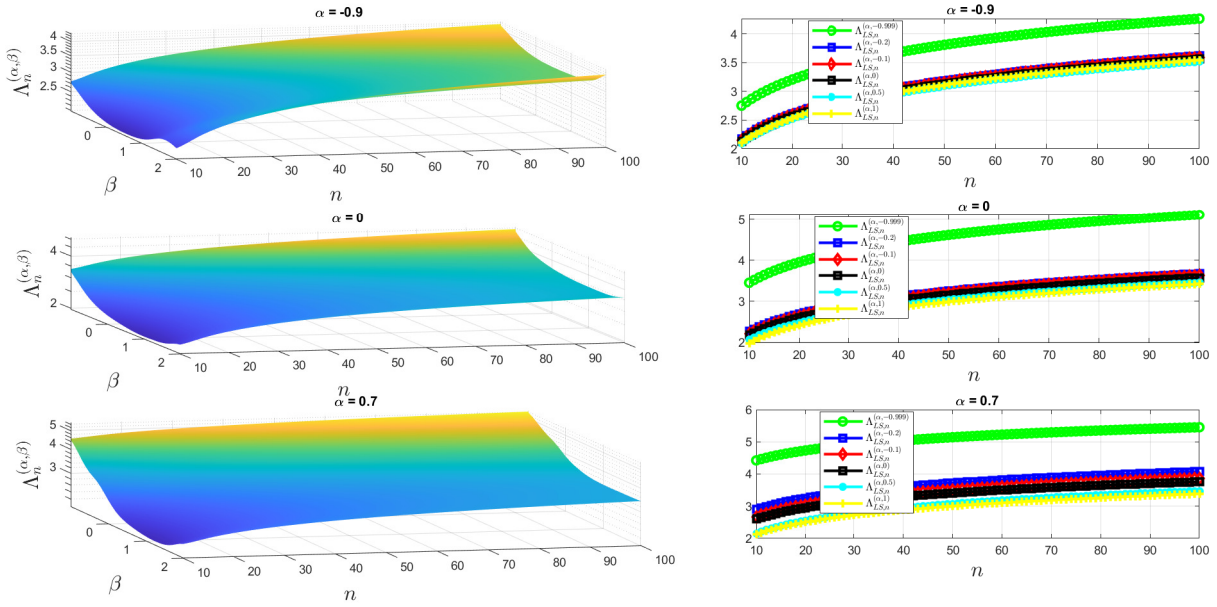

Figure 6: The left plots display the surfaces of the Lebesgue constant for JGR-based interpolation, sketched over the planar region defined by the discrete set $\{(n,\beta) : n = 10(10)100, \beta \in [-0.999, -0.99, -0.9(0.1)2]\}$, with $\alpha \in \{-0.9, 0, 0.7\}$. These surfaces are constructed using a least-squares approximation with curves of the form $c_1 + c_2 \ln n$, for real parameters c_1 and c_2. The right plots illustrate the Lebesgue constant versus $n = 10(10)100$ for $\beta \in \{-0.999, -0.2, -0.1, 0, 0.5, 1\}$ and the same values of α.

Theorem 65. *Let* $\psi_k \in C^{n+1}[-1,1)$, $n > 1$, *and* $\left\|\psi_k^{(n+1)}\right\|_{[-1,1)} = A_{\psi_k,n} \in \mathbb{R}^+ \forall k$, *for some constant* $A_{\psi_k,n}$ *dependent on* n *and* k. *Then the truncation errors of the discretized integral constraints at each collocation point* $\tau_j \in \mathbb{S}_n$ *are bounded by:*

$$\left|\psi_k E_n\left(\tau_j, \xi_j\right)\right| \leq \frac{A_{\psi_k,n} 2^n \Gamma(n+\alpha+\beta+2)\left(1+\tau_j\right)}{(n+1)\Gamma(2n+\alpha+\beta+2)} \left\|\mathscr{J}_{n+1}^{(\alpha,\beta)}\right\|_{[-1,1)}, \quad k = 1(1)n_x, \quad j = 0(1)n, \tag{6.9}$$

where $-1 < \xi_j < 1 \forall j$, *and*

$$\left\|\mathscr{J}_{n+1}^{(\alpha,\beta)}\right\|_{[-1,1)} = \begin{cases} \dfrac{2(2n+\beta+q+2)}{2n+\alpha+\beta+2}\binom{n+q}{n} = O\left(n^q\right), & q \geq -1/2, \\ \dfrac{\left|2(n+\beta+1)J_n^{(\alpha,\beta)}(x')\right| + \left|2(n+1)J_{n+1}^{(\alpha,\beta)}(x'')\right|}{|2n+\alpha+\beta+2|} = O\left(n^{-1/2}\right), & q < -1/2, \end{cases} \tag{6.10}$$

with $q = \max(\alpha,\beta)$ *governing the growth rate of the scaled Jacobi polynomial norm,* x' *being the maximum point of* $J_n^{(\alpha,\beta)}(\tau)$ *nearest to* $x_0 = \dfrac{\beta-\alpha}{\alpha+\beta+1}$, *and* x'' *being the maximum point of* $J_{n+1}^{(\alpha,\beta)}(\tau)$ *nearest to* x_0. *Furthermore, as* $n \to \infty$, *we have:*

$$\left|\psi_k E_n\left(\tau_j, \xi_j\right)\right| \lesssim A_{\psi_k,n}\left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+1-q}}, \quad \forall q \geq -1/2, \tag{6.11}$$

and

$$\left|\psi_k E_n\left(\tau_j, \xi_j\right)\right| \lesssim A_{\psi_k,n}\left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+\frac{3}{2}}}, \quad \forall -1 < q < -1/2. \tag{6.12}$$

Proof. Eq. (3.116b) from [46] immediately yields:

$$(1+\tau)J_n^{(\alpha,\beta+1)}(\tau) = \frac{2(n+\beta+1)J_n^{(\alpha,\beta)}(\tau) + 2(n+1)J_{n+1}^{(\alpha,\beta)}(\tau)}{2n+\alpha+\beta+2}. \tag{6.13}$$

Therefore:

$$\left\|\mathscr{J}_{n+1}^{(\alpha,\beta)}\right\|_{[-1,1)} = \left\|\frac{2(n+\beta+1)}{2n+\alpha+\beta+2}J_n^{(\alpha,\beta)}(\tau) + \frac{2(n+1)}{2n+\alpha+\beta+2}J_{n+1}^{(\alpha,\beta)}(\tau)\right\|_{[-1,1)}. \tag{6.14}$$

Eq. (6.10) can be obtained by applying the triangle inequality to the right-hand side of Eq. (6.14) and

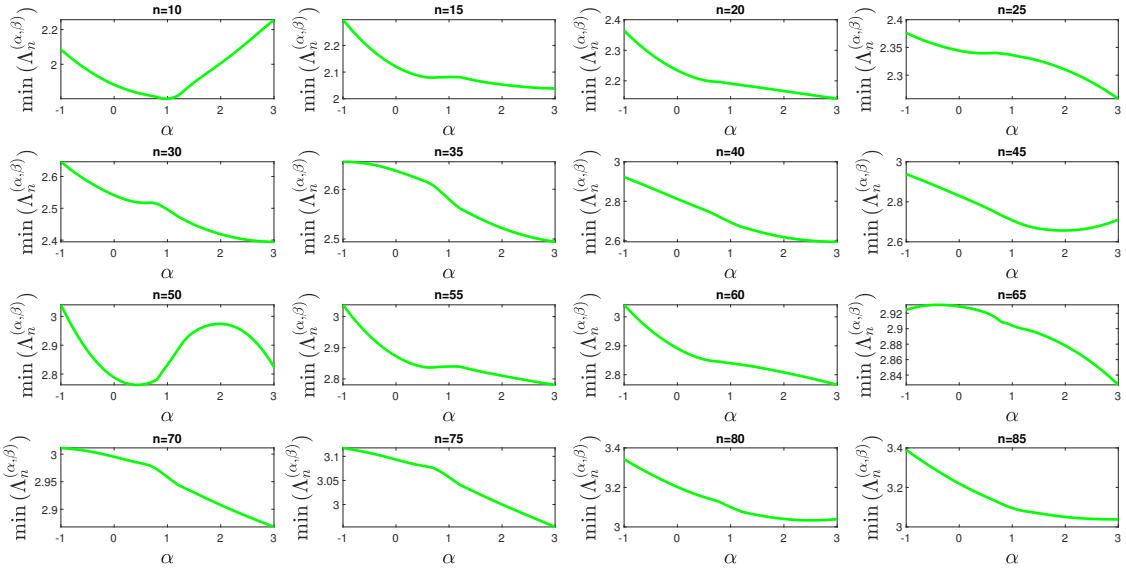

Figure 7: Minimum Lebesgue constant for JGR-Based Interpolation versus α: For each value of α, the Lebesgue constant is minimized over all β values in the range $\{-0.999, -0.99, -0.9(0.1)3\}$. The minimum Lebesgue constant at each $n = 10(5)85$ is plotted for $\alpha \in \{-0.999, -0.99, -0.9(0.1)3\}$.

utilizing the relation from [56, Eq. (7.32.2)]:

$$\left\|J_n^{(\alpha,\beta)}\right\|_{[-1,1)} = \max_{[-1,1]}\left|J_n^{(\alpha,\beta)}\right| = \begin{cases} \binom{n+q}{n} \sim n^q, & q = \max(\alpha,\beta) \geq -1/2, \\ \left|J_n^{(\alpha,\beta)}(x')\right| \sim n^{-1/2}, & q = \max(\alpha,\beta) < -1/2. \end{cases} \tag{6.15}$$

Inequality (6.9) can be obtained with the help of Eqs. (6.4) and (6.8). The asymptotic inequalities (6.11) and (6.12) follow directly from Lemma A1.

Similar bounds can be established for the truncation error of the discretized cost functional, as stated in the following theorem.

Theorem 66. *Let* $\eta \in C^{n+1}[-1,1)$ *and* $\left\|\eta^{(n+1)}\right\|_{[-1,1)} = A_{\eta,n} \in \mathbb{R}^+$*, for some constant* $A_{\eta,n}$ *dependent on n. Then the truncation error of the discretized cost functional is bounded by:*

$$\left|{}_\eta E_n(\zeta)\right| \leq \frac{A_{\eta,n} 2^{n+1}\Gamma(n+\alpha+\beta+2)}{(n+1)\Gamma(2n+\alpha+\beta+2)} \left\|\mathscr{J}_{n+1}^{(\alpha,\beta)}\right\|_{[-1,1)}, \tag{6.16}$$

where $-1 < \zeta < 1$*. Furthermore, as* $n \to \infty$*, we have:*

$$\left|{}_\eta E_n(\zeta)\right| \underset{\sim}{\leq} B_{\eta,n}\left(\frac{e}{2}\right)^n \frac{1}{n^{n+1-q}}, \quad \forall q \geq -1/2, \tag{6.17}$$

and

$$\left|{}_\eta E_n(\zeta)\right| \underset{\sim}{\leq} B_{\eta,n}\left(\frac{e}{2}\right)^n \frac{1}{n^{n+\frac{3}{2}}}, \quad \forall -1 < q < -1/2, \tag{6.18}$$

where $B_{\eta,n} = CA_{\eta,n}$*, for some positive constant C, and q is as defined by Eq.* (6.15)*.*

Proof. Inequality (6.16) can be obtained with the help of Eqs. (6.4) and (6.7). The asymptotic inequalities (6.17) and (6.18) follow directly from Lemma A1.

Although the previous theorems indicate that our method is convergent, we expect divergence as the mesh size grows large if the computations are carried out using floating-point arithmetic, extending the observation made in [28] for Gegenbauer polynomials to the more general Jacobi polynomial family. The next section highlights this fact with rigorous mathematical proofs.

6.3 Divergence Analysis for Large Mesh Grids

In this section, we analyze the convergence behavior of our method for large mesh grids, revealing some striking results regarding the limitations of typical collocation schemes when applied to the transformed FHOCI. The following corollary establishes the divergence of the method for large mesh sizes under certain conditions.

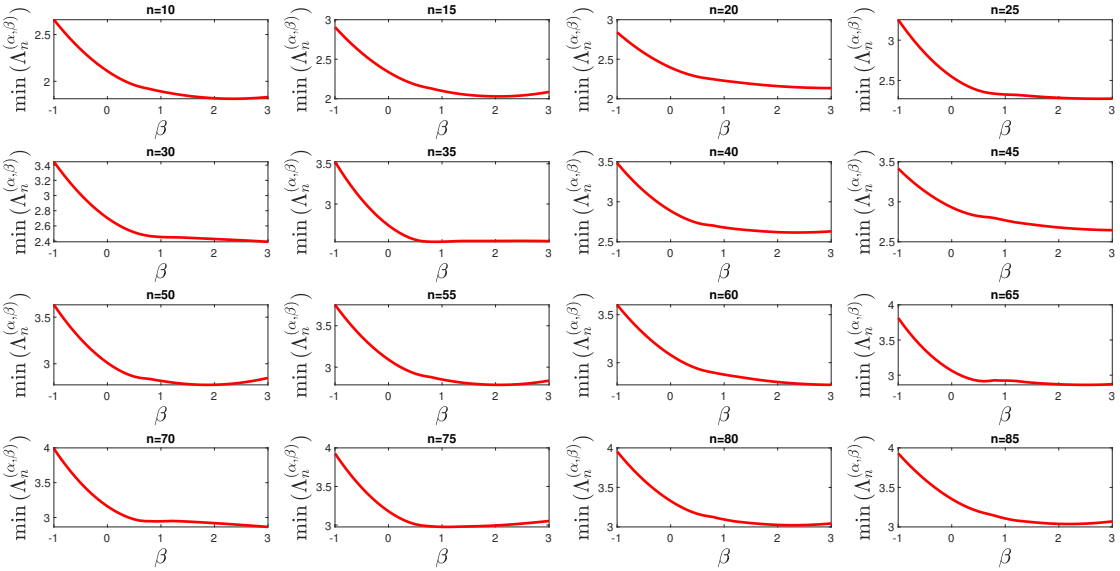

Figure 8: Minimum Lebesgue constant for JGR-Based Interpolation versus β: For each value of β, the Lebesgue constant is minimized over all α values in the range $\{-0.999,-0.99,-0.9(0.1)3\}$. The minimum Lebesgue constant at each $n = 10(5)85$ is plotted for $\beta \in \{-0.999,-0.99,-0.9(0.1)3\}$.

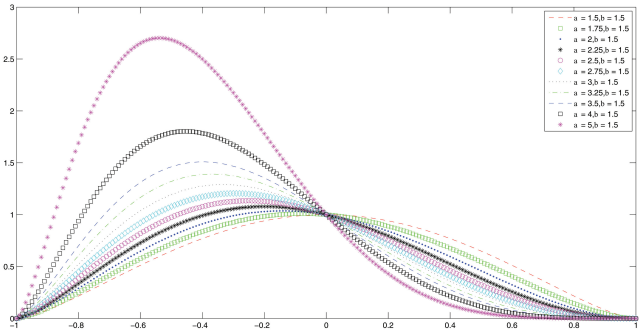

Figure 9: The profile of the Jacobi weight function $w^{(\alpha,\beta)}(x) = (1-x)^{\alpha}(1+x)^{\beta}$ for $\alpha \in \{1.5(0.25)3.5,4,5\}$ and $\beta = 1.5$. In the figure legend, 'a' and 'b' refer to α and β, respectively.

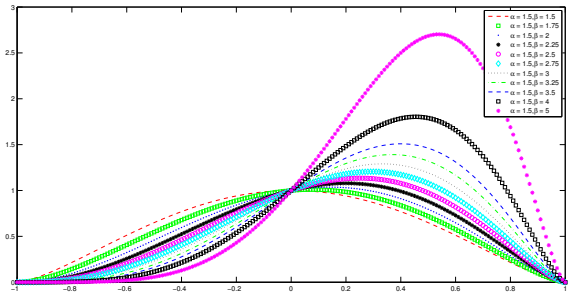

Figure 10: The profile of the Jacobi weight function $w^{(\alpha,\beta)}(x)$ for $\alpha = 1.5$ and $\beta \in \{1.5(0.25)3.5,4,5\}$.

Corollary 67. *Let $T \in \{T_{1,L}^{(\alpha,\beta)}, T_{2,L}^{(\alpha,\beta)}\}$, and suppose that $\exists \hat{k} \in \{1,\ldots,n_x\} : \psi_{\hat{k}} \in C^n[-1,1),\ 0 < \left\|f_{\hat{k}}\right\|_{[-1,1)} < \infty$, and $0 \le \left\|\dfrac{d^j}{d\tau^j} f_{\hat{k}}\right\|_{[-1,1)} < \infty \quad \forall j = 1,\ldots,n+1$. Then the upper truncation error bounds of the discretized integral constraints diverge at each collocation point as $n \to \infty$, for any map scaling parameter value L.*

Proof. By the General Leibniz Rule, the $(n+1)$st-derivative of $\psi_{\hat{k}}$ is given by:

$$\psi_{\hat{k}}^{(n+1)}(\tau) = \sum_{j=0}^{n+1} \binom{n+1}{j} T^{(n+2-j)}(\tau) \frac{d^j}{d\tau^j} f_{\hat{k}}(\tilde{\boldsymbol{x}}(\tau), \tilde{\boldsymbol{u}}(\tau)), \tag{6.19}$$

thus:

$$\left\|\psi_{\hat{k}}^{(n+1)}\right\|_{[-1,1)} = \sum_{j=0}^{n+1} \binom{n+1}{j} \left\|T^{(n+2-j)}\right\|_{[-1,1)} \left\|\frac{d^j}{d\tau^j} f_{\hat{k}}\right\|_{[-1,1)}. \tag{6.20}$$

Let $T = T_{1,L}^{(\alpha,\beta)}$, and notice that $\left(T_{1,L}^{(\alpha,\beta)}\right)^{(m)}(\tau) = \frac{2L(m)!}{(1-\tau)^{m+1}} \forall m \in \mathbb{Z}^+$, which is a monotonically increasing function for increasing values of τ as clearly seen from Figure 11. Therefore, $A_{\psi_{\hat{k}},n} = O\left(\left\|\left(T_{1,L}^{(\alpha,\beta)}\right)^{(n+2)}\right\|_{[-1,1)}\right)$. From Theorem 65:

$$\left|\psi_{\hat{k}} E_n(\tau_j,\xi_j)\right| \leq 2L(n+2)!\left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+1-q}}\left\|(1-\tau)^{-n-3}\right\|_{[-1,1)}, \quad \forall q \geq -1/2, \tag{6.21a}$$

and

$$\left|\psi_{\hat{k}} E_n(\tau_j,\xi_j)\right| \lesssim 2L(n+2)!\left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+\frac{3}{2}}}\left\|(1-\tau)^{-n-3}\right\|_{[-1,1)}, \quad \forall -1 < q < -1/2, \tag{6.21b}$$

whence we realize that the upper bound of $\left|\psi_{\hat{k}} E_n\right|$ at each collocation point τ_j diverges as $n \to \infty$. Consider now the case when $T = T_{2,L}^{(\alpha,\beta)}$. By a similar argument, notice first that $\left(T_{2,L}^{(\alpha,\beta)}\right)^{(m)}(\tau) = \frac{L(m-1)!}{(1-\tau)^m}\ \forall m \in \mathbb{Z}^+$ is also a monotonically increasing function for increasing values of τ as shown by Figure 11. Therefore, $A_{\psi_{\hat{k}},n} = O\left(\left\|\left(T_{2,L}^{(\alpha,\beta)}\right)^{(n+2)}\right\|_{[-1,1)}\right)$. From Theorem 65:

$$\left|\psi_{\hat{k}} E_n(\tau_j,\xi_j)\right| \leq L(n+1)!\left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+1-q}}\left\|(1-\tau)^{-n-2}\right\|_{[-1,1)}, \quad \forall q \geq -1/2, \tag{6.22a}$$

and

$$\left|\psi_{\hat{k}} E_n(\tau_j,\xi_j)\right| \lesssim L(n+1)!\left(\frac{e}{2}\right)^n \frac{1+\tau_j}{n^{n+\frac{3}{2}}}\left\|(1-\tau)^{-n-2}\right\|_{[-1,1)}, \quad \forall -1 < q < -1/2, \tag{6.22b}$$

from which we observe that the upper bound of $\left|\psi_{\hat{k}} E_n\right|$ at each collocation point τ_j diverges as $n \to \infty$.

Under a similar proof to that of Corollary 67, one can derive the following second divergence result.

Corollary 68. *Let $T \in \{T_{1,L}^{(\alpha,\beta)}, T_{2,L}^{(\alpha,\beta)}\}$, $\eta \in C^n[-1,1)$, $0 < \|g\|_{[-1,1)} < \infty$, and $0 \leq \left\|\frac{d^j}{d\tau^j} g\right\|_{[-1,1)} < \infty \forall j = 1,\ldots,n+1$, then the upper truncation error bound of the discretized cost functional diverges as $n \to \infty$, for any map scaling parameter value L.*

These results reveal an important limitation of our method: While it is expected to converge exponentially for small or moderate values of n, divergence becomes inevitable as n grows very large, regardless of the choice of the map scaling parameter L. This behavior is a consequence of the ill-conditioning introduced by the parametric mappings, which becomes increasingly severe as the mesh size grows.

The practical implication of this analysis is that our method should be used with appropriately sized mesh grids, typically in the range where exponential convergence is observed before the onset of divergence. This characteristic is not unique to our Jacobi-based approach but is shared by many PS methods applied to infinite-horizon problems using similar parametric mappings.

Figure 11 provides critical insight into the comparative behavior of the two parametric mappings $T_{1,L}^{(\alpha,\beta)}$ and $T_{2,L}^{(\alpha,\beta)}$. While both mappings exhibit divergence as $n \to \infty$, the derivatives of $T_{1,L}^{(\alpha,\beta)}$ grow more rapidly near $\tau = 1$, with $\left(T_{1,L}^{(\alpha,\beta)}\right)^{(m)}(\tau) = \frac{2L(m)!}{(1-\tau)^{m+1}}$ compared to $\left(T_{2,L}^{(\alpha,\beta)}\right)^{(m)}(\tau) = \frac{L(m-1)!}{(1-\tau)^m}$. This fundamental difference—an additional factor of 2 and higher-order singularity at $\tau = 1$—explains why $T_{1,L}^{(\alpha,\beta)}$ diverges faster than $T_{2,L}^{(\alpha,\beta)}$ for large n values. As demonstrated in the proof of Corollary 67, the ill-conditioning of $T_{1,L}^{(\alpha,\beta)}$ is more severe by a factor of $2/(1-\tau)$, which becomes particularly problematic as τ approaches 1. Consequently, $T_{2,L}^{(\alpha,\beta)}$ emerges as the superior choice for large-scale mesh implementations.

Remark 69 (Practical Guidance on Mesh Size Selection)**.** *Before selecting the mesh size for practical implementation, it is advisable to monitor convergence metrics iteratively. For example, we may begin with coarse grids (e.g., $n = 5-10$) and incrementally increase n while tracking key indicators such as the MRE, the AE_J, or the $MAE_{x,u}$. These metrics typically decrease exponentially until instability sets in. To avoid divergence, we can select the largest n before the onset of increasing or oscillating errors, as indicated by these metrics.*

6.4 Parameter Selection for Optimal Performance

One of the key advantages of our Jacobi-based approach over previous methods using Gegenbauer polynomials is the additional degree of freedom provided by the two parameters α and β. This flexibility allows for more refined tuning of the method's performance characteristics to match specific problem requirements.

Our analysis reveals that the choice of α and β affects several important aspects of the method's performance:

1. The distribution of collocation points, which influences the resolution of the approximation in different regions of the domain.

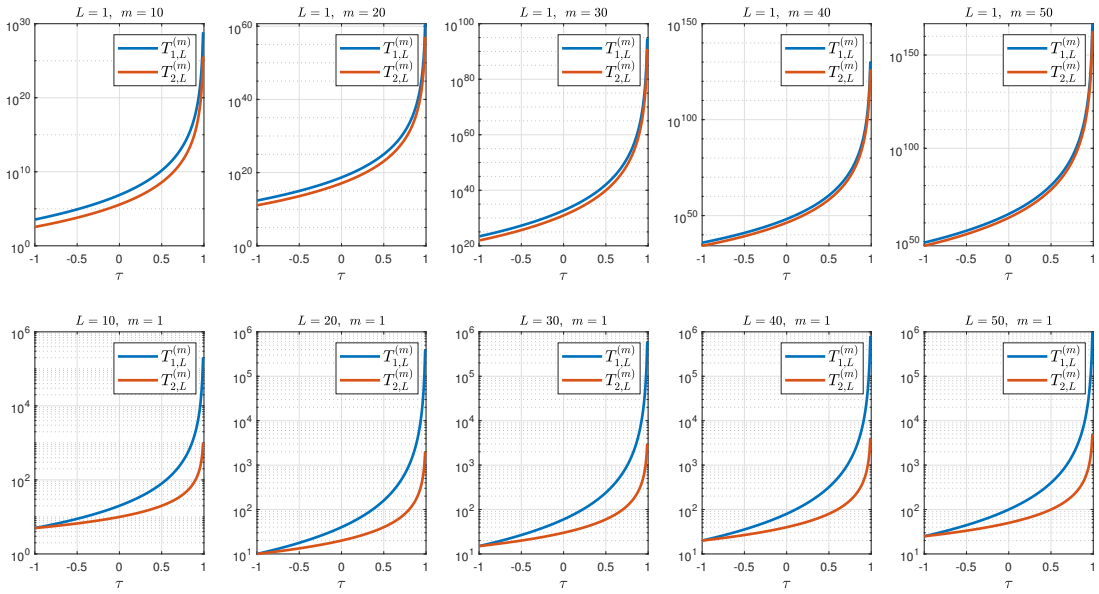

Figure 11: The mth-order derivatives of $T_{1,L}^{(\alpha,\beta)}$ and $T_{2,L}^{(\alpha,\beta)}$ versus τ in log-lin scale for several values of L and m. The superscript of $T_{i,L}^{(\alpha,\beta)}, i = 1,2$ has been omitted in the plots.

2. The conditioning of the integration matrices, which impacts the numerical stability of the discretization.
3. The convergence rate of the method, as indicated by the error bounds established in Theorems 65 and 66.
4. The range of mesh sizes for which the method exhibits stable convergence before the onset of divergence.

Through careful selection of these parameters, it is possible to optimize the method's performance for specific problem characteristics. In particular, our analysis and numerical experiments indicate that certain combinations with $\alpha \neq \beta$ can yield more accurate approximations than those achievable with Gegenbauer polynomials (where $\alpha = \beta = \lambda - 0.5$), Legendre polynomials (where $\alpha = \beta = 0$), or Chebyshev polynomials (where $\alpha = \beta = -0.5$).

This finding represents a significant advancement over previous approaches, as it expands the toolkit available for solving IHOCs with high precision. The optimal parameter values depend on the specific characteristics of the problem being solved, including the smoothness of the solution, the behavior of the system dynamics, and the nature of the cost functional.

In the following section, we present numerical examples that demonstrate the practical performance of our method and validate the theoretical findings established here.

7 Numerical Examples

To validate the theoretical findings presented in the previous section and demonstrate the practical effectiveness of our JGR-IPS, we now examine two numerical examples. These carefully selected test cases illustrate the method's performance across different problem types and parameter configurations, with particular emphasis on comparing the results obtained using Jacobi polynomials with those from Gegenbauer, Legendre, and Chebyshev polynomials. The first example is a practical application concerning spacecraft attitude maneuvers, while the second is a benchmark problem commonly used to evaluate numerical methods for infinite-horizon OC. Numerical experiments were performed using MATLAB R2020a, running on a personal laptop equipped with 64-bit Windows 10 Education, a 1.19 GHz Core i5 CPU, and 8 GB of RAM. The NLPs resulting from the JGR-IPS12 procedures were solved using MATLAB's `fmincon` solver, specifically with the `sqp` algorithm (referred to as fmincon-sqp). Optimal state and control variables for all numerical examples were precisely calculated within MATLAB using 15 digits of internal computational precision.

For Examples 1 and 2, the `fmincon` solver was run with stopping criteria TolFun = TolX = 10^{-15}. All tests used parameter values $L \in \{0.25(0.25)10\}$ and $\{\alpha,\beta\} \subseteq \{-0.9(0.1)2\}$. We used two sets of initial guesses: $\Omega_1 = \{(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) : \tilde{\boldsymbol{x}}_0 = 1_{n_x}, \tilde{\boldsymbol{u}}_0 = 1_{n_u}\}$ and $\Omega_2 = \{(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) : \tilde{\boldsymbol{x}}_0 = 0.5 1_{n_x}, \tilde{\boldsymbol{u}}_0 = 0.5 1_{n_u}\}$. For simplicity, Ω denotes $\Omega_1 \cup \Omega_2$.

7.1 Example 1: Attitude Maneuvers of an Asymmetric Rigid-Body Spacecraft

Our first example concerns the attitude maneuvers of an asymmetric rigid-body spacecraft, a problem of significant practical importance in aerospace engineering. The objective is to determine the OC strategy for reorienting the spacecraft from an initial attitude to a desired final attitude while minimizing control effort. The dynamics of the spacecraft are described by the following equations:

$$\dot{x}_1 = \frac{I_2 - I_3}{I_1} x_2 x_3 + \frac{u_1}{I_1}, \tag{7.1a}$$

$$\dot{x}_2 = \frac{I_3 - I_1}{I_2} x_3 x_1 + \frac{u_2}{I_2}, \tag{7.1b}$$

$$\dot{x}_3 = \frac{I_1 - I_2}{I_3} x_1 x_2 + \frac{u_3}{I_3}, \tag{7.1c}$$

where x_i are the angular velocities, I_i are the principal moments of inertia, and u_i are the control torques. The IHOC is formulated as:

$$\min_{\boldsymbol{x},\boldsymbol{u}} J = \frac{1}{2} \int_0^\infty \sum_{i=1}^{3} \left(x_i(t)^2 + u_i(t)^2 \right) dt, \tag{7.2a}$$

subject to the dynamics (7.1a)-(7.1c) and the initial conditions:

$$x_1(0) = 0.01, \quad x_2(0) = 0.005, \quad x_3(0) = 0.001. \tag{7.2b}$$

This example was previously solved in [50] using an extended modal series method. This technique converts a nonlinear TPBVP, derived from the maximum principle, into a sequence of linear time-invariant TPBVPs. Recursively solving these problems yields the OC law and trajectory as uniformly convergent series. Additionally, this example was also solved in [51] using a Feed Forward Neural Network Scheme.

We applied our JGR-IPS to this problem using the spacecraft's principal moments of inertia: $I_1 = 86.24\,\text{kg} \cdot \text{m}^2$, $I_2 = 85.07\,\text{kg} \cdot \text{m}^2$, and $I_3 = 113.59\,\text{kg} \cdot \text{m}^2$. We tested various combinations of the parameters α, β, L, and n.

Table 2 presents the optimal cost values obtained using different polynomial families and parameter combinations across various mesh sizes n. The JGR-IPS12 method demonstrably outperforms [50]'s approach, as evidenced by our method consistently achieving smaller objective function values for varying n. Specifically, the lowest objective function values were recorded for $\alpha \in [1.6, 1.9]$ and $\beta \in [0.6, 1.8]$ with $n = 10, 15$. This observation aligns with our general "Rule of Thumb" discussed in Section 6.1.

Similarly, Table 3 indicates that the JGR-IPS12 method yields superior *MRE*s. The tabulated data also reveals a noteworthy observation: Jacobi polynomials with $\alpha \in [1.6, 1.9]$ and $\beta \in [0.6, 1.8]$ for $n = 10, 15$ exhibit higher convergence rates compared to other polynomial families. This includes Chebyshev polynomials ($\alpha = \beta = -0.5$), Legendre polynomials ($\alpha = \beta = 0$), and Gegenbauer polynomials ($\alpha = \beta = \lambda - 0.5$). The data collectively illustrates the method's convergence behavior across different parameter combinations and confirms our theoretical prediction of exponential convergence for small to moderate mesh sizes, irrespective of the parameter choice.

The success of JGR-IPS12 in this context can be attributed to several key factors:

1. The system dynamics' integral form facilitates increased accuracy through numerically stable quadratures.
2. The parametric logarithmic mapping $T_{2,L}^{(\alpha,\beta)}$ is preferred over $T_{1,L}^{(\alpha,\beta)}$ due to its reduced sensitivity and slower growth near $\tau = 1$.
3. The scaling parameter L, when optimally selected, contributes to faster convergence rates.

Figure 12 further illustrates the approximate optimal state and control trajectories obtained using JGR-IPS1. The smooth and physically realistic nature of these trajectories provides additional validation for the effectiveness of our approach.

7.2 Example 2: Linear Quadratic Regulator Problem

Our second example is a LQR problem with an infinite horizon, which has an analytical solution that can be used to assess the accuracy of our numerical method. The problem is formulated as:

$$\min_{x_1,x_2,u} J = \int_0^\infty \left(x_1^2(t) + 0.5 x_2^2(t) + 0.25 u^2(t) \right) dt, \tag{7.3a}$$

subject to the dynamics:

$$\dot{x}_1 = x_2, \tag{7.3b}$$

$$\dot{x}_2 = 2x_1 - x_2 + u, \tag{7.3c}$$

and the initial conditions:

$$x_1(0) = -4, \quad x_2(0) = 4. \tag{7.3d}$$

The exact state and control variables to this problem are given by:

$$\boldsymbol{x}^*(t) = \exp(\mathcal{N} t)\boldsymbol{x}(0), \tag{7.4a}$$

$$u^*(t) = -\boldsymbol{H}\boldsymbol{x}^*(t), \tag{7.4b}$$

where

$$\mathcal{N} = \begin{bmatrix} 0 & 1 \\ -2.82842712474619 & -3.557647291327851 \end{bmatrix}, \tag{7.4c}$$

$$\boldsymbol{H} = [4.828427124746193, 2.557647291327851]; \tag{7.4d}$$

Table 2: *fval* of [50]'s method and JGR-IPS12 obtained using $n = 10, 15$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega$.

Example 1				
[50]'s method	JGR-IPS1		JGR-IPS2	
	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_1$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_2$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_1$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_2$
$fval/(M)$	$fval/(n)/(\alpha)/(L)/(\beta)$	$fval/(n)/(\alpha)/(L)/(\beta)$	$fval/(n)/(\alpha)/(L)/(\beta)$	$fval/(n)/(\alpha)/(L)/(\beta)$
0.005432195475/(3)	5.35003584754812$e-$05/(10)/(1.9)/(2.75)/(1.8)	5.35003584754812$e-$05/(10)/(1.9)/(2.75)/(1.8)	2.93725040738347$e-$05/(10)/(1.9)/(7.5)/(1.8)	2.93725040736794$e-$05/(10)/(1.9)/(7.5)/(1.8)
	4.18050600008906$e-$06/(15)/(1.7)/(0.25)/(0.7)	4.18050600008899$e-$06/(15)/(1.7)/(0.25)/(0.7)	5.45632819380131$e-$07/(15)/(1.6)/(0.25)/(0.6)	7.31733271230786$e-$07/(15)/(1.7)/(0.5)/(0.7)

Table 3: Table 3 presents the *MRE* for [50]'s method and JGR-IPS12, obtained using $n = 10, 15$ and initial guesses from Ω.

Example 1				
[50]'s method	JGR-IPS1		JGR-IPS2	
	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_1$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_2$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_1$	$(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_2$
$MRE/(M)$	$MRE/(n)/(\alpha)/(L)/(\beta)$	$MRE/(n)/(\alpha)/(L)/(\beta)$	$MRE/(n)/(\alpha)/(L)/(\beta)$	$MRE/(n)/(\alpha)/(L)/(\beta)$
6.4000$e-$13/(3)	3.4694$e-$18/(10)/(1.9)/(2.75)/(1.8)	2.6021$e-$18/(10)/(1.9)/(2.75)/(1.8)	3.9031$e-$18/(10)/(1.9)/(7.5)/(1.8)	7.3726$e-$18/(10)/(1.9)/(7.5)/(1.8)
	4.5103$e-$17/(15)/(1.7)/(0.25)/(0.7)	5.1608$e-$17/(15)/(1.7)/(0.25)/(0.7)	4.8139$e-$17/(15)/(1.6)/(0.25)/(0.6)	5.5403$e-$17/(15)/(1.7)/(0.5)/(0.7)

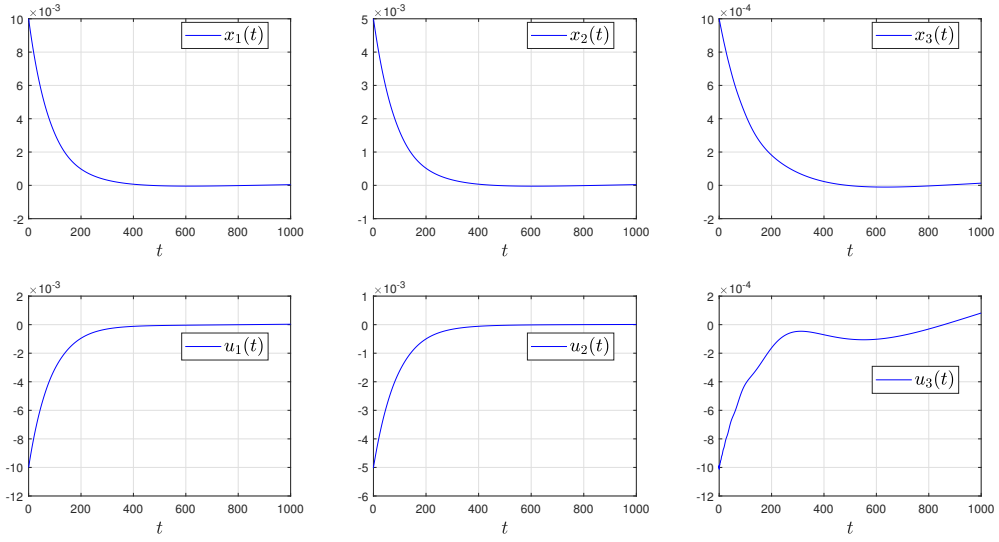

Figure 12: Numerical simulations for Example 1 performed using the JGR-IPS1 technique. All state and control variables were defined on the interval $[0, 1000]$. For the presented method, we used $n = 25$, $\alpha = \beta = 0$, and $L = 9.75$. Figures were generated by evaluating the solution at 10001 equally spaced points between 0 and 1000, starting with the initial guess $\tilde{\boldsymbol{x}}_n = \tilde{\boldsymbol{u}}_n = \mathbf{1}_n$.

cf. [53,52,39]. Using MATLAB's Symbolic Math Toolbox, the optimal cost functional value for this linear quadratic regulator problem was determined to be $J^* = 19.85335656362790$, rounded to sixteen significant digits.

We applied our JGR-IPS method to this problem, exploring various combinations of the parameters α and β. The results were then compared with the analytical solution. Only the parametric mapping $T_{2,L}^{(\alpha,\beta)}$ was thoroughly tested in this example with different values of the scaling parameter L, due to its slower divergence rate, as noted in Section 6.3.

Table 4 presents the AE_J for various polynomial families, parameter combinations, and mesh sizes. Our results demonstrate that the proposed method achieves high accuracy even with relatively small mesh sizes. The table also provides a direct comparison between our current method, JGR-IPS2, our previous work [28], and the transformed LGR approach of [41]. It is evident that the proposed method generally produces lower AE_J values than those in [41], with the exception of $n = 100$. Aligning with our divergence analysis, we observed an unusual drop in precision as the mesh size increases from moderate to large-scale. In particular, the results confirm our theoretical prediction of exponential convergence for moderate mesh sizes followed by divergence for very

large mesh sizes. Interestingly, complete machine accuracy for the exact J^* was recorded as early as $n = 20$ for both, the JGR-IPS2 and our previous method [28], highlighting incredibly accurate numerical techniques with exponential convergence rates for coarse meshes. It's important to note that the Legendre case ($\alpha = \beta = 0$) achieved the highest accuracy for the JGR-IPS2 method across all scenarios except for $n = 100$. In this specific case, collocation at values of α and β where $\alpha \neq \beta$ provided increased precision. Furthermore, the presented method demonstrates superiority over our previous method [28] only for $n = 30, 60$, and 90; both methods exhibit comparable accuracy for $n = 20$, while our former method [28] performs better otherwise. One may attribute this slight gain in accuracy by our previous method [28] to the exceedingly accurate SR interpolation algorithm developed there, which improved the direct IPS method for IHOCs using Gegenbauer polynomials and GGR points; cf. Remark 11.

Table 5 presents the CPU times (in seconds) for computing the AE_J, demonstrating that the JGR-IPS2 method consistently achieves lower computational times compared to [41], highlighting its superior efficiency. Table 6 displays further the $MAE_{x,u}$. A comparison with the results from [55] is provided in Table 7, showing that our presented method offers superior accuracy in most cases.

Figure 13 shows the approximate optimal state and control trajectories obtained using our JGR-IPS2 with specific parameter values, compared with the analytical solution. The excellent agreement between the numerical and analytical solutions confirms the accuracy of our approach. Figures 14, 15, and 16 present the $MAE_{x,u}$ for various parameter combinations, illustrating the convergence behavior of the JGR-IPS2 method. In Figure 14, for fixed n and L, the global minimum of $MAE_{x,u}$ typically occurs near $\alpha = 0$, suggesting that Legendre polynomials ($\alpha = \beta = 0$) are often optimal among Jacobi polynomials. Similarly, Figure 15 shows that the minimum $MAE_{x,u}$ generally occurs near $\beta = 0$ for fixed n and L. Stability analysis indicates that Jacobi polynomials remain stable for α near -1, but instability emerges as α approaches 2 (Figure 14). Conversely, instability is common for β near -1 or 2, while stability is generally observed in the intermediate range (Figure 15). Figure 16 does not yield a consistent pattern for a general rule of thumb across the tested (α, β, L) combinations.

7.3 Discussion of Results

The numerical examples presented in the previous subsection validate the theoretical findings established in Section 6 and demonstrate several key advantages of our JGR-IPS:

1. The method achieves exponential convergence for moderate mesh sizes, allowing for high-accuracy approximations with relatively few collocation points.

Example 2			
n	[41]'s Method AE_J	[28]'s Method $AE_J/\lambda/L$	JGR-IPS2 $AE_J/\alpha/\beta/L$
10	3.92e-05	1.0658e-14/0.5/4.25	2.4869e-14/0/0/2.5
20	1.76e-06	0/0.5/5.75	0/0/0/3.25
30	2.73e-07	1.0658e-14/0.5/2.5	3.5527e-15/0/0/6
40	7.24e-08	1.0303e-13/0.5/6	1.1724e-13/0/0/6.75
50	2.63e-08	5.1514e-13/0.5/5	6.6080e-13/0/0/5.75
60	1.19e-08	8.3844e-13/0.5/5.5	8.3489e-13/0/0/6
70	6.45e-09	1.2967e-12/0.5/10	2.2631e-12/0/0/5.5
80	4.06e-09	1.7977e-12/0.5/10	5.7376e-12/0/0/8.5
90	2.90e-09	1.4021e-9/0/3.5	1.5227e-10/0/0/1.5
100	2.29e-09	5.3783e-08/0.8/4	3.6658e-07/-0.3/0.7/0.75

Table 4: The AE_J of [41]'s method, our previous method in [28] using the logarithmic mapping $T_{2,L}^{(\lambda)}$, and the lowest AE_J gained by JGR-IPS2 using $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_1$. All approximations were rounded to 5 significant digits.

Example 2		
n	[41]'s Method CPU time	JGR-IPS2 CPU time
10	0.15	0.15
20	0.35	0.18
30	0.57	0.17
40	1.01	0.16
50	1.37	0.17
60	1.96	0.16
70	2.51	0.19
80	4.65	0.36
90	4.26	0.34
100	5.41	0.58

Table 5: The CPU times for running the code to compute AE_J of [41]'s method and the lowest AE_J obtained by the JGR-IPS2 using $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_1$.

Example 2				
n	α	β	L	$MAE_{x,u}$
10	0	0	2.5	8.5317e-07
20	0	0	3.25	1.6512e-06
30	0	0	6	1.2799e-06
40	0	0	6.75	4.0971e-06
50	0	0	5.75	3.5581e-06
60	0	0	6	2.1720e-06
70	0	0	5.5	1.4486e-05
80	0	0	8.5	7.4694e-06
90	0	0	1.5	1.6675e-04
100	-0.3	0.7	0.75	1.1630e-03

Table 6: The $MAE_{x,u}$ of JGR-IPS2 using $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_1$ computed at the collocation points. All approximations were rounded to 5 significant digits.

2. Certain combinations of Jacobi parameters α and β with $\alpha \neq \beta$ yield more accurate approximations than those provided by Gegenbauer, Legendre, or Chebyshev polynomials, confirming the benefit of the additional degree of freedom in parameter selection.

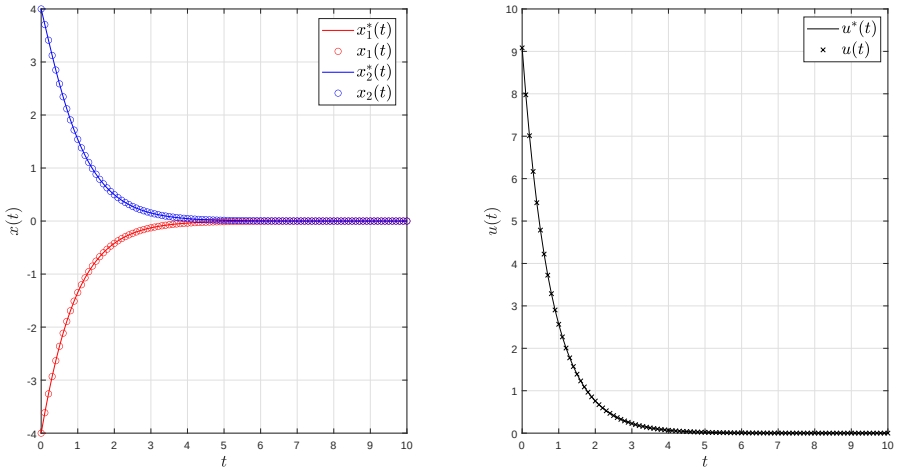

Figure 13: The plots of the exact optimal states and control of Example 2 and their collocated approximations acquired by means of JGR-IPS2 on the domain $[0,10]$ with $n=30, \alpha=\beta=0, L=4.25$, and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_1$. All plots were produced with 101 equally spaced points between 0 and 10.

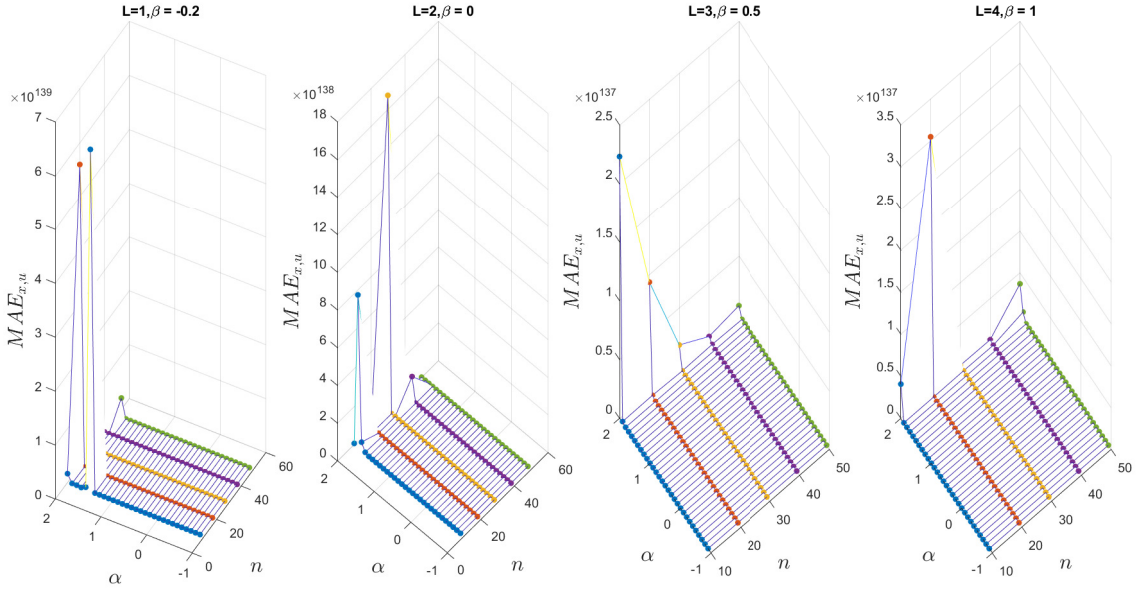

Figure 14: The $MAE_{x,u}$ of the JGR-IPS2 at the collocation points using $n = 10(10)50$, $\alpha = -0.9(0.1)2$, $(L,\beta) = (1,-0.2),(2,0),(3,0.5),(4,1)$, and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{u}}_0) \in \Omega_1$.

3. The method successfully handles both practical engineering problems (Example 1) and benchmark problems with analytical solutions (Example 2), demonstrating its versatility and robustness.
4. The observed divergence for very large mesh sizes confirms our theoretical prediction and highlights the importance of selecting appropriate mesh sizes for optimal performance.
5. The simplified computational approach without the switching technique for barycentric weights performs adequately across various parameter combinations, validating our decision to eliminate this complexity from the implementation.

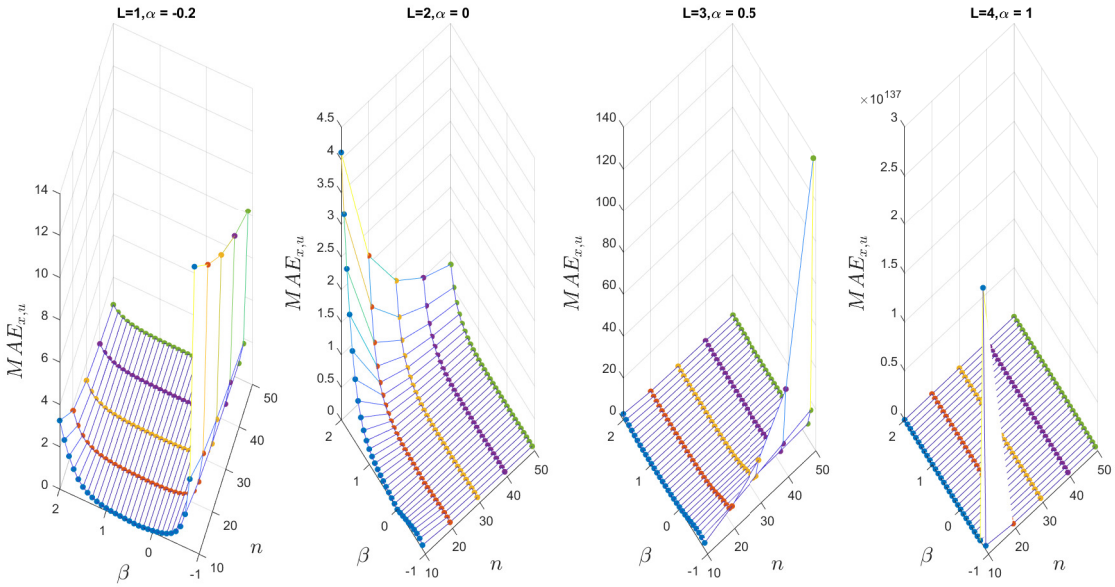

Figure 15: The $MAE_{x,u}$ of the JGR-IPS2 at the collocation points using $n = 10(10)50$, $\beta = -0.9(0.1)2$, $(L,\alpha) = (1,-0.2),(2,0),(3,0.5),(4,1)$, and $(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{u}}_0) \in \Omega_1$.

Example 2		
t	[55]'s Method $n=10$	JGR-IPS2 $n=10,\alpha=0,\beta=0,L=2.5$
0	0	0
1	6.86391e-04	1.86805e-05
2	7.46026e-04	8.79501e-05
3	2.54228e-06	4.69135e-05
4	9.00173e-04	3.32519e-05
5	2.08183e-04	7.15694e-06
6	8.95379e-04	1.17352e-05
7	1.89575e-04	3.20300e-07
8	7.03556e-04	7.69009e-06
9	7.33797e-04	1.00914e-05
10	3.75724e-05	7.93869e-06
15	6.67898e-04	6.6117e-05
20	6.60049e-04	1.66080e-04
fval	19.853452	19.8533565636279

Table 7: Absolute error of the first state component for [55]'s method and JGR-IPS2 with $(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{u}}_0) \in \Omega_1$, rounded to six significant digits.

These results collectively establish the effectiveness of our JGR-IPS for solving IHOCs and demonstrate its advantages over previous approaches based on more specialized polynomial families.

8 Conclusion

In this paper, we have presented a novel direct IPS method for solving IHOCs using Jacobi polynomials. Our approach extends previous work that utilized Gegenbauer polynomials by employing the more general and flexible Jacobi polynomial family, which offers an additional degree of freedom through its two parameters α and β. Through rigorous theoretical analysis and numerical experiments, we have demonstrated several key advantages of this approach.

First, our method transforms IHOCs into finite-horizon problems in integral form using parametric mappings, which are then discretized using rational collocations based on Jacobi polynomials and JGR nodes. This approach avoids the precision degradation typically associated with numerical differentiation procedures and allows for efficient solution of the resulting NLP.

Second, we have established comprehensive error bounds and convergence properties for our method, showing that it achieves exponential convergence for moderate mesh sizes. Our analysis also reveals the limitations of the method for very large mesh sizes, where divergence becomes inevitable due to the ill-conditioning introduced by the parametric mappings. This finding extends similar observations made for Gegenbauer polynomials to the more general Jacobi polynomial family.

Third, we have demonstrated that certain combinations of Jacobi parameters with $\alpha \neq \beta$ yield more accurate approximations than those provided by Gegenbauer, Legendre, or Chebyshev polynomials. This result highlights the benefit of the additional degree of freedom in parameter selection offered by the Jacobi polynomial approach. The optimal parameter values

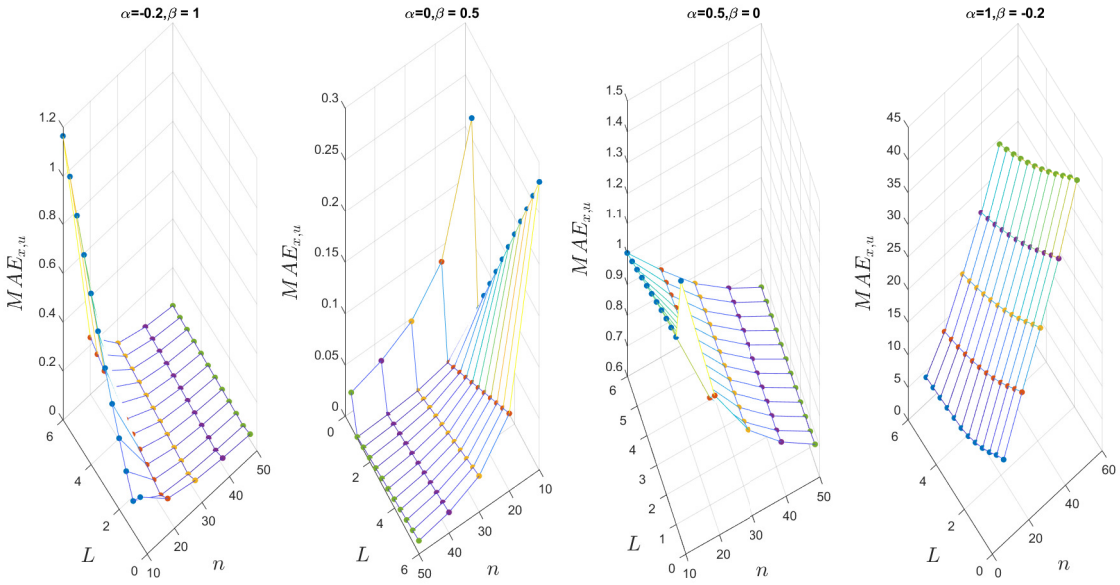

Figure 16: The $MAE_{x,u}$ of the JGR-IPS2 at the collocation points using $n = 10(10)50$, $L = 0.5(0.5)6$, $(\alpha,\beta) = (-0.2,1),(0,0.5),(0.5,0),(1,-0.2)$, and $(\tilde{\boldsymbol{x}}_0,\tilde{\boldsymbol{u}}_0) \in \Omega_1$.

depend on the specific characteristics of the problem being solved, providing a flexible framework that can be adapted to various applications.

Fourth, we have implemented a simplified computational approach that eliminates the need for the switching technique used in previous work for computing barycentric weights. Our analysis shows that for many parameter combinations within the Jacobi family, the standard formula performs adequately without requiring the more complex switching approach, reducing the computational complexity of the method.

Finally, we have validated our theoretical findings through numerical examples, including a practical application to spacecraft attitude maneuvers. These examples demonstrate the effectiveness of our method in solving complex IHOCs and confirm its advantages over previous approaches.

While the spacecraft attitude maneuver example demonstrates the efficacy of the proposed Jacobi PS method, its applicability extends to a wide range of IHOCs across diverse domains. For instance, in finance, the method can optimize long-term portfolio management strategies under stochastic constraints. In robotics, it can facilitate motion planning for autonomous systems operating over extended time horizons, ensuring energy-efficient trajectories. Similarly, in energy systems, the approach can address optimal control of renewable energy grids, balancing supply and demand indefinitely. These potential applications underscore the versatility of the proposed method, positioning it as a powerful tool for tackling complex, long-term optimization challenges in various fields. Future research directions may also include the development of adaptive mesh refinement strategies that can utilize the flexibility of Jacobi polynomials, the exploration of alternative parametric mappings that may offer improved conditioning for large mesh sizes, and the application of our method to a wider range of practical problems in various fields. Additionally, the extension of this approach to handle state and control constraints more efficiently represents an important area for further investigation.

In addition to the specific contributions of this study, the proposed Jacobi PS framework has broader implications in both aerospace and finance. In aerospace engineering, Jacobi PS methods have been successfully applied to optimal trajectory design and flight-control problems, demonstrating their effectiveness in handling complex dynamic constraints [29]. In the field of finance, Jacobi-family spectral and PS approaches, including barycentric and Gauss-Jacobi schemes, have been employed for option pricing under fractional models [57]. These applications highlight the versatility of the proposed method and underscore its potential impact beyond the immediate scope of this work.

In conclusion, our JGR-IPS provides a powerful and flexible tool for solving IHOCs, offering improved accuracy and adaptability compared to previous approaches based on more specialized polynomial families. The theoretical foundations and practical implementation presented in this paper establish a solid

framework for the continued development and application of this method in various scientific and engineering domains.

A Useful Lemma

Lemma A1. *For fixed $\alpha > -1$ and $\beta > -1$, the term $(n+1)!K_n^{(\alpha,\beta+1)} = O(n2^n(n/e)^n)$, as $n \to \infty$.*

Proof. Notice that:

$$(n+1)!K_n^{(\alpha,\beta+1)} = \frac{(n+1)\Gamma(2n+\alpha+\beta+2)}{2^n\Gamma(n+\alpha+\beta+2)}. \quad \text{(A.1)}$$

Also, Stirling's formula to the factorial function can be written as follows:

$$\sqrt{2\pi}x^{x+1/2}e^{-x} < \Gamma(x+1) < \sqrt{2\pi}xx^xe^{-x}e^{1/(12x)} \quad \forall x \in [1,\infty). \quad \text{(A.2)}$$

Applying Eq. (A.2) to the ratio of Gamma functions in Eq. (A.1) gives:

$$\frac{\Gamma(2n+\alpha+\beta+2)}{\Gamma(n+\alpha+\beta+2)} < \left(1+\frac{n}{n+\alpha+\beta+1}\right)^{n+\alpha+\beta+3/2}$$
$$(2n+\alpha+\beta+1)^n e^{-n}e^{1/(12(2n+\alpha+\beta+1))} = O\left(4^n\left(\frac{n}{e}\right)^n\right), \quad \text{as } n \to \infty. \quad \text{(A.3)}$$

By combining this result with Eq. (A.1), we can readily show that:

$$(n+1)!K_n^{(\alpha,\beta+1)} = \frac{n+1}{2^n}O\left(4^n\left(\frac{n}{e}\right)^n\right) = O(n2^n(n/e)^n). \quad \text{(A.4)}$$

Acknowledgments

We express our sincere gratitude to the editor and reviewers for their thorough evaluation and valuable feedback that has significantly improved this manuscript. This work was carried out without any funding.

Conflict of interest

The authors declare no conflicts of interest.

References

[1] Orszag, Steven A, Journal of Fluid Mechanics, 50 (1971) 689-703.
[2] Patterson Jr, GS and Orszag, Steven A, The Physics of Fluids, 14 (1971) 2538-2541.
[3] Gong, Qi and Kang, Wei and Bedrossian, Nazareth S and Fahroo, Fariba and Sekhavat, Pooya and Bollino, Kevin, 2007 46th IEEE Conference on Decision and Control, (2007) 4128-4142.
[4] Gan, Wenyang and Su, Lixia and Chu, Zhenzhong, Sensors, 23 (2023) 2350.
[5] Pryde, Martin and Nehaoua, Lamri and Alfayad, Samer and Hadj-Abdelkader, Hicham and Arioui, Hichem, The 63rd IEEE Conference on Decision and Control (CDC 2024), (2024).
[6] Elgindy, Kareem T., Unmanned Systems, 13 (2025) 1203-1221.
[7] Wang, Lixiang and Ye, Dong and Kong, Xianren and Xiao, Yan, Advances in Space Research (2025)
[8] Fornberg, Bengt, Cambridge university press; (1998).
[9] Hesthaven, Jan S and Gottlieb, Sigal and Gottlieb, David, Cambridge University Press; (2007).
[10] Canuto, C and Houssanini, M Y and Quarteroni, A and Zang, T A, Springer Berlin, Heidelberg; (1988).
[11] Canuto, Claudio and Hussaini, M Yousuff and Quarteroni, Alfio and Zang, Thomas A, Springer Science & Business Media; (2007).
[12] Clenshaw, Charles W and Curtis, Alan R, Numerische Mathematik, 2 (1960) 197-205.
[13] El-Gendi, SE, The Computer Journal, 12 (1969) 282-287.
[14] Lee, June-Yub and Greengard, Leslie, SIAM Journal on Scientific Computing, 18 (1997) 403-429.
[15] Greengard, Leslie, SIAM Journal on Numerical Analysis, 28 (1991) 1071-1080.
[16] Elgindy, Kareem T and Smith-Miles, Kate A, Journal of Computational and Applied Mathematics, 237 (2013) 307-325.
[17] Elgindy, Kareem T, Numerical Methods for Partial Differential Equations, 32 (2016) 307-349.
[18] Elgindy, Kareem T and Dahy, Sayed A, Mathematical Methods in the Applied Sciences, 41 (2018) 6226-6251.
[19] Elgindy, Kareem T and Refat, Hareth M, Applied Numerical Mathematics, 128 (2018) 98-124.
[20] Ling, Chen and Caputo, Michael R, Dynamic Games and Applications, 2 (2012) 313-334.
[21] Barucci, Emilio and Gozzi, Fausto, Journal of economics, 74 (2001) 1-38.
[22] Ross, I Michael and Karpenko, Mark, Annual Reviews in Control, 36 (2012) 182-197.
[23] Gao, Xiaoyang and Li, Tieshan and Shan, Qihe and Xiao, Yang and Yuan, Liang'en and Liu, Yifan, Journal of Ambient Intelligence and Humanized Computing, (2019) 1-13.
[24] Wang, Ding and Ha, Mingming and Zhao, Mingming, Artificial Intelligence Review, (2022) 1-22.
[25] ParandehGheibi, Ali and Roozbehani, Mardavij and Dahleh, Munther A and Ozdaglar, Asuman, 6 (2015) 129-152.
[26] Janová, Jitka and Hampel, David, Central European Journal of Operations Research, 24 (2016) 297-307.
[27] Pang, Bo and Cui, Leilei and Jiang, Zhong-Ping, Biological Cybernetics, (2022) 1-19.
[28] Elgindy, Kareem T and Refat, Hareth M, Aims Mathematics, 8 (2023) 3561-3605.
[29] Williams, Paul, Journal of Guidance, Control, and Dynamics, 27 (2004) 293-297.
[30] Imani, Ahmad and Aminataei, Azim and Imani, Ali and others, International Journal of Mathematics and Mathematical Sciences, 2011(2011).

[31] Kazem, Saeed, Applied Mathematical Modelling, 37 (2013) 1126-1136.
[32] Doha, EH and Bhrawy, AH, Applied Numerical Mathematics, 58 (2008) 1224-1244.
[33] Mirshojaei, SS and Shivanian, E and Ghayebi, B and Ghoncheh, SJ, Journal of Nonlinear Mathematical Physics, 32 (2025) 1-22.
[34] Nemati, Somayeh and Lima, Pedro M and Torres, Delfim FM, Numerical Algorithms, 86 (2021) 675-691.
[35] Doha, EH and Bhrawy, AH and Abdelkawy, MA, Journal of Computational and Nonlinear Dynamics, 10 (2015) 021016.
[36] Zhang, Tiangong and Li, Huiyuan and Wang, Zhongqing, Applied Numerical Mathematics, 176 (2022) 159-181.
[37] Garg, Divya and Hager, William and Rao, Anil, AIAA Guidance, Navigation, and Control Conference, (2002) 7890.
[38] Garg, Divya and Hager, William W and Rao, Anil V, Automatica, 47 (2011) 829-837.
[39] Garg, Divya and Patterson, Michael A and Francolin, Camila and Darby, Christopher L and Huntington, Geoffrey T and Hager, William W and Rao, Anil V, Computational Optimization and Applications, 49 (2011) 335-358.
[40] Tang, Xiaojun and Chen, Jie, IEEE/CAA Journal of Automatica Sinica, 3 (2016) 174-183.
[41] Shahini, Mehdi and Mehrpouya, MA, IMA Journal of Mathematical Control and Information, 35 (2018) 341-356.
[42] Tang, Xiaojun and Shi, Yang and Wang, Li-Lian, Automatica, 78 (2017) 333-340.
[43] Ali, Mushtaq Salh and Shamsi, Mostafa and Khosravian-Arab, Hassan and Torres, Delfim FM and Bozorgnia, Farid, Journal of Vibration and Control, 25 (2019) 1080-1095.
[44] Tang, Xiaojun and Xu, Heyong, Applied Mathematical Modelling, 68 (2019) 137-151.
[45] Yang, Yin and Zhang, Jiaqi and Liu, Huan and O. Vasilev, Aleksandr, Mathematical Methods in the Applied Sciences, 44 (2021) 2806-2824.
[46] Shen, Jie and Tang, Tao and Wang, Li-Lian, Springer Science & Business Media: (2011).
[47] Elgindy, Kareem T Applied Numerical Mathematics, 113 (2017) 1-25.
[48] Wang, Haiyong and Huybrechs, Daan and Vandewalle, Stefan, Mathematics of Computation, 83 (2014) 2893-2914.
[49] Elgindy, Kareem T and Smith-Miles, Kate A, Journal of Computational and Applied Mathematics, 242 (2013) 82-106.
[50] Jajarmi, Amin and Pariz, Naser and Effati, Sohrab and Kamyad, Ali Vahidian, Journal of Zhejiang University SCIENCE C, 12 (2011) 667-677.
[51] Mortezaee, Marziyeh and Nazemi, Alireza, Neural Processing Letters, 51 (2020) 449-471.
[52] Fahroo, Fariba and Ross, I Michael, Journal of Guidance, Control, and Dynamics, 31 (2008) 927-936.
[53] Kirk, Donald E, Englewood Cliffs, N.J.: Prentice-Hall: (1970).
[54] Nikooeinejad, Z and Delavarkhalafi, Ali and Heydari, M, International Journal of Control, 91 (2018) 725-739.
[55] Mamehrashi, Kamal and Nemati, Ali, International Journal of Computer Mathematics, 97 (2020) 1529-1544.
[56] Szegö, Gabor, American Mathematical Soc.: (1939).
[57] Nteumagné, BF and Pindza, Edson and Mare, Eben, Scientific Research Publishing, 4 (2014) 35-46.