A Unified Framework for the Solution of Optimal Control Problems via Pseudospectral Methods

Divya Garg[*]
Michael A. Patterson[†]
Christopher L. Darby[‡]
Danielle Pisano[§]
Department of Mechanical and Aerospace Engineering
University of Florida
Gainesville, FL 32611

David Benson[¶]
Charles Stark Draper Laboratory
Cambridge, MA 02139

Geoffrey T. Huntington[‖]
Blue Origin, LLC.
Kent, WA 98032

Anil V. Rao[**]
Department of Mechanical and Aerospace Engineering
University of Florida
Gainesville, FL 32611

Abstract

An improved unified framework is presented for the solution of optimal control problems via pseudospectral methods. This new framework utilizes a strategy that distinguishes between interpolation points (for the state) and collocation points (for the dynamics). Two methods are presented in this new framework, entitled the Gauss pseudospectral method (GPM) and the Radau pseudospectral method (RPM), which implement carefully chosen, unique combinations of interpolation and collocation points. These methods are demonstrated on two example problems that have been used in a previously developed framework. Using these historical examples, this new framework shows significantly more accurate results (in both the control and costate solutions) over the previous framework, which relied on a strategy that defined the collocation points as the interpolation points. Finally, it is found that this new framework produces accurate and smooth costates using the simple algebraic costate mapping equations that arise from the GPM and RPM.

[*] M.S. Student, Dept. of Mechanical and Aerospace Engineering. E-mail: divyagarg2002@ufl.edu

[†] Ph.D. Student, Dept. of Mechanical and Aerospace Engineering. E-mail: mpatterson@ufl.edu

[‡] Ph.D. Student, Dept. of Mechanical and Aerospace Engineering. E-mail: cdarby@ufl.edu

[§] M.S.E. Student, Dept. of Mechanical and Aerospace Engineering. E-mail: dpisano@ufl.edu

[¶] Senior Member of the Technical Staff, GN&C Systems Division. E-mail: dbenson@draper.com

[‖] GN& C Engineer. E-mail: ghuntington@blue.aero

[**] Assistant Professor, Dept. of Mechanical and Aerospace Engineering. E-mail: anilvrao@ufl.edu. Corresponding Author.

1 Introduction

Over the last decade, pseudospectral methods have risen to prominence in the numerical solution of optimal control problems.[1–17] Pseudospectral methods are a class of *direct collocation* where the optimal control problem is transcribed to a nonlinear programming problem (NLP) by parameterizing the state and control using global polynomials and collocating the differential-algebraic equations using nodes obtained from a Gaussian quadrature. It is noted that some researchers prefer the terminology orthogonal collocation,[18–20] but the terms "pseudospectral" and "orthogonal collocation" are the same.

The three most commonly used set of collocation points are *Legendre-Gauss* (LG), *Legendre-Gauss-Radau* (LGR), and *Legendre-Gauss-Lobatto* (LGL) points. These three sets of points are obtained from the roots of a Legendre polynomial and/or linear combinations of a Legendre polynomial and its derivatives. All three sets of points are defined on the domain $[-1, 1]$, but differ significantly in that the LG points include *neither* of the endpoints, the LGR points include *one* of the endpoints, and the LGL points include *both* of the endpoints. In addition, the LGR points are asymmetric relative to the origin and are not unique in that they can be defined using either the initial point or the terminal point. Because of the chronology of development in the aerospace industry and its relative straightforwardness, the most popular pseudospectral method has been the *Legendre pseudospectral method* (LPM) where collocation is performed at the LGL points.[1, 3–6, 10–13] In more recent years, however, two other pseudospectral methods have increased in popularity. These other two methods are the *Gauss pseudospectral method* (GPM),[14–16] where collocation is performed at LG points, and the *Radau pseudospectral method* (RPM),[17] where collocation is performed at LGR points.

In recent years, researchers and engineers have become increasingly confused about the validity of formulating a pseudospectral method using either LG and LGR collocation. This confusion arises from the following apparent contradictions in the following literature: (i) the research documented in Refs. 14–16 describing the aforementioned *Gauss pseudospectral method* (GPM) where collocation at LG points is employed. In addition, the research of Refs. 14–16 has been followed by research that solves complex optimal control problems using the GPM,[21–24] thus showing the validity of the GPM; (ii) the research documented in Ref. 17 demonstrating the validity of collocation at LGR points [i.e., the *Radau pseudospectral method* (RPM)]; (iii) research documented in Refs. 12 and 13, contradicting the research of (i) and (ii), making it appear wrongly that the only valid pseudospectral approach is via collocation at Gauss-Lobatto points. In particular, Ref. 12 extends the LPM to LG and LGR collocation points and states that, based on this approach, collocation at LG and LGR points is invalid while Ref. 13 states unequivocally that "Among other things, this theory indicates [20] that the proper PS method for solving generic finite-horizon optimal control problems must be based on Gauss-Lobatto points."[13] Finally, Ref. 12 states unequivocally that "These facts are illustrated by simple examples and counter examples which reveal when and why

PS methods based on LGR and LG points fail."[12]

The need for this paper arises directly from the contradictions given in (i)–(iii) above. In particular, in this paper we will show that the GPM and the RPM are indeed valid collocation methods, thereby demonstrating that proper pseudospectral methods can be formulated using LG and LGR collocation points. Furthermore, we will show that our framework using the GPM and RPM is different from the LPM-based framework given in Refs. 12 and 13, thereby demonstrating that LGL, LG, and LGR collocation cannot be unified into a single framework. Instead, our framework is based on the requirement that proper LG and LGR collocation methods require that the *interpolation points* (i.e., the points used in the Lagrange polynomial basis to approximate the state) must *differ* from the *collocation points* (i.e., the points used to collocate the dynamics). By distinguishing between interpolation and collocation in this manner, we put forth the GPM and RPM formulations which are a proper way to utilize LG and LGR collocation points. Our framework then unifies the GPM[14–16] and the RPM[17,25] while showing that the LPM and its variations fall outside of our framework. The results of this paper demonstrate that the GPM and the RPM are highly accurate methods both for the computation of primal (i.e., state and control) solutions and dual (i.e., costate) solutions.

2 Continuous Bolza Problem

Without loss of generality, consider the following optimal control problem in Bolza form. Determine the state, $\mathbf{x}(\tau) \in \mathbb{R}^n$, control, $\mathbf{u}(\tau) \in \mathbb{R}^m$, initial time, t_0, and final time, t_f, that minimize the cost functional

$$J = \Phi(\mathbf{x}(-1), t_0, \mathbf{x}(1), t_f) + \frac{t_f - t_0}{2} \int_{-1}^{1} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) d\tau \tag{1}$$

subject to the constraints

$$\frac{d\mathbf{x}}{d\tau} = \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \tag{2}$$

$$\boldsymbol{\phi}(\mathbf{x}(-1), t_0, \mathbf{x}(1), t_f) = \mathbf{0} \tag{3}$$

$$\mathbf{C}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \leq \mathbf{0} \tag{4}$$

The optimal control problem of Eqs. (1)–(4) will be referred to as the *continuous Bolza problem*. It is noted that the optimal control problem of Eqs. (1)–(4) can be transformed from the time interval $\tau \in [-1, 1]$ to the time interval $t \in [t_0, t_f]$ via the affine transformation

$$t = \frac{t_f - t_0}{2} \tau + \frac{t_f + t_0}{2} \tag{5}$$

3 LG, LGR, and LGL Collocation Points

The LG, LGR, and LGL collocation points lie on the open interval $\tau \in (-1, 1)$, the half open interval $\tau \in [-1, 1)$ or $\tau \in (-1, 1]$, and the closed interval $\tau \in [-1, 1]$, respectively. A depiction of these three sets of collocation points is shown in Fig. 1 where it is seen that the LG points contain neither -1 or 1, the LGR points contain only *one* of the points -1 or 1 (in this case, the point -1), and the LGL point contain *both* -1 and 1. Denoting K as the number of collocation points and $P_K(\tau)$ as the k^{th}-degree Legendre polynomial, the LG points are the roots of $P_K(\tau)$, the LGR points are the roots of $(P_{K-1}(\tau) + P_K(\tau))$, and the LGL points are the roots of $\dot{P}_{K-1}(\tau)$ together with the points -1 and 1. The polynomials whose roots are the respective points are summarized as follows:

LG:	Roots obtained from $P_K(\tau)$
LGR:	Roots obtained from $(P_{K-1}(\tau) + P_K(\tau))$
LGL:	Roots obtained from $\dot{P}_{k-1}(\tau)$ together with the points -1 and 1

It is seen from Fig. 1 that the LG and LGL points are symmetric about the origin whereas the LGR points are asymmetric. In addition, the LGR points are not unique in that two sets of points exist (one including the point -1 and the other including the point 1). The LGR points that include the terminal endpoint are often called the *flipped* LGR points. In this paper, however, we use the standard set of LGR points as defined above and consistent with the usage given in Ref. 12.

4 Collocation Points, Interpolation Points, and Discretization Points

A key aspect of the framework that we develop in this paper relies on the distinction between *collocation points* and *interpolation points*.[*] First, we define the *collocation points* to be those points at which the dynamic constraints are enforced. In other words, the collocation points for a given method are simply the LG, LGR, or LGL points (depending upon the method chosen). Next, we define the *interpolation points* as those points that define the Lagrange polynomial basis used to approximate the state. Lastly, we define the *discretization points* to be those points at which the state is solved for in a given method. It is important to understand that, when using a particular set of collocation points, the state is approximated using the interpolation points, the dynamics are collocated at only the collocation points, and the state is solved for in the NLP at the discretization points.

In order to maintain clarity and consistency, in this paper the parameter N will be used to denote the collocation points plus *both* endpoints (i.e., N is equal to the number of collocation points *plus* any missing endpoints). Finally, we note that the previously developed framework given in Refs. 11, 12 and 13 does *not* distinguish between interpolation points and collocation points, but forces these two sets of points to be identical.

[*] A theoretical rationale for distinguishing between collocation and interpolation points can be found in Refs. 16, 25.

5 Pseudospectral Methods Using LGL, LG, and LGR Collocation Points

In this section we provide a detailed description of pseudospectral methods using the LGL, LG, and LGR points. In particular, we present the one accepted approach for LGL collocation and present two *different* approaches for LG and LGR collocation. The approach for LGL collocation is referred to as the *Legendre Pseudospectral Method* (LPM) and was originally developed in Ref. 1. The first approach presented for both LG and LGR collocation, which we refer to as LPM@LG and LPM@LGR, are variations of the LPM described in Ref. 12. The second approach for LG collocation, called the *Gauss pseudospectral method* (GPM), is that developed in the works of Refs. 14–16 while the second approach for LGR collocation, called the *Radau pseudospectral method*, is that developed in the work of Refs. 17.[†] Table 1 shows how the various approaches considered in this paper utilize collocation points, interpolation points, and discretization points.

5.1 Legendre Pseudospectral Method (LPM)

As noted, in the Legendre pseudospectral method (LPM) collocation points *and* interpolation are performed at the LGL points. The state in the LPM is approximated as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=1}^{N} \mathcal{L}_i(\tau)\mathbf{X}(\tau_i) \tag{6}$$

where the Lagrange polynomials $\mathcal{L}_i(\tau),\ (i = 1, \ldots, N)$ are defined as

$$\mathcal{L}_i(\tau) = \prod_{\substack{j=1 \\ j\neq i}}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{7}$$

The time derivative of the state is then given as

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=1}^{N} \dot{\mathcal{L}}_i(\tau)\mathbf{X}(\tau_i) \tag{8}$$

The approximation to the time derivative of the state given in Eq. (8) is then applied at the N LGL (i.e., collocation) points $(\tau_1, \ldots, \tau_N)$ as

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=1}^{N} \dot{\mathcal{L}}_i(\tau_k)\mathbf{X}(\tau_i) = \sum_{i=1}^{N} D_{ki}^{LPM}\mathbf{X}(\tau_i), \qquad (k = 1, \ldots, N) \tag{9}$$

where $D_{ki}^{LPM},\ (k, i = 1, \ldots, N)$ is the $N \times N$ *differentiation matrix*. As will become more apparent later, it is *extremely* important to realize that D_{ki}^{LPM} is a *square* matrix (see Ref. 12 for the numerical values in

[†] We note that in this paper the LPM@LG and LPM@LGR will not be referred to as methods because, as correctly stated in Refs. 12 and 13, these variations of the LPM are indeed invalid. On the other hand, the *GPM and the RPM are valid* and, thus, will be referred to as methods.

D_{ki}^{LPM}). The continuous-time dynamics given in Eq. (2) are then collocated at the N LGL points as

$$\sum_{i=1}^{N} D_{ki}^{LPM}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) = \mathbf{0}, \qquad (k = 1, \ldots, N) \tag{10}$$

Next, the continuous-time cost functional is approximated using a Gauss-Lobatto quadrature as

$$J \approx \boldsymbol{\Phi}(\mathbf{X}(\tau_1), \mathbf{X}(\tau_N), \tau_1, \tau_N; t_0, t_f) + \frac{t_f - t_0}{2}\sum_{i=1}^{N} w_i^{LGL} g(\mathbf{X}(\tau_i), \mathbf{U}(\tau_i), \tau_i; t_0, t_f) \tag{11}$$

where w_i^{LGL}, $(i = 1, \ldots, N)$ are the LGL *weights*. The continuous-time boundary conditions are then approximated as

$$\boldsymbol{\phi}(\mathbf{X}(\tau_1), \mathbf{X}(\tau_N), \tau_1, \tau_N; t_0, t_f) = \mathbf{0} \tag{12}$$

Finally, the path constraints are approximated at the N LGL collocation points as

$$\mathbf{C}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) \leq \mathbf{0}, \qquad (k = 1, \ldots, N) \tag{13}$$

5.2 Variation of LPM for LG Points (LPM@LG)

The variation of the LPM for LG points, denoted, LPM@LG,[11–13] is essentially the LPM *applied at the* $N-2$ *LG points* $(\tau_2, \ldots, \tau_{N-1})$. In other words, the LPM@LG is obtained by making the following changes to the LPM:

- Approximation of the state using interpolation at *only* the LG points;
- Collocation of the dynamics at the LG points *without* taking into account the influence of the initial and terminal points $\tau_1 = -1$ and $\tau_N = 1$;
- Incorrect application of the boundary conditions and endpoint cost *at the first and last LG points*

The details of the LPM@LG are given as follows. First, the state in the LPM@LG is approximated as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=2}^{N-1} \mathcal{L}_i(\tau)\mathbf{X}(\tau_i) \tag{14}$$

where the Lagrange polynomials $\mathcal{L}_i(\tau)$, $(i = 2, \ldots, N-1)$ are defined as

$$\mathcal{L}_i(\tau) = \prod_{\substack{j=2 \\ j\neq i}}^{N-1} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{15}$$

The time derivative of the state is then given as

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=2}^{N-1} \dot{\mathcal{L}}_i(\tau)\mathbf{X}(\tau_i) \tag{16}$$

The approximation to the time derivative of the state given in Eq. (16) is then applied at the $N-2$ LG collocation points $(\tau_2, \ldots, \tau_{N-1})$ as

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=2}^{N-1} \dot{\mathcal{L}}_i(\tau_k)\mathbf{X}(\tau_i) = \sum_{i=2}^{N-1} D_{ki}^{LPM@LG}\mathbf{X}(\tau_i), \qquad (k = 2, \ldots, N-1) \tag{17}$$

where $D_{ki}^{LPM@LG} = \dot{\mathcal{L}}_i(\tau_k),\ (k, i = 2, \ldots, N-1)$ is the $(N-2) \times (N-2)$ *LPM@LG differentiation matrix*. Again, as with the LPM, it is *extremely* important to realize that $D_{ki}^{LPM@LG}$ is a *square* matrix whose entries are given in Ref. 12. The continuous-time dynamics given in Eq. (2) are then collocated at the $N-2$ LG points as

$$\sum_{i=2}^{N-1} D_{ki}^{LPM@LG}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) = \mathbf{0}, \qquad (k = 2, \ldots, N-1) \tag{18}$$

Next, the continuous-time cost functional is approximated using a Gauss quadrature as

$$J \approx \mathbf{\Phi}(\mathbf{X}(\tau_2), \mathbf{X}(\tau_{N-1}), \tau_2, \tau_{N-1}; t_0, t_f) + \frac{t_f - t_0}{2}\sum_{i=2}^{N-1} w_i^{LG} g(\mathbf{X}(\tau_i), \mathbf{U}(\tau_i), \tau_i, t_0, t_f) \tag{19}$$

where $w_i^{LG},\ (i = 2, \ldots, N-1)$ are the LG *weights*. We note that the endpoint cost Φ in Eq. (19) is applied at the *first and last LG points* τ_2 and τ_{N-1} and is *not* applied at the boundary points $\tau_1 = -1$ and $\tau_N = 1$. Furthermore, similar to the way that the endpoint cost is applied at the *first and last LG points*, the continuous-time boundary conditions are also approximated *at the first and last LG points* as

$$\boldsymbol{\phi}(\mathbf{X}(\tau_2), \mathbf{X}(\tau_{N-1}), \tau_2, \tau_{N-1}; t_0, t_f) = \mathbf{0} \tag{20}$$

Finally, the path constraints are approximated at the $N-2$ LG collocation points as

$$\mathbf{C}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) \leq \mathbf{0}, \qquad (k = 2, \ldots, N-1) \tag{21}$$

We complete the description of the LPM@LG by stating that *nowhere* in the LPM@LG are the boundary points $\tau_1 = -1$ and $\tau_N = 1$ *ever* used.

5.3 Variation of LPM for LGR Points (LPM@LGR)

Similar to the LPM@LG, the LPM@LGR is the LPM *applied at the $N-1$ LGR points* $(\tau_1, \ldots, \tau_{N-1})$. In other words, the LPM@LGR is obtained by making the following changes to the LPM:

- Approximation of the state using interpolation at *only* the LGR points;
- Collocation of the dynamics at the LGR points *without* taking into account the influence of the terminal point $\tau_N = 1$;
- Incorrect application of the boundary conditions and endpoint cost *at the last LGR point*

The details of the LPM@LG are given as follows. First, the state in the LPM@LGR method is approximated as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=1}^{N-1} \mathcal{L}_i(\tau)\mathbf{X}(\tau_i) \tag{22}$$

where the Lagrange polynomials $\mathcal{L}_i(\tau),\ (i = 1, \ldots, N-1)$ are defined as

$$\mathcal{L}_i(\tau) = \prod_{\substack{j=1 \\ j\neq i}}^{N-1} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{23}$$

The time derivative of the state is then given as

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=1}^{N-1} \dot{\mathcal{L}}_i(\tau)\mathbf{X}(\tau_i) \tag{24}$$

The approximation to the time derivative of the state given in Eq. (24) is then applied at the $N-1$ LGR collocation points $(\tau_1, \ldots, \tau_{N-1})$ as

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=1}^{N-1} \dot{\mathcal{L}}_i(\tau_k)\mathbf{X}(\tau_i) = \sum_{i=1}^{N-1} D_{ki}^{LPM@LGR}\mathbf{X}(\tau_i), \qquad (k = 1, \ldots, N-1) \tag{25}$$

where $D_{ki}^{LPM@LGR} = \dot{\mathcal{L}}_i(\tau_k),\ (k, i = 1, \ldots, N-1)$ is the $(N-1) \times (N-1)$ *LPM@LGR differentiation matrix*. Again, as with the LPM and the LPM@LG methods, it is *extremely* important to realize that $D_{ki}^{LPM@LGR}$ is a *square* matrix whose entries are given in Ref. 12. The continuous-time dynamics given in Eq. (2) are then collocated at the $N-1$ LGR points as

$$\sum_{i=1}^{N-1} D_{ki}^{LPM@LGR}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k) = \mathbf{0}, \qquad (k = 1, \ldots, N-1) \tag{26}$$

Next, the continuous-time cost functional is approximated using a Gauss-Radau quadrature as

$$J \approx \mathbf{\Phi}(\mathbf{X}(\tau_1), \mathbf{X}(\tau_{N-1}), \tau_1, \tau_{N-1}; t_0, t_f) + \frac{t_f - t_0}{2} \sum_{i=1}^{N-1} w_i^{LGR} g(\mathbf{X}(\tau_i), \mathbf{U}(\tau_i), \tau_i) \tag{27}$$

where $w_i^{LGR},\ (i = 1, \ldots, N-1)$ are the LGR *weights*. We note that, in a manner similar to the LPM@LG method, the endpoint cost Φ in Eq. (27) is applied using the *first and last LGR point* τ_1 and τ_{N-1} and is *not applied* at the boundary points $\tau_1 = -1$ and $\tau_N = 1$. Furthermore, similar to the way that the endpoint cost is applied at the *first and last LGR points*, the continuous-time boundary conditions are also approximated at the *first and last LGR points* as

$$\boldsymbol{\phi}(\mathbf{X}(\tau_1), \mathbf{X}(\tau_{N-1}), \tau_1, \tau_{N-1}; t_0, t_f) = \mathbf{0} \tag{28}$$

Finally, the path constraints are approximated at the $N-1$ LGR points as

$$\mathbf{C}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) \leq \mathbf{0}, \qquad (k = 1, \ldots, N-1) \tag{29}$$

We complete the description of the LPM@LGR method by stating that *nowhere* in the LPM@LGR method is the final boundary point $\tau_N = 1$ *ever* used.

5.4 Gauss Pseudospectral Method (GPM)

Before we begin the description of the Gauss pseudospectral method (GPM),[14–16] we note that *the GPM is not the same method as the LPM@LG*. In particular, the GPM utilizes *both* the collocation points (i.e., the LG points) *and* the endpoints $\tau_1 = -1$ and $\tau_N = 1$. First, in the GPM the state is approximated using the initial point and the LG points as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=1}^{N-1} \mathcal{L}_i(\tau)\mathbf{x}(\tau_i) \tag{30}$$

where the Lagrange polynomials $\mathcal{L}_i(\tau)$ $(i = 1, \ldots, N-1)$ are defined as

$$\mathcal{L}_i(\tau) = \prod_{\substack{j=1 \\ j\neq i}}^{N-1} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{31}$$

The time derivative of the state is then given as

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=1}^{N-1} \dot{\mathcal{L}}_i(\tau)\mathbf{X}(\tau_i) \tag{32}$$

The approximation to the time derivative of the state given in Eq. (32) is then applied at the $N-2$ LG collocation points $(\tau_2, \ldots, \tau_{N-1})$ as

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=1}^{N-1} \dot{\mathcal{L}}_i(\tau_k)\mathbf{X}(\tau_i) = \sum_{i=1}^{N-1} D_{ki}^{GPM}\mathbf{X}(\tau_i), \qquad (k = 2, \ldots, N-1) \tag{33}$$

where D_{ki}^{GPM}, $(k = 2, \ldots, N-1; i = 1, \ldots, N-1)$ is the $(N-2)\times(N-1)$ *GPM differentiation matrix*.[‡] Unlike the LPM@LG, the GPM differentiation matrix is *not square* (it has more columns than rows) because the state is approximated using a different set of points than are used to collocate the dynamics. In particular, the dynamics are collocated at the $N-2$ LG points as

$$\sum_{i=1}^{N-1} D_{ki}^{GPM}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) = \mathbf{0}, \qquad (k = 2, \ldots, N-1) \tag{34}$$

Next, in order to account for the initial and terminal points (i.e., the *boundary points* $\tau_1 = -1$ and $\tau_N = 1$), an additional variable $\mathbf{X}(\tau_N)$ is defined via a Gauss quadrature as

$$\mathbf{X}(\tau_N) \equiv \mathbf{X}(\tau_1) + \frac{t_f - t_0}{2}\sum_{i=2}^{N-1} w_i^{LG}\mathbf{f}(\mathbf{X}(\tau_i), \mathbf{U}(\tau_i), \tau_i; t_0, t_f) \tag{35}$$

The reader should observe that the quadrature equation given in Eq. (35) does *not* appear in the LPM@LG. Next, the continuous-time cost functional is approximated using a Gauss quadrature as

$$J \approx \mathbf{\Phi}(\mathbf{X}(\tau_1), \mathbf{X}(\tau_{K+2}), \tau_1, \tau_{K+2}; t_0, t_f) + \frac{t_f - t_0}{2}\sum_{i=2}^{N-1} w_i^{LG} g(\mathbf{X}(\tau_i), \mathbf{U}(\tau_i), \tau_i; t_0, t_f) \tag{36}$$

[‡] Repeating again for clarity, $D_{ki}^{GPM} \neq D_{ki}^{LPM@LG}$. In fact, these two matrices are not even the same size.

where w_i^{LG}, $(i = 2, \ldots, N-1)$ are the Legendre-Gauss *weights*. Again, unlike the LPM@LG, the endpoint cost in the GPM is evaluated at the *boundary points* $\tau_1 = -1$ and $\tau_N = 1$ and is *not* applied at the first and last LG points. Furthermore, similar to the way that the endpoint cost is evaluated at the boundary points, the continuous-time boundary conditions are also approximated at the *boundary points* as

$$\boldsymbol{\phi}(\mathbf{X}(\tau_1), \mathbf{X}(\tau_N), \tau_1, \tau_N; t_0, t_f) = \mathbf{0} \tag{37}$$

Finally, the path constraints are approximated at the $N-2$ LG points as

$$\mathbf{C}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) \leq \mathbf{0}, \qquad (k = 2, \ldots, N-1) \tag{38}$$

We complete the description of the GPM by stating that, *unlike the LPM@LG*, in the GPM the initial and terminal points $\tau_1 = -1$ and $\tau_N = 1$ *are* included in the formulation [which is seen clearly by examining the endpoint cost in Eq. (36) and the boundary conditions in Eq. (37)].

5.5 Radau Pseudospectral Method (RPM)

Before we begin the description of the Radau pseudospectral method (RPM),[17,25] we note that *the RPM is not the same method as the LPM@LGR*. In particular, the RPM utilizes *both* the collocation points and the endpoint $\tau_N = \tau_{K+1} = 1$. First, the state in the RPM is approximated as

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=1}^{N} \mathcal{L}_i(\tau)\mathbf{x}(\tau_i) \tag{39}$$

where the Lagrange polynomials $\mathcal{L}_i(\tau)$ $(i = 1, \ldots, N)$ are defined as

$$\mathcal{L}_i(\tau) = \prod_{\substack{j=1 \\ j \neq i}}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{40}$$

The time derivative of the state is then given as

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=1}^{N} \dot{\mathcal{L}}_i(\tau)\mathbf{X}(\tau_i) \tag{41}$$

The approximation to the time derivative of the state given in Eq. (41) is then applied at the $N-1$ LGR collocation points $(\tau_1, \ldots, \tau_{N-1})$ as

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=1}^{N} \dot{\mathcal{L}}_i(\tau_k)\mathbf{X}(\tau_i) = \sum_{i=1}^{N} D_{ki}^{RPM}\mathbf{X}(\tau_i), \qquad (k = 1, \ldots, N-1) \tag{42}$$

where D_{ki}^{RPM}, $(k = 1, \ldots, N-1; i = 1, \ldots, N)$ is the $(N-1) \times N$ *RPM differentiation matrix*.[§] Unlike the LPM@LGR, the RPM differentiation matrix is *not square* (it has more columns than rows) because the state

[§] Repeating again for clarity, $D_{ki}^{RPM} \neq D_{ki}^{LPM@LGR}$. In fact, these two matrices are not even the same size.

is approximated using a different set of points than are used to collocate the dynamics. In particular, the dynamics are collocated at the $N-1$ LGR points as

$$\sum_{i=1}^{N} D_{ki}^{RPM}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) = \mathbf{0}, \qquad (k = 1, \ldots, N-1) \tag{43}$$

Next, the continuous-time cost functional is approximated using a Gauss-Radau quadrature as

$$J \approx \mathbf{\Phi}(\mathbf{X}(\tau_1), \mathbf{X}(\tau_N), \tau_1, \tau_N; t_0, t_f) + \frac{t_f - t_0}{2}\sum_{i=1}^{N-1} w_i^{LGR} g(\mathbf{X}(\tau_i), \mathbf{U}(\tau_i), \tau_i; t_0, t_f) \tag{44}$$

Again, unlike the LPM@LGR, the endpoint cost in the RPM is evaluated at the *boundary points* $\tau_1 = -1$ and $\tau_N = 1$ and is *not applied* at the first and last LGR points. Furthermore, similar to the way that the endpoint cost is applied at the boundary points, the continuous-time boundary conditions are also approximated at the *boundary points* as

$$\boldsymbol{\phi}(\mathbf{X}(\tau_1), \mathbf{X}(\tau_N), \tau_1, \tau_N; t_0, t_f) = \mathbf{0} \tag{45}$$

Finally, the path constraints are approximated at the $N-1$ LGR points as

$$\mathbf{C}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) \leq \mathbf{0}, \qquad (k = 1, \ldots, N-1) \tag{46}$$

We complete the description of the RPM by stating that, *unlike the LPM@LGR*, in the RPM the terminal point $\tau_N = 1$ *is* included in the formulation [this last fact is seen clearly by examining the endpoint cost in Eq. (44) and the boundary conditions in Eq. (45)].

6 Dual Solutions Using LPM, LPM@LG/LGR, and GPM/RPM

The next critical step in the current analysis is to discuss how the dual solutions to the optimal control problem (i.e., the costates) are computed from the NLP solutions using the LPM, LPM@LG/LGR, and GPM/RPM. This discussion is divided into two parts: (1) a discussion of the costate computations using LPM, LPM@LG, and LPM@LGR and (2) a discussion of the costate computation using the GPM and the RPM.

6.1 Dual Solutions Using LPM, LPM@LG, and LPM@LGR

Ref. 12 proposes a unified costate mapping approach for computing discrete dual solutions of the continuous-time optimal control problem using the LPM, LPM@LG, and LPM@LGR. The authors of this paper attempted to reproduce the results given in Ref. 12 using our best understanding of this unified costate mapping approach, but were unsuccessful. We then contacted the authors of Ref. 12 to find out how this unified costate mapping approach is implemented, but they were unwilling to provide us with any information that would enable us to reproduce the costates.

Given our inability to ascertain how the unified costate mapping approach is applied in practice, we decided to estimate costates using the approach described in Ref. 4. In particular, Ref. 4 describes a two-step approach as follows. First, the costate is calculated using the equation[4]

$$\mathbf{\Lambda}_k = \tilde{\mathbf{\Lambda}}_k / w_k \tag{47}$$

where $\tilde{\mathbf{\Lambda}}_k$ is the KKT multiplier of the associated collocated dynamic constraint and w_k is the corresponding quadrature weight. More generally, Eq. (47) can be written for the LPM, LPM@LG, and LPM@LGR as

$$\begin{array}{lll} \text{LPM :} & \mathbf{\Lambda}_k^{LPM} = \tilde{\mathbf{\Lambda}}_k^{LPM} / w_k^{LGL} & , \quad (k = 1, \ldots, N) \\ \text{LPM@LG :} & \mathbf{\Lambda}_k^{LPM@LG} = \tilde{\mathbf{\Lambda}}_k^{LPM@LG} / w_k^{LG} & , \quad (k = 2, \ldots, N-1) \\ \text{LPM@LGR :} & \mathbf{\Lambda}_k^{LPM@LGR} = \tilde{\mathbf{\Lambda}}_k^{LPM@LGR} / w_k^{LGR} & , \quad (k = 1, \ldots, N-1) \end{array} \tag{48}$$

The second step of the procedure is to filter the result of Eq. (48) through a three-point FIR digital filter.[¶] As will be seen, the aforementioned two-step procedure produces costate estimates that are identical to those given in Ref. 12 for the examples considered later in this paper.

6.2 Dual Solutions Using GPM and RPM

The approach for computation of the dual solution in the GPM and RPM differs from that of the LPM@LG or LPM@LGR. In particular, the costates for the GPM are computed at the K LG points and the endpoints as[14–16]

$$\text{GPM:} \quad \begin{array}{l} \mathbf{\Lambda}_N = \tilde{\mathbf{\Lambda}}_N \\ \mathbf{\Lambda}_k = \tilde{\mathbf{\Lambda}}_N + \tilde{\mathbf{\Lambda}}_k / w_k^{LG}, \qquad (k = 2, \ldots, N-1) \\ \mathbf{\Lambda}_1 = \tilde{\mathbf{\Lambda}}_N - \displaystyle\sum_{i=2}^{N-1} D_{i,1}^{GPM} \tilde{\mathbf{\Lambda}}_i \end{array} \tag{49}$$

where $\mathbf{\Lambda}_k; (k = 2, \ldots, N-1)$ are the costates at the LG points (and $\tilde{\mathbf{\Lambda}}_k; (k = 2, \ldots, N-1)$ are the corresponding KKT multipliers) $\mathbf{\Lambda}_N$ is the costate at the terminal point [and $\tilde{\mathbf{\Lambda}}_N$ is the corresponding KKT multiplier, associated with Eq. (35)], $\mathbf{\Lambda}_1$ is the initial costate, and where $D_{i,1}^{GPM}$ $(i = 2, \ldots, N-1)$ are the elements in the first column of the GPM differentiation matrix. The costates for the RPM are computed at the $N-1$ LGR points and the terminal point as[16,17]

$$\text{RPM:} \quad \begin{array}{l} \mathbf{\Lambda}_k = \tilde{\mathbf{\Lambda}}_k / w_k^{LGR}, \qquad (k = 1, \ldots, N-1) \\ \mathbf{\Lambda}_N = - \displaystyle\sum_{i=1}^{N-1} D_{i,N}^{RPM} \tilde{\mathbf{\Lambda}}_i \end{array} \tag{50}$$

[¶] The authors of this paper gratefully acknowledge Tom P. Thorvaldsen of The Charles Stark Draper Laboratory, Inc. for supplying us with the mathematical equations so that we could develop an independent implementation of the three-point digital filtering algorithm. The filtering algorithm and a MATLAB® implementation of the filter are shown in the Appendix.

where $D_{i,N}^{RPM},\ (i=1,\ldots,N-1)$ are the elements in the N^{th} column of the RPM differentiation matrix.[26]

7 Examples Using LPM, LPM@LG/LGR, and GPM/RPM

We now consider two examples to compare the solutions obtained using the LPM, LPM@LG, LPM@LGR, GPM, and RPM. In order to reproduce results that have already been published by the authors of Ref. 12, the examples studied in this section are taken from Ref. 12. It is noted that the original results for these examples are reprinted from Ref. 12 while our independent results are obtained using the software *OptimalPrime*[27] developed at The University of Florida by Mr. Michael Patterson, the second author of this paper.

7.1 Example 1: Particle Moving in a Linear Resistive Medium

Consider the following example, taken from Ref. 12 but originally found in Ref. 28. Minimize the cost functional

$$J=\int_0^1 x_2(t)u(t)dt \tag{51}$$

subject to the dynamic constraints

$$\begin{array}{rcl}\dot{x}_1(t) & = & x_2(t)\\ \dot{x}_2(t) & = & -x_2(t)+u(t)\end{array}, \tag{52}$$

the path and control constraints

$$\begin{array}{rcl}x_2(t) & \geq & 0\\ u(t) & \in & [0,2]\end{array}, \tag{53}$$

and the boundary conditions

$$\begin{array}{rcl}(x_1(0),x_2(0)) & = & (0,1)\\ (x_1(1),x_2(1)) & = & (1,1)\end{array} \tag{54}$$

7.1.1 Solution Using LPM, LPM@LG, and LPM@LGR

It is noted that the optimal control for this example is $u^*(t)\equiv 1$. Quoting directly from page 9 of Ref. 12,

> "Since the optimal control is a polynomial, a correct PS [pseudospectral] method is expected to achieve exact performance for sufficiently large N."

The authors of Ref. 12 then demonstrate for this example that the LPM produces the correct solution while the LPM@LG and LPM@LGR solutions produce, "disastrous solutions." For use later in the discussion, the LPM, LPM@LG, and LPM@LGR control solutions obtained in Ref. 12 for 10 and 30 collocation points are reprinted[‖] (with *a minor* edit to reflect the terminologies "LPM@LG" and "LPM@LGR" used in this paper) in Figs. 2a (LPM) and 2b (LPM@LG and LPM@LGR). The authors of this paper then reproduced the results given in Fig. 2 independently using the pseudospectral optimal control software *OptimalPrime*.[27][**] The results using *OptimalPrime* obtained by applying the LPM, LPM@LG, and LPM@LGR for 10 and 30 collocation points are shown in Fig. 3a (for 10 collocation points) and 3b (for 30 collocation points). It is seen that the *OptimalPrime* results are *identical* to the results obtained in Ref. 12 (see again Fig. 2). From these results, we support the conclusions made by the authors of Refs. 12 and 13 that LG and LGR collocation will indeed be invalid *if such methods are based on the Legendre pseudospectral method*.

Next, it was desired to obtain costate solutions using LPM, LPM@LG, and LPM@LGR. Fig. 4 shows the unfiltered costates we obtained for the LPM, LPM@LG, and LPM@LGR using Eq. (48) for 30 collocation points (i.e., the costates obtained *before* applying the aforementioned digital filter) while Fig. 5 shows the costates obtained in Ref. 12. It is seen that the unfiltered LPM@LG and LPM@LGR solutions that we obtained bared no resemblance to those given in Ref. 12 and 13.

Next, the costates shown in Fig. 4 were filtered using our implementation of the aforementioned digital filter (see Appendix) and the results are shown in Fig. 6. Comparing Fig. 6 and 5, it is seen that our filtered costates are *identical* to those given in Fig. 5 (i.e., those shown in Fig. 5 and given in Ref. 12). Furthermore, it is seen that, even though filtering improves the result, the filtered costates are still not constant. Instead, the filtered costates still show small variations about the optimal solutions of $\lambda_1^*(t) \equiv -2$ and $\lambda_2^*(t) \equiv -1$. Thus, while filtering the costates improves the solution, the result is still suboptimal.

7.1.2 Solution Using GPM and RPM

Next, suppose we solve Example 1 using the *Gauss Pseudospectral Method* (GPM) and *Radau Pseudospectral Method* (RPM) as described in Sections 5.4 and 5.5, respectively. Recall again that the GPM and the RPM are *not* the same methods as the LPM@LG and LPM@LGR, respectively. Instead, the GPM and RPM are *completely different methods* based on using different sets of interpolation and collocation points.

Figs. 7a and 7b shows the optimal control obtained by applying the GPM (Fig. 7a) and RPM (Fig. 7b) for 10 and 30 collocation points, respectively, using the software *OptimalPrime*. *Unlike the results obtained using the LPM@LG and LPM@LGR*, the GPM and RPM solutions are in excellent agreement with the optimal control $u^*(t) \equiv 1$. Now, although we have provided an extensive discussion and comparison of the differences between the LG and LGR framework proposed in Ref. 12 and the framework proposed in this paper, one may still ask for clarification as to why the results obtained using the GPM and RPM are much

[‖] It is noted that the material in Ref. 12 is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Furthermore, the material taken from Ref. 12 is used under the "doctrine of fair use" in conjunction with scholarly analysis.

[**] It is noted that *OptimalPrime* is designed to implement the LPM, GPM, and RPM, but, with a few simple changes to the source code, can be made to implement the LPM@LG and LPM@LGR.

better than those obtained using the LPM@LG or LPM@LGR. In light of the results obtained in Fig. 7, we remind the reader that the GPM and RPM utilize different interpolation points and collocation points and arise from a foundation that is distinct from that of LPM-based methods. In particular, Refs. 16 and 25 provide comparisons showing why the GPM and RPM produce better costates than the LPM@LG and LPM@LGR methods, respectively.

Similar to the high-quality state and control solutions, the GPM and RPM produce equally high-quality dual solutions. In particular, Fig. 8 shows the GPM and RPM costates for $N = 10$ and $N = 30$ (i.e., 10 and 30 collocation points) obtained by applying Eqs. (49) and (50). Again, unlike the results obtained for the LPM@LG and LPM@LGR, it is seen that the costates are in excellent agreement with the optimal solutions $\lambda_1^*(t) \equiv -2$ and $\lambda_2^*(t) \equiv -1$. It is noted that the GPM and RPM costates were obtained by applying Eqs. (49)–(50) *exactly* as they appear [i.e., the KKT multipliers were extracted from the solution of the NLP and Eqs. (49)–(50) were used *verbatim*].

7.2 Example 2: Orbit-Raising Problem

Consider the following optimal control problem known in the literature as the *orbit-raising problem* that has been studied extensively (e.g., see Ref. 29), but in this case is taken from Ref. 12. Minimize the cost functional

$$J = -r(t_f) \tag{55}$$

subject to the dynamic constraints

$$\begin{aligned} \dot{r} &= v_r \\ \dot{\theta} &= v_\theta / r \\ \dot{v}_r &= v_\theta^2/r - \mu/r^2 + a \sin\beta \\ \dot{v}_\theta &= -v_r v_\theta / r + a\cos\beta \end{aligned} \tag{56}$$

and the boundary conditions

$$\begin{aligned} (r(0), \theta(0), v_r(0), v_\theta(0)) &= (1, 0, 0, 1) \\ (v_r(t_f), v_\theta(t_f)) &= (0, \sqrt{\mu/r(t_f)}) \end{aligned} \tag{57}$$

where

$$a \equiv a(t) = \frac{T}{m_0 - |\dot{m}|t} \tag{58}$$

It is noted for this example that $\mu = 1$, $T = 0.1405$, $m_0 = 1$, $\dot{m} = 0.0749$, and $t_f = 3.32$.

The orbit-raising problem was solved using the LPM, GPM, and RPM.[1] The state solutions for the three methods obtained using the software *OptimalPrime*[27] are shown in Fig. 9 for $N = 64$ (i.e., 64 *total*

[1] We do not consider the LPM@LG or LPM@LGR because, as we have discussed and has been discussed in Ref. 12, these two variations of the LPM are invalid. In addition, no results using the LPM@LG or LPM@LGR are provided in any papers written by the authors of Ref. 12 or written by others against which we could compare our LPM@LG and LPM@LGR results.

points) while the control solutions are shown in Fig. 10 (where it is noted that the controls shown Fig. 10 include an unwrapping of the angle in order to present smooth results). It is seen that, qualitatively, the three methods produce quite similar values of the state and control. Next, Fig. 11 shows the unfiltered costates for $N = 64$ (i.e., the costates obtained using Eq. (48) and *prior* to the application of the aforementioned digital filter). Clearly the unfiltered LPM costates look *nothing like* the costates obtained using either the GPM or the RPM. In fact, for this problem is it known $\lambda_\theta^* \equiv 0$. It is seen that the GPM and the RPM clearly produce the correct result for λ_θ^* while the LPM produces an incorrect result. Given the results between the three methods, it would clearly be more difficult to verify optimality using the LPM as compared to either the GPM or the RPM.

Next, as was done in Example 1, the unfiltered *OptimalPrime* costates shown in Fig. 11 were filtered using the aforementioned digital filter (see Appendix). The filtered *OptimalPrime* costates are shown in Fig. 12 alongside a reprint of the costates obtained in Ref. 12. First, it is seen that, even with filtering, the costates are still not as good as those obtained with either the GPM or the RPM. The fact that the GPM and RPM costates are better is seen by examining Fig. 11 where the values λ_θ obtained for the GPM and RPM costates are zero (recalling that the optimal solution is $\lambda_\theta^* \equiv 0$) whereas even the filtered LPM costate still drifts away from zero near the endpoints. Also, similar to the results obtained for Example 1, *our* filtered costates are *identical* to those obtained in Ref. 12 (in fact, an overlay plot of the two figures reveals that every nuance in both plots is *identical*).

8 Costate Estimation for LPM and Its Variations

While the authors of this paper acknowledge that we do not have a complete understanding of the unified costate mapping approach described in Refs. 5 and Refs. 11–13, we have shown in this paper that we are able to reproduce the results of the unified costate mapping approach by using the two-step approach where we apply Eq. (48) and filter the resulting costate with the aforementioned FIR digital filter. Furthermore, while we are able to reproduce results on the two examples that are identical to those obtained in Refs. 12 and 13, we fully acknowledge that two examples do not provide a proof that our alternate procedure is indeed correct. However, the results in this paper show that our filtering technique produces results that are indistinguishable from previously published results. Until actual numerical values for the LPM costate are published, our two-step approach is the best validation method available. Our alternate approach for costate computation using the LPM (and its variations LPM@LG and LPM@LGR) is shown schematically in Fig. 13.

9 Key Differences Between LPM-Based Variations and GPM/RPM

The following key differences arise directly from the LPM@LG/LGR methods and the GPM/RPM. First, in the LPM@LG and LPM@LGR all approximations and subsequent collocation equations use information strictly at the *collocation* points and use *no* information at the missing boundary points (i.e., both boundary points for LG the terminal boundary point for LGR). Table 2 summarizes these key differences for the LPM@LG/LGR and GPM/RPM pairs. It is seen that, because no information is used at the uncollocated boundary point(s), the LPM@LG and LPM@LGR are improperly formulated. Contrariwise, the GPM and RPM properly account for endpoint effects at $\tau_1 = -1$ and $\tau_N = 1$ while obtaining accurate state, control, and costate results, thereby resulting in *proper* formulations of LG and LGR pseudospectral methods. *It is extremely important to note that one cannot simply apply the LPM at the LG or LGR points*. In fact, this study demonstrates quite clearly the care that is needed when formulating a pseudospectral method using LG or LGR collocation. Specifically, when using LG or LGR collocation it is necessary to account for the influence of the interpolated but uncollocated endpoint(s) and apply the boundary conditions at the endpoints of the time interval and to *not* apply the boundary conditions at the first and last collocation points.

10 Our Unified Framework for Pseudospectral Methods

We now present *our* unified framework for pseudospectral methods using the GPM and RPM. The basis for our framework for LG and LGR collocation lies in the distinction between *collocation points* and *interpolation points* and the proper application of these points in the formulation of a pseudospectral method. As has been shown, collocation points and interpolation points are *not* the same. In particular, in a properly formulated LG collocation method (i.e., the *GPM*) the state approximation includes the collocation points *plus the initial point*, but the dynamics are collocated at only the LG points. Similarly, in a properly formulated LGR collocation method (i.e., the *RPM*) the state approximation includes *all of the points*, but the dynamics are collocated at only the LGR points.

Using a framework where the interpolation points are different from the collocation points, it is possible to pose properly formulated pseudospectral methods using LG and LGR collocation. First, the left-hand side of the GPM collocation conditions [see Eq. (33)] includes the influence of the initial point via the differentiation matrix. Furthermore, because LG points are missing both endpoints, a quadrature equation is required [see Eq. (35)] in order to account for the influence of the terminal point [again, see Eq. (35)]. Similarly, the left-hand side of the collocation conditions for the RPM includes the influence of the terminal point via the differentiation matrix. Because the RPM is missing only one endpoint, however, no quadrature equation is required in the RPM.

Fig. 14 presents our framework that unified the GPM and the RPM while placing the LPM, LPM@LG, and LPM@LGR outside of this framework. The LPM falls outside of our framework because the LPM collocates the the dynamics at *all* of the interpolation points. Collocating at all of the interpolation points can result in poor costate estimates or require filtering.[4] In fact, it has been shown in the literature that the LPM costates using Eq. (48) are "noisy", having "saw-tooth" behavior similar to the results presented earlier in this paper and results that have been presented elsewhere.[16,25] This "saw-tooth" behavior is what leads to the desire to filter the costates. Finally, it is noted that this "saw-tooth" behavior is exhibited not only in the LPM, but is also exhibited in the LPM@LG and LPM@LGR. Referring again to Table 1, it is seen that the LPM (and its variations) fall outside of the framework we propose in this paper.

11 Trade-Offs Between LPM, GPM, and RPM

An issue of importance is the trade-off between using the LPM, GPM, and RPM. First, the benefit of using the LPM is that, because the control is discretized at the boundaries, it is possible to incorporate a control constraint at the boundary points. Next, the RPM produces a complete set of state and costate information, but produces a control at only one endpoint. While it may appear to be a deficiency, the reality is that not having a control at the terminal point is irrelevant for applications such as model-predictive or receding horizon problems because the control from the NLP solution will be implemented for only a small fraction of the entire time interval. Moreover, the fact that the RPM produces highly accurate primal and dual solutions makes it reliable as an offline tool to verify the optimality of solution while simultaneously making it attractive as a method for applications such as guidance (i.e., real-time terminal point control). Finally, the GPM produces the state and costate at the interior points and the endpoints, but produces values of the control at only the LG points. The fact that the GPM does not provide endpoint controls can be a concern, particularly for problems where a small number of collocation points is used and the distances between the initial and terminal points are sufficiently far from the first and last LG points. Thus, the ability to use the GPM for real-time applications (e.g., model-predictive or receding horizon control) becomes an issue because it is generally the case that in a real-time implementation it is desired to use as few collocation points as possible. It is noted, however, that real-time applications using the GPM can be accommodated by integrating the Hamiltonian system (i.e., the state-costate system) using the accurate initial costate provided by the GPM.[14] Moreover, the GPM does provide an exact mapping between the KKT conditions of the NLP and the discretized first-order optimality conditions from the calculus of variations[14–16] which enables a user to verify optimality of solutions.

12 Conclusions

A unified framework has been presented for the solution of optimal control problems via pseudospectral methods. The framework has been developed by using a different set of points to discretize the state than are used to collocate the right-hand side of the dynamics. Using this approach, methods were presented for collocation at Legendre-Gauss (LG) and Legendre-Gauss-Radau (LGR) points. These two methods are called the *Gauss pseudospectral method* (GPM) and the *Radau pseudospectral method* (RPM) and have been shown to be proper formulations of LG and LGR collocation. To demonstrate the validity of our framework, we compare our results against a previously developed framework where LG and LGR collocation is obtained via variations on collocation at *Legendre-Gauss-Lobatto* (LGL) points. The results of this paper show that our framework is a significant improvement over the previously developed framework because it produces highly accurate primal (i.e., state and control) and dual (i.e., costate) solutions. The results of this research demonstrate that the GPM and the RPM can be unified into a single framework and are valid LG and LGR pseudospectral methods. Moreover, the approach presented here highlights the need to distinguish between interpolation points, collocation points, and discretization points. Two examples are considered that demonstrate the proposed framework and how this framework compares against a previously developed framework, where the previous framework suggests that LG and LGR collocation are invalid pseudospectral methods. The framework presented here provides an excellent analysis of the various methods (and variations therein for the case of LGL collocation) and will be a useful reference for researchers and engineers developing and using numerical methods for solving complex optimal control problems.

Acknowledgments

Support for this research to three of the authors (Patterson, Rao, and Pisano) by the NASA Florida Space Grant Consortium is gratefully acknowledged. The authors also gratefully acknowledge Mr. Tom Thorvaldsen of The Charles Stark Draper Laboratory, Inc. for supplying us with the algorithm for the three-point FIR digital filter used to generate some of the results in this article.

Disclaimer

The views and opinions expressed in this paper are those of the authors and do not reflect those of The National Aeronautics and Space Administration, The University of Florida, The Charles Stark Draper Laboratory, Inc., any other individual, or any other commercial, academic, not-for-profit, or government agency. Finally, it is noted that the material taken from Ref. 12 has been duly cited and used in this research under (i) the "doctrine of fair use" in conjunction with scholarly analysis and (ii) based on the fact that material in Ref. 12 is declared a work of the U.S. Government and is *not* subject to copyright protection in the United States.

Appendix: Three-Point Finite-Impulse-Response Digital Filter

The algorithm for the three point Finite-Impulse-Response (FIR) was developed by T. P. Thorvaldsen of Draper Laboratory. The algorithm is given as follows. Consider a time series of data (T, Q) where T contains a vector of values of the ordinate $T = (t_1, \ldots, t_N)$ and Q contains a vector of values of the abscissa $Q = (q_1, \ldots, q_N)$. It is desired to filter the data in Q. A general filter is given as a rational function of z^{-1} as

$$G(z) = \frac{\sum_{i=0}^{n} b_{i+1} z^{-i}}{\sum_{i=0}^{d} a_{i+1} z^{-i}} \tag{59}$$

The filter used in this research (as given by Thorvaldsen) has the following design constraints:

(a) The DC gain must be unity;

(b) The filter must produce zero gain at 1/2 the sampling frequency;

(c) The filter must be symmetric

As a result, we have $n = 2$, $d = 0$ resulting in the numerator having three coefficients (b_1, b_2, b_3) while the denominator has a single coefficient $a_1 = 1$. The design constraints of (a) through (c) above leads to the following three scalar equations:

$$b_1 + b_2 + b_3 = 1 \tag{60}$$

$$b_2 = 1/2 \tag{61}$$

$$b_1 - b_3 = 0 \tag{62}$$

Solving for b_1, b_2, and b_3 gives $b_1 = b_3 = 0.25$ and $b_2 = 0.5$. The filter is then given as

$$G(z) = b_1 + b_2 z^{-1} + b_3 z^{-2} \tag{63}$$

where $b_1 = b_3 = 0.25$ and $b_2 = 0.5$. Using $G(z)$ above, the following steps produce filtered values of Q, denoted $\tilde{Q}$:

(a) Eliminate the first two and last two data points in T and Q called T_{reduced} and Q_{reduced}, respectively;

(b) Filter Q_{reduced} using $G(z)$ above, resulting in $\tilde{Q}_{\text{reduced}}$

(c) Shift all data in $\tilde{Q}_{\text{reduced}}$ back one cycle to account for the time delay introduced by $G(z)$. Call this shifted set of data $\tilde{Q}_{\text{reduced}}^{\text{shifted}}$

(d) Compute the slopes between the following points in $\tilde{Q}^{\text{shifted}}_{\text{reduced}}$:

   (i) The last two points and the next to last two points

   (ii) The first two points and the second and third points

(e) Compute the average slopes in (i) and the average slopes in (ii);

(f) Extrapolate the data on both ends of $\tilde{Q}^{\text{shifted}}_{\text{reduced}}$ to the missing data points using the proper average slope, resulting in $\tilde{Q}$;

(g) Output the quantity $\tilde{Q}$

The MATLAB® code that implements the above procedure is given below.

```
function X_final_out = FIR_Filter(Col_Pts,Cos_in)

% -----------------------------------------------------
% Three-Point FIR Digital Filter (for Costate Filtering)

% Filter Designer: Tom P Thorvaldsen, Draper Laboratory
% Code Written By: Danielle Pisano, September 2008

% Permission is hereby granted, free of charge, to any person obtaining a copy
% of this software and associated documentation files (the "Software"), to deal
% in the Software without restriction, including without limitation the rights
% to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
% copies of the Software, and to permit persons to whom the Software is
% furnished to do so, subject to the following conditions:

% The above copyright notice and this permission notice shall be included in
% all copies or substantial portions of the Software.

% THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
% IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
% FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
% AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
% LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
% OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
% THE SOFTWARE.
% -----------------------------------------------------

% -------
% Inputs:
% -------
%    Col_Pts:      Collocation Points (Column vector of length N)
%    Cos_In:       Unfiltered Costates (matrix of size m by N, where m=# of costates)
% -------
% Output:
% -------
%    X_final_out:  Filtered Value of Costate
%
% Define variables
n=2;        % n = number values removed at beginning and end of input vector
p=2;        % p = number of values added to beginning of input by filter function
k=n+p-1;    % k = number of values to be added to beginning and end of filtered vector
if length(Cos_in) ˜= length(Col_Pts)
    error(’FIR_Filter input vectors must be the same length.’)
end
[r,c]=size(Cos_in);         % r = number of rows for input matrices
                            % c = number of cols for input matrices
j=1;                        % j = column increment counter

Filt_inp_b=[.25 .5 .25];    % Set numerator coefficient vector: b=[b1 b2 b3]
Filt_inp_a=1;               % Where b1-b3=0, b2-f/2=0, f=1, b1+b2+b3=1
```

```
X_final_out=[];             % Initialize final output matrix to empty
X_out_j=[];
while j<c+1
    Mod_Cos_in=Cos_in(n+1:end-n,j);     % Remove n points from each end of input vector
    % Apply filter
    Filt_Out=filter(Filt_inp_b,Filt_inp_a,Mod_Cos_in);
    % Manipulate filter output
    Mod_Filt_Out=Filt_Out(p+1:end,1);   % Remove points added by filter at beginning of filtered values
    % Extrapolate new endpoints base on average slopes near endpoints
    % At start of vector
    Slp_1=(Mod_Filt_Out(2,1)-Mod_Filt_Out(1,1))/(Col_Pts(k+2,1)-Col_Pts(k+1,1));
    Slp_2=(Mod_Filt_Out(3,1)-Mod_Filt_Out(2,1))/(Col_Pts(k+3,1)-Col_Pts(k+2,1));
    Avg_Slp=((Slp_1)+(Slp_2))/2;
    % Create new beginning points
    Begin_Pts=[];       % Initialize beginning point vector to empty
    i=1;                % i = point increment counter
    while i<k+1
        Pt_i=(Avg_Slp*(Col_Pts(i,1)-Col_Pts(k+1,1)))+Mod_Filt_Out(1,1);
        Begin_Pts=[Begin_Pts;Pt_i];
        i=i+1;
    end
    % At end of vector
    Slp_1=(Mod_Filt_Out(end-1,1)-Mod_Filt_Out(end,1))/(Col_Pts(end-(k+1),1)-Col_Pts(end-k,1));
    Slp_2=(Mod_Filt_Out(end-2,1)-Mod_Filt_Out(end-1,1))/(Col_Pts(end-(k+2),1)-Col_Pts(end-(k+1),1));
    Avg_Slp=((Slp_1)+(Slp_2))/2;
    % Create new end points
    i=0;                % Reset i counter
    End_Pts=[];         % Initialize end point vector to empty
    while i<k
        Pt_i=(Avg_Slp*(Col_Pts(end-i,1)-Col_Pts(end-k,1)))+Mod_Filt_Out(end,1);
        End_Pts=[Pt_i;End_Pts];
       i=i+1;
    end
    X_out_j=[Begin_Pts;Mod_Filt_Out;End_Pts];        % Concatenate points to filtered vector
    X_final_out=[X_final_out, X_out_j];              % Add columns to create final matrix
    j=j+1;
end
```

References

[1] Elnagar, G., Kazemi, M., and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp. 1793–1796.

[2] Elnagar, G. and Kazemi, M., "Pseudospectral Chebyshev Optimal Control of Constrained Nonlinear Dynamical Systems," *Computational Optimization and Applications*, Vol. 11, No. 2, 1998, pp. 195–217.

[3] Fahroo, F. and Ross, I. M., "A Spectral Patching Method for Direct Trajectory Optimization," *Journal of Astronautical Sciences*, Vol. 48, No. 2-3, Apr.-Sept. 2000, pp. 269–286.

[4] Fahroo, F. and Ross, I. M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 270–277.

[5] Ross, I. M. and Fahroo, F., *Lecture Notes in Control and Information Sciences*, chap. Legendre Pseudospectral Approximations of Optimal Control Problems, Springer-Verlag, 2003.

[6] Rao, A. V., "Extension of a Pseudospectral Legendre Method for Solving Non-Sequential Multiple-Phase Optimal Control Problems," AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2003-5634*, Austin, Texas, August 11–14 2003.

[7] Williams, P., "Jacobi Pseudospectral Method for Solving Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, 2004, pp. 293–297.

[8] Williams, P., "Application of pseudospectral methods for receding horizon control," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, 2004, pp. 310–314.

[9] Williams, P., "Hermite-Legendre-Gauss-Lobatto Direct Transcription Methods in Trajectory Optimization," *Americal Astronautical Society*, Spaceflight Mechanics Meeting, August 2005.

[10] Ross, I. M. and Fahroo, F., "Pseudospectral Knotting Methods for Solving Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, 2004, pp. 397–405.

[11] Fahroo, F. and Ross, I. M., "On Discrete-Time Optimality Conditions for Pseudospectral Methods," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2006-6304, Keystone, Colorado, August 2006.

[12] Ross, I. M. and Fahroo, F., "Advances in Pseusospectral Methods," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2008-7309, Honolulu, Hawaii, August 2008.

[13] Ross, I. M. and Fahroo, F., "Convergence of the Costates Do Not Imply Convergence of the Controls," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 1492–1497.

[14] Benson, D. A., *A Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, MIT, 2004.

[15] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, November-December 2006, pp. 1435–1440.

[16] Huntington, G. T., *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2007.

[17] Kameswaran, S. and Biegler, L. T., "Convergence Rates for Direct Transcription of Optimal Control Problems Using Collocation at Radau Points," *Computation Optimization and Applications*, Vol. 41, No. 1, 2008, pp. 81–126.

[18] Reddien, G. W., "Collocation at Gauss Points as a Discretization in Optimal Control," *SIAM Journal on Control and Optimization*, Vol. 17, No. 2, March 1979.

[19] Cuthrell, J. E. and Biegler, L. T., "On the Optimization of Differential-Algebraic Processes," *AIChe Journal*, Vol. 33, No. 8, August 1987, pp. 1257–1270.

[20] Cuthrell, J. E. and Biegler, L. T., "Simultaneous Optimization and Solution Methods for Batch Reactor Control Profiles," *Computers and Chemical Engineering*, Vol. 13, No. 1/2, 1989, pp. 49–62.

[21] Huntington, G. T., Benson, D. A., and Rao, A. V., "Optimal Configuration of Tetrahedral Spacecraft Formations," *The Journal of the Astronautical Sciences*, Vol. 55, No. 2, April-June 2007, pp. 141–169.

[22] Huntington, G. T. and Rao, A. V., "Optimal Reconfiguration of Spacecraft Formations Using the Gauss Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, May-June 2008, pp. 689–698.

[23] Rao, A. V., Scherich, A. E., Cox, S., and Mosher, T. E., "A Concept for Operationally Responsive Space Mission Planning Using Aeroassisted Orbital Transfer," *2008 Responsive Space Conference*, AIAA Paper RS6-2008-7309, Los Angeles, California, April 2008.

[24] Darby, C. L. and Rao, A. V., "An Initial Examination of Using Pseudospectral Methods for Time-Scale and Differential Geometric Analysis of Nonlinear Optimal Control Problems," *2008 Guidance, Navigation, and Control Conference*, AIAA, Honolulu, Hawaii, August 2008.

[25] Huntington, G. T. and Rao, A. V., "A Comparison of Accuracy and Computational Efficiency of Three Pseudospectral Methods," *Proceedings of Guidance, Navigation and Control Conference*, AIAA Paper 2007-6405, Hilton Head, South Carolina, August 2007.

[26] Garg, D. and Rao, A. V., "A Comparison of Accuracy of Three Pseudospectral Methods," Working Paper, Department of Mechanical and Aerospace Engineering, University of Florida, September 2008.

[27] Patterson, M. A., "OptimalPrime: A MATLAB Package for Solving Multiple-Phase Optimal Control Problems Using Pseudospectral Methods," Dept. of Mechanical and Aerospace Engineering, University of Florida, August 2008.

[28] Gong, Q., Kang, W., and Ross, I. M., "A Pseudospectral Method for the Optimal Control of Feedback Linearizable Systems," *IEEE Transactions on Automatic Control*, Vol. 51, No. 7, July 2006, pp. 1115–1129.

[29] Bryson, A. E. and Ho, Y.-C., *Applied Optimal Control*, Hemisphere Publishing, New York, 1975.

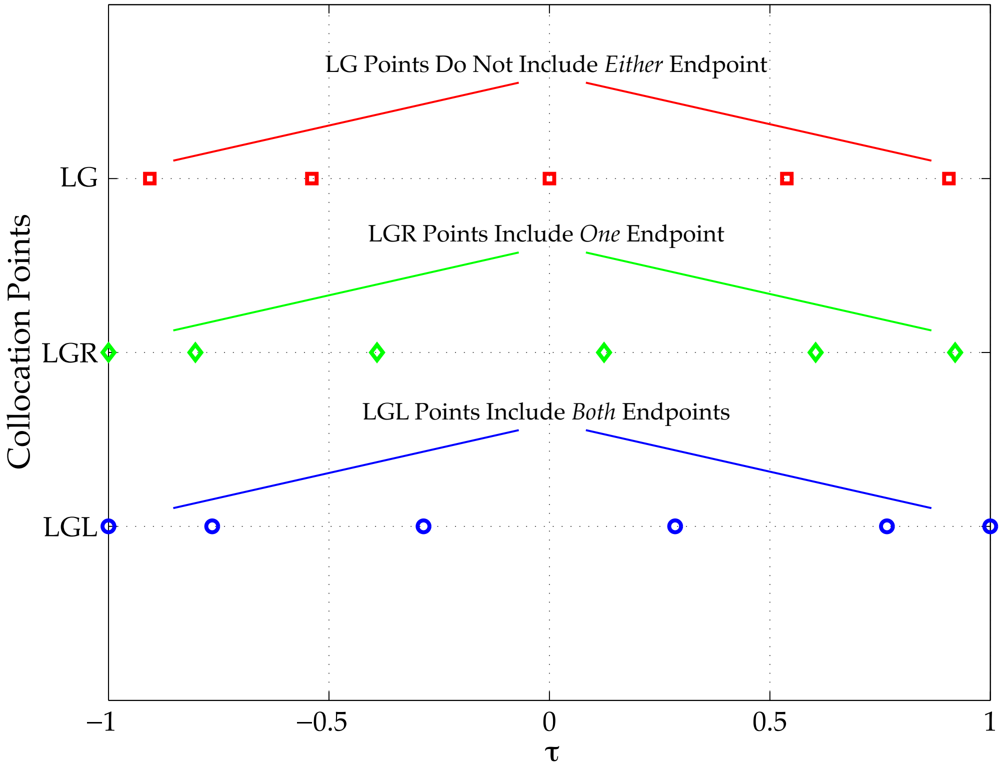
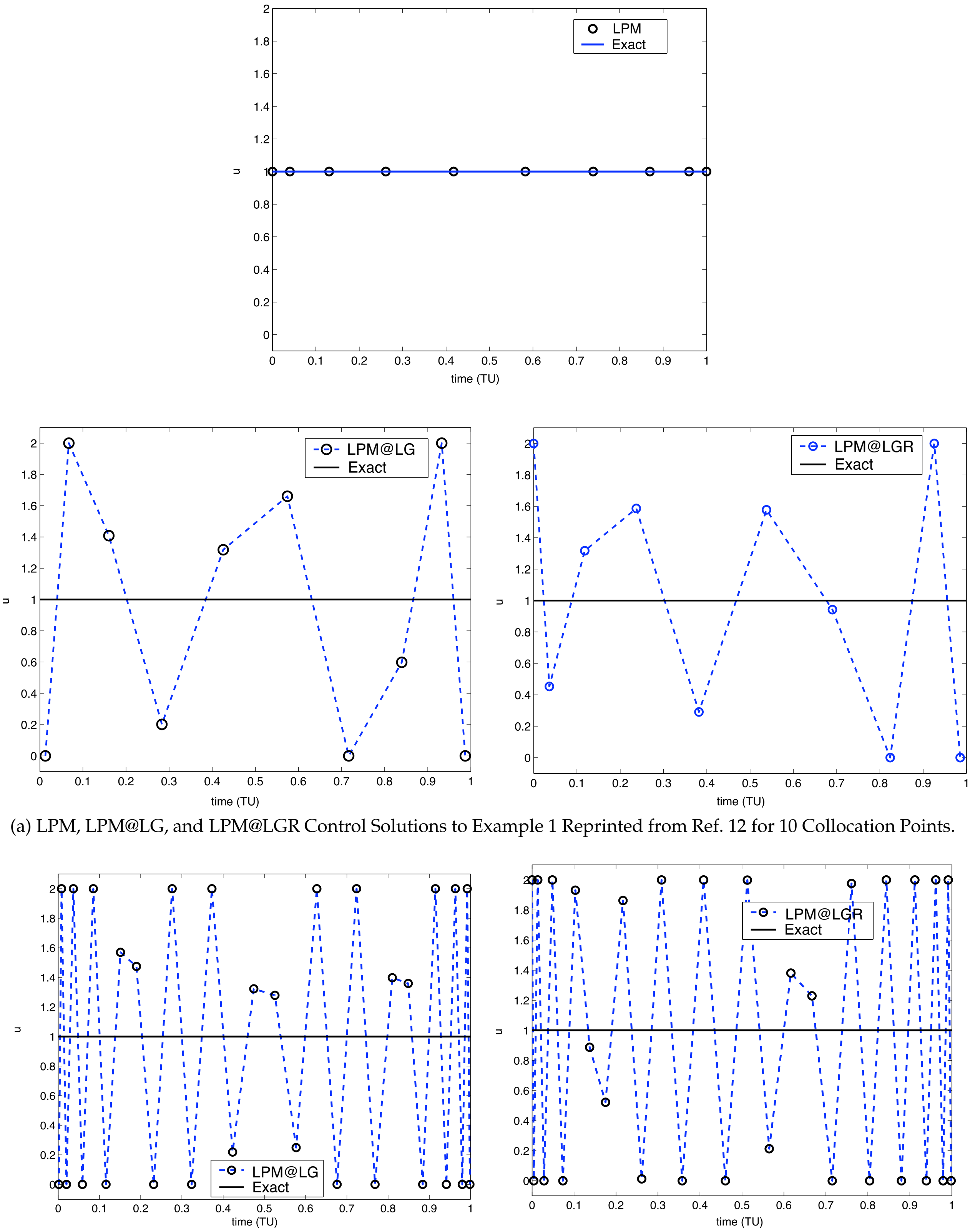


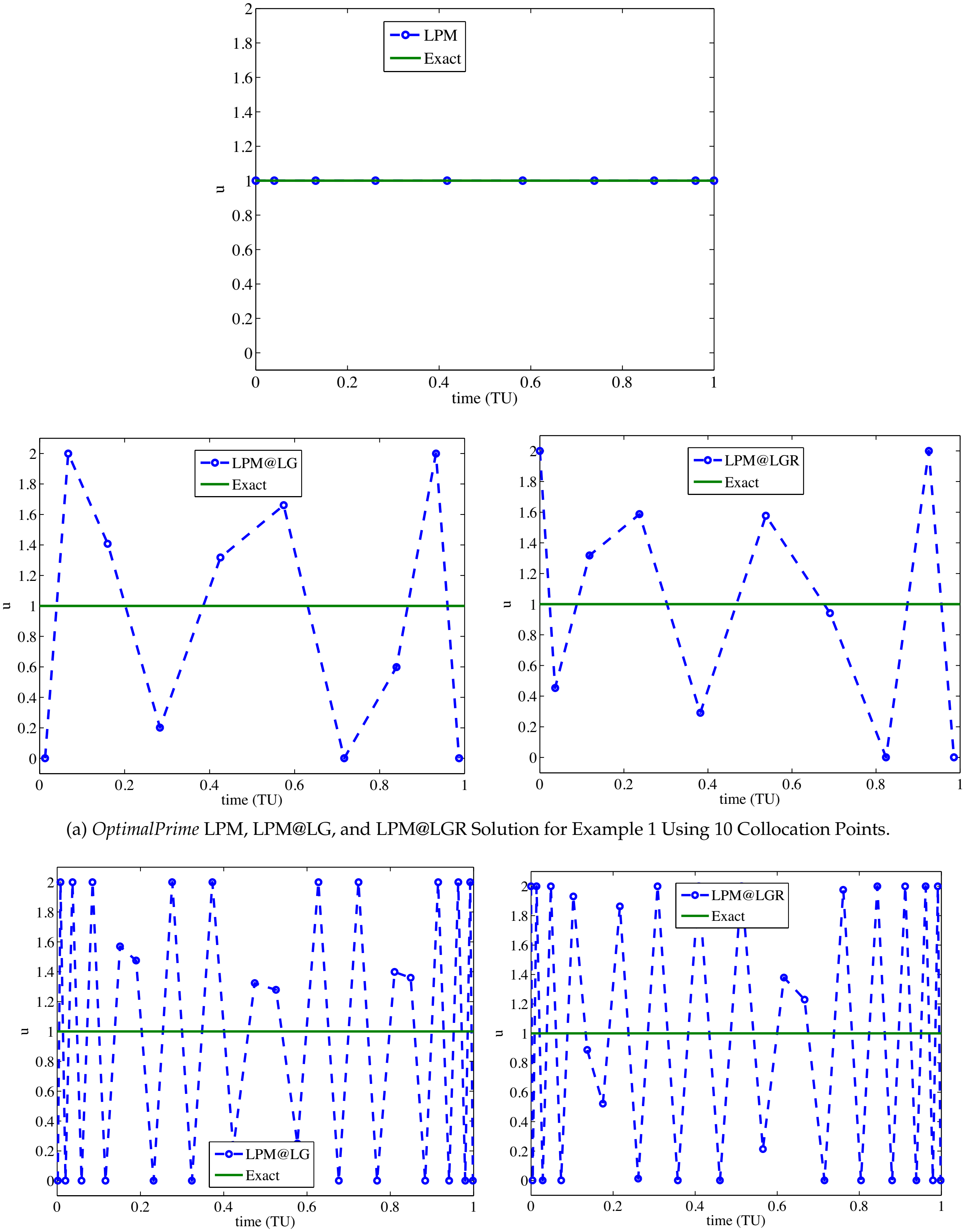
Figure 1: Schematic Showing the Differences Between LGL, LGR, and LG Collocation Points.



(a) LPM, LPM@LG, and LPM@LGR Control Solutions to Example 1 Reprinted from Ref. 12 for 10 Collocation Points.

(b) LPM, LPM@LG, and LPM@LGR Control Solutions to Example 1 Reprinted from Ref. 12 (also found in Ross 13) for 30 Collocation Points.

Figure 2: Reprint of Solutions from Ref. 12 of LPM, LPM@LG, and LPM@LGR Control Solutions to Example 1 for 10 and 30 Collocation Points.



(a) *OptimalPrime* LPM, LPM@LG, and LPM@LGR Solution for Example 1 Using 10 Collocation Points.

(b) *OptimalPrime* LPM@LG, and LPM@LGR Solution for Example 1 Using 30 Collocation Points.

Figure 3: *OptimalPrime* LPM@LG, and LPM@LGR Control Solutions to Example 1 for 10 and 30 Collocation Points.

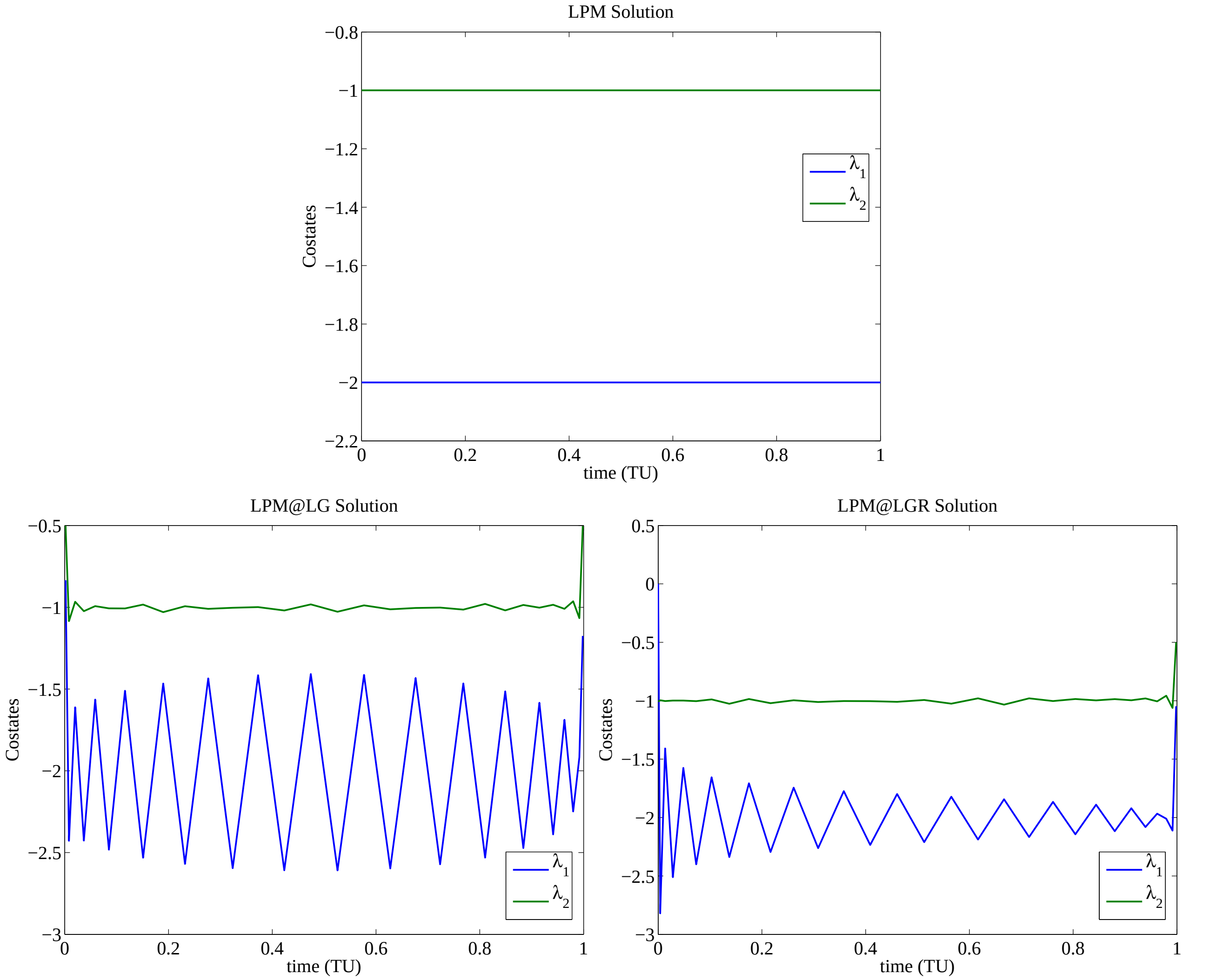


Figure 4: *OptimalPrime* Costate Solutions Using LPM, LPM@LG, and LPM@LGR by Applying Eq. (48) for Example 1 Using 30 Collocation Points.

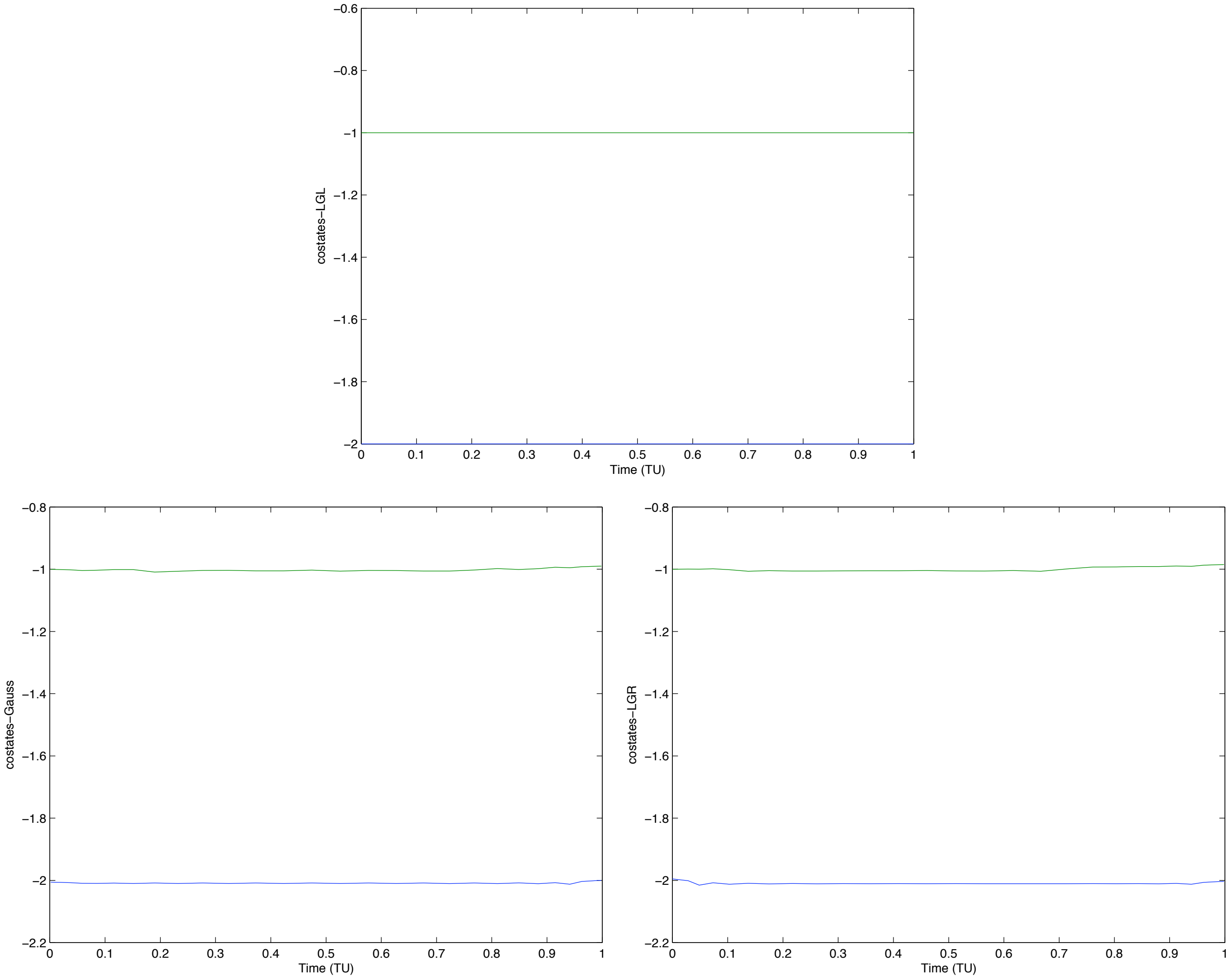


Figure 5: Costate Solutions for LPM, LPM@LG, and LPM@LGR Reprinted from Ref. 12 for Example 1 Using 30 Collocation Points.

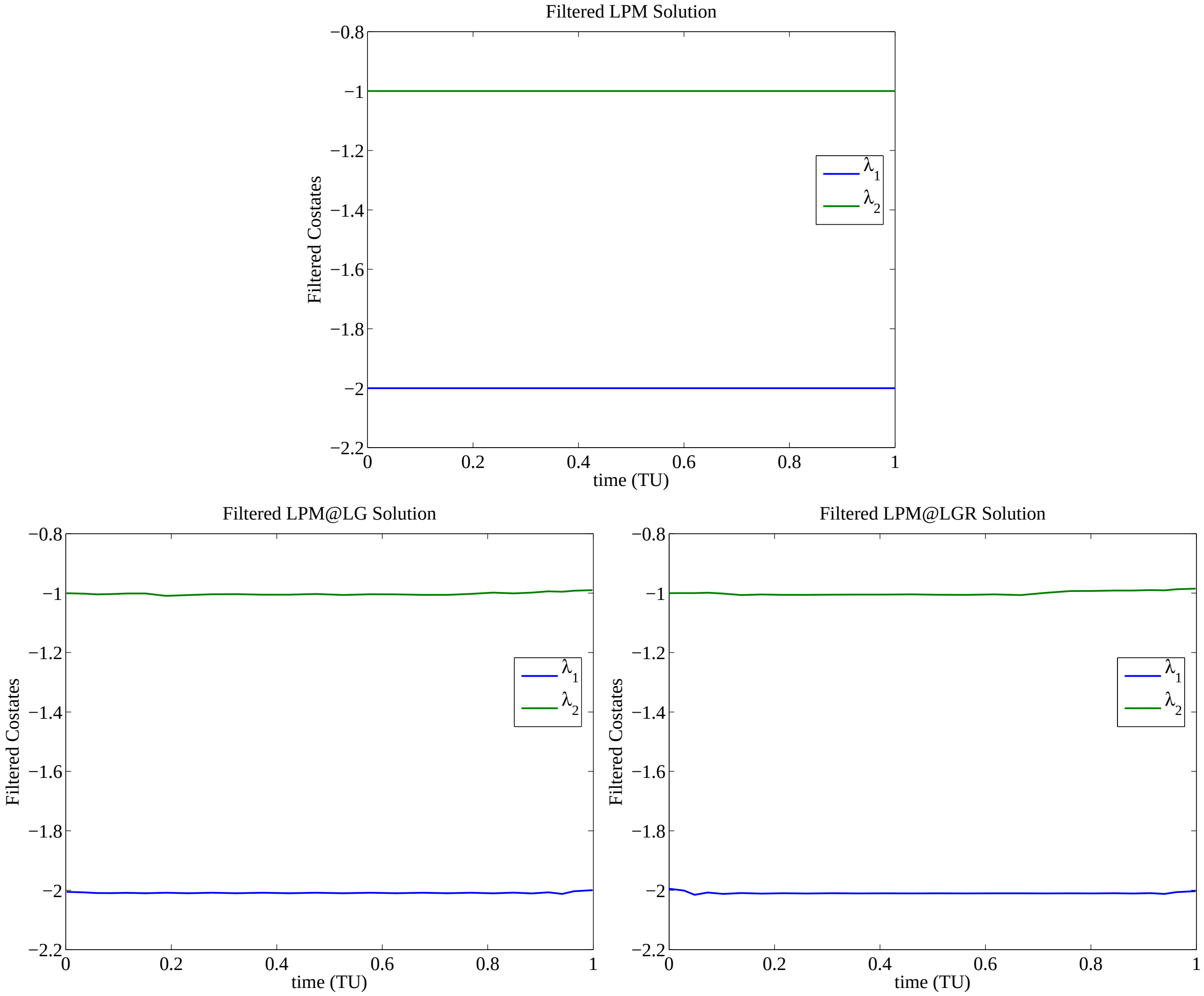
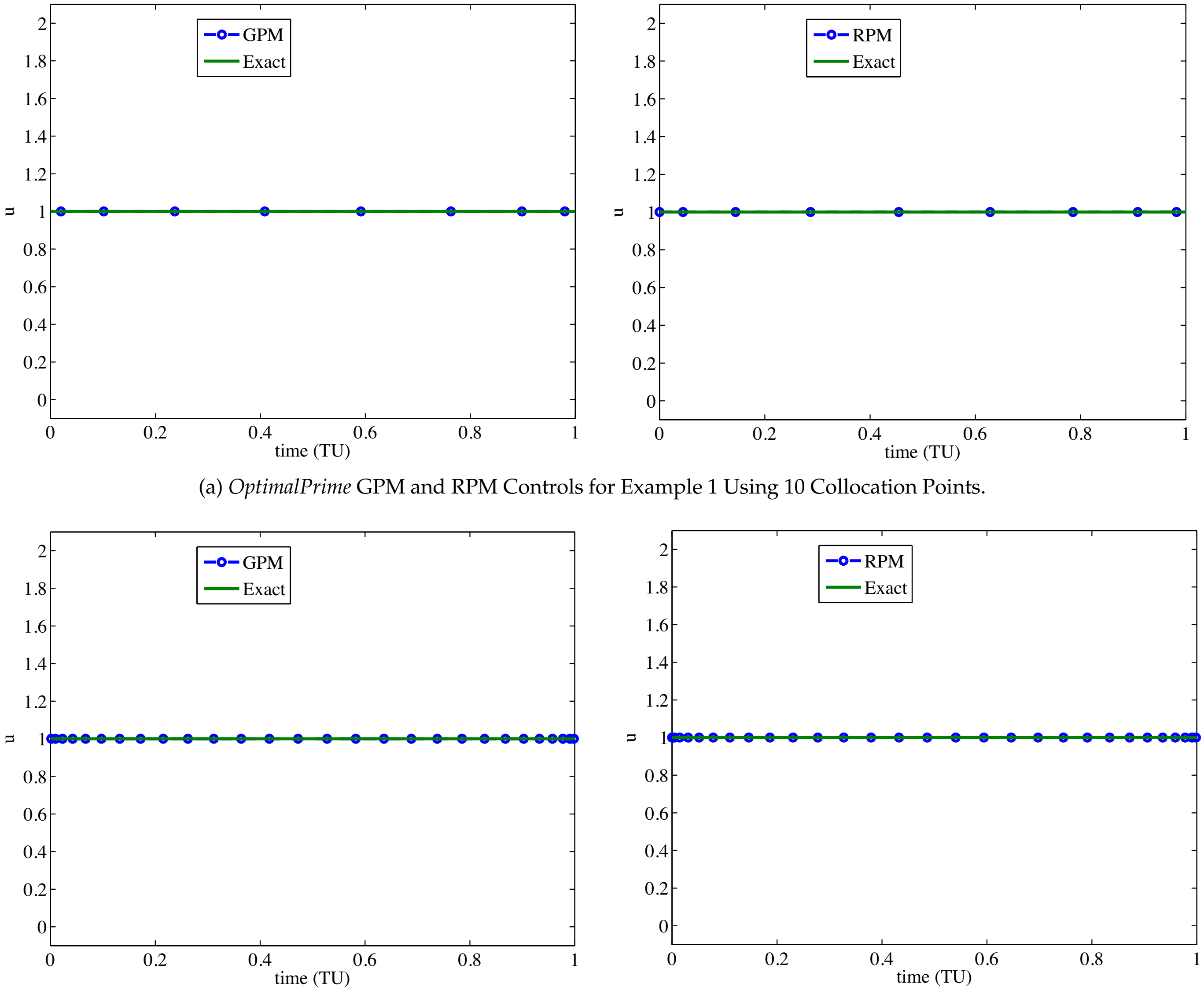


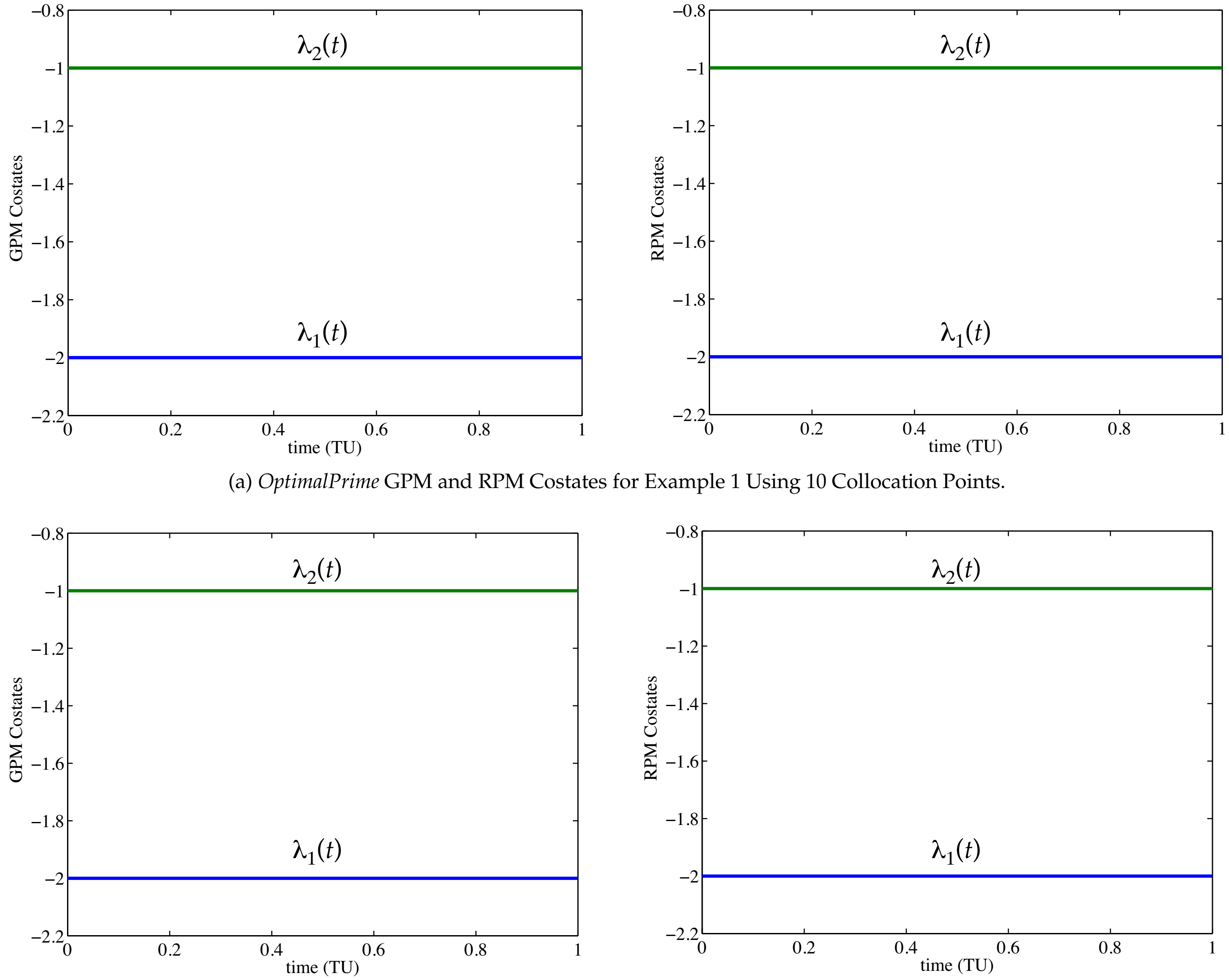
Figure 6: **Filtered** *OptimalPrime* Costate Solutions Using LPM, LPM@LG, and LPM@LGR for Example 1 Using 30 Collocation Points. **Note: These Results are Identical to Those Given in Fig. 5 Which Are Found in Refs. 12 and 13.**



(a) *OptimalPrime* GPM and RPM Controls for Example 1 Using 10 Collocation Points.

(b) *OptimalPrime* GPM and RPM Controls for Example 1 Using 30 Collocation Points.

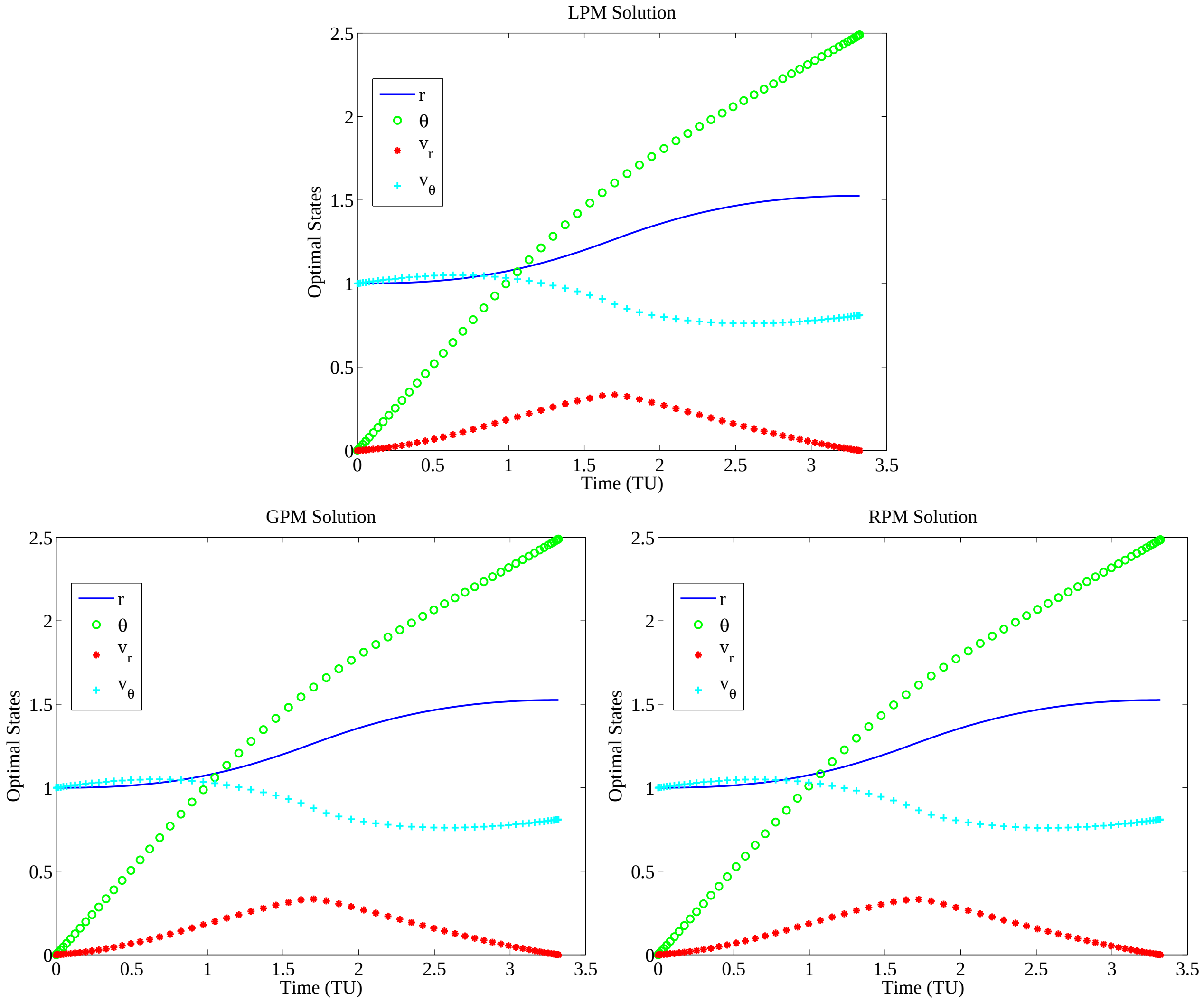
Figure 7: *OptimalPrime* GPM and RPM Controls for Example 1 Using 10 and 30 Collocation Points.



(a) *OptimalPrime* GPM and RPM Costates for Example 1 Using 10 Collocation Points.

(b) *OptimalPrime* GPM and RPM Costates for Example 1 Using 30 Collocation Points.

Figure 8: *OptimalPrime* GPM and RPM Costates for Example 1 Using 10 and 30 Collocation Points.


LPM Solution

GPM Solution

RPM Solution

Figure 9: *OptimalPrime* LPM, GPM, and RPM State Solutions for Example 2 Using 64 Collocation Points.

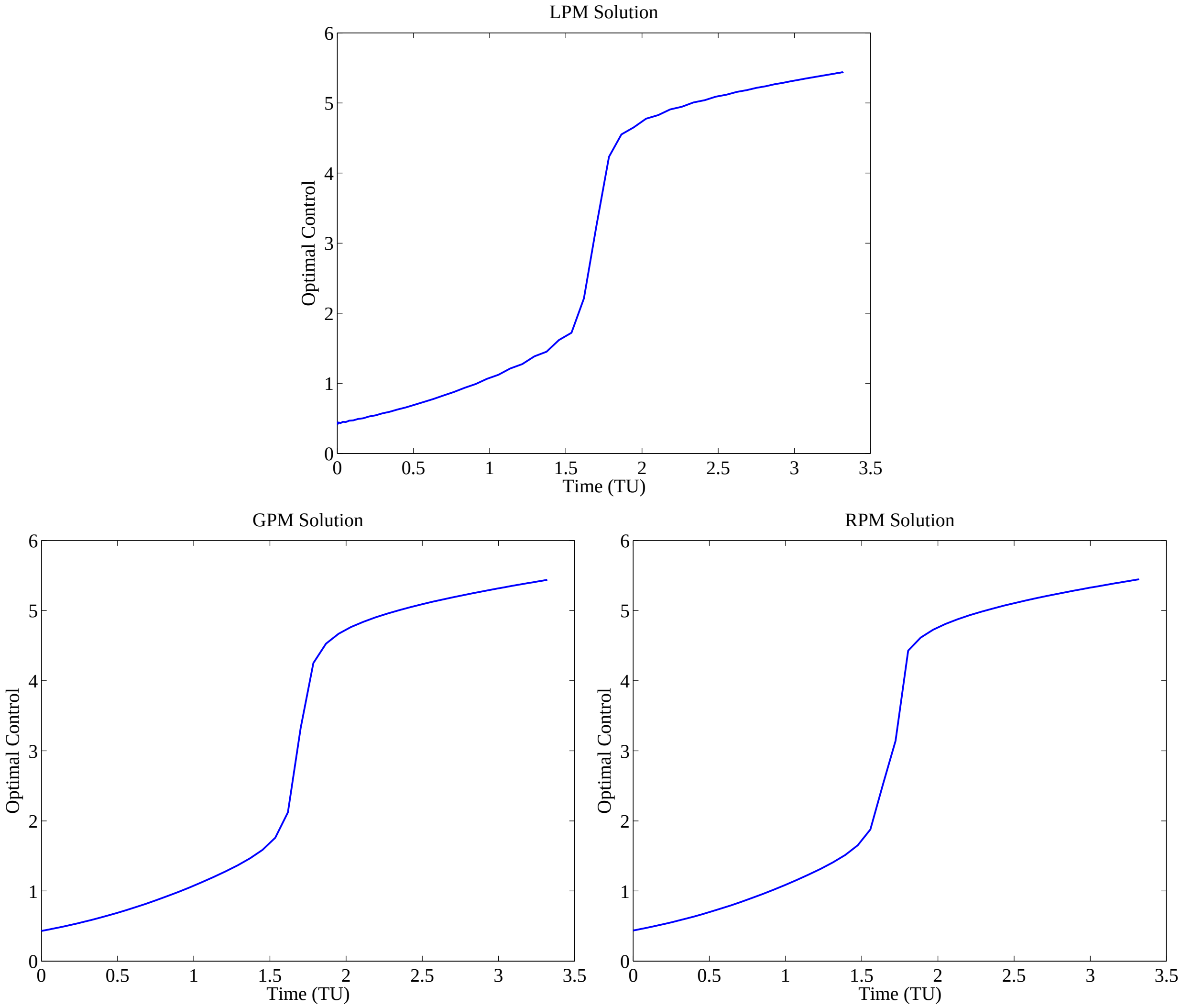


Figure 10: *OptimalPrime* LPM, GPM, and RPM Control Solutions (Modified via an Unwrapping of the NLP Solution) for Example 2 Using 64 Collocation Points.

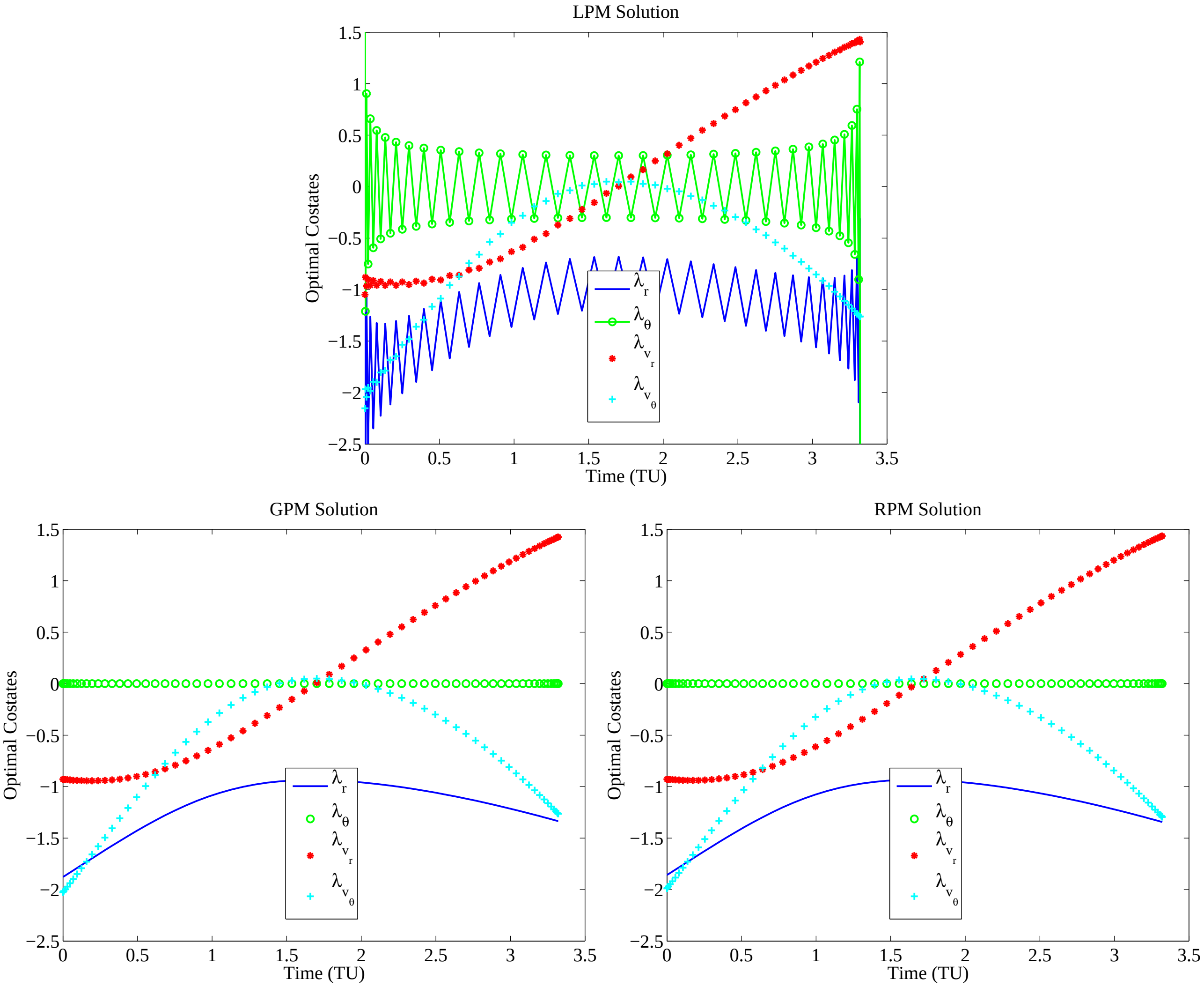


Figure 11: *OptimalPrime* LPM, GPM, and RPM Costate Solutions for Example 2 Using 64 Collocation Points.

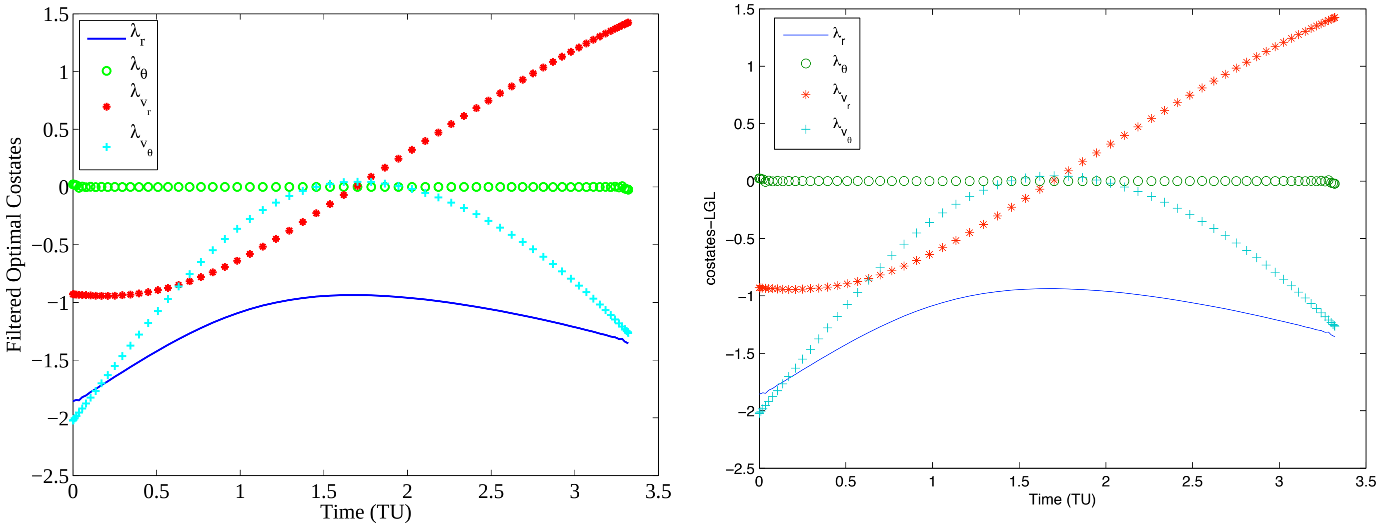


Figure 12: Filtered *OptimalPrime* LPM Costates (Left Plot) Alongside LPM Costates Reprinted from Ref. 12 (Right Plot) for Example 2 Using 64 Collocation Points. **Note: The Results in the *Left* Plot (i.e., The Filtered *OptimalPrime* Costates) are Identical to Those Given in the *Right* Plot where the Right Plot is Taken from Ref. 12.**

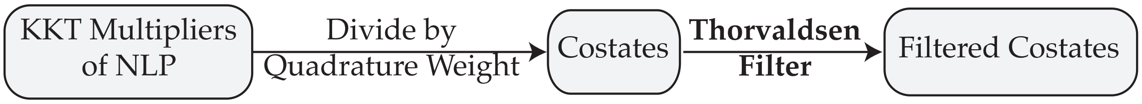


Figure 13: Alternate Two-Step Approach to "Covector Mapping Theorem" to Obtain Costates Using LPM-Based Methods.

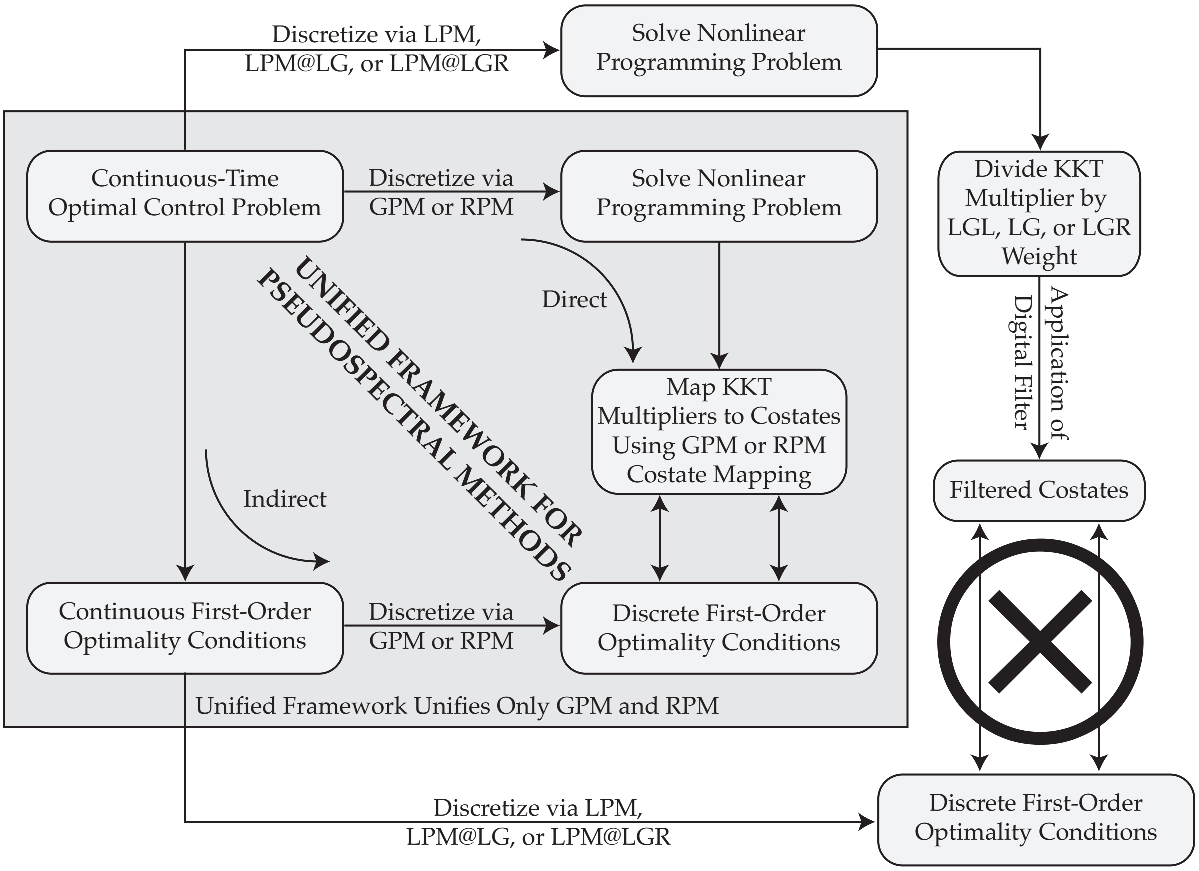


Figure 14: Unified Framework for GPM and RPM Shown Alongside LPM, LPM@LG, and LPM@LGR, Showing That LPM, LPM@LG, and LPM@LGR, Do Not Satisfy Conditions of Unified Framework.

Table 1: Differences Between Collocation Points and Interpolation Points for LPM, LPM@LG, LPM@LGR, GPM, and RPM.

	Collocation Points	Interpolation Points	Discretization Points
LPM	LGL Points: $(\tau_1, \ldots, \tau_N)$	LGL Points: $(\tau_1, \ldots, \tau_N)$	LGL Points: $(\tau_1, \ldots, \tau_N)$
LPM@LG	LG Points: $(\tau_2, \ldots, \tau_{N-1})$	LG Points: $(\tau_2, \ldots, \tau_{N-1})$	LG Points: $(\tau_2, \ldots, \tau_{N-1})$
LPM@LGR	LGR Points: $(\tau_1, \ldots, \tau_{N-1})$	LGR Points: $(\tau_1, \ldots, \tau_{N-1})$	LGR Points: $(\tau_1, \ldots, \tau_{N-1})$
GPM	LG Points: $(\tau_2, \ldots, \tau_{N-1})$	LG Points $\oplus$ Initial Point: $(\tau_1, \ldots, \tau_{N-1})$	LG Points $\oplus$ Both Boundary Points $(\tau_1, \ldots, \tau_N)$
RPM	LGR Points: $(\tau_1, \ldots, \tau_{N-1})$	LGR Points $\oplus$ Terminal Boundary Point: $(\tau_1, \ldots, \tau_N)$	LGR Points $\oplus$ Terminal Boundary Point $(\tau_1, \ldots, \tau_N)$

Table 2: Differences in Sizes of Differentiation Matrices for LPM, LPM@LG, LPM@LGR, GPM, and RPM.

Method	Size of Differentiation Matrix	Matrix Type	Point of Application of Boundary Conditions	Quadrature Condition
LPM@LG	$N \times N$	Square	First and Last LGL Points (τ_1 and τ_N)	No
LPM@LG	$(N-2) \times (N-2)$	Square	First and Last LG Points (τ_2 and τ_{N-1})	No
LPM@LGR	$(N-1) \times (N-1)$	Square	First and Last LGR Points (τ_1 and τ_{N-1})	No
GPM	$(N-2) \times (N-1)$	Non-Square	Boundary Points ($\tau_1 = -1$ and $\tau_N = 1$)	Yes
RPM	$(N-1) \times N$	Non-Square	Boundary Points ($\tau_1 = -1$ and $\tau_N = 1$)	No