Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems

by

Geoffrey Todd Huntington

B. S. Aerospace Engineering, University of California, Los Angeles, 2001

S. M. Aerospace Engineering, Massachusetts Institute of Technology, 2003

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2007

Author ....
Department of Aeronautics and Astronautics
May 11, 2007

Certified by ....
Jonathan P. How, Ph.D., Professor
Department of Aeronautics and Astronautics, MIT
Thesis Committee Co-Chair

Certified by ....
Anil V. Rao, Ph. D., Assistant Professor
Dept. of Mechanical and Aerospace Engineering, University of Florida
Thesis Committee Co-Chair

Accepted by ....
Jaime Peraire, Ph.D., Professor
Department of Aeronautics and Astronautics, MIT
Chair, Committee on Graduate Students

Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems

by

Geoffrey Todd Huntington

Author ....
Department of Aeronautics and Astronautics
May 11, 2007

Certified by ....
Jonathan P. How, Ph.D., Professor
Department of Aeronautics and Astronautics, MIT
Thesis Committee Co-Chair

Certified by ....
Anil V. Rao, Ph. D., Assistant Professor
Dept. of Mechanical and Aerospace Engineering, University of Florida
Thesis Committee Co-Chair

Certified by ....
David W. Miller, Ph.D., Professor
Department of Aeronautics and Astronautics, MIT
Thesis Advisor

Certified by ....
Raymond J. Sedwick, Ph.D., Principal Research Scientist
Department of Aeronautics and Astronautics, MIT
Thesis Advisor

Certified by ....
Ronald J. Proulx, Ph.D., Principal Member of the Technical Staff
Charles Stark Draper Laboratory
Thesis Advisor

Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems

by

Geoffrey Todd Huntington

Submitted to the Department of Aeronautics and Astronautics
on May 11, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in
Aeronautics and Astronautics

Abstract

As optimal control problems become increasingly complex, innovative numerical methods are needed to solve them. Direct transcription methods, and in particular, methods involving orthogonal collocation have become quite popular in several field areas due to their high accuracy in approximating non-analytic solutions with relatively few discretization points. Several of these methods, known as pseudospectral methods in the aerospace engineering community, have also established costate estimation procedures which can be used to verify the optimality of the resulting solution. This work examines three of these pseudospectral methods in detail, specifically the Legendre, Gauss, and Radau pseudospectral methods, in order to assess their accuracy, efficiency, and applicability to optimal control problems of varying complexity. Emphasis is placed on improving the Gauss pseudospectral method, where advancements to the method include a revised pseudospectral transcription for problems with path constraints and differential dynamic constraints, a new algorithm for the computation of the control at the boundaries, and an analysis of a local versus global implementation of the method. The Gauss pseudospectral method is then applied to solve current problems in the area of tetrahedral spacecraft formation flying. These optimal control problems involve multiple finite-burn maneuvers, nonlinear dynamics, and nonlinear inequality path constraints that depend on both the relative and inertial positions of all four spacecraft. Contributions of this thesis include an improved numerical method for solving optimal control problems, an analysis and numerical comparison of several other competitive direct methods, and a greater understanding of the relative motion of tetrahedral formation flight.

Thesis Committee Co-Chair: Jonathan P. How, Ph.D., Professor
Title: Department of Aeronautics and Astronautics, MIT

Thesis Committee Co-Chair: Anil V. Rao, Ph. D., Assistant Professor
Title: Dept. of Mechanical and Aerospace Engineering, University of Florida

[This page intentionally left blank.]

Acknowledgments

For my part I know nothing with any certainty, but the sight of the stars makes me dream...

- V. van Gogh

Looking back on my experience as a doctoral candidate, only now do I realize that a Ph.D. is more about the journey than the end result. In that respect, I feel an even greater sense of accomplishment and growth than I do for the results in this thesis. Part of this was a result of my own maturation and increasing respect for past accomplishments in the field, but it was mostly due to the fact that I had two excellent co-advisors. Prof. Anil Rao, your tireless effort and attention to detail is inspiring and taught me that you should take nothing for granted; to truly know a topic, you must fully understand even the smallest facets of the problem. Prof. Jon How, your ability to cut straight to the root of a problem gave me direction, especially when I needed it most. To the rest of my committee: Prof. Dave Miller, Dr. Ray Sedwick, and Dr. Ron Proulx, your comments both in and out of committee meetings were invaluable and helped me identify what the key research areas were.

In the course of my research, I read piles of papers and books, but there really was no substitute for just picking up the phone and calling the author. To all the fellow academics who helped me understand the vast amount of literature, I thank you. Specifically, I'd like to thank Dr. John Betts, Prof. Larry Biegler, Juan Arrieta-Camacho, Prof. Qi Gong, Prof. Wei Kang, Dr. Mark Milam, Jeremy Schwartz, Prof. William Hager, Prof. Claudio Canuto, Prof. Phillip Gill, Prof. Gamal Elnagar, and Prof. Mohammad Kazemi. I'd especially like to thank my old officemate, Dr. David Benson. I'll never forget all the days of bouncing both balls and ideas off each other!

A large part of what made my doctoral experience so enjoyable was the assurance of funding, compliments of Draper Labs. So to George Schmidt and the rest of the education department at Draper, thank you for sticking with me to the end! As for the Aerospace Controls Lab at MIT, you made sharing a room with 20 people actually enjoyable, and for that I thank you!

Lastly, I have to mention my incredible support network of friends and family. To the IM sports crew, you were my stress relief. To all the taco nite attendees, your thirst for margaritas and homemade taco shells kept me true to my ”work hard, play hard” motto. To my ever-supportive family, thank you for offering whatever was needed all along the way. When I make it to space, it will be because of you. And the best for last: to my lovely bride Katie, the wait is over. I'm coming home.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., and was partially funded by the NASA Goddard Spaceflight Center under Cooperative Agreement NAS5-NCC5-730.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Geoffrey Todd Huntington ....

ASSIGNMENT

Draper Laboratory Report Number T-1583

In consideration of the research opportunity and permission to prepare my thesis by and at The Charles Stark Draper Laboratory, Inc., I hereby assign my copyright to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

Author's Signature Date

[This page intentionally left blank.]

Contents

List of Figures

List of Tables

[This page intentionally left blank.]

Chapter 1

Introduction

Optimal control problems date back to the 17th century with Johann Bernoulli's famous *brachystochrone* problem.[125] Greek for "shortest time", Bernoulli posed this problem to his contemporaries, which attempts to find the path along which an object moves between two points in a vertical plane under its own weight in the shortest time. Several esteemed mathematicians including Wilhelm Gottfried Leibniz, the Marquis de l'Hôpital, Isaac Newton and both Johann and Jakob Bernoulli submitted solutions to the brachystochrone problem, marking the beginnings of optimal control theory.

With over 300 years of research in this area, many significant advancements have been made. Highlights of these advancements include the creation of calculus of variations, first elaborated by Euler in 1733 in the *Elementa Calculi Variatonum*[32] from which it got its name. Also, in the 1950's, Richard Bellman pioneered work in dynamic programming[73] which led to sufficient conditions for optimality using the Hamilton-Jacobi-Bellman equation. Lev Pontryagin's development of the maximum (minimum) principle[82] in 1962 provided a method to determine the optimal control for constrained problems, often resulting in "bang-bang" solutions. The availability of commercial computers in the 1950's fundamentally transformed the field by enabling efficient numerical solutions.[21] Present-day numerical methods for solving optimal control problems are plentiful and vary greatly in their approach and complexity. These methods discretize the continuous-time problem in some fashion and solve the resulting approximate, finite-dimensional problem to a specified tolerance.

Numerical methods for solving optimal control problems fall into two general categories:

indirect methods and direct methods. In an indirect method, first-order necessary conditions for optimality are derived from the optimal control problem via the calculus of variations[73] and Pontryagin's minimum principle.[82] These necessary conditions form a Hamiltonian boundary-value problem (HBVP), which is then solved numerically for extremal trajectories.[3] The optimal solution is then found by choosing the extremal trajectory with the lowest cost. The primary advantages of indirect methods are a high accuracy in the solution and the assurance that the solution satisfies the first-order optimality conditions. However, indirect methods have several disadvantages. First, the HBVP must be derived analytically, which can often be non-trivial. Second, because indirect methods typically have small radii of convergence, an extremely good initial guess of the unknown boundary conditions is generally required. Furthermore, many indirect methods require an accurate initial guess of the costate, which is often non-intuitive and difficult to obtain. Finally for problems with path constraints, it is necessary to have *a priori* knowledge of the constrained and unconstrained arcs or switching structure. BNDSCO[80] is a commercial program for solving multipoint boundary value problems that implements an indirect multiple shooting algorithm.

In a direct method, the continuous-time optimal control problem is transcribed directly to a nonlinear programming problem (NLP) without formulating an alternate set of optimality conditions. The resulting NLP can be solved numerically by well developed algorithms,[10,45] which attempt to satisfy a set of conditions (called Karush-Kuhn-Tucker (KKT) conditions) associated with the NLP. Direct methods have the advantage that the first-order necessary conditions do not need to be derived. Furthermore, they have much larger radii of convergence than indirect methods and thus, do not require as good an initial guess, and do not require an initial guess for the costate. Lastly, the switching structure does not need to be known *a priori*. However, many direct methods have the disadvantage of providing either an inaccurate costate or providing no costate information whatsoever, meaning it is uncertain whether the optimal solution to the NLP is truly an optimal solution to the original optimal control problem. Well-known software packages employing direct methods include *Optimal Trajectories by Implicit Simulation* (OTIS),[119] *Sparse Optimal Control Software* (SOCS),[11] *Graphical Environment for Simulation and Optimization* (GESOP),[122] *Direct Collocation* (DIRCOL),[121] *Nonlinear Trajectory Generation* (NTG),[78,79] and *Direct* *and Indirect Dynamic Optimization* (DIDO).[95]

The category of direct methods is quite broad and encompasses some very different techniques.[64] In particular, the choice of what quantities to discretize and how to approximate the continuous-time dynamics varies widely amongst the different direct methods. Two of the more common types of direct methods are *control parameterization* and *state and control parameterization*[115] techniques. In a control parameterization method,[16,102,109,130] the control alone is approximated and the differential equations are solved via numerical integration. Examples of control parameterization include shooting methods and multiple shooting methods. In state and control parameterization methods,[14,37,56,57,81,107] the state is discretized within the NLP as well as the control, and the continuous-time differential equations are converted into algebraic constraints. These constraints are then imposed in the NLP formulation, which avoid the sensitivity issues of direct shooting methods at the expense of a larger NLP.

In recent years, considerable attention has been brought to a class of state and control parameterization methods called *pseudospectral*[23,33,43] or *orthogonal collocation*[26,116] methods. In a pseudospectral method, a finite basis of global interpolating polynomials is used to approximate the state and control at a set of discretization points. The time derivative of the state in the dynamic equations is approximated by the derivative of the interpolating polynomial and is then constrained to be equal to the vector field of the dynamic equations at a set of collocation points. While any set of unique collocation points can be chosen, generally speaking an *orthogonal collocation* is chosen, i.e., the collocation points are chosen to be the roots of an orthogonal polynomial (or linear combinations of such polynomials and their derivatives). Because pseudospectral methods are generally implemented via orthogonal collocation, the terms *pseudospectral* and *orthogonal collocation* are essentially interchangeable (thus researchers in one field use the term *pseudospectral*[43] while others use the term *orthogonal collocation*[116]). One advantage to pseudospectral methods is that for smooth problems, pseudospectral methods typically have faster convergence rates than other methods, exhibiting so called "spectral accuracy".[112] For non-smooth problems or problems where modeling changes are desired, the optimal control problem can be divided into phases and orthogonal collocation can be applied globally within each phase. A vast amount of work has been done on using pseudospectral methods to solve non-smooth optimal control problems (see Refs. 27, 34, 38, 85, 92, 95).

Pseudospectral methods in optimal control arose from spectral methods which were traditionally used to solve fluid dynamics problems.[23, 43] Meanwhile, seminal work in orthogonal collocation methods for optimal control problems date back to 1979 with the work of Ref. 87 and some of the first work using orthogonal collocation methods in engineering can be found in the chemical engineering literature.[27] More recent work in chemical and aerospace engineering have used collocation at the Legendre-Gauss-Radau (LGR) points,[41, 71, 72] which is termed the *Radau pseudospectral method* (RPM) in this thesis. Within the aerospace engineering community, several well-known pseudospectral methods have been developed for solving optimal control problems such as the *Chebyshev pseudospectral method* (CPM),[117, 118] the *Legendre pseudospectral method* (LPM),[33] and the *Gauss pseudospectral method* (GPM).[8] The CPM uses Chebyshev polynomials to approximate the state and control, and performs orthogonal collocation at the Chebyshev-Gauss-Lobatto (CGL) points. An enhancement to the Chebyshev pseudospectral method that uses a Clenshaw-Curtis quadrature was developed in Ref. 40. The LPM uses Lagrange polynomials for the approximations, and Legendre-Gauss-Lobatto (LGL) points for the orthogonal collocation. A costate estimation procedure for the Legendre pseudospectral method was developed by in Ref. 39 and recently updated in Ref. 49. Recent work by Williams shows several variants of the standard LPM. The Jacobi pseudospectral method[126] is a more general pseudospectral approach that uses Jacobi polynomials to find the collocation points, of which Legendre polynomials are a subset. Another variant, called the Hermite-LGL method,[128] uses piecewise cubic polynomials rather than Lagrange polynomials, and collocates at a subset of the LGL points. These new methods have not yet become mainstream optimization techniques because of their novelty, but the new software tool DIRECT[127] could expedite this process. Lastly, in the Gauss pseudospectral method, the state is approximated using a basis of Lagrange polynomials similar to the LPM, and the optimal control problem is orthogonally collocated at the interior Legendre-Gauss (LG) points. Much of the analysis done in this thesis centers on pseudospectral methods and their accuracy and efficiency in numerically solving optimal control problems.

1.1 Motivation for Research

The research presented in this thesis is motivated by fundamental questions in two different areas.

1.1.1 Advancement of Numerical Methods for Optimal Control Problems

The Gauss pseudospectral method in its current form is one of the newest numerical approaches in the literature today,[8] although it bears resemblance to work done in 1979.[87] The current method was originally developed for problems with integral dynamic equations, and was then adapted to accommodate differential dynamic equations. The critical contribution by Benson, which distinguishes the GPM among the other pseudospectral methods, was to show that for an optimal control problem with either integral or differential dynamics, the Karush-Kuhn-Tucker (KKT) conditions of the NLP are exactly equivalent to the discretized form of the first-order optimality conditions of the HBVP. Hence, a solution to the NLP also satisfies the optimality conditions traditionally used in *indirect* methods, thereby eliminating one of the primary disadvantages of direct methods. Benson also showed that this equivalence for the integral formulation existed in the presence of path constraints, but did not explicitly show equivalence for problems with both path constraints and the more common differential dynamic constraints. Therefore the first motivation for this thesis is to complete the equivalence proof for the general formulation of the optimal control problem involving both differential dynamic constraints and path constraints.

Costate Mapping using GPM

When analyzing the solution to optimal control problems, it is desirable for the solution to include not only state and control approximations, but a costate approximation as well. The ability to obtain accurate costate estimates is useful for verifying the optimality of solutions, determining the sensitivity of the state with respect to cost, and performing mesh refinement.[14] Indirect methods naturally include the costate in their problem formulation, so any indirect solution contains a costate estimate. However, direct methods do not explicitly approximate the costate since the HBVP equations are not formulated. Despite this, a costate estimate can be determined for many direct methods in several ways. Some of these estimates are based on solving an approximation to the costate dynamics in post-processing.[58,76] Other estimates are based on relationships between the KKT multipliers of the NLP and the continuous costate found by a sensitivity analysis,[103] or relating the KKT conditions of the NLP to the continuous costate dynamics.[55,107] Or, for some pseudospectral methods, the KKT multipliers can be algebraically mapped to the discrete costate via a simple computation. Costate mapping procedures have been documented in Ref. 71 for the RPM, Refs. 39, 49 for the LPM, and Ref. 8 for the GPM. Although an exact mapping between the KKT multipliers and the discrete costate has not been proven for the RPM and LPM, research has been done on these pseudospectral methods that discusses the (rapid) convergence rates of the NLP solution to the solution of the continuous-time optimality conditions.[48,49,71] Benson developed a costate mapping procedure for the GPM and proved it was an *exact* map due to the equivalence between the KKT conditions and HBVP optimality conditions. The aforementioned equivalence in the GPM brings about a certain mathematical elegance that can possibly contain some undiscovered properties. Another motivation for this research is to simply explore what additional benefits can be gleaned from the costate mapping.

Analysis and Comparison of Various Pseudospectral Methods

With so many different approaches to solve optimal control problems, it is clear that the academic community is fragmented on which approach may be the best for a general problem. And it is likely that no one method works the best on all problems. However, very few papers have been published that compare the performance of various methods. Survey papers, such as Ref. 12, stop short of providing comparisons between various methods. Numerical comparisons between methods are often difficult to perform fairly because many direct methods are integrated with a specific NLP solver, thereby making it difficult to distinguish between differences between the methods and differences due to the NLP solvers. However, some research has been done that compares direct methods. Ref. 81 is primarily focused on presenting enhanced mesh refinement and scaling strategies, but contains a brief comparison of the accuracy of direct implicit integration schemes (like Hermite-Simpson integration) and pseudospectral methods. Williams created the software tool DIRECT in order to more effectively compare various trajectory optimization algorithms. The results of a thorough comparison involving 21 test problems of several different direct methods is presented in Ref. 127. Ref. 42 presents a comparison between several pseudospectral approaches, but uses unorthodox variants of pseudospectral methods that are inconsistent with a large majority of the literature. Based on this very limited body of research, there exists a greater need to examine the accuracy and efficiency of these direct methods from a numerical and theoretical standpoint.

Analysis and Comparison of Local versus Global Orthogonal Collocation Methods

Interestingly, when the Gauss pseudospectral method was first introduced to the academic community, it received quite a bit of resistance and skepticism. Critics noted that historically, *local* methods, i.e., those that approximate the dynamics locally using piecewise polynomials, have been prosperous due to their property of *local support*,[14] which bounds the maximum error that can be generated locally along the trajectory. Consequently one can determine the maximum error between the exact optimal solution and the computed solution for the entire trajectory. While local methods have a long history in solving optimal control problems, recent research in pseudospectral methods suggests that local collocation may neither be the most accurate nor the most computationally efficient approach.[8,48,66,68,89] Instead, recent work has shown great success in the application of *global* collocation (i.e., collocation using a global polynomial across the entire time interval as the basis for approximation). By using appropriate discretization points and interpolating polynomials, one can also bound the maximum error generated by the global polynomial approximation, shown in Chapter 2. In fact, Ref. 81 specifically compares global pseudospectral methods to other local direct collocation methods, and also experiments with the order of the local collocation methods (i.e., the number of collocation points per segment) and its effect on the solution accuracy. In light of the recent results that promote global orthogonal collocation and the long history of the use local collocation, it is important to gain a better understanding as to how these two different philosophies work in practice.

Computation of the Boundary Control Using the Gauss Pseudospectral Method

Other critics have highlighted the fact that since the Gauss pseudospectral method performs collocation at the interior LG points, there is no explicit value for the control at the boundary points. Consequently, the value for the boundary control can be quite arbitrary. Moreover, traditional extrapolation techniques (such as spline extrapolation[14] or Lagrange extrapolation[71]) may violate control path constraints and certainly will be sub-optimal. Further motivation for this research lies in the need to improve the accuracy of the boundary control for the Gauss pseudospectral method.

1.1.2 Optimization of Formation Flying Maneuvers

Spacecraft formation flying is defined as a set of more than one spacecraft whose states are coupled through a common control law.[101] Formation flying has been identified as an enabling technology for many future space missions.[100] In particular, space missions using multiple spacecraft, as compared with using a single spacecraft, allow simultaneous measurements to be taken at specific relative locations, thereby improving science return. An important aspect that is critical to the successful implementation of formation flying missions is trajectory design (also called path planning or guidance[100]). An excellent survey of methods used to design formation flying trajectories can be found in Ref. 100, which also provides an extensive list of references on formation flying guidance. Formation flying trajectory design has two main categories: *stationkeeping*, i.e., to maintain a relative spacecraft formation for a specified portion of the trajectory and *reconfiguration*, i.e., to maneuver a spacecraft formation from one configuration (involving the formation geometry, relative motion, and/or orientation) to a second configuration.

A particular class of Earth-orbit formations that have been studied extensively are those involving four spacecraft.[17,25,47,53,62,66,67] Four-spacecraft formations are desirable because they use relatively few spacecraft yet are still capable of taking measurements in three-dimensions at relatively large inter-spacecraft distances (e.g., a spacing of several kilometers). However, because of complex mission constraints, it is often difficult to determine feasible trajectories and controls. Because of the already difficult task of determining feasible solutions, it is even more of a challenge to determine solutions that minimize a specified performance metric (e.g., trajectories that minimize fuel). Fuel-optimal solutions (if one can be found) are often non-intuitive, but are very worthwhile as they can lead to significant fuel savings which can result in a longer mission duration or increased payload capacity.

Several papers have attempted to solve tetrahedral trajectory optimization problems similar to the ones considered in this research. Ref. 75 uses a two-step approach that begins with genetic algorithms (a global heuristic optimization method) and afterwards refines the problem using Lawden's primer vector theory.[5] Ref. 54 also uses a two-step approach, but combines a direct SQP method with a genetic algorithm. Ref. 113 employs a three-step approach that uses simulated annealing (another global heuristic method) as its initial step. In each of these previous works, the problem is separated into steps where, in general, one step optimizes the orbital transfer portion of the trajectory and the other step optimizes the relative position constraints in a sequential but separate optimization procedure that uses the trajectory from the first step. While dividing the problem into parts and optimizing each part separately makes the problem more tractable, it also reduces the solution search space. In this work the entire problem is formulated as a *single unified* numerical optimization procedure. Other papers use a single numerical optimization procedure, but necessitate significant simplifying assumptions on either the dynamic model[111] or the search space[60] in order to make the problem more manageable. Lastly, Refs. 18, 19 use linearized dynamics, but incorporate them into an MPC format, exhibiting a capability for real-time optimization of formation flying maneuvers.

The formation flying problems posed in this thesis are based on the proposed NASA Multi-scale Magnetospheric (MMS) Mission, which will attempt to make fundamental advancements in the understanding of the Earths magnetosphere and its dynamic interaction with solar wind by measuring magnetic and electric fields.[62,63] The mission intends to have a four-spacecraft formation that forms a tetrahedral geometry when taking science data. Very little is known about the characteristics of the fuel-optimal maneuvers for this type of mission, so this research is primarily interested in gaining a better perspective on the location, number, and duration of maneuvers needed to optimally reconfigure the spacecraft between different configurations while simultaneously satisfying certain nonlinear geometric path constraints along some region of the orbit.

1.2 Contributions & Thesis Summary

This section briefly describes the contents of the chapters in this thesis, and specifically highlights the significant contributions.

In order to better understand the theoretical contributions in this thesis, there are several mathematical concepts that are worth describing in detail within the thesis. Chapter 2 not only presents this material in a concise manner, but also discusses the rationale and advantages for using these mathematical constructs.

Chapter 3 presents the Gauss pseudospectral method in its most current form. This has been slightly modified from the original formulation in Ref. 6 in order to simplify the notation, and provide a more complete NLP solution, which includes both path constraints and differential dynamics in the optimal control problem formulation. Next, the equivalence between the KKT conditions and the HBVP first-order optimality conditions is proved for this new formulation, and a corresponding costate mapping theorem is derived. Lastly, this chapter provides a discussion on past and present attempts to prove the convergence of the NLP solution towards the exact optimal solution as the number of discretization points increases towards infinity.

Chapter 4 directly addresses some of the critics' concerns that the Gauss pseudospectral method is an ill-conceieved approach because it does not provide an explicit value for the boundary control from the NLP. Traditional methods involving extrapolation are presented, along with a new procedure for computing a highly accurate boundary control from the NLP solution. These approaches are compared on several example problems including a complex multi-stage launch vehicle problem.

Chapter 5 directly addresses some of the other critics' concerns regarding local versus global approaches to optimal control problems, as mentioned earlier in this chapter. This chapter outlines the rationale and procedure for each approach and references the current work in both approaches. The comparison is conducted using the Gauss pseudospectral method, which is implemented in both a local and global fashion. The comparison is made on two example problems that have characteristics that might suggest the use of a local approach. The computational accuracy and efficiency are analyzed for both examples.

In Chapter 6, a comparison is made between three commonly used pseudospectral methods: the Legendre, Radau, and Gauss pseudospectral methods. In order to provide a fair comparison, in this study the NLP solver SNOPT[45] is used for each of the discretizations. Furthermore, all three methods are implemented using the same version of MATLAB® and the initial guesses provided for all examples are identical. Using this equivalent setup for all three methods, the goal of the study is to assess the similarities and differences in the accuracy and and computational performance between the three methods. Three examples are used to make the comparison. The first two examples are designed to be sufficiently simple so that the key features of each method can be identified and analyzed. The third example, a commonly used problem in aerospace engineering, is designed to provide an assessment as to how the three methods compare on a more realistic problem. A great deal of the emphasis of this study is to understand when one method may perform better than another method and to identify why such an improved performance is attained in such circumstances.

Chapter 7 considers several spacecraft reconfiguration problems. The first problem is what's typically known as an *orbit insertion* problem or initialization problem,[75] meaning the four spacecraft must optimally maneuver from an initial parking orbit to the desired mission orbit and satisfy certain formation configuration constraints upon reaching the mission orbit. As formation flying missions continue throughout their mission lifetime, disturbances will cause the spacecraft to naturally drift apart from one another, thus creating a "degraded" formation. The second scenario addresses this issue by examining the problem of optimally reconfiguring a four-spacecraft formation from an initial degraded formation to a formation that satisfies a set of mission constraints along a region of the orbit (representing the scientific area of interest for the mission). Both the orbit insertion problem and the reconfiguration problem are posed as optimal control problems. These complex optimal control problems are highly nonlinear and have no analytic solutions. Consequently, they are transcribed into an NLP using the Gauss pseudospectral method. A greater understanding for the relative motion of spacecraft formations, the location of minimum-fuel maneuvers, and minimum-fuel tetrahedral geometries is developed in this chapter.

Finally, Chapter 8 summarizes the significant contributions of this thesis and suggests potential future research directions.

Chapter 2

Mathematical Background

This chapter introduces several of the theoretical and mathematical concepts used in this thesis. Many of the advancements in this thesis are founded on optimal control theory, and a significant portion of this chapter explains the fundamentals of optimal control. This work also largely relies on numerical approximations, and several approaches to function approximation and quadrature approximation are described in detail.

2.1 Optimal Control

As explained concisely by Kirk in Ref. 73, the objective of an optimal control problem is *to determine the control signals that will cause a process to satisfy the physical, geometric, or design constraints and at the same time minimize (or maximize) some performance criterion.* This section describes the optimal control problem in a more mathematical framework, and discusses common approaches to solving optimal problems.

2.1.1 Continuous Bolza Problem

Consider the following general optimal control problem. Determine the control, $\mathbf{u}(t) \in \mathbb{R}^m$, that minimizes the Bolza cost functional

$$J = \Phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)dt \tag{2.1}$$

involving the state, $\mathbf{x}(t) \in \mathbb{R}^n$, the initial time, t_0, and (free or fixed) final time, t_f, subject to the dynamic constraint

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \qquad t \in [t_0, t_f] \tag{2.2}$$

the boundary condition

$$\boldsymbol{\phi}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = \mathbf{0} \tag{2.3}$$

and inequality path constraint

$$\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{0}, \qquad t \in [t_0, t_f] \tag{2.4}$$

In Eqs. (2.1)-(2.4), the functions $\mathbf{\Phi}$, g, $\mathbf{f}$, $\boldsymbol{\phi}$, and $\mathbf{C}$ are defined as follows:

$$\begin{aligned}
\Phi &: \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R} \\
g &: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R} \\
\mathbf{f} &: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n \\
\boldsymbol{\phi} &: \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^q \\
\mathbf{C} &: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^c
\end{aligned} \tag{2.5}$$

The problem of Eqs. (2.1)-(2.4) is referred to as the continuous Bolza problem.

The Bolza problem is defined on the time interval $t \in [t_0, t_f]$. Certain numerical techniques (like pseudospectral methods) require a fixed time interval, such as $[-1, 1]$. The independent variable can be mapped to the general interval $\tau \in [-1, 1]$ via the affine transformation

$$\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0} \tag{2.6}$$

Note that this mapping is still valid with free initial and final times. Using Eq. (2.6), the Bolza problem can be redefined as follows. Minimize the cost functional

$$J = \Phi(\mathbf{x}(\tau_0), t_0, \mathbf{x}(\tau_f), t_f) + \frac{t_f - t_0}{2} \int_{\tau_0}^{\tau_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) d\tau \tag{2.7}$$

subject to the constraints

$$\frac{d\mathbf{x}}{d\tau} = \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \tag{2.8}$$

$$\boldsymbol{\phi}(\mathbf{x}(\tau_0), t_0, \mathbf{x}(\tau_f), t_f) = \mathbf{0} \tag{2.9}$$

$$\mathbf{C}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \leq \mathbf{0} \tag{2.10}$$

The problem of Eqs. (2.7)–(2.10) is referred to as the *transformed continuous Bolza problem.*

2.1.2 Indirect Approach

The transformed Bolza problem of Eqs. (2.7)–(2.10) historically has been solved using a branch of mathematics called *calculus of variations* to obtain a set of first-order necessary conditions for optimality.[20,73,74] A solution to the optimality conditions is called an extremal solution, and second-order conditions can be checked to ensure that the extremal solution is a minimum. The fundamental theorem of the calculus of variations is[73]

If $\mathbf{x}^*$ is an extremal, the variation of the cost, J, must vanish on $\mathbf{x}^*$; that is,

$$\delta J(\mathbf{x}^*, \delta\mathbf{x}) = 0, \quad \forall \text{ admissible } \delta\mathbf{x} \tag{2.11}$$

The first-order necessary conditions are found by taking the first-order variation of the *augmented* cost functional, which is created by adjoining the constraints to the cost functional via adjoint variables (i.e., costate, $\boldsymbol{\lambda}(\tau) \in \Re^n$) and Lagrange multipliers, $\boldsymbol{\nu} \in \Re^q$ and $\boldsymbol{\mu}(\tau) \in \Re^c$ as

$$\begin{aligned} J_a = \; & \Phi(\mathbf{x}(\tau_0), t_0, \mathbf{x}(\tau_f), t_f) - \boldsymbol{\nu}^T \boldsymbol{\phi}(\mathbf{x}(\tau_0), t_0, \mathbf{x}(\tau_f), t_f) + \frac{t_f - t_0}{2} \int_{-1}^{1} \Big[g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \\ & -\boldsymbol{\lambda}^T(t) \left(\frac{d\mathbf{x}}{d\tau} - \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \right) - \boldsymbol{\mu}^T(\tau)\mathbf{C}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f) \Big] d\tau \end{aligned} \tag{2.12}$$

The variation with respect to each free variable is then set to zero, as in Eq. (2.11) and results in a set of first-order necessary conditions for optimality. This approach is commonly called "indirect", as it indirectly solves the original problem by formulating and solving this alternate set of optimality conditions. Often, the first-order optimality conditions are simplified by defining an augmented Hamiltonian functional, $\mathcal{H}$, as

$$\mathcal{H}(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu},\mathbf{u},\tau;t_0,t_f) = g(\mathbf{x},\mathbf{u},\tau;t_0,t_f) + \boldsymbol{\lambda}^T\mathbf{f}(\mathbf{x},\mathbf{u},\tau;t_0,t_f) - \boldsymbol{\mu}^T\mathbf{C}(\mathbf{x},\mathbf{u},\tau;t_0,t_f) \tag{2.13}$$

which includes the costate, $\boldsymbol{\lambda}(\tau) \in \mathbb{R}^n$, and Lagrange multiplier function associated with the path constraint, $\boldsymbol{\mu}(\tau) \in \mathbb{R}^c$. For brevity, the explicit dependence on time, τ, for the state, control, costate, and Lagrange multiplier has been dropped. The first-order optimality conditions are also referred to as the Hamiltonian boundary value problem (HBVP):

$$\begin{aligned}
\frac{d\mathbf{x}^T}{d\tau} &= \frac{t_f - t_0}{2}\mathbf{f}^T(\mathbf{x},\mathbf{u},\tau;t_0,t_f) = \frac{t_f - t_0}{2}\frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}} \\
\frac{d\boldsymbol{\lambda}^T}{d\tau} &= \frac{t_f - t_0}{2}\left(-\frac{\partial g}{\partial \mathbf{x}} - \boldsymbol{\lambda}^T\frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \boldsymbol{\mu}^T\frac{\partial \mathbf{C}}{\partial \mathbf{x}}\right) = -\frac{t_f - t_0}{2}\frac{\partial \mathcal{H}}{\partial \mathbf{x}} \\
\mathbf{0}^T &= \frac{\partial g}{\partial \mathbf{u}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{f}}{\partial \mathbf{u}} - \boldsymbol{\mu}^T\frac{\partial \mathbf{C}}{\partial \mathbf{u}} = \frac{\partial \mathcal{H}}{\partial \mathbf{u}} \\
&\boldsymbol{\phi}(\mathbf{x}(\tau_0),t_0,\mathbf{x}(\tau_f),t_f) = \mathbf{0} \\
\boldsymbol{\lambda}(\tau_0)^T &= -\frac{\partial \Phi}{\partial \mathbf{x}(\tau_0)} + \boldsymbol{\nu}^T\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(\tau_0)} \\
\boldsymbol{\lambda}(\tau_f)^T &= \frac{\partial \Phi}{\partial \mathbf{x}(\tau_f)} - \boldsymbol{\nu}^T\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}(\tau_f)} \\
\mathcal{H}(t_0) &= \frac{\partial \Phi}{\partial t_0} - \boldsymbol{\nu}^T\frac{\partial \boldsymbol{\phi}}{\partial t_0} \\
\mathcal{H}(t_f) &= -\frac{\partial \Phi}{\partial t_f} + \boldsymbol{\nu}^T\frac{\partial \boldsymbol{\phi}}{\partial t_f} \\
\mu_j(\tau) = 0\,,\ \text{when}\quad &C_j(\mathbf{x},\mathbf{u},\tau;t_0,t_f) < 0\,,\quad j = 1,\ldots,c \\
\mu_j(\tau) \le 0\,,\ \text{when}\quad &C_j(\mathbf{x},\mathbf{u},\tau;t_0,t_f) = 0\,,\quad j = 1,\ldots,c
\end{aligned} \tag{2.14}$$

where $\boldsymbol{\nu} \in \mathbb{R}^q$ is Lagrange multiplier associated with the boundary condition $\boldsymbol{\phi}$. For some problems, the control cannot be uniquely determined, either implicitly or explicitly, from these optimality conditions. In such cases, the weak form of Pontryagin's minimum principle can be used which solves for the permissible control that globallly minimizes the augmented Hamiltonian in Eq. (2.13). If $\mathcal{U}$ is the set of permissible controls, then Pontryagin's minimum principle[73] states that the optimal control, $\mathbf{u}^* \in \mathcal{U}$, satisfies the following:

$$\mathcal{H}(\mathbf{x}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \tau; t_0, t_f) \leq \mathcal{H}(\mathbf{x}^*, \mathbf{u}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \tau; t_0, t_f), \quad \forall \mathbf{u} \in \mathcal{U}, \quad \tau \in [-1, 1] \tag{2.15}$$

Note that the equations described in this section constitute a set of necessary conditions for optimality, but they are not sufficient conditions. A second order sufficiency check can be implemented to confirm that the extremal solution is the desired minimum or maximum.

The necessary conditions for optimality involve not only the state and control, but the costate as well, emphasizing the importance of the costate in an extremal solution. Without a costate, one cannot check the necessary and sufficient conditions to ensure that an optimal solution has been found. Furthermore, the costate provides information regarding the sensitivity of the minimum cost with respect to the corresponding extremal state.[73] Specifically, let $\delta J^*(\mathbf{x}^*(\tau), \tau, \delta\mathbf{x}(\tau))$ denote the first-order approximation to the change in the minimum cost that results when the state at time τ deviates from $\mathbf{x}^*(\tau)$ by an amount $\delta\mathbf{x}(\tau)$. Then

$$\delta J^*(\mathbf{x}^*(\tau), \tau, \delta\mathbf{x}(\tau)) = \boldsymbol{\lambda}^{*T}(\tau)\delta\mathbf{x}(\tau) \tag{2.16}$$

Thus the sensitivities represented by the costate are often used to redesign the original optimal control problem if the constraints on the problem are flexible.

2.1.3 Direct Approach

The optimality conditions of Eq. (2.14) are often not trivial to formulate. Furthermore, numerical methods that solve these equations generally require an accurate guess for the costate, which is often non-intuitive. For these reasons and others suggested in Chapter 1, *direct methods* have become a very popular alternative to indirect methods in recent years. Rather than formulate a set of optimality conditions, direct methods *transcribe* or convert the infinite-dimensional optimal control problem into a finite-dimensional optimization problem with algebraic constraints, also known as a nonlinear program (NLP).

As explained by Betts in Ref. 14, a direct transcription method has three fundamental steps:

1. convert the dynamic system into a problem with a *finite* set of variables and algebraic constraints, then
2. solve the finite-dimensional problem using a parameter optimization method, then
3. assess the accuracy of the finite-dimensional approximation and if necessary repeat the transcription and optimization steps.

Some direct methods discretize only the control, and propagate the dynamics across the interval using the control approximation. Any additional constraints are checked to ensure feasibility, and along with the cost, help determine the search direction for NLP. These methods are called "shooting methods", named after the early application of aiming a cannon such that the cannonball hit its target.[14] Other methods discretize both the state and control, and are hence called *state and control parameterization methods.* These methods are further subdivided into *local* and *global* methods. Local methods break the dynamics into subintervals at the points $t_0 \leq t_1, \ldots, t_i, \ldots, t_N \leq t_f$, and attempt to find the state and control that satisfy

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \int_{t_i}^{t_{i+1}} \mathbf{f}(\mathbf{x}, \mathbf{u}, t)dt \tag{2.17}$$

where the integral is then replaced with some quadrature approximation:

$$\int_{t_i}^{t_{i+1}} \mathbf{f}(\mathbf{x}, \mathbf{u}, t)dt \approx h_i \sum_{j=1}^{K} \beta_j \mathbf{f}(\mathbf{x}_j, \mathbf{u}_j, \tau_j), \quad t_i \leq \tau_j \leq t_{i+1} \tag{2.18}$$

Euler, Runge-Kutta, Trapezoidal, and Hermite-Simpson methods are all popular *one-step* local methods. Adams schemes, such as Adams-Moulton and Adams-Bashforth schemes are multi-step methods, meaning their integration steps involve more than just $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. Multi-step methods have the general form:

$$\mathbf{x}_{i+k} = \sum_{j=0}^{k-1} \alpha_j \mathbf{x}_{i+j} + h \sum_{j=0}^{k} \beta_j \mathbf{f}_{i+j} \tag{2.19}$$

where α_j and β_j are known constants. Global schemes go even further than multi-step schemes in that they span the entire problem interval. Global, or *pseudospectral* methods are extensively used in the numerical solution to partial differential equations, with the most common types of methods being Galerkin, Tau, and Collocation methods.[23] Pseudospectral methods, as explained by Fornberg in Ref. 43, approximate the solution, $\mathbf{x}(t)$, by a finite sum, $\mathbf{X}(t) = \sum_{k=1}^{M} a_k\phi_k(t)$. The two main questions that arise are then:

1. from which function class should $\phi_k(t),\ k = 1, \ldots, M$ be chosen, and

2. how should the expansion coefficients a_k be determined.

The $\phi_k(t),\ k = 1, \ldots, M$ are called *trial* functions[23] (also called expansion or approximating functions), and are the basis for the truncated series expansion of the solution. These trial functions are most commonly trigonometric functions or orthogonal polynomials such as Legendre polynomials. The a_k's are determined from *test* functions, which try to to ensure that the differential equations are satisfied as closely as possible. Tau, Galerkin, and Collocation methods each use different test functions to determine the expansion coefficients. In a Tau method, the expansion coefficients are selected so that the boundary conditions are satisfied and the residual, $\mathbf{R}_M(t) = \dot{\mathbf{X}}(t) - \mathbf{f}(\mathbf{X}(t), \mathbf{U}(t), t)$ is orthogonal to the basis functions. In other words, the inner product between the residual and the basis functions is zero, as

$$< \mathbf{R}_M(t), \phi_i(t) > \equiv \int_{t_0}^{t_f} \mathbf{R}_M(t)\phi_i(t)dt = 0, \quad \forall i = 1, \ldots, M \tag{2.20}$$

In a Galerkin method, the original basis functions are combined into a new set, $\tilde{\phi}_i(t),\ i = 1, \ldots, M$, in which all the functions satisfy the boundary conditions. The expansion coefficients are those in which the residual is orthogonal to the new basis functions, or

$$< \mathbf{R}_M(t), \tilde{\phi}_i(t) > = 0, \quad \forall i = 1, \ldots, M \tag{2.21}$$

Finally, in a collocation method, the test functions are the Dirac delta functions. Rather than requiring the residual to be orthogonal to the basis functions, the residual must equal zero at a suitably chosen set of *collocation* points, as

$$\mathbf{R}_M(t_k) = \mathbf{0}, \quad \forall k = 1, \ldots, M \tag{2.22}$$

The expansion coefficients are selected so that Eq. (2.22) is satisfied, in addition to the boundary conditions. The Gauss pseudospectral method is a collocation method.

Unlike indirect methods, in direct methods there is no need to discretize and approximate the costate. However, if an accurate costate estimate can be generated, this information can help validate the optimality of the solution from a direct approach. Consequently, many direct methods attempt to produce a costate approximation based on the Lagrange multipliers involved in the NLP. One of the key attributes of the Gauss pseudospectral method is that it produces unusually accurate costate estimates as compared to other direct methods. This is explained in detail in Chapter 3.

2.2 Numerical Approximation Methods

Analytic solutions to optimal control problems are often limited to simple, well-understood problems. Consequently, most optimal control problems are solved numerically. In a direct method, the original infinite-dimensional optimal control problem is discretized and approximated. In an indirect method, the HBVP is discretized and approximated. However, both formulations require numerical approximation techniques. Since this thesis focuses on pseudospectral methods, this section describes the mathematics of the approximation methods used specifically in pseudospectral direct transcription methods.

2.2.1 Global Polynomial Approximations

Pseudospectral methods employ global interpolating polynomials to approximate the state across the entire interval, $\tau \in [-1, 1]$. For the pseudospectral methods considered in this thesis, the approximations use Lagrange interpolating polynomials as the basis functions. These polynomials are defined using a set of M support points $\tau_1, \dots, \tau_M$ on the time interval $\tau \in [-1, 1]$. The state, control, and costate of the optimal control problem can be approximated as[28]

$$y(\tau) \approx Y(\tau) = \sum_{i=1}^{M} \mathcal{L}_i(\tau) Y(\tau_i), \tag{2.23}$$

where $Y(\tau)$ is a $(M-1)^{\text{th}}$ order polynomial approximation and $\mathcal{L}_i(\tau)$, $(i = 1, \ldots, M)$ is the set of Lagrange interpolating polynomials, defined as

$$\mathcal{L}_i(\tau) = \prod_{j=1, j\neq i}^{M} \frac{\tau - \tau_j}{\tau_i - \tau_j} = \frac{g(\tau)}{(\tau - \tau_i)\dot{g}(\tau)} \tag{2.24}$$

where $g(\tau)$ creates the trial function that determines the locations of the support points (often this trial function is related to Legendre or Chebyshev polynomials) and $\dot{g}(\tau)$ is the time derivative of $g(\tau)$. Lagrange polynomials work well for collocation methods, since it can be shown that

$$\mathcal{L}_i(\tau_j) = \begin{cases} 1 & , \quad i = j \\ 0 & , \quad i \neq j \end{cases} \tag{2.25}$$

resulting in the property that $y(\tau_j) = Y(\tau_j)$, $(j = 1, \ldots, M)$, i.e., the function approximation is equal to the true function at the M points. In many pseudospectral methods, the state may be approximated with one basis of Lagrange polynomials, while the control may be approximated with a different basis of Lagrange polynomials. In fact, the Gauss pseudospectral method does not even use the same *number* of support points for the state and control. This is explained in detail in Chapter 3.

When discretizing the continuous-time interval, an intuitive discretization scheme is to break the interval at equidistant support points. However, for polynomial approximation, a uniform mesh has some very undesirable properties. As mentioned earlier, the $(M-1)^{\text{th}}$ order polynomial approximation uses M support points. One would hope that as the number of support points increases, the error between the polynomial approximation and the true function decreases. However, for uniform support points, this is not the case. As the order of the polynomial approximation increases for uniformly spaced support points, the *Runge phenomenon* surfaces, meaning the approximation error near the boundaries actually increases as the order increases. Fig. 2-1 depicts the Runge phenomenon for a Lagrange approximation to the function $1/(1+16x^2)$ using 25 equidistant points, where it is clear that the approximation is quite poor near the boundaries. Fortunately, there exist sets of non-uniform points that eliminate the Runge phenomenon and can guarantee that the polynomial approximation error monotonically decreases as the number of support points

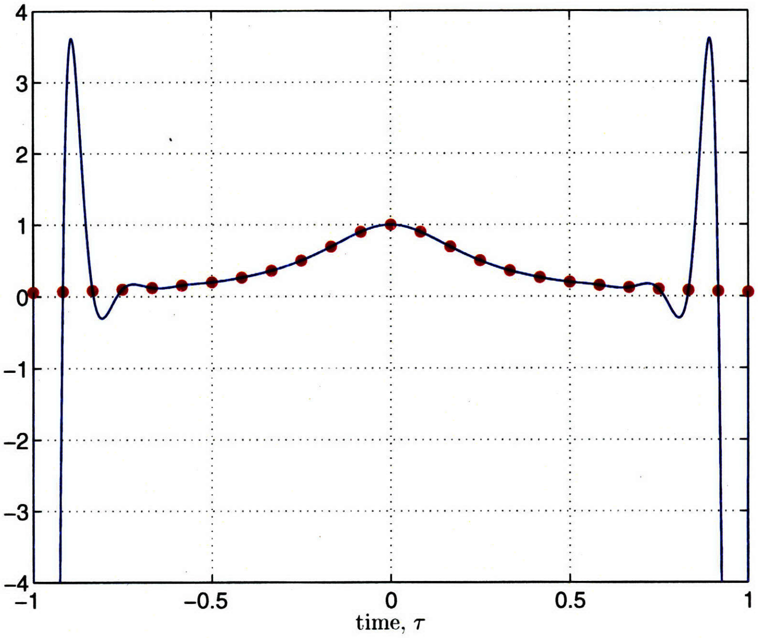


Figure 2-1: Lagrange polynomial approximation to the function $1/(1+16x^2)$ using 25 equidistant points on the interval $[-1, 1]$.

is increased. Support points based on the roots of Legendre and Chebyshev polynomials have this property, and have the characteristic that the spacing between the support points is denser towards the boundaries. As is explained in Chapter 3, the support points of the Gauss pseudospectral method are based on the roots of Legendre polynomials. As illustrative examples, a comparison is shown between equidistant points and the points used in the GPM for two functions in Figs. 2-2 and 2-3. The function in Fig. 2-2 is t^9 and the function in Fig. 2-3 is e^t. In these figures, it is clear that for large numbers of support points, the approximation error using uniform spacing actually increases, while the support points based on Legendre polynomials remains at machine precision accuracy.

2.2.2 Quadrature Approximation

In addition to the choice of support points used to approximate the state, control and costate, pseudospectral methods use another set of points to accurately approximate the dynamic aspects of the optimal control problem. For example, the integral within the cost functional of Eq. (2.7) and the dynamic constraints of Eq. (2.8) must be discretized and

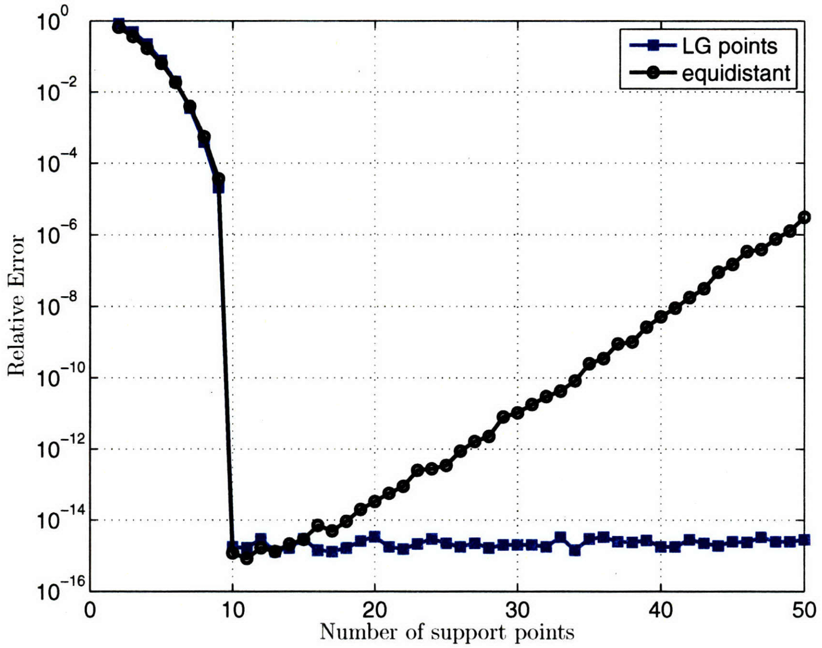


Figure 2-2: Approximation accuracy of t^9 for uniform and non-uniform spacing as the number of support points is increased

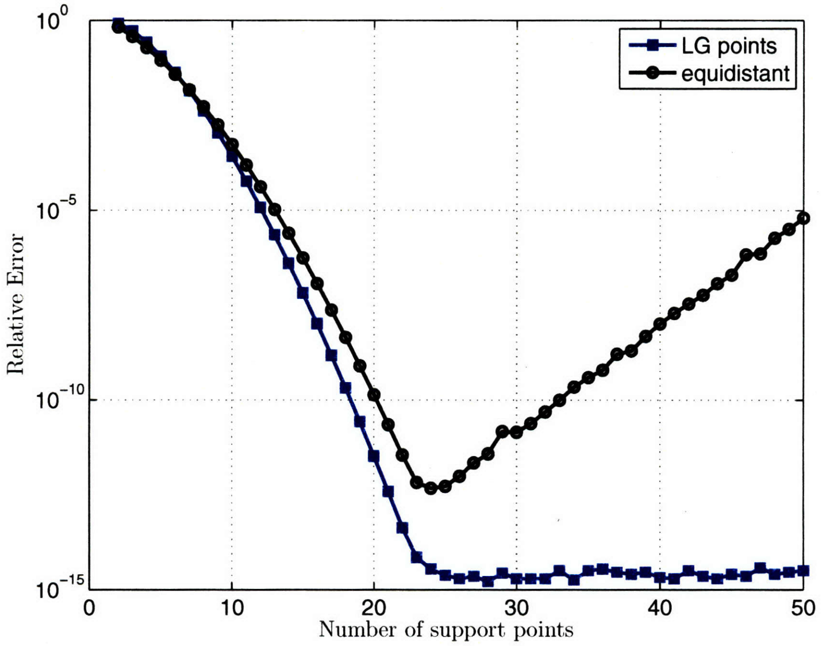


Figure 2-3: Approximation accuracy of e^t for uniform and non-uniform spacing as the number of support points is increased

approximated as accurately as possible. Consequently, the points are chosen to minimize the error in the quadrature approximation[29] to an integral. The general form for the quadrature approximation in pseudospectral methods is

$$\int_a^b f(\tau)d\tau \approx \sum_{i=1}^{K} w_i f(\tau_i) \tag{2.26}$$

where $\tau_1, \ldots, \tau_K$ are the quadrature points on the interval $\tau \in [-1, 1]$ and w_i, $(i = 1, \ldots, K)$ are quadrature weights. For an arbitrary set of unique points $\tau_1, \ldots, \tau_K$, the quadrature approximation is exact for polynomials of degree $K - 1$ or less. However, by appropriately choosing the points, the accuracy of the quadrature can be improved greatly. It is well-known that the quadrature approximation with the highest accuracy for a given number of points K is the Gauss quadrature. In particular, the Gauss quadrature is exact for all polynomials of degree $2K-1$ or less. In a Gauss quadrature, the K quadrature points are the Legendre-Gauss (LG) points, defined as the roots of the K^{th}-degree Legendre polynomial, $P_K(t)$ where

$$P_K(\tau) = \frac{1}{2^K K!}\frac{d^K}{d\tau^K}\left[\left(\tau^2 - 1\right)^K\right] \tag{2.27}$$

These non-uniform points are located on the interior of the interval $[-1, 1]$, and are clustered towards the boundaries, as depicted in Fig. 2-4. The corresponding Gauss quadrature weights are then found by the formula

$$w_i = \int_{-1}^{1} \mathcal{L}_i(\tau)d\tau = \frac{2}{(1 - \tau_i^2)\left[\dot{P}_K(\tau_i)\right]^2}, \quad (i = 1, \ldots, K) \tag{2.28}$$

where $\dot{P}_K$ is the derivative of the K^{th}-degree Legendre polynomial.

Another approach is to use Legendre-Gauss-Radau (LGR) points, which lie on the interval $\tau \in [-1, 1)$. By forcing one of the points to lie at the boundary, the degree of freedom is reduced by one, thus making this choice of points accurate to $2K - 2$. The K LGR points are defined as the roots of $P_K(\tau) + P_{K-1}(\tau)$, or the summation of the Kth and $(K - 1)$th order Legendre polynomials. The corresponding weights for the LGR points are

$$\begin{aligned} w_1 &= \frac{2}{K^2}, \\ w_i &= \frac{1}{(1-\tau_i)[\dot{P}_{K-1}(\tau_i)]^2}, \quad i = 2, \ldots, K \end{aligned} \tag{2.29}$$

The standard set of LGR points includes the initial point but not the final point. This set of LGR points works well for infinite horizon problems.[42] For finite-horizon problems, the LGR points (and corresponding weights) are often flipped on the interval,[72] meaning the set includes the final point but not the initial point ($\tau \in (-1, 1]$). This set of LGR points can be found from the roots of $P_K(\tau) - P_{K-1}(\tau)$.

A third set of points often used in pseudospectral methods is the set of Legendre-Gauss-Lobatto (LGL) points, which lie on the interval $\tau \in [-1, 1]$. In this approach, quadrature points are forced to lie at both boundaries, reducing the degree of freedom by two degrees, thus making this quadrature scheme accurate to $2K-3$. The K LGL points are the roots of $(1-\tau^2)\dot{P}_{K-1}(\tau)$ (where $\dot{P}_{K-1}(\tau)$ is the derivative of the $(K-1)$th order Legendre polynomial, $P_{K-1}(\tau)$). The corresponding weights for the LGL points are

$$w_i = \frac{2}{K(K-1)} \frac{1}{[P_{K-1}(\tau_i)]^2}, \quad (i = 1, \ldots, K) \tag{2.30}$$

An example of the point locations for the LG, LGR, and LGL points is shown in Fig. 2-4, where each set includes 10 points. Furthermore, the quadrature approximation accuracy of the LG points, LGR points, and LGL points can be seen in Figs . 2-5 and 2-6. Fig. 2-5 shows the quadrature approximation accuracy for the example function t^9. As seen in this figure, the approximation to this 9^{th} order polynomial achieves machine precision accuracy for 5 LG points $(2K-1)$, 6 LGR $(2K-2)$, and 6 LGL points $(2K-3)$. Fig. 2-6 shows the quadrature approximation error for the function e^t for all three previously mentioned Legendre points. Notice the rapid convergence of the error, even for C^∞ functions such as e^t. This rate of convergence is called *spectral*, meaning that the rate of convergence for an N^{th}-order approximation is $O(N^{-m})$ for every m for C^∞ functions. Pseudospectral methods exploit this rapid convergence rate to obtain very accurate solutions with a few number of points.

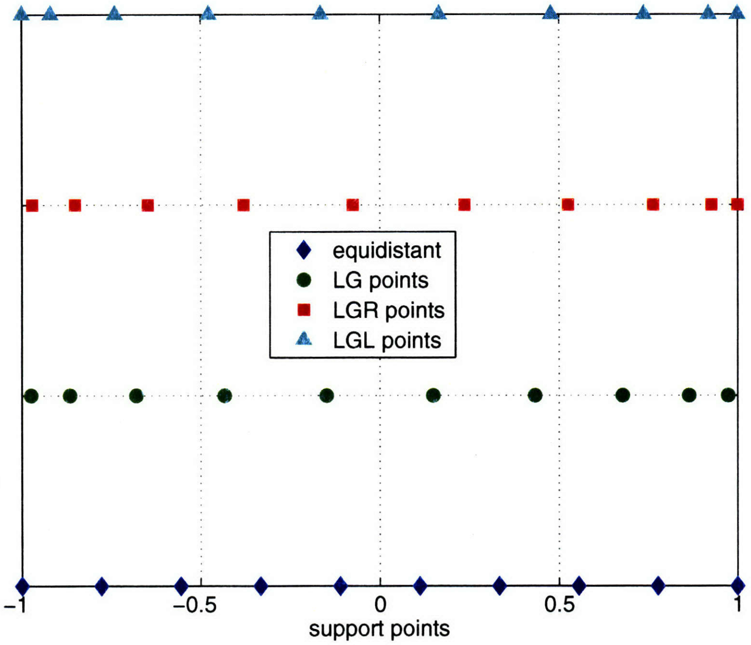


Figure 2-4: Legendre-Gauss, Legendre-Gauss-Radau, and Legendre-Gauss-Lobatto collocation points for $K = 10$ on the interval $[-1, 1]$

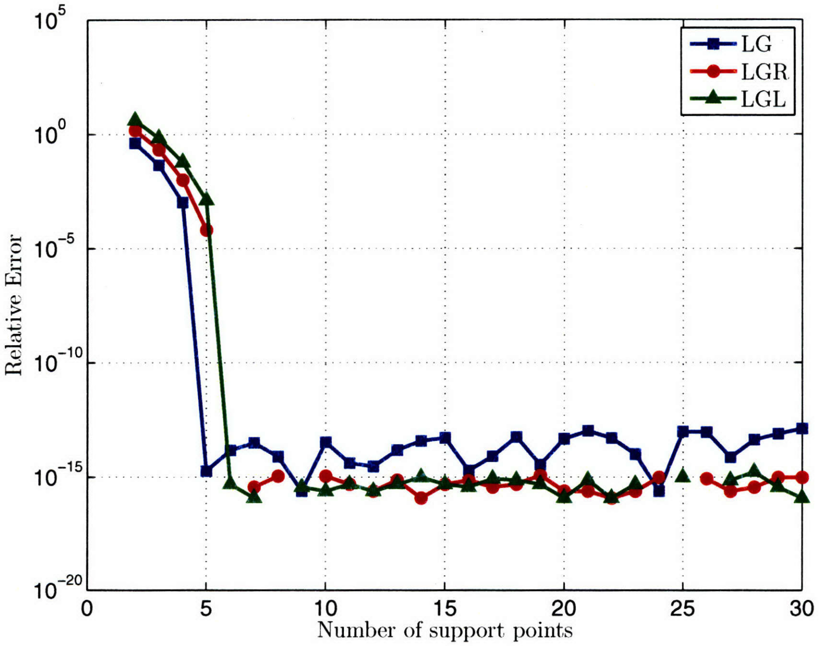


Figure 2-5: Accuracy of the quadrature approximation to t^9 using LG, LGL, and LGR points as the number of points increases

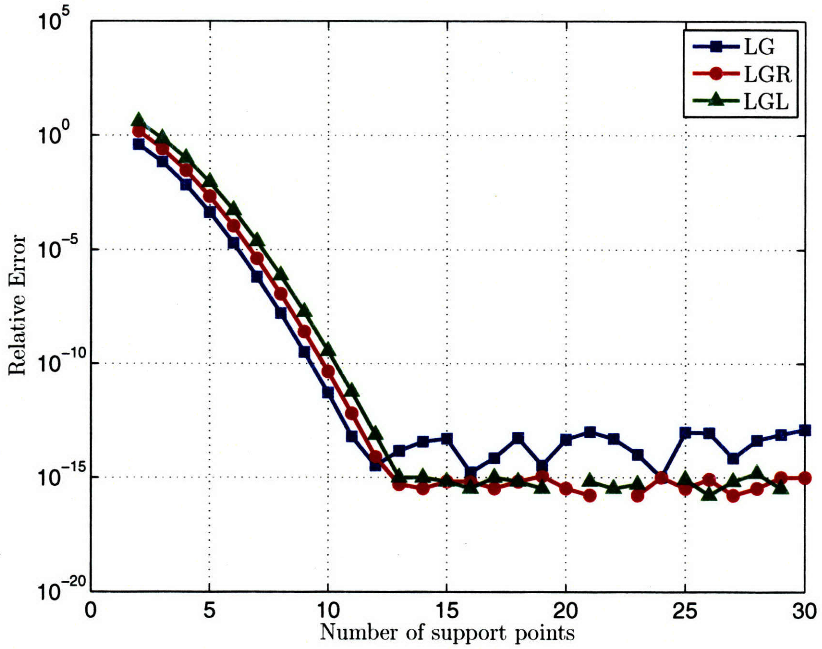


Figure 2-6: Accuracy of the quadrature approximation to e^t using LG, LGL, and LGR points as the number of points increases

2.2.3 Orthogonal Collocation

Pseudospectral methods, like all direct methods, transform the dynamic equations into algebraic conditions. Pseudospectral methods accomplish this by orthogonal collocation, meaning they collocate the derivative of the state approximation (Eq. (2.23)) with the vector field, $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$ of Eq. (2.2), at a set of points along the interval corresponding to the roots of orthogonal polynomials (e.g., Legendre polynomials). Let us assume there are K *collocation* points and M points used to approximate the state. Mathematically, the derivative of the state approximation at the k-th collocation point, τ_k, is

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=1}^{M} \dot{\mathcal{L}}_i(\tau_k)\mathbf{X}(\tau_i) = \sum_{i=1}^{M} D_{ki}\mathbf{X}(\tau_k), \quad (k = 1, \ldots, K) \tag{2.31}$$

where the differentiation matrix, $D \in \mathbb{R}^{K \times M}$, is defined as

$$D_{ki} = \begin{cases} \dfrac{\dot{g}(\tau_k)}{(\tau_k - \tau_i)\dot{g}(\tau_i)}, & \text{if } k \neq i \\ \dfrac{\ddot{g}(\tau_i)}{2\dot{g}(\tau_i)}, & \text{if } k = i \end{cases} \tag{2.32}$$

Note that because the derivative of the state approximation is an algebraic expression, the differential equation is replaced by a set of algebraic conditions evaluated at the collocation points. The continuous dynamics of Eq. (2.8) are then transcribed into the following set of K algebraic equations (or *residuals*) via orthogonal collocation:

$$\mathbf{R}_k = \sum_{i=1}^{M} D_{ki}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) = \mathbf{0}, \quad (k = 1, \ldots, K) \tag{2.33}$$

which can then be implemented in the nonlinear program. The specific details and theory behind the Gauss pseudospectral method is presented in the next chapter.

Chapter 3

The Gauss Pseudospectral Method

The Gauss pseudospectral method was originally developed by Benson[6] in an effort to improve the costate estimation of the Legendre pseudospectral method. It was originally formulated for optimal control problems involving integral dynamic constraints, but was later adapted to be used with the more common differential dynamic constraints.

This chapter describes the Gauss pseudospectral method in detail, where the notation has been streamlined and expanded to include problems with state and/or control path constraints. Specific attention is drawn to the costate estimation procedure, and an equivalence between the NLP multipliers and the discretized costate of the HBVP is mapped out. Furthermore, a new estimate for the boundary control is presented and analyzed. Lastly, the issue of the convergence of pseudospectral methods is addressed.

3.1 Direct Transcription Formulation

The Gauss pseudospectral transcription converts the transformed continuous Bolza problem of Chapter 2 into a nonlinear program. This is accomplished using many of the mathematical approximation tools expained in the second half of Chapter 2.

The Gauss pseudospectral method, like all pseudospectral methods, approximates the state using a basis of global interpolating polynomials. These global polynomials are based on a set of discrete points across the interval. One of the primary distinctions between pseudospectral methods is the choice of discrete points used in the NLP formulation. Specifically for the Gauss pseudospectral method, let $\mathcal{K} = \{\tau_1, \ldots, \tau_K\}$ be the set of K Legendre-Gauss (LG) points which correspond to the roots of the K^{th} degree Legendre polynomial, $P_K(\tau)$, defined in the previous chapter. These LG points points lie on the *interior* of the interval $(-1, 1)$ such that $\tau_1, \ldots, \tau_K$ are strictly increasing. Next, suppose that, for a given value of K, one can create a superset, $\mathcal{K}_0$, that appends the point $\tau_0 = -1$ to the set $\mathcal{K}$, for a total of $K + 1$ points on the interval $[-1, 1)$. Let $\mathbf{x}(\tau)$ be the state as a function of the independent variable τ where $\tau \in [-1, 1]$, as seen in the transformed continuous Bolza problem. An approximation to the state, $\mathbf{X}(\tau)$, is formed with a basis of $K + 1$ Lagrange interpolating polynomials $\mathcal{L}_i(\tau)$ $(i = 0, \ldots, K)$ as follows:

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=0}^{K} \mathcal{L}_i(\tau)\mathbf{x}(\tau_i) \tag{3.1}$$

where

$$\mathcal{L}_i(\tau) = \prod_{j=0, j\neq i}^{K} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{3.2}$$

The use of Lagrange polynomials results in a state approximation that is equal to the true state at all the points within $\mathcal{K}_0$, or mathematically, $\mathbf{x}(\tau_i) = \mathbf{X}(\tau_i)$, $(i = 0, \ldots, K)$.

Note that the time interval in the transformed optimal control problem spans $\tau \in [-1, 1]$, yet the state approximation of Eq. (3.1) does not specifically include a discrete point at the terminal time, $\tau_f = 1$. Optimal control problems often include constraints involving the terminal state, so it is advantagous to include a discrete variable $\mathbf{X}_f$ in the nonlinear program. Therefore, in order to fully discretize the time interval for the NLP, another set of points, called *discretization points* or *nodes*, is created as a superset of $\mathcal{K}_0$. The set of nodes, $\mathcal{N}$, includes the K interior LG points, $\tau_1, \ldots, \tau_K$, the initial point, $\tau_0 \equiv -1$, and the final point, $\tau_f \equiv 1$. The state at the final time, $\mathbf{X}_f$, is included in the NLP discretization, although it is not specifically part of the state approximation of Eq. (3.1). Since $\mathbf{X}_f$ is absent in the state approximation, it must be constrained in other ways to ensure that it satisfies the state dynamic equations of Eq. (2.8). This is accomplished by including an additional constraint that relates the final state to the initial state via a Gauss quadrature. According to the state dynamics,

$$\mathbf{x}(\tau_f) = \mathbf{x}(\tau_0) + \int_{-1}^{1} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \tag{3.3}$$

which can be discretized and approximated as

$$\mathbf{X}(\tau_f) - \mathbf{X}(\tau_0) - \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) = \mathbf{0} \tag{3.4}$$

where w_k are the Gauss weights and τ_k are the LG points. At first glance, the non-intuitive handling of the terminal state seems unnecessarily complicated, however by formulating the problem in this manner, unique mathematical properties can be derived which allow for a very accurate approximation to the costate. This is presented more formally in the next section.

Eq. (3.4) forces the terminal state to satisfy the dynamic constraints, but additional constraints must be posed to ensure that the entire discrete state satisfies the dynamic equations. This is accomplished via orthogonal collocation, as described in Chapter 2. Collocation methods require the dynamic equations of Eq. (2.2) to be satisfied exactly at a set of collocation points. The left-hand side of the dynamic equations is approximated by differentiating the state approximation of Eq. (3.1) at the Legendre-Gauss points in the manner of Eq. (2.31) as follows:

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=0}^{K} \dot{\mathcal{L}}_i(\tau_k) \mathbf{X}(\tau_i) = \sum_{i=0}^{K} D_{ki} \mathbf{X}(\tau_k), \quad (k = 1, \ldots, K) \tag{3.5}$$

The differentiation matrix, $D \in \mathbb{R}^{K \times (K+1)}$, can be determined offline from Eq. (2.32), where $g(\tau) = (1 + \tau) P_N(\tau)$.

$$D_{ki} = \dot{\mathcal{L}}_i(\tau_k) = \begin{cases} \dfrac{(1+\tau_k)\dot{P}_K(\tau_k) + P_K(\tau_k)}{(\tau_k - \tau_i)\left[(1+\tau_i)\dot{P}_K(\tau_i) + P_K(\tau_i)\right]} & , \quad i \neq k \\[2ex] \dfrac{(1+\tau_i)\ddot{P}_K(\tau_i) + 2\dot{P}_K(\tau_i)}{2\left[(1+\tau_i)\dot{P}_K(\tau_i) + P_K(\tau_i)\right]} & , \quad i = k \end{cases} \tag{3.6}$$

where τ_k, $(k = 1, \ldots, K)$ are the points in the set $\mathcal{K}$, and τ_i, $(i = 0, \ldots, K)$ are the points in the set $\mathcal{K}_0$. Unlike many pseudospectral methods, the Gauss pseudospectral method does not collocate at every point in the discretization set, $\mathcal{N}$. To emphasize this point, the set of LG points, $\mathcal{K}$, is termed the set of *collocation points.* Recall the discretization set ($\mathcal{N}$) contains the LG points plus the initial point and final point. Therefore, for the Gauss pseudspectral method, there are two more discretization points than collocation points ($K = N - 2$). The K collocation equations require Eq. (3.5) to be equal to the right-hand side of the dynamic equations at the collocation points:

$$\sum_{i=0}^{K} D_{ki}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k), \tau_k; t_0, t_f) = \mathbf{0}, \quad (k = 1, \dots, K) \tag{3.7}$$

As is shown shortly, collocating strictly on the interior of the interval leads to a unique mathematical equivalence used to approximate the costate.

A discrete approximation for the control, $\mathbf{U}_k$, $(k = 1, \dots, K)$ is required for the collocation constraints of Eq. (3.7) as well as the quadrature constraint of Eq. (3.4). The time derivative of the control is not approximated, so the control approximation need not be of the form Eq. (2.23). In fact, any approximation that has the property $\mathbf{u}(\tau_k) = \mathbf{U}_k$, $(k = 1, \dots, K)$ is an equivalent control approximation in the eyes of the NLP. However, for consistency, this work approximates the control at the K collocation points using a basis of K Lagrange interpolating polynomials $\tilde{\mathcal{L}}_i(\tau)$, $(i = 1, \dots, K)$ as

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{i=1}^{K} \tilde{\mathcal{L}}_i(\tau)\mathbf{U}(\tau_i) \tag{3.8}$$

where τ_i, $(i = 1, \dots, K)$ are the LG points belonging to the set $\mathcal{K}$.

Lastly, the integral term in the cost functional of Eq. (2.7) can be approximated with a Gauss quadrature as before, resulting in

$$J = \Phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k g(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \tag{3.9}$$

3.1.1 Gauss Pseudospectral Discretization of Transformed Continuous Bolza Problem

From the previous discussion, the Gauss pseudospectral discretization of the transformed continuous Bolza optimal control problem in Chap 2 can now be stated formally. Determine the discretized state, $\mathbf{X}_i$, $i \in \mathcal{N}$, control, $\mathbf{U}_k$, $k \in \mathcal{K}$, initial time, t_0, and final time, t_f that minimizes the cost function

$$J = \Phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k g(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \tag{3.10}$$

subject to the algebraic collocation constraints expressed as a residual function,

$$\mathbf{R}_k \equiv \sum_{i=0}^{K} D_{ki}\mathbf{X}_i - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) = \mathbf{0} \quad (k = 1, \ldots, N) \tag{3.11}$$

quadrature constraint expressed as a residual,

$$\mathbf{R}_f \equiv \mathbf{X}_f - \mathbf{X}_0 - \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) = \mathbf{0} \tag{3.12}$$

boundary constraint,

$$\phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) = \mathbf{0} \tag{3.13}$$

and path constraint,

$$\mathbf{C}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \leq \mathbf{0} \quad (k = 1, \ldots, N) \tag{3.14}$$

The cost function of Eq. (3.10) and the constraints of Eq. (3.11)–(3.14) define an NLP whose solution is an approximate solution to the continuous Bolza problem.

To recap, the variables that are used in the GPM discretization are as follows:

NLP Variables Corresponding to State:	$(\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_K, \mathbf{X}_f)$
NLP Variables Corresponding to Control:	$(\mathbf{U}_1, \ldots, \mathbf{U}_K)$

Similarly, the constraints in the GPM are as follows:

Constraints Corresponding to Dynamics:	$(\mathbf{R}_1, \ldots, \mathbf{R}_K, \mathbf{R}_f)$
Constraints Corresponding to Path Constraints:	$(\mathbf{C}_1, \ldots, \mathbf{C}_K)$
Constraints Corresponding to Boundary Conditions:	$\boldsymbol{\phi}$

3.1.2 A Comment on the Final State: $\mathbf{X}_f$

As seen in the previous section, the state is discretized at all N discretization points, but the actual state approximation of Eq. (3.1) includes only $N-1$ points in the set $\mathcal{K}_0$. The final state is not explicitly in the state approximation. It is, however, implicitly in the state approximation due to the quadrature constraint of Eq. (3.4). Although it is not immediately obvious, the inclusion of the quadrature constraint produces the same value for $\mathbf{X}_f$ as one would get by extrapolating the state approximation of Eq. (3.1) to t_f. In this section, it is shown mathematically that these two expressions are consistent with one another. Specifically, it is shown that the NLP final state, $\mathbf{X}_f$ is the same as the extrapolated state $\mathbf{X}(t_f)$:

Lemma 1. *Using the Gauss pseudospectral transcription for an optimal control problem, the following two formulations for $\mathbf{X}_f$ are equivalent:*

$$\mathbf{X}_f = \mathbf{X}_0 + \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \tag{3.15}$$

and

$$\mathbf{X}_f = \mathbf{X}(t_f) = \sum_{i=0}^{K} \mathbf{X}(t_i) \mathcal{L}_i(t_f), \tag{3.16}$$

Proof of Lemma 1. *Beginning with Eq. (3.15):*

$$\mathbf{X}_f = \boldsymbol{X}_0 + \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \tag{3.17}$$

From the collocation constraints of Eq. (3.11), the vector field at the Gauss points is equal to the derivative of the state approximation, therefore $\mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f)$ can be replaced in

Eq. (3.17):

$$\mathbf{X}_f = \mathbf{X}_0 + \sum_{k=1}^{K} w_k \left(\sum_{i=0}^{K} \mathbf{X}(\tau_i) D_{ki} \right) \tag{3.18}$$

Recall that $D_{ki} = \dot{\mathcal{L}}_i(\tau_k)$. Hence, Eq. (3.18) is a Gauss quadrature for a polynomial of degree less than $2K - 1$ and can be replaced exactly with an integral

$$\mathbf{X}_f = \mathbf{X}_0 + \int_{-1}^{1} \sum_{i=0}^{K} \mathbf{X}(\tau_i) \dot{\mathcal{L}}_i(\tau) d\tau \tag{3.19}$$

This integral can be evaluated, resulting in

$$\mathbf{X}_f = \mathbf{X}_0 + \sum_{i=0}^{K} \mathbf{X}(\tau_i) \mathcal{L}_i(\tau_f) - \sum_{i=0}^{K} \mathbf{X}(\tau_i) \mathcal{L}_i(\tau_0) \tag{3.20}$$

According to the definition of Lagrange polynomials in Eq. (3.2), the final term in Eq. (3.20) is equal to $\mathbf{X}_0$, so the first and last terms cancel, resulting in

$$\mathbf{X}_f = \sum_{i=0}^{K} \mathbf{X}(\tau_i) \mathcal{L}_i(\tau_f) \tag{3.21}$$

which simplifies to

$$\mathbf{X}_f = \mathbf{X}(t_f) \tag{3.22}$$

from Eq. (3.16).

□

The state approximation and the quadrature constraint are consistent with one another and do not produce conflicting values for the state at the final time.

It is noted that in his original formulation, Benson did not explicitly include $\mathbf{X}_f$ or Eq. (3.12) into the NLP, but rather used Eq. (3.12) to eliminate $\mathbf{X}_f$ from the problem. This has the benefit of reducing the number of NLP nodes by one, but also makes the other constraints more complicated. For example, if $\mathbf{X}_f$ is not explicitly in the discretization, then any simple terminal constraint on the state would be a function of the state at *all* the discretization points in the NLP according to Eq. (3.12). Because simple terminal constraints are often found in optimal control problems, this work retains that simplicity by explicitly including $\mathbf{X}_f$ and Eq. (3.12) in the NLP formulation.

3.1.3 Discontinuities & Phases

Some optimal control problems have known discontinuities in either the state or control. Common examples are a mass drop during a launch vehicle ascent problem, or a "bang-bang" control solution. The spectral accuracy described in Chapter 2 only applies to smooth problems, meaning the primary advantage of using a global polynomial approximation is eliminated for discontinuous problems. Chapter 5 discusses this effect in more detail. However, for problems with known discontinuities, a common procedure to recover the spectral accuracy is to divide the trajectory into *phases*, where the dynamics are transcribed within each phase and then connected together by additional phase interface (a.k.a *linkage*) constraints.[85,95] Other approaches involve using a smoothing filter[34] to eliminate the oscillatory Gibb's phenomenon often associated with a polynomial approximation to a discontinuous function.

The Gauss pseudospectral method has been used to solve several multi-phase problems (see Ref. 6,66,68). The extension to a P-phase problem involves repeating the structure for the one-phase formulation P times. In addition, the terminal constraints for the first phase, the initial and terminal constraints for the interior phases, and the initial constraints for the P^{th} phase are re-characterized as interior point constraints. These interior point constraints include any continuity conditions in the state or time between adjacent phases. For our formulation, it is assumed that the control is allowed to be discontinuous across phases, but this too can be constrained if necessary. Eq. (3.13) of Section 3.1.1 can then be replaced by the following set of equations:

$$\begin{aligned} &\boldsymbol{L}^{(1)}(\mathbf{X}_0^{(1)}, t_0^{(1)}) &&= \mathbf{0}, \\ &\boldsymbol{L}_{(r)}^{(r+1)}(\mathbf{X}_f^{(r)}, t_f^{(r)}, \mathbf{X}_0^{(r+1)}, t_0^{(r+1)}) &&= \mathbf{0}, \\ &\boldsymbol{L}^{(P)}(\mathbf{X}_f^{(P)}, t_f^{(P)}) &&= \mathbf{0} \end{aligned} \tag{3.23}$$

where $r = 1, \dots, P-1$ phases. It is noted that these phases do not need to be sequential, and can in fact be parallel. The incorporation of parallel phases is utilized in the formation flying problems of Chapter 7, but is represented simply as ϕ for the rest of this chapter in order to simplify the equations.

3.2 Costate Approximation

As mentioned previously in the discussion on optimal control theory, the costate, $\lambda(\tau)$ from the HBVP equations of Eq. (2.14), plays a very important role in determining the optimality of a given solution. This section describes the mathematics behind the costate approximation for the Gauss pseudospectral method. Specifically, the optimality conditions for the NLP are developed, which involve a set of Lagrange multipliers. Next, the discretized HBVP equations are formulated, which include a discrete costate. The Lagrange multipliers are shown to have an exact mapping to the discrete costate, thus proving the equivalence between the NLP optimality conditions and the HBVP optimality conditions. Benson originally showed this equivalence for a simplified problem in Ref. 6. This section advances Benson's work by showing equivalence for problems with state and control inequality path constraints. The expanded results shown here were published in Ref. 8.

Although the costate is not an explicit variable in the NLP formulation of the previous section, an interpolating polynomial is defined to represent it. Recall that the state approximation, $X(\tau)$ involved the support points in the set $\mathcal{K}_0$, which included the K collocation points and the initial point, τ_0. The costate approximation, $\Lambda(\tau)$, utilizes the points in a new set, $\mathcal{K}_f$, which includes the K collocation points and the final point, τ_f. A basis of $K+1$ Lagrange interpolating polynomials, $\mathcal{L}_i^{\dagger}(\tau)$ $(i = 1, \ldots, K+1)$ and $(\tau_{K+1} = \tau_f)$, approximates the costate as

$$\lambda(\tau) \approx \Lambda(\tau) = \sum_{i=1}^{K+1} \lambda(\tau_i)\mathcal{L}_i^{\dagger}(\tau) \tag{3.24}$$

where

$$\mathcal{L}_i^{\dagger}(\tau) = \prod_{j=1, j\neq i}^{K+1} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{3.25}$$

As with the state approximation, Eq. (3.24) can be differentiated to determine an approximation to $\dot{\lambda}(\tau)$. The costate is approximated using $K+1$ points, yet collocation is performed at the K *collocation points* using

$$\dot{\lambda}(\tau_k) \approx \dot{\Lambda}(\tau_k) = \sum_{i=1}^{K+1} \lambda(\tau_i) D^\dagger_{ki} \quad (k = 1, \ldots, K) \tag{3.26}$$

where $D^\dagger_{ki} = \dot{\mathcal{L}}^\dagger_i(\tau_k)$ represents the elements of the differential approximation matrix $D^\dagger \in \mathbb{R}^{K \times K+1}$. $D^\dagger$ is referred to as the adjoint of the differential approximation matrix D. Gauss weights are used to prove the following lemma that relates elements of the differential approximation matrix, D, to elements of the adjoint differential approximation matrix, $D^\dagger$, as originally shown in Ref. 6:

Lemma 2. *[6] The relationship between elements in the differential approximation matrix D and elements in its adjoint $D^\dagger$ is*

$$D_{ik} = -\frac{w_k}{w_i} D^\dagger_{ki} \quad (i, k = 1, \ldots, K) \tag{3.27}$$

where w_i and w_k, $(i, k = 1, \ldots, K)$ are the Gauss weights.[29] *Notice the elements of D corresponding to the initial point, τ_0 (the first column of D), and elements of $D^\dagger$ corresponding to the final point, τ_f (the last column of $D^\dagger$), are omitted in this relationship.*

Proof of Lemma 2 (6). *Consider the integration by parts formula for the product of $\dot{p}(\tau)$ and $q(\tau)$ with $\tau \in [-1, 1]$ where $p(\tau)$ and $q(\tau)$ are defined to be polynomials of degree K, i.e.,*

$$\int_{-1}^{1} \dot{p}(\tau) q(\tau) d\tau = p(\tau) q(\tau) \Big|_{-1}^{1} - \int_{-1}^{1} p(\tau) \dot{q}(\tau) d\tau$$

Because $p(\tau)$ and $q(\tau)$ are polynomials of degree K, the functions $\dot{p}(\tau)q(\tau)$ and $p(\tau)\dot{q}(\tau)$ are polynomials of degree $2K - 1$ and, thus, can be replaced exactly by a Gauss quadrature.[29] *Then, using the exact derivatives of the polynomials $\dot{p}(\tau)$ and $\dot{q}(\tau)$ at the Gauss points in Eqs. (3.5) and (3.26), respectively, the integration by parts formula is replaced exactly by*

$$\begin{aligned} &\sum_{j=1}^{K} \left[\sum_{l=0}^{K} D_{jl} p(\tau_l) \right] q(\tau_j) w_j = \\ &\quad [p(1)q(1) - p(-1)q(-1)] - \sum_{j=1}^{K} p(\tau_j) \left[\sum_{l=1}^{K+1} D^\dagger_{jl} q(\tau_l) \right] w_j \end{aligned} \tag{3.28}$$

Because Eq. (3.28) must hold for all polynomials $p(\tau)$ and $q(\tau)$ of degree K or less, it must hold for the set of K^{th} degree Lagrange interpolating polynomials $\mathcal{L}_p(\tau)$ $(p = 0, \ldots, K)$ and $\mathcal{L}_q^\dagger(\tau)$ $(q = 1, \ldots, K+1)$ as defined in Eqs. (3.2) and (3.25). To show the relationship between the elements of the differential approximation matrix D and the elements of the adjoint $D^\dagger$, a subset of these Lagrange polynomials, $\mathcal{L}_k(\tau)$ and $\mathcal{L}_i^\dagger(\tau)$ $(i, k = 1, \ldots, K)$ from Eqs. (3.2) and (3.25), respectively, are substituted into Eq. (3.28) to obtain

$$\begin{gathered}\sum_{j=1}^{K}\left[\sum_{l=0}^{K} D_{jl}\mathcal{L}_k(\tau_l)\right]\mathcal{L}_i^\dagger(\tau_j)w_j = \\ \left[\mathcal{L}_k(1)\mathcal{L}_i^\dagger(1) - \mathcal{L}_k(-1)\mathcal{L}_i^\dagger(-1)\right] - \sum_{j=1}^{K}\mathcal{L}_k(\tau_j)\left[\sum_{l=1}^{K+1} D_{jl}^\dagger\mathcal{L}_i^\dagger(\tau_l)\right]w_j\end{gathered} \tag{3.29}$$

Now from the properties of Lagrange polynomials in Eq. (2.25) it is seen that

$$\begin{aligned}\mathcal{L}_k(-1) &= 0 \qquad (k = 1, \ldots, K)\\ \mathcal{L}_i^\dagger(1) &= 0 \qquad (i = 1, \ldots, K)\end{aligned}$$

Eq. (3.29) then simplifies to

$$\sum_{j=1}^{K}\sum_{l=1}^{K} D_{jl}\mathcal{L}_k(\tau_l)\mathcal{L}_i^\dagger(\tau_j)w_j = -\sum_{j=1}^{K}\sum_{l=1}^{K} D_{jl}^\dagger\mathcal{L}_k(\tau_j)\mathcal{L}_i^\dagger(\tau_l)w_j \tag{3.30}$$

Then, once again using the Lagrange polynomial properties of Eq. (2.25) in Eq. (3.30), it can be seen that the product $\mathcal{L}_k(\tau_l)\mathcal{L}_i^\dagger(\tau_j)$ is zero unless $k = l$ and $i = j$. Similarly, the product $\mathcal{L}_k(\tau_j)\mathcal{L}_i^\dagger(\tau_l)$ is zero unless $k = j$ and $i = l$. Therefore, the following result is obtained at the Gauss points:

$$D_{ik}w_i = -D_{ki}^\dagger w_k \qquad (i, k = 1, \ldots, K) \tag{3.31}$$

Rearranging Eq. (3.31) yields Eq. (3.27).

□

The following relationships involving the differentiation matrix are also important in the costate analysis of the Gauss pseudospectral transcription method, originally shown in Ref. 6.

Lemma 3. *[6] The elements of the differential approximation matrix, D, corresponding to the LG points are related to the elements D corresponding to the initial time, τ_0, by the equation*

$$D_{i,0} = -\sum_{k=1}^{K} D_{ik} \tag{3.32}$$

Proof of Lemma 3 (6). *This relationship can be shown by examining the derivative of a constant function, $f(t) = c$. Applying the differential approxmation matrix results in*

$$\dot{f}(t_i) = cD_{i,0} + c\sum_{k=1}^{K} D_{ik} = 0 \tag{3.33}$$

This relation is exact for all values of K because the function $f(t) = c$ is a polynomial of degree 0. *The result can be simplified to Eq. (3.32).*

□

A similar relationship can be drawn for the adjoint differentiation matrix, $D^{\dagger}$.

Lemma 4. *[6] The elements of the adjoint differential approximation matrix, $D^{\dagger}$, corresponding to the LG points are related to the elements $D^{\dagger}$ corresponding to the final time, τ_f, by the equation*

$$D^{\dagger}_{i,K+1} = -\sum_{k=1}^{K} D^{\dagger}_{ik} \tag{3.34}$$

Proof of Lemma 4 (6). *This relationship can also be shown by examining the derivative of a constant function, $f(t) = c$. See Lemma 3.*

□

3.2.1 KKT Conditions of the Transcribed NLP

The solution to a nonlinear program satisfies a set of first-order optimality conditions called the Karush-Kuhn-Tucker (KKT) conditions. The KKT conditions corresponding to the NLP described in Section 3.1.1 can be obtained using the augmented cost function or Lagrangian.[9] The augmented cost function is formed using the Lagrange multipliers $\tilde{\mathbf{\Lambda}}_k \in \mathbb{R}^n$, $\tilde{\boldsymbol{\mu}}_k \in \mathbb{R}^c$,

$k = 1, \ldots, K$, $\tilde{\boldsymbol{\Lambda}}_{\boldsymbol{F}} \in \mathbb{R}^n$, and $\tilde{\boldsymbol{\nu}} \in \mathbb{R}^q$ as

$$\begin{aligned} J_a = \ & \Phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k g(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) - \tilde{\boldsymbol{\nu}}^T \boldsymbol{\phi}(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) \\ & - \sum_{k=1}^{K} \tilde{\boldsymbol{\mu}}_k^T \mathbf{C}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) - \sum_{k=1}^{K} \tilde{\boldsymbol{\Lambda}}_k^T \left(\sum_{i=0}^{K} D_{ki}\mathbf{X}_i - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \right) \\ & - \tilde{\boldsymbol{\Lambda}}_{\boldsymbol{F}}^T \left(\mathbf{X}_f - \mathbf{X}_0 - \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \right) \end{aligned} \tag{3.35}$$

where the tilde notation is used to highlight multipliers related to the solution of the NLP. The KKT conditions are found by setting equal to zero the derivatives of the Lagrangian with respect to $\mathbf{X}_0$, $\mathbf{X}_k$, $\mathbf{X}_f$, $\mathbf{U}_k$, $\tilde{\boldsymbol{\Lambda}}_k$, $\tilde{\boldsymbol{\mu}}_k$, $\tilde{\boldsymbol{\Lambda}}_{\boldsymbol{F}}$, $\tilde{\boldsymbol{\nu}}$, t_0, and t_f.[9]

First, setting equal to zero the partial derivatives of the Lagrangian with respect to the Lagrange multipliers, $\tilde{\boldsymbol{\Lambda}}_k$, $\tilde{\boldsymbol{\mu}}_k$, $\tilde{\boldsymbol{\Lambda}}_{\boldsymbol{F}}$, and $\tilde{\boldsymbol{\nu}}$ results in Eqs. (3.11)–(3.14). Next, suppose

$$\mathbf{C}(\mathbf{x}, \mathbf{u}, \tau; t_0, t_f) = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_c \end{bmatrix} \tag{3.36}$$

Then, applying the complementary slackness condition,[9]

$$\begin{aligned} &\tilde{\mu}_{jk} = 0 \ , \text{ when } \quad C_{jk} < 0 \\ &\tilde{\mu}_{jk} \leq 0 \ , \text{ when } \quad C_{jk} = 0 \end{aligned} \qquad (j = 1, \cdots, c), \ (k = 1, \cdots, K) \tag{3.37}$$

where $C_{jk} \equiv C_j(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f)$ and $\tilde{\mu}_{jk}$ is the j^{th} component of $\tilde{\boldsymbol{\mu}}_k$.

The partial derivative of the augmented cost function with respect to the state at the k^{th} LG point, $\mathbf{X}_k$, $k = 1, \ldots, K$, is given as

$$\begin{aligned} \frac{\partial J_a}{\partial \mathbf{X}_k} = \ & \frac{t_f - t_0}{2} w_k \frac{\partial g_k}{\partial \mathbf{X}_k} - \tilde{\boldsymbol{\mu}}_k^T \frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k} - \sum_{i=1}^{K} D_{ik} \tilde{\boldsymbol{\Lambda}}_i^T + \\ & + \frac{t_f - t_0}{2} \tilde{\boldsymbol{\Lambda}}_k^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \frac{t_f - t_0}{2} w_k \tilde{\boldsymbol{\Lambda}}_F^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} \end{aligned} \tag{3.38}$$

where simplifying notation $g_k \equiv g(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f)$, $\mathbf{f}_k \equiv \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f)$, and $\mathbf{C}_k \equiv \mathbf{C}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_k)$ is used. Setting Eq. (3.38) equal to zero and dividing through by w_k results in

$$\mathbf{0}^T = \frac{t_f - t_0}{2}\left[\frac{\partial g_k}{\partial \mathbf{X}_k} - \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k} + \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k}\right] - \sum_{i=1}^{K} D_{ik}\frac{\tilde{\mathbf{\Lambda}}_i^T}{w_k} \tag{3.39}$$

Applying Lemma 2 to Eq. (3.39) and adding and subtracting the term $\sum_{i=1}^{K} D_{ki}^{\dagger}\tilde{\mathbf{\Lambda}}_F^T$, results in

$$\begin{aligned}\mathbf{0}^T = {} & \frac{t_f - t_0}{2}\left[\frac{\partial g_k}{\partial \mathbf{X}_k} - \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k} + \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k}\right] \\ & + \sum_{i=1}^{K} D_{ki}^{\dagger}\left(\frac{\tilde{\mathbf{\Lambda}}_i^T}{w_i} + \tilde{\mathbf{\Lambda}}_F^T\right) - \sum_{i=1}^{K} D_{ki}^{\dagger}\tilde{\mathbf{\Lambda}}_F^T\end{aligned} \tag{3.40}$$

Applying Lemma 4 to the last term in Eq. (3.40) results in

$$\begin{aligned}\mathbf{0}^T = {} & \frac{t_f - t_0}{2}\left[\frac{\partial g_k}{\partial \mathbf{X}_k} - \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k} + \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k}\right] \\ & + \sum_{i=1}^{K} D_{ki}^{\dagger}\left(\frac{\tilde{\mathbf{\Lambda}}_i^T}{w_i} + \tilde{\mathbf{\Lambda}}_F^T\right) + D_{k,K+1}^{\dagger}\tilde{\mathbf{\Lambda}}_F^T\end{aligned} \tag{3.41}$$

The next KKT condition is found by computing the partial derivative of the augmented cost function with respect to the control $\mathbf{U}_k$ as

$$\begin{aligned}\frac{\partial J_a}{\partial \mathbf{U}_k} = {} & \frac{t_f - t_0}{2} w_k \frac{\partial g_k}{\partial \mathbf{U}_k} - \tilde{\boldsymbol{\mu}}_k^T \frac{\partial \mathbf{C}_k}{\partial \mathbf{U}_k} \\ & + \frac{t_f - t_0}{2}\tilde{\mathbf{\Lambda}}_k^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k} + \frac{t_f - t_0}{2} w_k \tilde{\mathbf{\Lambda}}_F^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k}\end{aligned} \tag{3.42}$$

Setting Eq. (3.42) equal to zero and multiplying through by $\frac{2}{(t_f - t_0)w_k}$, gives

$$\mathbf{0}^T = \frac{\partial g_k}{\partial \mathbf{U}_k} + \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k} - \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{U}_k} \tag{3.43}$$

The next KKT condition is found by computing the partial derivative of the augmented cost function with respect to the initial state, $\mathbf{X}_0$, as

$$\frac{\partial J_a}{\partial \mathbf{X}_0} = \frac{\partial \Phi}{\partial \mathbf{X}_0} - \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial \mathbf{X}_0} - \sum_{k=1}^{K} D_{k,0} \tilde{\boldsymbol{\Lambda}}_k^T + \tilde{\boldsymbol{\Lambda}}_F^T \tag{3.44}$$

Applying Lemmas 2 and 3 to Eq. (3.44),

$$\mathbf{0}^T = \frac{\partial \Phi}{\partial \mathbf{X}_0} - \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial \mathbf{X}_0} - \sum_{k=1}^{K} \sum_{i=1}^{K} \frac{w_i}{w_k} D_{ik}^{\dagger} \tilde{\boldsymbol{\Lambda}}_k^T + \tilde{\boldsymbol{\Lambda}}_F^T \tag{3.45}$$

Adding and subtracting the term $\sum_{i=1}^{K} w_i \sum_{k=1}^{K} D_{ik}^{\dagger} \tilde{\boldsymbol{\Lambda}}_F$ in Eq. (3.45) yields

$$\begin{aligned} \mathbf{0}^T = & \frac{\partial \Phi}{\partial \mathbf{X}_0} - \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial \mathbf{X}_0} + \tilde{\boldsymbol{\Lambda}}_F^T \\ & - \sum_{i=1}^{K} w_i \left[\sum_{k=1}^{K} D_{ik}^{\dagger} \left(\frac{\tilde{\boldsymbol{\Lambda}}_k^T}{w_k} + \tilde{\boldsymbol{\Lambda}}_F^T \right) - \sum_{k=1}^{K} D_{ik}^{\dagger} \tilde{\boldsymbol{\Lambda}}_F^T \right] \end{aligned} \tag{3.46}$$

Applying Lemma 4 to Eq. (3.46) results in

$$\begin{aligned} \mathbf{0}^T = & \frac{\partial \Phi}{\partial \mathbf{X}_0} - \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial \mathbf{X}_0} + \tilde{\boldsymbol{\Lambda}}_F^T \\ & - \sum_{i=1}^{K} w_i \left[\sum_{k=1}^{K} D_{ik}^{\dagger} \left(\frac{\tilde{\boldsymbol{\Lambda}}_k^T}{w_k} + \tilde{\boldsymbol{\Lambda}}_F^T \right) + D_{i,K+1}^{\dagger} \tilde{\boldsymbol{\Lambda}}_F^T \right] \end{aligned} \tag{3.47}$$

Suppose now that the new variable, $\tilde{\boldsymbol{\Lambda}}_0$, is defined as

$$\tilde{\boldsymbol{\Lambda}}_0^T \equiv -\frac{\partial \Phi}{\partial \mathbf{X}_0} + \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial \mathbf{X}_0} \tag{3.48}$$

Then, replacing the term in the square brackets in Eq. (3.47) using (3.41) produces

$$\begin{aligned} \mathbf{0}^T = & -\tilde{\boldsymbol{\Lambda}}_0^T + \tilde{\boldsymbol{\Lambda}}_F^T \\ & + \frac{t_f - t_0}{2} \sum_{i=1}^{K} w_i \left[\frac{\partial g_i}{\partial \mathbf{X}_i} - \frac{2}{t_f - t_0} \frac{\tilde{\boldsymbol{\mu}}_i^T}{w_i} \frac{\partial \mathbf{C}_i}{\partial \mathbf{X}_i} + \left(\frac{\tilde{\boldsymbol{\Lambda}}_i^T}{w_i} + \tilde{\boldsymbol{\Lambda}}_F^T \right) \frac{\partial \mathbf{f}_i}{\partial \mathbf{X}_i} \right] \end{aligned} \tag{3.49}$$

The next KKT condition is found by taking the partial derivative of the augmented cost function with respect to the final state, $\mathbf{X}_f$, as

$$\mathbf{0}^T = \frac{\partial J_a}{\partial \mathbf{X}_f} = \frac{\partial \Phi}{\partial \mathbf{X}_f} - \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial \mathbf{X}_f} - \tilde{\boldsymbol{\Lambda}}_F^T \tag{3.50}$$

The last two KKT conditions are found by computing the partial derivatives of the augmented cost function with respect the initial and final times. First, computing the partial derivative of the augmented cost function with respect to the initial time results in

$$\begin{aligned}\frac{\partial J_a}{\partial t_0} = {} & \frac{\partial \Phi}{\partial t_0} - \frac{1}{2}\sum_{k=1}^{K} w_k g_k + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k \frac{\partial g_k}{\partial t_0} - \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial t_0} \\ & - \sum_{k=1}^{K} \tilde{\boldsymbol{\mu}}_k^T \frac{\partial \mathbf{C}_k}{\partial t_0} - \frac{1}{2}\sum_{k=1}^{K} \tilde{\boldsymbol{\Lambda}}_k^T \mathbf{f}_k + \frac{t_f - t_0}{2}\sum_{k=1}^{K} \tilde{\boldsymbol{\Lambda}}_k^T \frac{\partial \mathbf{f}_k}{\partial t_0} \\ & - \frac{1}{2}\sum_{k=1}^{K} w_k \tilde{\boldsymbol{\Lambda}}_F^T \mathbf{f}_k + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k \tilde{\boldsymbol{\Lambda}}_F^T \frac{\partial \mathbf{f}_k}{\partial t_0}\end{aligned} \tag{3.51}$$

Note that the functions g, $\mathbf{f}$, and $\mathbf{C}$ are now explicit functions of t_0 and t_f due to the transformation from $t \in [t_0, t_f]$ to $\tau \in [-1, 1]$ in Eq. (2.6). The term $\frac{1}{2}\sum_{k=1}^{K} w_k \frac{2}{t_f - t_0} \frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k} \mathbf{C}_k$ (which equals zero due to the complementary slackness condition of Eq. (3.37)) is added to Eq. (3.51). The resulting equation is

$$\begin{aligned}0 = {} & \frac{\partial \Phi}{\partial t_0} - \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial t_0} - \frac{1}{2}\sum_{k=1}^{K} w_k \left[g_k + \left(\frac{\tilde{\boldsymbol{\Lambda}}_k^T}{w_k} + \tilde{\boldsymbol{\Lambda}}_F^T \right) \mathbf{f}_k - \frac{2}{t_f - t_0} \frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k} \mathbf{C}_k \right] \\ & + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k \left[\frac{\partial g_k}{\partial t_0} + \left(\frac{\tilde{\boldsymbol{\Lambda}}_k^T}{w_k} + \tilde{\boldsymbol{\Lambda}}_F^T \right) \frac{\partial \mathbf{f}_k}{\partial t_0} - \frac{2}{t_f - t_0} \frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k} \frac{\partial \mathbf{C}_k}{\partial t_0} \right]\end{aligned} \tag{3.52}$$

Similarly, computing the partial derivative of the augmented cost function with respect to the final time produces

$$
\begin{aligned}
\frac{\partial J_a}{\partial t_f} = \quad & \frac{\partial \Phi}{\partial t_f} + \frac{1}{2}\sum_{k=1}^{K} w_k g_k + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k \frac{\partial g_k}{\partial t_f} - \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial t_f} \\
& - \sum_{k=1}^{K} \tilde{\boldsymbol{\mu}}_k^T \frac{\partial \mathbf{C}_k}{\partial t_f} + \frac{1}{2}\sum_{k=1}^{K} \tilde{\mathbf{\Lambda}}_k^T \mathbf{f}_k + \frac{t_f - t_0}{2}\sum_{k=1}^{K} \tilde{\mathbf{\Lambda}}_k^T \frac{\partial \mathbf{f}_k}{\partial t_f} \\
& + \frac{1}{2}\sum_{k=1}^{K} w_k \tilde{\mathbf{\Lambda}}_F^T \mathbf{f}_k + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k \tilde{\mathbf{\Lambda}}_F^T \frac{\partial \mathbf{f}_k}{\partial t_f}
\end{aligned}
\tag{3.53}
$$

Combining terms and subtracting the complementary slackness term results in

$$
\begin{aligned}
0 = \quad & \frac{\partial \Phi}{\partial t_f} - \tilde{\boldsymbol{\nu}}^T \frac{\partial \phi}{\partial t_f} + \frac{1}{2}\sum_{k=1}^{K} w_k \left[g_k + \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_f^T \right) \mathbf{f}_k - \frac{2}{t_f - t_0} \frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k} \mathbf{C}_k \right] \\
& + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k \left[\frac{\partial g_k}{\partial t_f} + \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_f^T \right) \frac{\partial \mathbf{f}_k}{\partial t_f} - \frac{2}{t_f - t_0} \frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k} \frac{\partial \mathbf{C}_k}{\partial t_f} \right]
\end{aligned}
\tag{3.54}
$$

The solution to the NLP of Eqs. (3.10)–(3.12) must satisfy the set of KKT conditions described below:

$$\sum_{i=0}^{K}\mathbf{X}_i D_{ki} = \frac{t_f - t_0}{2}\mathbf{f}_k \tag{3.55}$$

$$\begin{aligned}\sum_{i=1}^{K}\left(\frac{\tilde{\boldsymbol{\Lambda}}_i^T}{w_i} + \tilde{\boldsymbol{\Lambda}}_F^T\right) D_{ki}^{\dagger} + \tilde{\boldsymbol{\Lambda}}_F^T D_{k,K+1}^{\dagger} &= \\ \frac{t_f - t_0}{2}\left(-\frac{\partial g_k}{\partial \mathbf{X}_k} - \left(\frac{\tilde{\boldsymbol{\Lambda}}_k^T}{w_k} + \tilde{\boldsymbol{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k}\right)&\end{aligned} \tag{3.56}$$

$$\mathbf{0}^T = \frac{\partial g_k}{\partial \mathbf{U}_k} + \left(\frac{\tilde{\boldsymbol{\Lambda}}_k^T}{w_k} + \tilde{\boldsymbol{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k} - \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{U}_k} \tag{3.57}$$

$$\boldsymbol{\phi}(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) = 0 \tag{3.58}$$

$$\tilde{\boldsymbol{\Lambda}}_0^T = -\frac{\partial \Phi}{\partial \mathbf{X}_0} + \tilde{\boldsymbol{\nu}}^T\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}_0} \tag{3.59}$$

$$\tilde{\boldsymbol{\Lambda}}_F^T = \frac{\partial \Phi}{\partial \mathbf{X}_f} - \tilde{\boldsymbol{\nu}}^T\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}_f} \tag{3.60}$$

$$-\frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k\frac{\partial \tilde{\mathcal{H}}_k}{\partial t_0} + \frac{1}{2}\sum_{k=1}^{K} w_k\tilde{\mathcal{H}}_k = \frac{\partial \Phi}{\partial t_0} - \tilde{\boldsymbol{\nu}}^T\frac{\partial \boldsymbol{\phi}}{\partial t_0} \tag{3.61}$$

$$\frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k\frac{\partial \tilde{\mathcal{H}}_k}{\partial t_f} + \frac{1}{2}\sum_{k=1}^{K} w_k\tilde{\mathcal{H}}_k = -\frac{\partial \Phi}{\partial t_f} + \tilde{\boldsymbol{\nu}}^T\frac{\partial \boldsymbol{\phi}}{\partial t_f} \tag{3.62}$$

$$\mathbf{C}_k \leq \mathbf{0} \tag{3.63}$$

$$\tilde{\mu}_{jk} = 0\ , \ \text{when} \quad C_{jk} < 0 \tag{3.64}$$

$$\tilde{\mu}_{jk} \leq 0\ , \ \text{when} \quad C_{jk} = 0 \tag{3.65}$$

$$\mathbf{X}_f = \mathbf{X}_0 + \frac{(t_f - t_0)}{2}\sum_{k=1}^{K} w_k\mathbf{f}_k \tag{3.66}$$

$$\tilde{\boldsymbol{\Lambda}}_F^T = \tilde{\boldsymbol{\Lambda}}_0^T + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k\left(-\frac{\partial g_k}{\partial \mathbf{X}_k} - \left(\frac{\tilde{\boldsymbol{\Lambda}}_k^T}{w_k} + \tilde{\boldsymbol{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k}\right) \tag{3.67}$$

where the shorthand notation $g_k \equiv g(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f)$, $\mathbf{f}_k \equiv \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f)$, $\tilde{\mathcal{H}}_k \equiv \tilde{\mathcal{H}}(\mathbf{X}_k, \boldsymbol{\Lambda}_k, \boldsymbol{\mu}_k, \mathbf{U}_k, \tau_k; t_0, t_f)$, and $C_{jk} \equiv C_j(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f)$ is used. Note that the discrete augmented Hamiltonian, $\tilde{\mathcal{H}}_k$, is defined here as

$$\tilde{\mathcal{H}}_k \equiv g_k + \left(\frac{\tilde{\boldsymbol{\Lambda}}_k^T}{w_k} + \tilde{\boldsymbol{\Lambda}}_F^T\right)\mathbf{f}_k - \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\mathbf{C}_k \tag{3.68}$$

3.2.2 Gauss Pseudospectral Discretization of the First-Order Necessary Conditions

This section describes in detail the discretized equations resulting from the Gauss pseudospectral transcription of the first-order necessary conditions of the HBVP in Chapter 2. In practice this step is not necessary in order to solve the problem, but by transcribing the HBVP, one can extract a mapping between this set of equations and the equations in the previous section. The entire set of continuous-time necessary conditions of the transformed Bolza problem in Eq. (2.14) are discretized as follows. The state, control, and costate are approximated using Eqs. (3.1), (3.8), and (3.24), respectively. The continuous-time first-order optimality conditions of Eq. (2.14) are discretized using the variables $\mathbf{X}_0 \equiv \mathbf{X}(\tau_0)$, $\mathbf{X}_k \equiv \mathbf{X}(\tau_k) \in \mathbb{R}^n$, $(k = 1, \ldots, K)$, and $\mathbf{X}_f \equiv \mathbf{X}(\tau_f)$ for the state, $\mathbf{U}_k \equiv \mathbf{U}(\tau_k) \in \mathbb{R}^m$, $(k = 1, \ldots, K)$ for the control, $\mathbf{\Lambda}_0 \equiv \mathbf{\Lambda}(\tau_0)$, $\mathbf{\Lambda}_k \equiv \mathbf{\Lambda}(\tau_k) \in \mathbb{R}^n$, $(k = 1, \ldots, K)$, and $\mathbf{\Lambda}_f \equiv \mathbf{\Lambda}(\tau_f)$ for the costate, and $\boldsymbol{\mu}_k \equiv \boldsymbol{\mu}(\tau_k) \in \mathbb{R}^c$, $(k = 1, \ldots, K)$, for the Lagrange multiplier associated with the path constraints. The other unknown variables in the problem are the initial and final times, $t_0 \in \mathbb{R}$, $t_f \in \mathbb{R}$, and the Lagrange multiplier, $\boldsymbol{\nu} \in \mathbb{R}^q$. The total number of variables is then given as $(2n + m + c)K + 4n + q + 2$. These variables are used to discretize the continuous necessary conditions of Eq. (2.14). Recall that the derivative of the state is approximated using Lagrange polynomials based on $K + 1$ points consisting of the K LG points and the initial time, τ_0, while the derivative of the costate is approximated using Lagrange polynomials based on $K + 1$ points consisting of the K LG points and the final time, τ_f. The resulting algebraic equations that approximate the continuous necessary conditions are given as

$$\sum_{i=0}^{K} \mathbf{X}_i D_{ki} = \frac{t_f - t_0}{2} \mathbf{f}_k \tag{3.69}$$

$$\sum_{i=1}^{K} \boldsymbol{\Lambda}_i^T D_{ki}^{\dagger} + \boldsymbol{\Lambda}_f^T D_{k,K+1}^{\dagger} = \frac{t_f - t_0}{2} \left(-\frac{\partial g_k}{\partial \mathbf{X}_k} - \boldsymbol{\Lambda}_k^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \boldsymbol{\mu}_k^T \frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k} \right) \tag{3.70}$$

$$\mathbf{0}^T = \frac{\partial g_k}{\partial \mathbf{U}_k} + \boldsymbol{\Lambda}_k^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{U}_k} - \boldsymbol{\mu}_k^T \frac{\partial \mathbf{C}_k}{\partial \mathbf{U}_k} \tag{3.71}$$

$$\boldsymbol{\phi}(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) = 0 \tag{3.72}$$

$$\boldsymbol{\Lambda}_0^T = -\frac{\partial \Phi}{\partial \mathbf{X}_0} + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}_0} \tag{3.73}$$

$$\boldsymbol{\Lambda}_f^T = \frac{\partial \Phi}{\partial \mathbf{X}_f} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}_f} \tag{3.74}$$

$$-\frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \frac{\partial \mathcal{H}_k}{\partial t_0} + \frac{1}{2} \sum_{k=1}^{K} w_k \mathcal{H}_k = \frac{\partial \Phi}{\partial t_0} - \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial t_0} \tag{3.75}$$

$$\frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \frac{\partial \mathcal{H}_k}{\partial t_f} + \frac{1}{2} \sum_{k=1}^{K} w_k \mathcal{H}_k = -\frac{\partial \Phi}{\partial t_f} + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial t_f} \tag{3.76}$$

$$\mu_{jk} = 0\,, \text{ when } \quad C_{jk} < 0 \tag{3.77}$$

$$\mu_{jk} \leq 0\,, \text{ when } \quad C_{jk} = 0 \tag{3.78}$$

for $k = 1, \ldots, K$ and $j = 1, \ldots, c$. Note that the augmented Hamiltonian, $\mathcal{H}_k$, is defined as

$$\mathcal{H}_k \equiv g_k + \boldsymbol{\Lambda}_k^T \mathbf{f}_k - \boldsymbol{\mu}_k^T \mathbf{C}_k \tag{3.79}$$

The final two equations that are required are

$$\mathbf{X}_f = \mathbf{X}_0 + \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \mathbf{f}_k \tag{3.80}$$

$$\boldsymbol{\Lambda}_f^T = \boldsymbol{\Lambda}_0^T + \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \left(-\frac{\partial g_k}{\partial \mathbf{X}_k} - \boldsymbol{\Lambda}_k^T \frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \boldsymbol{\mu}_k^T \frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k} \right) \tag{3.81}$$

Equations (3.80) and (3.81) are required in order to link the initial and final state and costate with the interior points. The total number of equations in Eqs. (3.69)–(3.81) is $(2n + m + c)K + 4n + q + 2$ (the same as the number of unknown variables). The solution to these nonlinear algebraic equations is an indirect solution to the optimal control problem.

Note that Eqs. (3.75)-(3.76) were derived from the original necessary conditions using the relationships:

$$\mathcal{H}(t_0) = -\frac{t_f - t_0}{2}\int_{-1}^{1}\frac{\partial\mathcal{H}}{\partial t_0}d\tau + \frac{1}{2}\int_{-1}^{1}\mathcal{H}d\tau \tag{3.82}$$

$$\mathcal{H}(t_f) = \frac{t_f - t_0}{2}\int_{-1}^{1}\frac{\partial\mathcal{H}}{\partial t_f}d\tau + \frac{1}{2}\int_{-1}^{1}\mathcal{H}d\tau \tag{3.83}$$

3.2.3 Costate Mapping Theorem

A costate estimate for the Bolza problem results from a comparison of the discretized necessary conditions from Section 3.2.2 to the optimality or Karush-Kuhn-Tucker conditions of the nonlinear programming problem in Section 3.2.1. The theorem presented here is an expansion of the one developed in Ref. 6 and is published in Ref. 8.

Theorem 1 (Gauss Pseudospectral Costate Mapping Theorem). *The Karush-Kuhn-Tucker (KKT) conditions of the NLP are exactly equivalent to the discretized form of the continuous first-order necessary conditions of the Bolza problem when using the Gauss pseudospectral discretization. Furthermore a costate estimate at the initial time, final time, and the Gauss points can be found from the KKT multipliers,* $\tilde{\mathbf{\Lambda}}_k$, $\tilde{\boldsymbol{\mu}}_k$, $\tilde{\mathbf{\Lambda}}_F$, *and* $\tilde{\boldsymbol{\nu}}$,

$$\begin{gathered}\mathbf{\Lambda}_k = \frac{\tilde{\mathbf{\Lambda}}_k}{w_k} + \tilde{\mathbf{\Lambda}}_F \\ \boldsymbol{\mu}_k = \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k}{w_k} \\ \boldsymbol{\nu} = \tilde{\boldsymbol{\nu}} \\ \mathbf{\Lambda}_0 = \tilde{\mathbf{\Lambda}}_0\,, \quad \mathbf{\Lambda}(t_f) = \tilde{\mathbf{\Lambda}}_F\end{gathered} \tag{3.84}$$

Proof of Theorem 1. *Using the substitution of Eq. (3.84), it is seen that Eqs. (3.55)-(3.67) are exactly the same as Eqs. (3.69)-(3.81).*

□

Theorem 1 indicates that solving the NLP derived from the Gauss pseudospectral transcription of the optimal control problem is equivalent to applying the Gauss pseudospectral

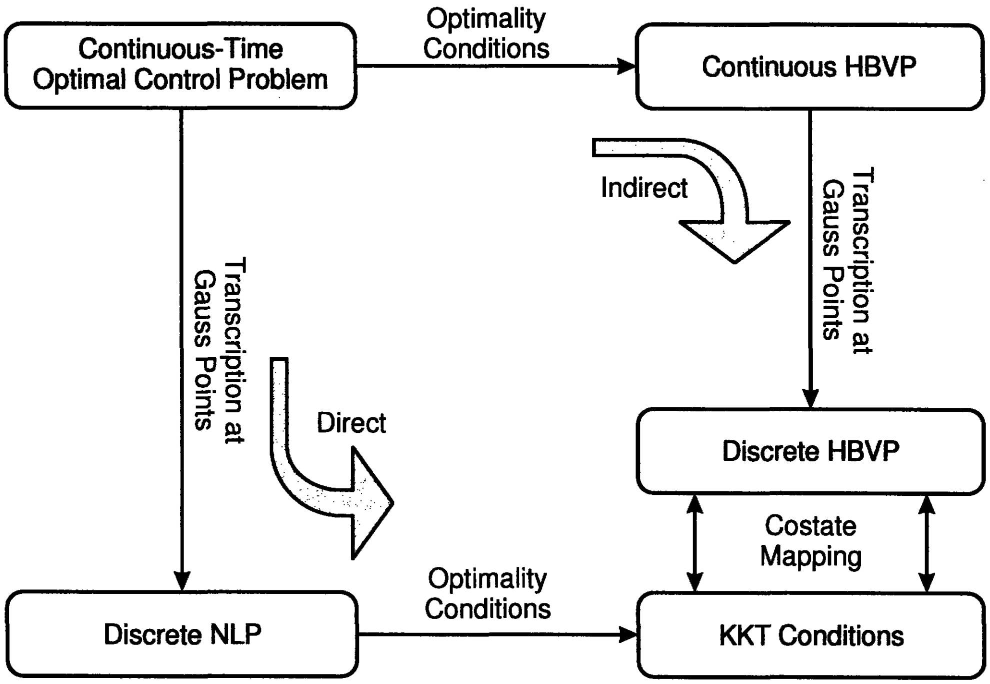


Figure 3-1: Flow diagram of the Gauss pseudospectral transcription (both direct and indirect approaches).

discretization to the continuous-time, first-order optimality conditions. Fig. 3-1 shows the solution path for both the direct and indirect methods. In the indirect path, the optimal control problem is converted into a continuous-time Hamiltonian boundary-value problem (HBVP) using the first order necessary conditions. The continuous-time HBVP is then discretized via pseudospectral transcription at the LG points. Conversely, along the direct path, the optimal control problem is discretized via the Gauss pseudospectral transcription at the LG points, which results in an NLP. The solution to the NLP satisfies the KKT conditions. The costate mapping relates *exactly* the KKT conditions from the direct method to the discretized necessary conditions from the indirect method. Note that in the proof of Theorem 1, it is assumed that Pontryagin's equation(Eq. (2.15)) is not necessary to determine the control.

To show equivalence, Theorem 1 uses the KKT multipliers $\tilde{\mathbf{\Lambda}}_k$, $\tilde{\boldsymbol{\mu}}_k$, $\tilde{\mathbf{\Lambda}}_F$, and $\tilde{\boldsymbol{\nu}}$, as well as a created variable, $\tilde{\mathbf{\Lambda}}_0$, defined in Eq. (3.48). Defining $\tilde{\mathbf{\Lambda}}_F$ in this manner makes the equivalence easier to see, but if one were to use Eq. (3.48) in practice, it would require formulating the transversality condition, which may involve complicated partial derivatives. Instead, it can be shown that an estimate for the initial costate can be computed directly from the other NLP multipliers, $\tilde{\mathbf{\Lambda}}_k$ and $\tilde{\mathbf{\Lambda}}_F$, thus eliminating the need for Eq. (3.48).

Lemma 5. *This relationship between the initial costate approximation and the NLP multipliers is:*

$$\tilde{\mathbf{\Lambda}}_0 = \tilde{\mathbf{\Lambda}}_F - \sum_{i=1}^{K} D_{i0}\tilde{\mathbf{\Lambda}}_i \tag{3.85}$$

Proof of Lemma 5. *This relationship can be derived by examining the KKT conditions in Section 3.2.1. It was shown in Eq. (3.56) that the following condition holds for the derivative of the augmented cost function with respect to the state at the Gauss points:*

$$\begin{aligned} &\sum_{i=1}^{K}\left(\frac{\tilde{\mathbf{\Lambda}}_i^T}{w_i} + \tilde{\mathbf{\Lambda}}_F^T\right) D_{ki}^{\dagger} + \tilde{\mathbf{\Lambda}}_F^T D_{k,K+1}^{\dagger} = \\ &\frac{t_f - t_0}{2}\left(-\frac{\partial g_k}{\partial \mathbf{X}_k} - \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k}\right), \quad (k = 1, \ldots, K) \end{aligned} \tag{3.86}$$

Using Lemmas 2 and 4 to convert $D^{\dagger}$ into D results in

$$\begin{aligned} &-\sum_{i=1}^{K}\left(\frac{\tilde{\mathbf{\Lambda}}_i^T}{w_i} + \tilde{\mathbf{\Lambda}}_F^T\right)\frac{w_i}{w_k} D_{ik} + \tilde{\mathbf{\Lambda}}_F^T \sum_{i=1}^{K}\frac{w_i}{w_k} D_{ik} = \\ &\frac{t_f - t_0}{2}\left(-\frac{\partial g_k}{\partial \mathbf{X}_k} - \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k}\right), \quad (k = 1, \ldots, K) \end{aligned} \tag{3.87}$$

Canceling like terms on the left-hand side of Eq. (3.87), results in

$$-\sum_{i=1}^{K}\frac{\tilde{\mathbf{\Lambda}}_i^T}{w_k} D_{ik} = \frac{t_f - t_0}{2}\left(-\frac{\partial g_k}{\partial \mathbf{X}_k} - \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k}\right), \quad (k = 1, \ldots, K) \tag{3.88}$$

Furthermore, it was shown in Eq. (3.67) that $\tilde{\mathbf{\Lambda}}_F$ and $\tilde{\mathbf{\Lambda}}_0$ are related as

$$\tilde{\mathbf{\Lambda}}_F^T = \tilde{\mathbf{\Lambda}}_0^T + \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k\left(-\frac{\partial g_k}{\partial \mathbf{X}_k} - \left(\frac{\tilde{\mathbf{\Lambda}}_k^T}{w_k} + \tilde{\mathbf{\Lambda}}_F^T\right)\frac{\partial \mathbf{f}_k}{\partial \mathbf{X}_k} + \frac{2}{t_f - t_0}\frac{\tilde{\boldsymbol{\mu}}_k^T}{w_k}\frac{\partial \mathbf{C}_k}{\partial \mathbf{X}_k}\right) \tag{3.89}$$

Eq. (3.88) can be substituted into Eq. (3.89) to obtain

$$\tilde{\mathbf{\Lambda}}_F^T = \tilde{\mathbf{\Lambda}}_0^T - \sum_{k=1}^{K} w_k \sum_{i=1}^{K}\frac{\tilde{\mathbf{\Lambda}}_i^T}{w_k} D_{ik} \tag{3.90}$$

which can be rewritten as

$$\tilde{\mathbf{\Lambda}}_F = \tilde{\mathbf{\Lambda}}_0 - \sum_{k=1}^{K}\sum_{i=1}^{K} \tilde{\mathbf{\Lambda}}_i D_{ik} \tag{3.91}$$

Note that $\tilde{\mathbf{\Lambda}}_i$ is independent of the summation over the index k. Consequently, Eq. (3.91) further simplifies to

$$\tilde{\mathbf{\Lambda}}_F = \tilde{\mathbf{\Lambda}}_0 - \sum_{i=1}^{K} \tilde{\mathbf{\Lambda}}_i \sum_{k=1}^{K} D_{ik} \tag{3.92}$$

Then, applying Lemma 3 gives

$$\tilde{\mathbf{\Lambda}}_F = \tilde{\mathbf{\Lambda}}_0 + \sum_{i=1}^{K} \tilde{\mathbf{\Lambda}}_i D_{i0} \tag{3.93}$$

Rearranging,

$$\tilde{\mathbf{\Lambda}}_0 = \tilde{\mathbf{\Lambda}}_F - \sum_{i=1}^{K} \tilde{\mathbf{\Lambda}}_i D_{i0} \tag{3.94}$$

□

This approximation for the initial costate has a very small computational cost associated with it, and is preferred in practice over the transversality condition of Eq. (3.48).

3.3 Convergence

The equivalence described in the previous section provides a way to relate the discrete direct transcription with the discrete indirect transcription of the optimal control problem. Yet what is still missing is a proof that shows that a solution to the discrete NLP converges to the optimal solution of the original, continuous-time optimal control problem. This is by no means a trivial proof, and is currently being studied by several researchers.[36,49,71]

Most discussions on convergence in the literature refer to the ability of a numerical integrator to converge to the true solution when integrating a system of differential equations. Euler, trapezoidal, and Runge-Kutta methods all have certain convergence properties.[2,44] However, convergence of an integration method does not necessarily mean that the same discretization scheme will converge when used to solve an optimal control problem. Hager[56] showed that a "convergent" Runge-Kutta method may not actually converge to the continuous optimal solution when used to solve an optimal control problem. Conversely, Betts[15] showed that a "nonconvergent" scheme could be used to solve inequality constrained optimal control problems. The inclusion of a cost function adds an interesting but frustratingly difficult aspect to convergence proofs for optimal control problems.

Kameswaran and Biegler[71,72] have attempted to show convergence for their pseudospectral method. Their method is essentially the Radau pseudospectral method implemented in a local fashion, meaning the trajectory is broken into segments and each segment is pseudospectrally approximated. By implementing the RPM in a local manner, they are able to use the more traditional approach to convergence, namely that the segment step size diminishes to zero in the limit. However, this formulation is limited in the sense that they are able to show convergence, but not "spectral" convergence.

For global pseudospectral approaches, there is only one segment, so some other approach to convergence must be used. Rather, one can show convergence as the number of discretization points increases to infinity. This approach is commonly seen in the literature for the convergence of methods used to solve partial differential equations.[23,120] Interestingly, there is a large body of research that discusses convergence of pseudospectral methods for boundary value problems involving PDEs.[51,52,61,83,108] Canuto proved in Ref. 23 that pseudospectral methods converge spectrally for elliptic boundary value problems when using Legendre polynomials. However, there is still no mention of solving optimal control problems using pseudospectral methods.

Recall that the Gauss pseudospectral method is able to show equivalence between the KKT conditions of the NLP and the first-order optimality conditions of the discretized HBVP. This HBVP is very similar to the boundary value problems that are formulated in Canuto's book, as initially discussed in Ref. 90. Currently, it remains unclear if it is possible to formulate the HBVP in a manner that can properly use Canuto's approach to prove convergence. However, as the connection between pseudospectral BVPs and pseudospectral discretizations to OCPs grows stronger with further study, a better understanding of convergence may be uncovered.

Lastly, Gong & Ross take an entirely new approach to convergence which seems to work quite well for showing the convergence of the primal solution.[48] This method is based on Polak's theory of consistent approximations and assumes that there exists a quantity called an *accumulation point.*[49] This assumption is verified by their formulation of the optimal control problem. However, they use this same assumption to show the convergence of the costate, yet there is no justification for a costate accumulation point in their problem formulation. Hence, the convergence of the primal solution is valid, but convergence of the dual solution (i.e., costate) is under dispute.

3.4 Summary

This chapter describes a reformulated version of the Gauss pseudospectral method originally proposed in Ref. 6. This differential form can handle a more general optimal control problem with inequality state and/or control path constraints. The key properties of the Gauss pseudospectral method have been identified. In particular, the KKT conditions from the NLP obtained via the Gauss pseudospectral discretization are derived and shown to be identical to the first-order optimality conditions of the general continuous-time optimal control problem discretized via the Gauss pseudospectral method. As a result, the KKT multipliers of the NLP can be used to obtain an accurate estimate of the costate and an updated costate mapping procedure is presented.

Chapter 4

Improving the Boundary Control

As mentioned in Chapter 3, the GPM discretizes the control only at the LG points and does *not* disctretize the control at either the initial or the terminal point. Consequently, the solution of the NLP defined in the previous chapter does not produce values of the controls at the boundaries. The ability to obtain accurate initial and terminal controls can be critical in many applications, particularly in precision guidance and real-time computations. Control schemes like model-predictive control[99] (MPC) implement only the first portion of the control profile before the optimal control profile is recomputed at the next time step. Consequently, the accuracy of the initial control when using MPC is extremely important. Thus at first glance, the lack of an explicit initial control for the Gauss pseudospectral method would limit the effectiveness of this method. This chapter discusses traditional methods and a novel approach to generating accurate values for the initial control.

In order to implement the control from an NLP, the discrete control must be transformed into a continuous-time control via an interpolation scheme. There are several interpolation techniques commonly used in practice. For example, direct methods such as the trapezoidal method or Hermite-Simpson method,[14] use piecewise polynomial splines[30] to represent the continuous-time control profile, where it is required that the spline approximation, $\tilde{\mathbf{U}}(t)$, equal the NLP control at the M grid points:

$$\tilde{\mathbf{U}}(t_k) = \mathbf{U}_k, \quad \forall k = 1, \dots, M \tag{4.1}$$

If the grid points do not involve the boundary points (as in the GPM), the lack of control information at the boundaries can be overcome simply via extrapolation of the control at the LG points. For example, the Radau pseudospectral method discretizes the control only at the Radau points, meaning that one of the boundary points (either the initial or final point) is not an explicit point in the discretization. As seen in Eq. (6.16), the continuous-time control is commonly determined from a Lagrange polynomial extrapolation[71,72] rather than a spline approximation.

So which interpolation scheme should be used to determine the boundary controls for the GPM? In fact, multiple reasons exist that suggest *any* polynomial extrapolation may not be the best approach. First, since the GPM discretization of the control assumes no particular functional form, it is unclear which extrapolation would be the best function to use. Seccond, any reasonable extrapolation of the control may violate a path constraint which, in general, will render the extrapolated control infeasible. Third, even if the extrapolated control is feasible, it will not satisfy the required optimality conditions at the boundaries (i.e., the control will be suboptimal with respect to the NLP). The previously mentioned traditional extrapolation approaches suffer from these deficiencies. Consequently, it is both practical and most rigorous to develop a systematic procedure to compute the boundary controls. The next section develops an approach to compute the boundary controls from the primal and dual solutions of the NLP arising from the Gauss pseudospectral method.

4.1 Algorithm for Computation of Boundary Controls

This section proposes an algorithm for the computation of the initial and final control. Rather than simply extrapolating a continuous-time approximation to the boundaries, as done in the traditional approaches, this algorithm utilizes the mathematical equivalence between the optimality conditions of the NLP and the HBVP to solve for the boundary control. Using accurate boundary state and costate estimates from the NLP, one can solve for the optimal control at the boundaries. Because the approach for computing the initial control is identical to the approach for the terminal control, this section focuses on the computation of just the initial control.

First, the augmented Hamiltonian, $\tilde{\mathcal{H}}_a$, for the continuous-time optimal control problem is defined (in shorthand notation) as

$$\tilde{\mathcal{H}}_a(\mathbf{x},\mathbf{u},\boldsymbol{\lambda},\boldsymbol{\mu}) \equiv g + \boldsymbol{\lambda}^T\mathbf{f} - \boldsymbol{\mu}^T\mathbf{C} \tag{4.2}$$

Recall that, from the minimum principle of Pontryagin, at every instant of time the optimal control is the control $\mathbf{u}^*(\tau) \in \mathcal{U}$ that satisfies the condition

$$\tilde{\mathcal{H}}_a(\mathbf{x}^*,\mathbf{u}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*) \leq \tilde{\mathcal{H}}_a(\mathbf{x}^*,\mathbf{u},\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*) \tag{4.3}$$

where $\mathcal{U}$ is the feasible control set. Consequently, for a given instant of time τ where $\mathbf{x}^*(\tau)$, $\boldsymbol{\lambda}^*(\tau)$, and $\boldsymbol{\mu}^*(\tau)$ are known, Eq. (4.3) is a constrained optimization problem in the $\mathbf{u}(\tau) \in \mathbb{R}^m$. In order to solve this constrained optimization optimization problem at the *initial* time, it is necessary to know $\mathbf{x}^*(\tau_0)$, $\boldsymbol{\lambda}^*(\tau_0)$, and $\boldsymbol{\mu}^*(\tau_0)$.

Consider now the information that can be obtained by solving the NLP associated with the GPM. In particular, the primal solution to the NLP produces $\mathbf{X}(\tau_0)$ while the dual solution to the NLP can be manipulated algebraically to obtain the initial costate, $\boldsymbol{\Lambda}(\tau_0)$. However, because the NLP does not evaluate that path constraint at the boundaries, there is no mapping to the Lagrange multiplier $\boldsymbol{\mu}(\tau_0)$. This apparent impediment can be overcome by applying the minimum principle in a manner somewhat different from that given in Eq. (4.3). In particular, let $\mathcal{H}$ be the *Hamiltonian* (not the augmented Hamiltonian), where $\mathcal{H}$ is defined as

$$\mathcal{H}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) \equiv g + \boldsymbol{\lambda}^T\mathbf{f} \tag{4.4}$$

It is seen in Eq. (4.4) that the term involving the path constraint is not included. The Hamiltonian can be used in place of the augmented Hamiltonian in Eq. (4.3) due to complementary slackness, which states that the term with the path constraint is zero at the optimal solution. The path constraint is instead incorporated into the feasible control set. In particular, let $\mathcal{V}_0$

$$\mathcal{V}_0 = \mathcal{U} \bigcap \mathcal{C}_0 \tag{4.5}$$

which is the intersection of the original set of feasible controls at time τ_0, denoted $\mathcal{U}$, with the set of all controls at time τ_0 that satisfy the inequality constraint of Eq. (3.14), denoted $\mathcal{C}_0$. Under the assumption that an optimal solution does exist, the set $\mathcal{V}$ must be nonempty. Then, using the values of $\mathbf{X}(\tau_0)$ and $\mathbf{\Lambda}(\tau_0)$, the following modified optimization problem in m variables $\mathbf{U}(\tau_0) \in \mathbb{R}^m$ can be solved to obtain the initial control, $\mathbf{U}(\tau_0)$:

$$\begin{array}{cl} \text{minimize} & \mathcal{H}(\mathbf{X}(\tau_0), \mathbf{U}(\tau_0), \mathbf{\Lambda}(\tau_0), \tau_0; t_0, t_f) \\ \mathbf{U}(\tau_0) \in \mathcal{V}_0 & \end{array} \tag{4.6}$$

It is noted that, because $\mathcal{V}_0$ is restricted by the inequality path constraint at τ_0, the solution of $\mathbf{U}(\tau_0)$ is equivalent to the solution of the following problem:

$$\begin{array}{cl} \text{minimize} & \mathcal{H}(\mathbf{X}(\tau_0), \mathbf{U}(\tau_0), \mathbf{\Lambda}(\tau_0), \tau_0; t_0, t_f) \\ \mathbf{U}(\tau_0) \in \mathcal{U} & \\ \text{subject to} & \\ & \mathbf{C}(\mathbf{X}(\tau_0), \mathbf{U}(\tau_0), \tau_0; t_0, t_f) \leq \mathbf{0} \end{array} \tag{4.7}$$

Interestingly, if the constraint is *active*, then the initial path constraint multiplier, $\tilde{\boldsymbol{\mu}}(\tau_0)$, will also be determined by the minimization problem of Eq. (4.7). Finally, as alluded to above, the control at the terminal time, $\mathbf{U}(\tau_f)$, can be obtained by solving the minimization problem of Eq. (4.7) at $\tau = \tau_f$, i.e.,

$$\begin{array}{cl} \text{minimize} & \mathcal{H}(\mathbf{X}(\tau_f), \mathbf{U}(\tau_f), \mathbf{\Lambda}(\tau_f), \tau_f; t_0, t_f) \\ \mathbf{U}(\tau_f) \in \mathcal{U} & \\ \text{subject to} & \\ & \mathbf{C}(\mathbf{X}(\tau_f), \mathbf{U}(\tau_f), \tau_f; t_0, t_f) \leq \mathbf{0} \end{array} \tag{4.8}$$

4.2 Applications of Boundary Control Algorithm

The minimization problem posed in the previous section is now applied to three example problems. The problems considered include a problem with no path constraints, a problem with equalty path constraints, and a multiple-phase launch vehicle problem with equality and inequality path constraints.

These examples employ three different control interpolation schemes to convert the discrete NLP control to continuous time. Recall that any interpolation scheme is valid, as long as it is collocated with the NLP control values at the K LG points. The first approach uses a summation of K Lagrange polynomials, similar to the state approximation, denoted as "Lagrange" in Figs. 4-2–4-4. The second approach uses piecewise polynomial splines, computed using the "interp1" command in MATLAB[77] with the "spline" option. The third technique uses the boundary control algorithm to compute the initial and final control, and then fits a summation of $K + 2$ Lagrange polynomials through those points. This approach is denoted as "Pontryagin" in Figs. 4-2–4-4. The accuracy of these three methods is compared, as well as the rate at which the accuracy improves as the number of nodes, N, increases (recall from Chapter 3 that $N = K + 2$).

Another metric to measure the accuracy of the control interpolation is to actually implement the continuous-time control according to the dynamic equations and then evaluate the resulting cost. A desirable control law would produce a propagated cost that is close to the analytic cost.

4.2.1 Unconstrained Problem

Consider the following optimal control problem with two states and one control:

$$\text{minimize} \quad J = x_2(t_f) \tag{4.9}$$

subject to the dynamic constraints

$$\begin{aligned} \dot{x}_1 &= \tfrac{1}{2}x_1 + u \\ \dot{x}_2 &= x_1^2 + \tfrac{1}{2}u^2 \end{aligned} \tag{4.10}$$

and the initial conditions

$$x_1(0) = 1, \quad x_2(0) = 0 \tag{4.11}$$

where $t_f = 5$. This problem was taken from Ref. 72, and the solution to this problem is shown in Fig. 4-1. The analytic solution is

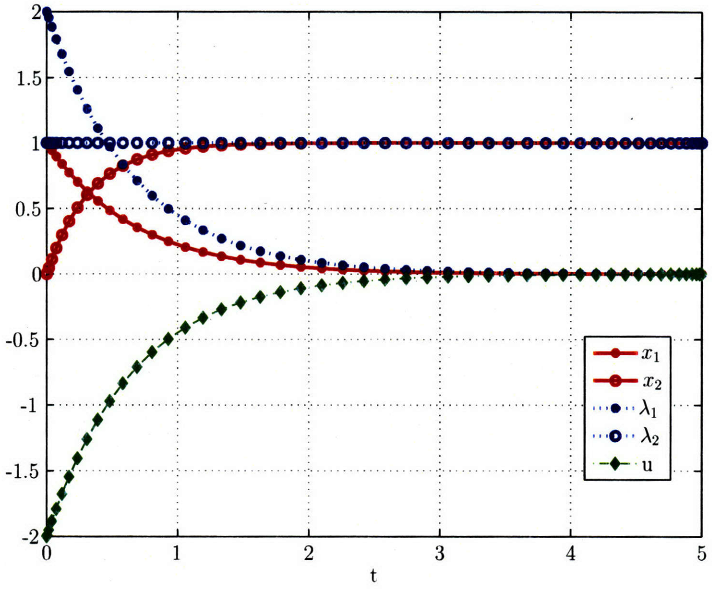


Figure 4-1: Solution for the unconstrained problem.

$$
\begin{aligned}
x_1(t) &= \frac{2}{2+e^3}e^{3/2t} + \frac{e^3}{2+e^3}e^{-3/2t} \\
x_2(t) &= \frac{2(e^{3t}-1)}{(2+e^3)^2} - \frac{e^6(e^{-3t}-1)}{(2+e^3)^2} \\
u(t) &= \frac{2}{2+e^3}e^{3/2t} - \frac{2e^3}{2+e^3}e^{-3/2t}
\end{aligned}
\tag{4.12}
$$

Fig. 4-2 depicts the error in the initial control for all three types of control interpolation as the number of nodes is increased. It is clear from the figure that using the boundary control algorithm ("pontryagin") improves both the accuracy of the initial control for a given number of nodes and the convergence rate. Similar results are shown for the final control in Fig. 4-3.

The error in the propagated cost is shown in Fig. 4-4. Again, the interpolation scheme involving the boundary control algorithm outperforms the other two approaches. Notice also that the spline interpolation method has a much slower convergence rate. This phenomenon is shown in the next example as well, and suggests that for some problems, a Lagrange interpolation scheme leads to more accurate solutions than the more common practice of using splines. Put in another light, this also suggests that, when using the boundary control algorithm, fewer nodes are needed to discretize the problem in order to get the same level of accuracy in the result.

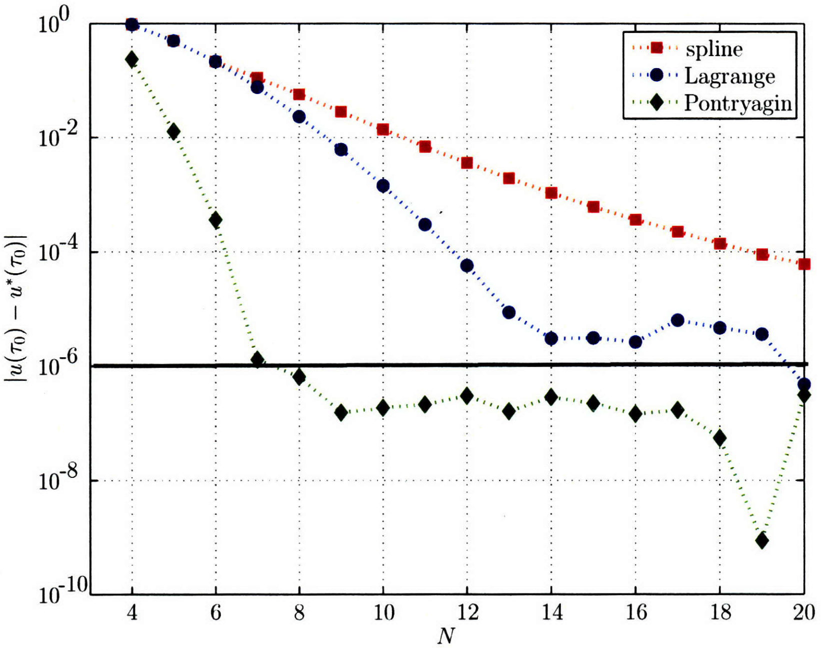


Figure 4-2: Error in the initial control as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin for the unconstrained problem.

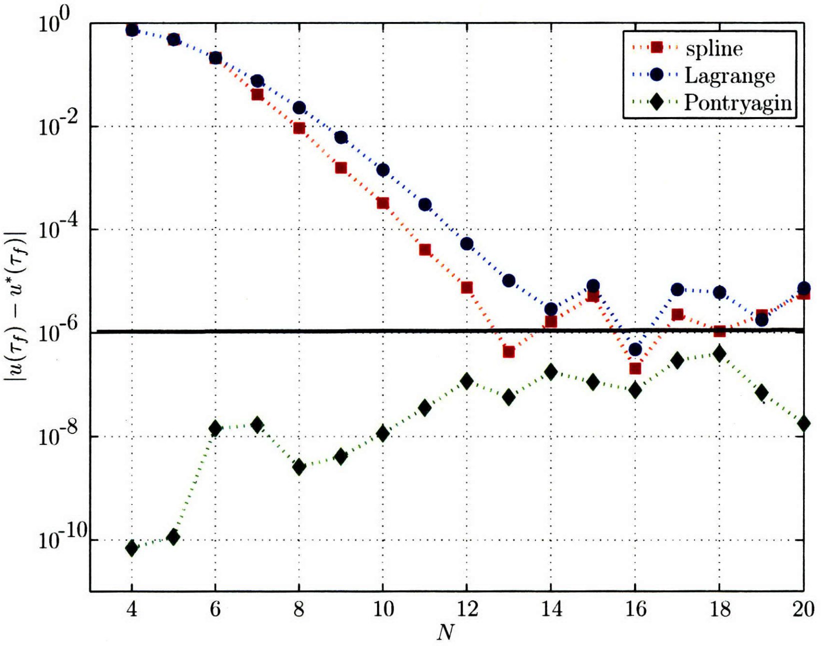


Figure 4-3: Error in the final control as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin for the unconstrained problem.

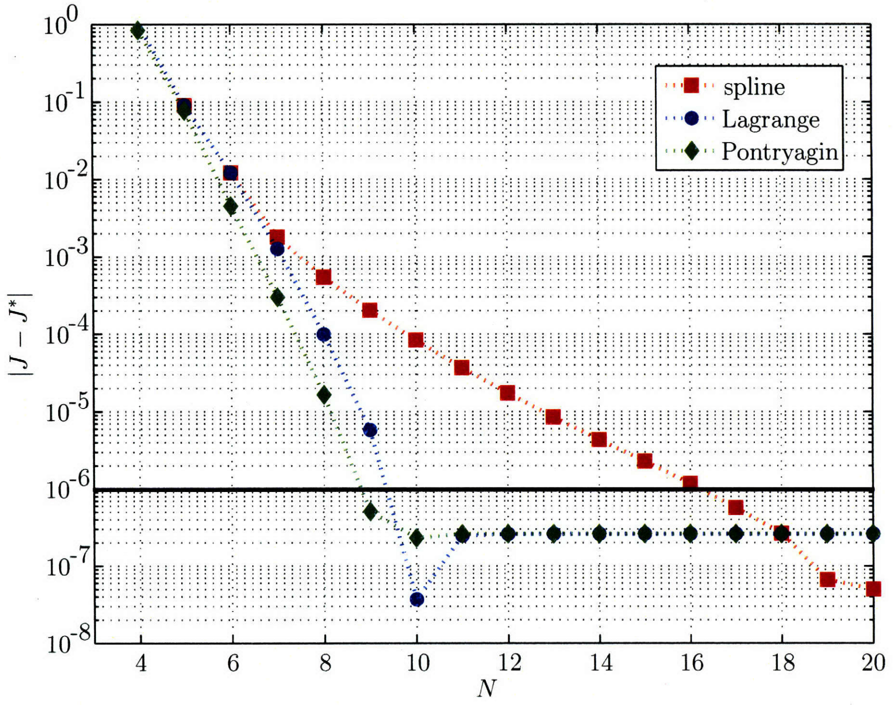


Figure 4-4: Error in the propagated cost as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin for the unconstrained problem.

4.2.2 Problem with a Path Constraint

Consider the following optimal control problem with three states and two controls:

$$\text{minimize} \quad J = -x_1(t_f) \tag{4.13}$$

subject to the dynamic constraints

$$\begin{aligned} \dot{x}_1 &= V u_1 \\ \dot{x}_2 &= V u_2 \\ \dot{V} &= a - g u_2 \end{aligned} \tag{4.14}$$

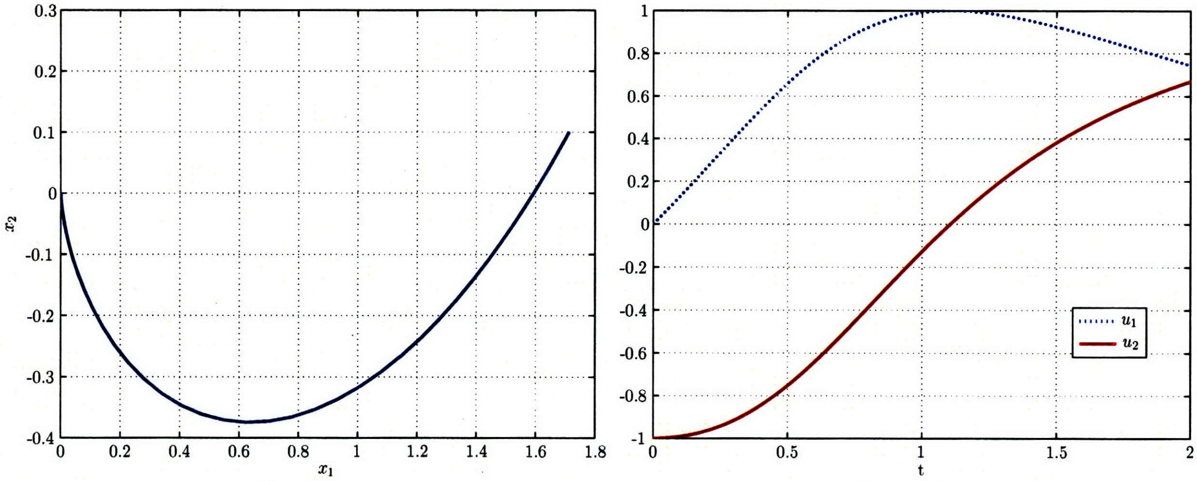
where

$$u_1^2 + u_2^2 = 1 \tag{4.15}$$

and the boundary conditions

$$\begin{aligned} x_1(0) = x_2(0) = V(0) = 0 \\ x_2(t_f) = 0.1 \end{aligned} \tag{4.16}$$

where $t_f = 2$, $g = 1$, and $a = 0.5g$. Commonly known as the "max range" problem, where x_1 is the downrange direction, this problem was taken from Ref.22. The NLP solution is compared to the boundary value problem solution using "bvp4c" in Matlab.[77] The state solution is shown in Fig. 4-5(a). The control solution is presented in Fig. 4-5(b).



(a) State Solution

(b) Control Solution

Figure 4-5: Solution for the path constrained problem.

Since there are two elements of control in this problem, Figs. 4-8 and 4-9 depict the error in the initial controls as the number of nodes is increased for all three types of control interpolation. Again, the boundary control algorithm improves the initial control by several orders of magnitude. Similar results are shown for the final controls in Figs. 4-10 and 4-11.

Recall in the boundary control algorithm presented in Section 4.1, the control path constraint is enforced at the boundaries as well. If one were to simply extrapolate the control using either splines or Lagrange polynomials, there would be no guarantee that the path constraint would be satisfied, and therefore one could potentially end up with an infeasible initial or final control. The boundary control algorithm guarantees a feasible boundary control. Figs. 4-6 and 4-7 show that for small values of N, the path constraint is indeed violated at the boundaries if "spline" or "Lagrange" extrapolation is used. However,

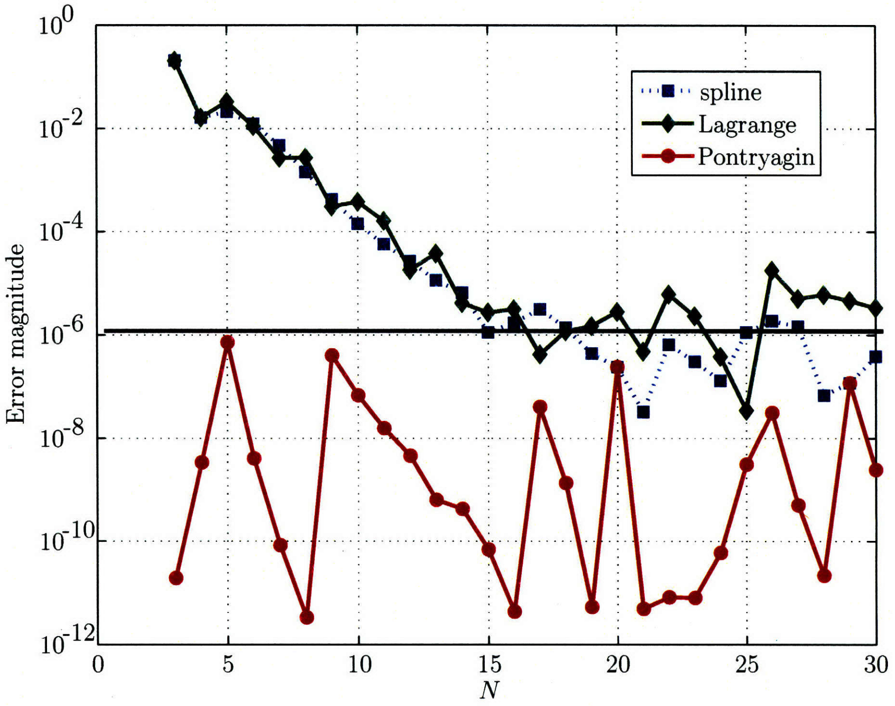


Figure 4-6: Error in the equality path constraint at τ_0 as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin in the path constrained problem.

by using the "Pontryagin" approach presented in this chapter, the path constraint is always satisfied at the boundaries, regardless of the number of nodes used.

The error in the propagated cost is shown in Fig. 4-12. As seen in Example 1, the boundary control algorithm ("Pontryagin") is the best performer, although for this example, the improvement over the standard Lagrange interpolation is slight. However, the spline interpolation method significantly underperforms, suggesting that splines are not the best interpolation scheme to use for this type of problem.

[h]

4.2.3 Launch Vehicle Ascent Problem

The problem considered in this section is the guidance for a Delta III launch vehicle during its ascent to orbit. The objective is to maneuver the launch vehicle from the ground to the target orbit while maximizing the remaining fuel in the upper stage. This problem has been rigorously studied and results are documented in Ref. 6.

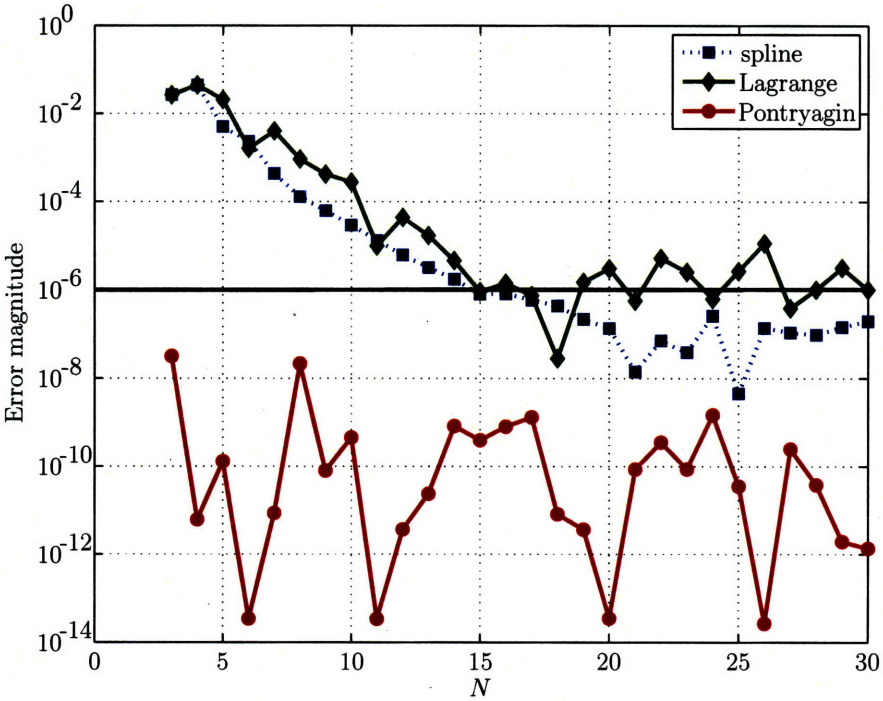


Figure 4-7: Error in the equality path constraint at τ_f as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin in the path constrained problem.

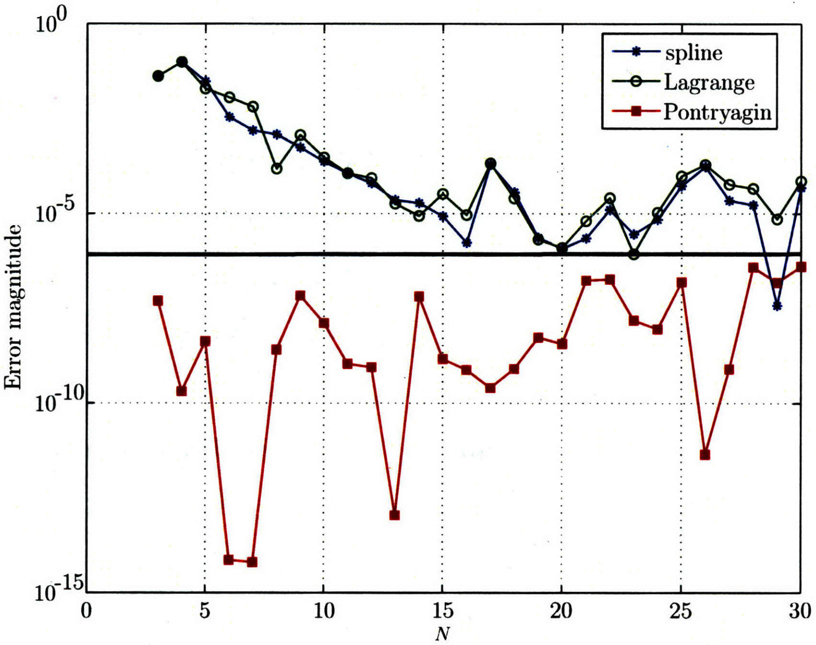


Figure 4-8: Error in the initial control, $u_1(t_0)$, as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin in the path constrained problem.

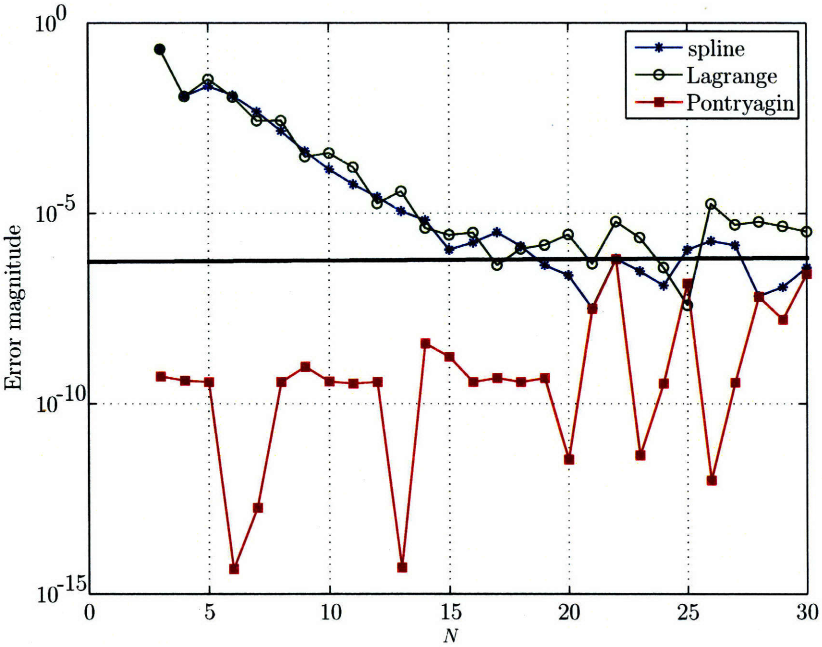


Figure 4-9: Error in the initial control, $u_2(t_0)$, as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin in the path constrained problem.

Vehicle Properties

The Delta III expendable launch vehicle has two stages along with nine strap-on solid rocket boosters. The flight of the vehicle can be divided into four distinct phases. The first phase begins with the rocket at rest on the ground and at time t_0, the main engine and six of the nine solid boosters ignite. When the boosters are depleted at time t_1, their remaining dry mass is ejected. The final three boosters are ignited, and along with the main engine, represent the thrust for phase two of the flight. These boosters are then ejected once their fuel is exhausted at time t_2, and the main engine alone creates the thrust for phase three. The fourth phase begins when the main engine fuel has been exhausted (MECO) and the dry mass associated with the main engine is ejected at time t_3. The thrust during phase four is from a second stage, which burns until the target orbit has been reached (SECO) at time t_4, thus completing the trajectory. The specific characteristics of these rocket motors can be seen in Table 4.1. Note that the solid boosters and main engine burn for their entire duration (meaning t_1, t_2, and t_3 are fixed), while the second stage engine is shut off when

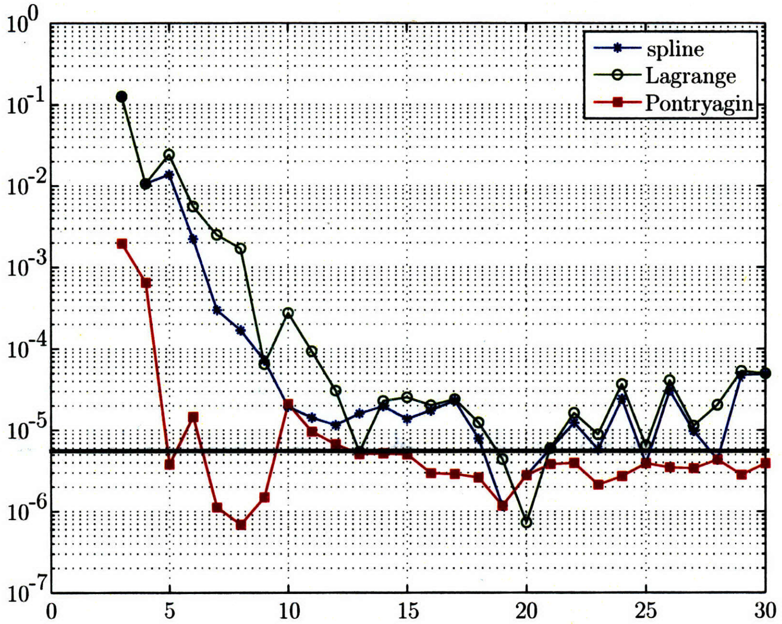


Figure 4-10: Error in the final control, $u_1(t_f)$, as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin in the path constrained problem.

the target orbit is achieved (t_4 is free).

Table 4.1: Properties of the Delta III launch vehicle

	Solid Boosters	Stage 1	Stage 2
Total Mass (kg)	19290	104380	19300
Propellant Mass (kg)	17010	95550	16820
Engine Thrust (N)	628500	1083100	110094
Isp (sec)	284	301.7	462.4
Number of Engines	9	1	1
Burn Time (sec)	75.2	261	700

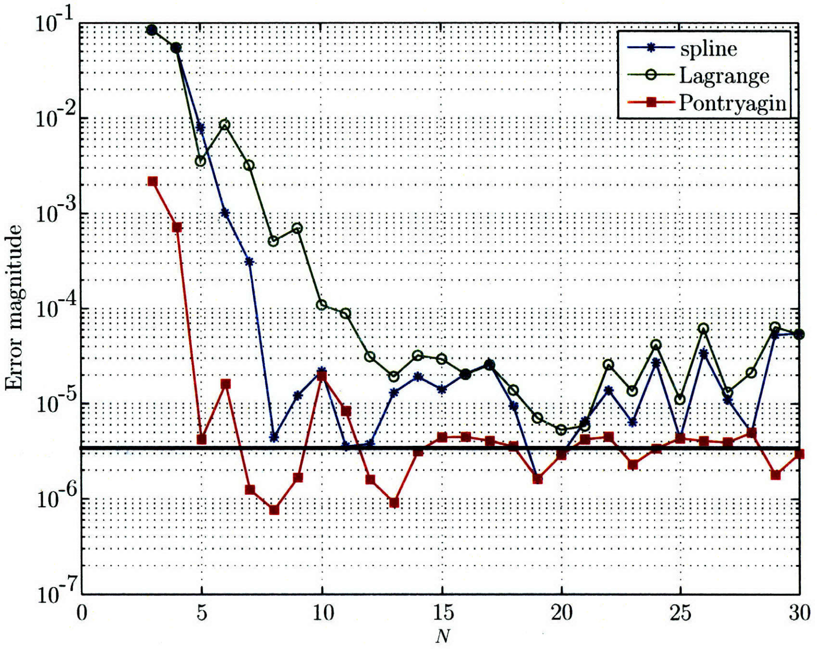


Figure 4-11: Error in the final control, $u_2(t_f)$, as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin in the path constrained problem.

Dynamic Model

The equations of motion for a non-lifting point mass in flight over a spherical rotating planet are expressed in Cartesian Earth centered inertial (ECI) coordinates as

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{\mu}{\|\mathbf{r}\|^3}\mathbf{r} + \frac{T}{m}\mathbf{u} + \frac{\mathbf{D}}{m} \\ \dot{m} &= -\frac{T}{g_0 I_{sp}} \end{aligned} \tag{4.17}$$

where $\mathbf{r}(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}^T$ is the position, $\mathbf{v} = \begin{bmatrix} v_x(t) & v_y(t) & v_z(t) \end{bmatrix}^T$ is the Cartesian ECI velocity, μ is the gravitational parameter, T is the vacuum thrust, m is the mass, g_0 is the acceleration due to gravity at sea level, I_{sp} is the specific impulse of the engine, $\mathbf{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$ is the thrust direction, and $\mathbf{D} = \begin{bmatrix} D_x & D_y & D_z \end{bmatrix}^T$ is the drag force. The drag force is defined as

$$\mathbf{D} = -\frac{1}{2} C_D A_{ref} \rho \|\mathbf{v}_{rel}\| \mathbf{v}_{rel} \tag{4.18}$$

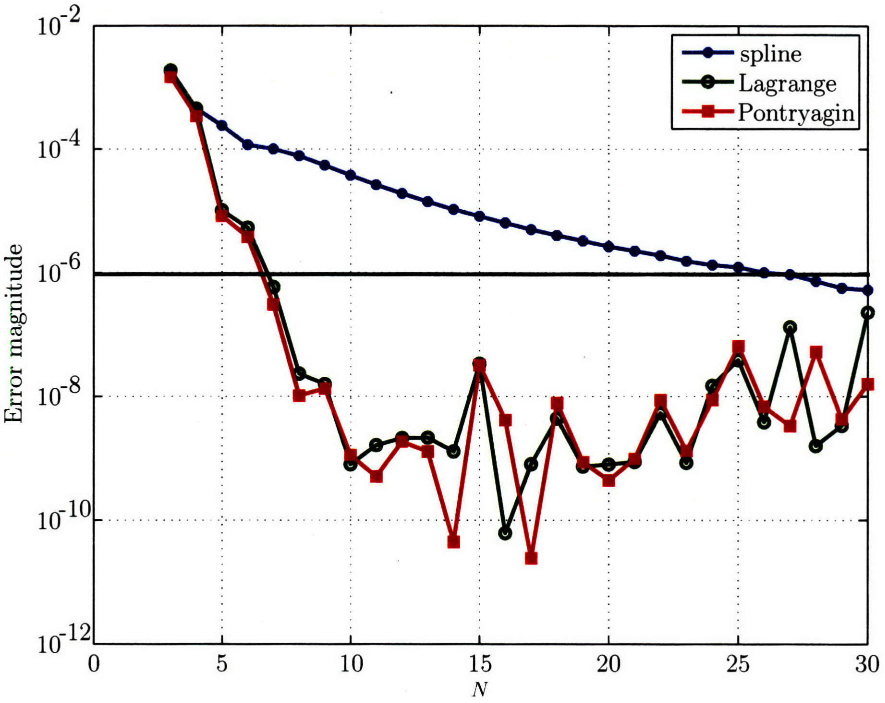


Figure 4-12: Error in the propagated cost as the number of nodes is increased for a spline extrapolation, a Lagrange extrapolation, or by using Pontryagin for the path constrained problem.

where C_D is the drag coefficient, A_{ref} is the reference area, ρ is the atmospheric density, and $\mathbf{v}_{rel}$ is the Earth relative velocity, where $\mathbf{v}_{rel}$ is given as

$$\mathbf{v}_{rel} = \mathbf{v} - \boldsymbol{\omega} \times \mathbf{r} \tag{4.19}$$

where $\boldsymbol{\omega}$ is the angular velocity of the Earth relative to inertial space. The atmospheric density is modeled as the exponential function

$$\rho = \rho_0 e^{-h/h_0} \tag{4.20}$$

where ρ_0 is the atmospheric density at sea level, $h = \|\mathbf{r}\| - R_e$ is the altitude, R_e is the equatorial radius of the Earth, and h_0 is the density scale height. The numerical values for these constants are found in Table 4.2.

Table 4.2: Constants used in the Delta III launch vehicle problem

Constant	Value
Payload Mass (kg)	4164
A_{ref} (m^2)	4π
C_d	0.5
ρ_0 (kg/m^3)	1.225
h_0 (km)	7.2
t_1 (s)	75.2
t_2 (s)	150.4
t_3 (s)	261
R_e (km)	6378.14
V_E (km/s)	7.905

Constraints

The launch vehicle starts on the ground at rest (relative to the Earth) at time t_0, so that the ECI initial conditions are

$$\begin{aligned} \mathbf{r}(t_0) &= \mathbf{r}_0 = \begin{bmatrix} 5605.2 & 0 & 3043.4 \end{bmatrix}^T \quad \text{km} \\ \mathbf{v}(t_0) &= \mathbf{v}_0 = \begin{bmatrix} 0 & 0.4076 & 0 \end{bmatrix}^T \quad \text{km/s} \\ m(t_0) &= m_0 = 301454 \quad \text{kg} \end{aligned} \tag{4.21}$$

which corresponds to the Cape Canaveral launch site. The terminal constraints define the target geosynchronous transfer orbit (GTO), which is defined in orbital elements as

$$\begin{aligned} a_f &= 24361.14\,\text{km}, \\ e_f &= 0.7308, \\ i_f &= 28.5\,\text{deg}, \\ \Omega_f &= 269.8\,\text{deg}, \\ \omega_f &= 130.5\,\text{deg} \end{aligned} \tag{4.22}$$

The orbital elements, a, e, i, Ω, and ω represent the semi-major axis, eccentricity, inclination, right ascension of the ascending node (RAAN), and argument of perigee, respectively. Note that the true anomaly, ν, is left undefined since the exact location within the orbit is not constrained. These orbital elements can be transformed into ECI coordinates via the transformation, T_{o2c} as shown in Ref. 114.

In addition to the boundary constraints, there exists both a state path constraint and a control path constraint in this problem. A state path constraint is imposed to keep the vehicle's altitude above the surface of the Earth, so that

$$|\mathbf{r}| \geq R_e \tag{4.23}$$

where R_e is the radius of the Earth, as seen in Table 4.2. Next, a path constraint is imposed on the control to guarantee that the control vector is unit length, so that

$$||\mathbf{u}||_2 = 1 \tag{4.24}$$

Lastly, each of the four phases in this trajectory is linked to the adjoining phases by a set of linkage conditions. These constraints force the position and velocity to be continuous and also account for the mass ejections, as

$$\begin{array}{rcll} \mathbf{r}^{(p)}(t_f) - \mathbf{r}^{(p+1)}(t_0) & = & \mathbf{0}, & \\ \mathbf{v}^{(p)}(t_f) - \mathbf{v}^{(p+1)}(t_0) & = & \mathbf{0}, & \qquad (p = 1, \ldots, 3) \\ m^{(p)}(t_f) - m_{dry}^{(p)} - m^{(p+1)}(t_0) & = & 0 & \end{array} \tag{4.25}$$

where the superscript (p) represents the phase number.

The optimal control problem is then to find the control, $\mathbf{u}$, that minimizes the cost function

$$J = -m^{(4)}(t_f) \tag{4.26}$$

subject to the conditions of Eqs. (4.17), (4.21), (4.22), (4.23), and (4.24). Lastly, it is noted that this problem is scaled canonically in order to make the NLP more tractable. Distances are scaled by the radius of the Earth, R_e, velocities are scaled by the Schuler period (the circular orbit velocity at the radius of the Earth, V_E), and mass is scaled by the total vehicle mass.

Results

This optimization problem was transcribed into an NLP via the Gauss pseudospectral transcription of Section 3.1.1 using 20 nodes in each phase. The NLP was then solved in MATLAB using the new software GPOCS[86] in conjunction with SNOPT.[45] The optimal altitude profile is shown in Fig. 4-13. The burn duration of the final stage is 665.3 seconds out of a possible 700 seconds, resulting in 668.2 kg of fuel remaining in the second-stage tank. The profiles for the three components of control are shown in Fig. 4-14.

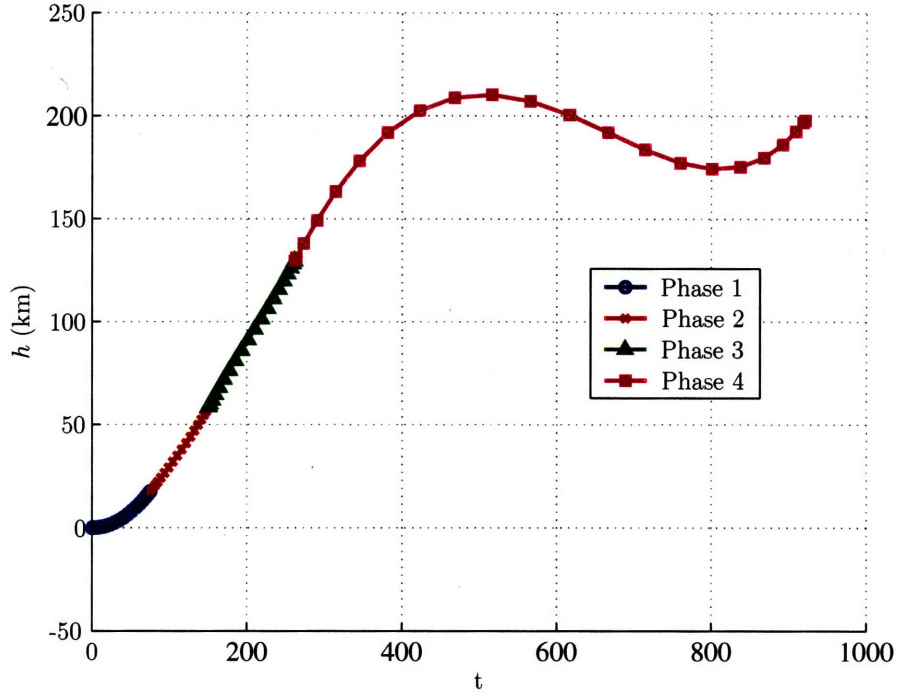


Figure 4-13: Optimal altitude profile for the Delta III launch vehicle example.

There is no analytic solution to this problem, so the NLP solution cannot be compared to an exact solution. However, it appears from Fig. 4-14 that the control profile should be continuous. Since there are no linkage constraints that require the control to be continuous at the phase boundaries, the extrapolation techniques can be examined to see which ones are closer to being continuous at the phase boundaries. For this example, two approaches are used. First, the control in each phase is extrapolated to the boundaries using a spline extrapolation. Second, the control at the boundaries is computed using the approach of Section 4.1. At the three interior phase boundaries (t_1, t_2, and t_3), the extrapolated final

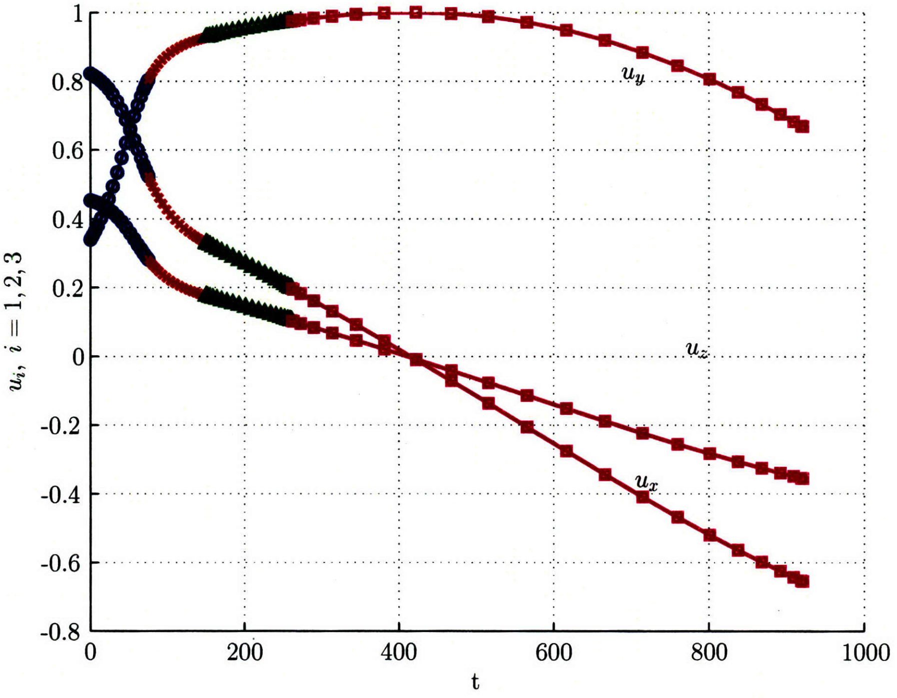


Figure 4-14: Profile of the three components of the optimal control for the Delta III launch vehicle example.

control from one phase is compared to the extrapolated initial control of the next phase, or $|u_i(t_k)^- - u_i(t_k)^+|$, where u_i, $(i = 1, 2, 3)$ are the three components of control and t_k, $(k = 1, 2, 3)$ are the three phase interfaces. The difference in extrapolated controls are shown for the spline extrapolation in Fig. 4-15 versus the number of nodes per phase. A similar graph is shown for the Pontryagin extrapolated controls in Fig. 4-16. In the spline extrapolation, the difference between the extrapolated controls is roughly 10^{-3} when using 10 nodes, and gradually becomes more accurate as the number of nodes increases. Using Pontryagin, on the other hand, produces extremely similar boundary controls ($\sim 10^{-6}$) for small numbers of nodes, which is a significant improvement over the spline approach. The results of Fig. 4-16 hover around the optimality tolerance of 10^{-6} for all the simulations tested.

4.3 Summary

This chapter described a method to obtain more accurate information about the control at the boundary through post-processing. A post-optimality computation is performed where

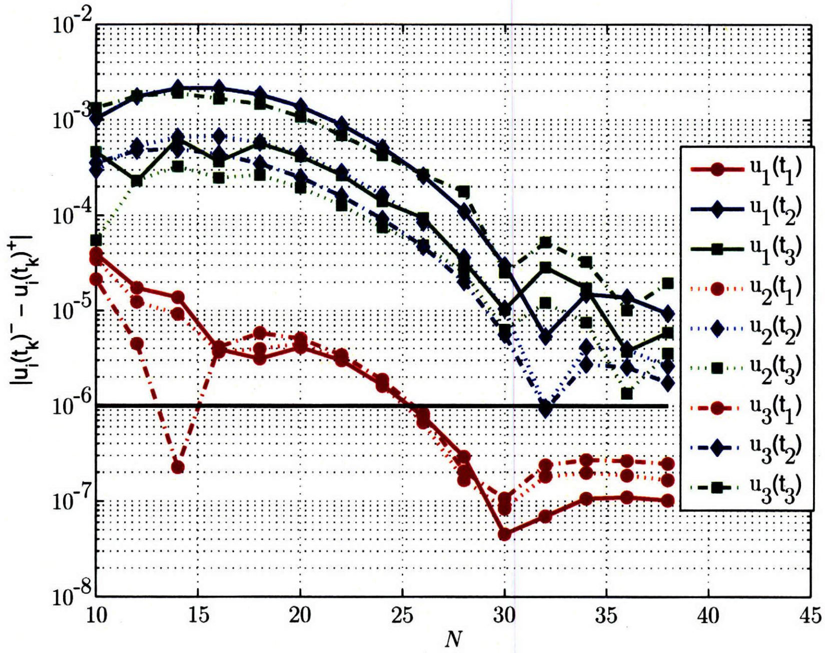


Figure 4-15: The difference in spline-extrapolated control, $|u_i(t_k)^- - u_i(t_k)^+|$, at all three interior phase interfaces in the launch vehicle example. Some differences are significantly greater than the optimality tolerance of $1e-6$.

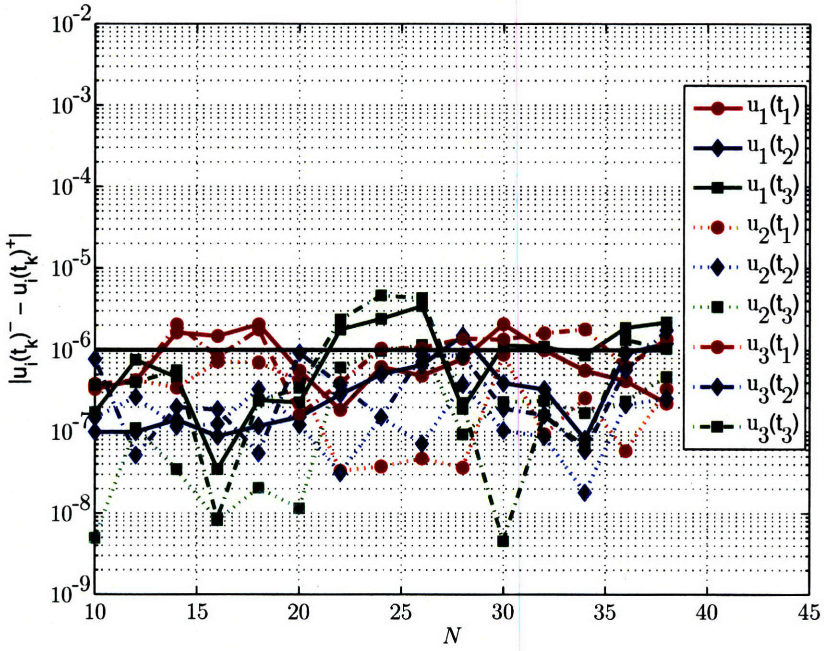


Figure 4-16: The difference in Pontryagin-extrapolated control, $|u_i(t_k)^- - u_i(t_k)^+|$, at all three interior phase interfaces in the launch vehicle example. Almost all differences lie below the optimality tolerance of $1e-6$.

the Pontryagin minimum principle is applied at the boundary points in order to obtain a more accurate control approximation at the initial and final times. The results show signifcant improvement in simulation when compared with a control approximation that uses various extrapolation techniques to compute the boundary control.

[This page intentionally left blank.]

Chapter 5

Local versus Global Orthogonal Collocation Methods

Orthogonal collocation methods are traditionally global in nature, meaning that a global interpolating polynomial spans the entire problem interval. However, orthogonal collocation methods have been used locally,[27,72] meaning the problem interval is divided into segments and a "global" interpolating polynomial spans the width of each segment. The segments are then connected by various linkage conditions, similar to the concept of *phases* presented in Chapter 3. The conceptual difference between phases and segments are that phase interfaces usually represent a physical event (such as a mass drop in a launch vehicle problem), whereas segment interfaces are simply a construct of the numerical approximation procedure, and tend to be more numerous. Furthermore, phases may occur in parallel, while segments are sequential in the independent variable (as seen in Chapter 7).

This chapter compares the theory and implementation of a global and local approach for the Gauss pseudospectral method. Proponents of the global approach tout the "spectral accuracy" (see Chapter 2) of the method for smooth problems, while proponents of the local approach claim the benefits of "local support". These two approaches are compared on two example problems. The first example is a smooth problem with nonlinear dynamics and a quadratic cost, while the second example is a "bang-bang" problem that is discontinuous in the control. Each method is evaluated on accuracy of the NLP solution, convergence rate, and computational efficiency.

5.1 Local GPM Approach: Segments

First, the description of the Gauss pseudospectral transcription of Chapter 3 is slightly modified to reflect the partitioning of the optimal control problem into segments. This local application of the GPM is a similar approach to that found in Refs. 26, 71, 104. The continuous Bolza problem of Eqs. (2.7)–(2.10) is transcribed to a nonlinear programming problem using the *local* Gauss pseudospectral method as follows. First, the original time interval $I = [t_0, t_f]$ is divided into S sub-intervals (or *segments*) I_s, $(s = 1, \ldots, S)$ such that $I_s = [t_{s-1}, t_s]$ and

$$\begin{aligned} \bigcap_{s=1}^{S} I_s &= \emptyset \\ \bigcup_{s=1}^{S} I_s &= I \end{aligned}$$

where $\emptyset$ is the empty set. Within each sub-interval, the time, $t^{(s)}$, is transformed to the interval $[-1, 1]$ via the affine transformation

$$\tau^{(s)} = \frac{2t^{(s)}}{t_s - t_{s-1}} - \frac{t_s + t_{s-1}}{t_s - t_{s-1}} \tag{5.1}$$

where $\tau^{(s)} \in [-1, 1]$ and denotes the transformed time in I_s. Each sub-interval I_s, is discretized according to the GPM. For example, the state is approximated using a basis of $N+1$ Lagrange interpolating polynomials, $\mathcal{L}_i(\tau)$, $(i = 0, \ldots, K)$, as in Eq. (3.1):

$$\mathbf{x}^{(s)}(\tau) \approx \mathbf{X}^{(s)}(\tau) = \sum_{i=0}^{K} \mathbf{X}^{(s)}(\tau_i)\mathcal{L}_i(\tau) \tag{5.2}$$

where K is the number of Legendre-Gauss points. The resulting NLP is as follows: Minimize

$$J = \Phi(\mathbf{X}_0^{(1)}, t_0^{(1)}, \mathbf{X}_f^{(S)}, t_f^{(S)}) + \sum_{s=1}^{S} \left[\frac{t_f^{(s)} - t_0^{(s)}}{2} \sum_{k=1}^{K} w_k^{(s)} g(\mathbf{X}_k^{(s)}, \mathbf{U}_k^{(s)}, \tau_k^{(s)}; t_{s-1}, t_s) \right] \tag{5.3}$$

subject to the dynamic constraint

$$\sum_{i=0}^{K} D_{ki}^{(s)} \mathbf{X}_i^{(s)} - \frac{t_f^{(s)} - t_0^{(s)}}{2} \mathbf{f}(\mathbf{X}_k^{(s)}, \mathbf{U}_k^{(s)}, \tau_k^{(s)}; t_{s-1}, t_s) = \mathbf{0}, \qquad (k = 1, \ldots, K), \quad (s = 1, \ldots, S) \tag{5.4}$$

and the Gauss quadrature

$$\mathbf{X}_f^{(s)} \equiv \mathbf{X}_0^{(s)} + \frac{t_f^{(s)} - t_0^{(s)}}{2} \sum_{k=1}^{K} w_k^{(s)} \mathbf{f}(\mathbf{X}_k^{(s)}, \mathbf{U}_k^{(s)}, \tau_k^{(s)}; t_{s-1}, t_s), \quad (s = 1, \ldots, S) \tag{5.5}$$

where $\mathbf{X}_k^{(s)} \equiv \mathbf{X}(\tau_k^{(s)}) \in \mathbb{R}^n$ and $\mathbf{U}_k^{(s)} \equiv \mathbf{U}(\tau_k^{(s)}) \in \mathbb{R}^m$. Next, the boundary constraint of Eq. (2.9) is expressed as

$$\boldsymbol{\phi}(\mathbf{X}_0^{(1)}, t_0^{(1)}, \mathbf{X}_f^{(S)}, t_f^{(S)}) = \mathbf{0} \tag{5.6}$$

Furthermore, the path constraint of Eq. (2.10) is evaluated at the LG points in each segment as

$$\mathbf{C}(\mathbf{X}_k^{(s)}, \mathbf{U}_k^{(s)}, \tau_k^{(s)}; t_{s-1}, t_s) \leq \mathbf{0} \quad (k = 1, \ldots, K), \ (s = 1, \ldots, S) \tag{5.7}$$

Lastly, the state can be constrained to be continuous across segment boundaries by including the following linkage conditions:

$$\boldsymbol{L}_{(s)}^{(s+1)}(\mathbf{X}_f^{(s)}, t_f^{(s)}, \mathbf{X}_0^{(s+1)}, t_0^{(s+1)}) = \mathbf{0}, \quad (s = 1, \ldots, S - 1) \tag{5.8}$$

The cost function of Eq. (5.3) and the algebraic constraints of Eqs. (5.4), (5.5), (5.6), (5.7), and (5.8) define an NLP whose solution is an approximate solution to the continuous Bolza problem.

In the case where *one* sub-interval is used (i.e., $S = 1$), the GPM is employed as a global collocation method. Conversely, in this analysis it is assumed that any case where $S \gg 1$, to be a local application of the method (although, strictly speaking, this is untrue). In the latter case, the sub-intervals are connected via linkage conditions on the state, the independent variable, and, possibly, the control. The analysis that follows compares the GPM in both global collocation (i.e., $S = 1$) and local collocation (i.e., $S \gg 1$) for a given number of total discretization points.

5.2 Global and Local Applications of the GPM

In this comparison, the Gauss pseudospectral method is applied as both a global and a local collocation method on two examples. As a global collocation method, the number of segments is set to unity (i.e., $S = 1$) and the accuracy is assessed as a function of the number of collocation points, N. As a local collocation method, the number of collocation points in a given sub-interval is fixed and the number of sub-intervals is varied. In the local approach, five nodes (three LG points plus two boundary points) are chosen for each sub-interval and the width of each sub-interval is constrained to be equal. Fig. 5-1 depicts the distribution of nodes and collocation points for both the global and local approaches using a total of 20 discretization points. Recall from Chapter 2 that the LG points are more dense near the endpoints than in the middle, which is readily apparent in Fig. 5-1. It is emphasized that, for both approaches, once the number of LG points per interval is determined, the location of each point inside the interval is immediately fixed. All examples in this chapter were solved using the TOMLAB® version[59] of the sparse NLP solver SNOPT[45] with default tolerances and comparable initial guesses for both the local and global approaches. A Pentium 4, 3.2 GHz desktop computer was used for all computations and any documented computation times are in reference to this machine.

5.2.1 Modified Hicks-Ray Reactor Problem

Consider the Hicks-Ray reactor problem[104] where it is desired to minimize the quadratic cost functional

$$\int_0^{10} \left[a_1(C(t) - \bar{C})^2 + a_2(T(t) - \bar{T})^2 + a_3(U(t) - \bar{U})^2\right] dt \tag{5.9}$$

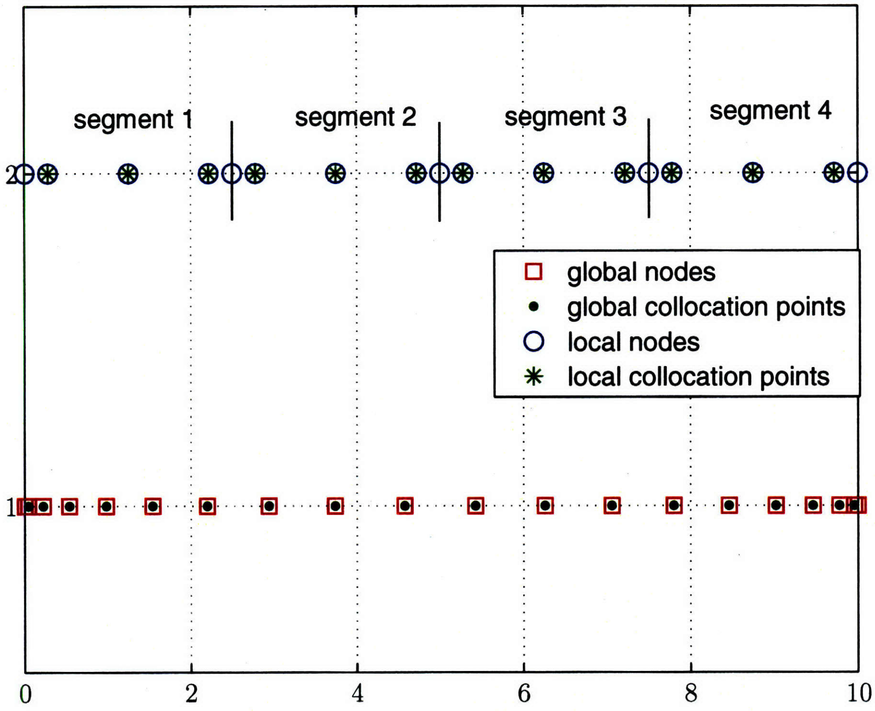


Figure 5-1: Distribution of nodes and collocation points for both the global and local approaches ($N = 20$).

subject to the nonlinear dynamic constraints

$$\begin{aligned}\dot{C}(t) &= \frac{1 - C(t)}{\theta} - k_{10}C(t)e^{-t/T(t)} && (5.10)\\ \dot{T}(t) &= \frac{y_f - T(t)}{\theta} + k_{10}C(t)e^{-t/T(t)} && (5.11)\\ & -\alpha(T(t) - y_c)U(t) && (5.12)\end{aligned}$$

and the initial conditions

$$\begin{aligned}C(0) &= 0.1367\\ T(0) &= 0.7293\end{aligned} \qquad (5.13)$$

The two states are denoted by $C(t)$ and $T(t)$, the control is denoted by $U(t)$, and all other parameters are constants. This problem has no analytic solution, however, this analysis compares the NLP solutions to a highly accurate extremal solution of the Hamiltonian boundary-value problem (HBVP) associated with Eqs. (5.9)–(5.13). This HBVP solution was found using the MATLAB boundary-value problem solver BVP4C[77] and is shown in Fig. 5-2. Finally, it is noted that the traditional form of this example includes the (active) path constraint $U(t) \geq 0$. However, due to the increased difficulty in accurately solving a path-constrained HBVP and the fact that this is simply an illustrative example, the path constraint was omitted for this analysis.

The results obtained by solving the Hicks-Ray reactor problem using GPM as both a global and a local collocation method are shown in Figs. 5-3 and 5-4, respectively. The "error" in these figures represent the maximum absolute error between the NLP state and the HBVP state across the discretized trajectory for a given number of nodes, N. Fig. 5-3 shows the error for the global approach as the number of nodes increases. It is seen from Fig. 5-3 that the error decreases at a spectral rate[43] as N increases and attains a minimum just below the feasibility tolerance of 10^{-6} for discretizations with at least 21 nodes. This rapid convergence is typical for pseudospectral methods and is one of the primary motivations for using a global approach. As for the local approach, Fig. 5-4 shows the error in the state as a function of the number of segments where five nodes (three LG points) were used per segment. The state errors using local collocation converge at a much slower rate than seen in the global approach. Notice that the data shown for the global approach end at 30 nodes while the data for the local approach extend to 25 segments, or 125 total nodes. For comparison, the vertical line in Fig. 5-4 represents the largest number of nodes used in the global scenario. Thus for 30 total nodes, the global method has an accuracy better than 10^{-6} (Fig. 5-3), while the local method produces an accuracy of $\sim 10^{-3}$ (Fig. 5-4). Often in practice, it is desirable to know how many nodes are required to achieve a certain level of accuracy. The data in this example show that to achieve an accuracy of 10^{-5}, one would need 16 nodes in the global approach and more than 125 total nodes in the local approach.

Although the global method outperforms the local method in accuracy, one must also consider the computational efficiency between the two approaches. First, it is noted that, for the same number of nodes, the global method results in a much more dense constraint Jacobian as compared to the local method since the collocation equations are a function of the state and control across the entire interval in the global method. However, as seen from Fig. 5-4, for a certain required accuracy, the local approach needs many more collocation points as compared to the global approach. This results in a much larger, albeit sparser Jacobian for the local approach. Table 5.1 summarizes the performance of each method for a desired accuracy. Specifically, Table 5.1 shows that, for a desired accuracy, the local method requires significantly more computational effort as compared to the global approach both in terms of CPU time and major iterations of the NLP solver. Table 5.1 also shows that the larger, sparser Jacobian of the local approach results in a higher number of nonzero elements than the smaller, more dense counterpart in the global approach, which is the likely contributer to the higher computational expense. This result is not intuitive, as one might expect the sparse NLP solver to perform faster on the large, sparse approach. For this example, the extremely high accuracy of global polynomials outweighs the more numerous but relatively inexpensive computations of the local approach.

Table 5.1: Computational expense for both the global and local approach in the Hicks-Ray reactor problem

			Global Approach			
Accuracy	Nodes Required	Non-Zeros	Jacobian Density (%)	CPU Time (s)	Major Iterations	Minor Iterations
1e-3	10	248	46	1.343	29	45
1e-4	12	348	44	1.234	31	55
1e-5	16	596	41	5.047	113	144
1e-6	21	996	40	6.531	115	171
			Local Approach			
Accuracy	Nodes Required	Non-Zeros	Jacobian Density (%)	CPU Time (s)	Major Iterations	Minor Iterations
1e-3	35	512	6.6	8.890	43	100
1e-4	80	1178	2.8	49.86	56	174
1e-5	165	2436	1.4	148.3	44	285

5.2.2 Problem with Discontinuous Control

In this problem, it is desired to minimize the cost functional

$$J = t_f \tag{5.14}$$

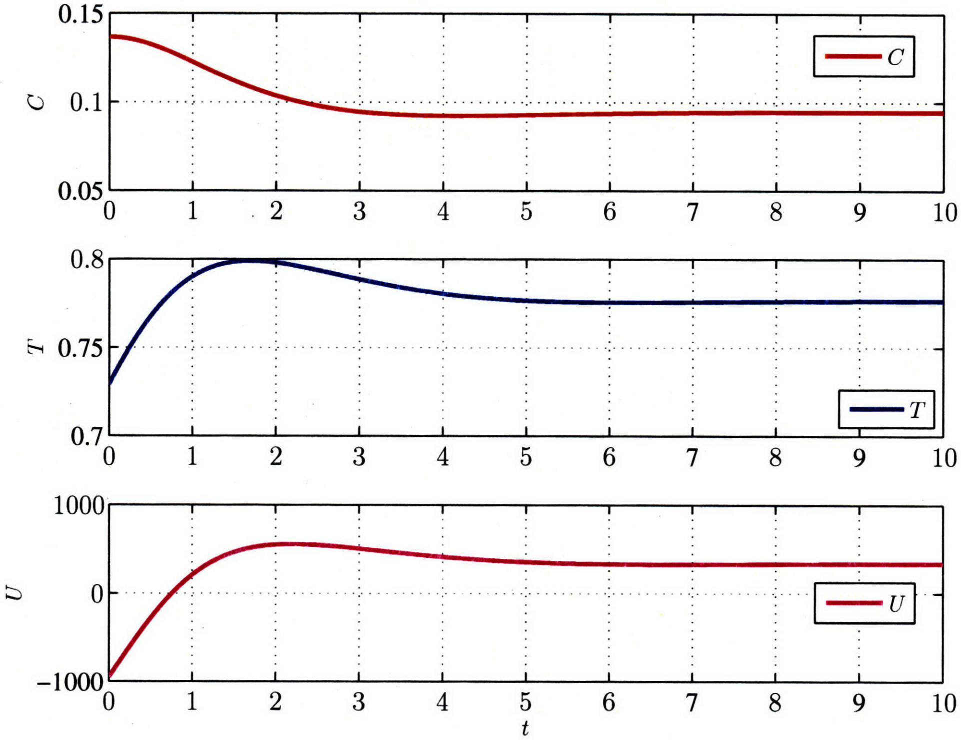


Figure 5-2: BVP solution to the Hicks-Ray reactor problem.

subject to the dynamic constraints

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= u(t) \end{aligned} \tag{5.15}$$

the boundary conditions

$$\begin{aligned} x_1(0) &= x_{10} \quad ; \quad x_2(0) = x_{20} \\ x_1(t_f) &= 0 \quad\;\; ; \quad x_2(t_f) = 0 \end{aligned} \tag{5.16}$$

and the inequality control constraint

$$|u(t)| \leq 1 \tag{5.17}$$

The key difference between this example and the previous example is that, for a wide variety of initial conditions, the optimal control for this problem is "bang-bang", i.e., it switches from either its maximum value to its minimum value (or vice-versa) at a time t_b (where

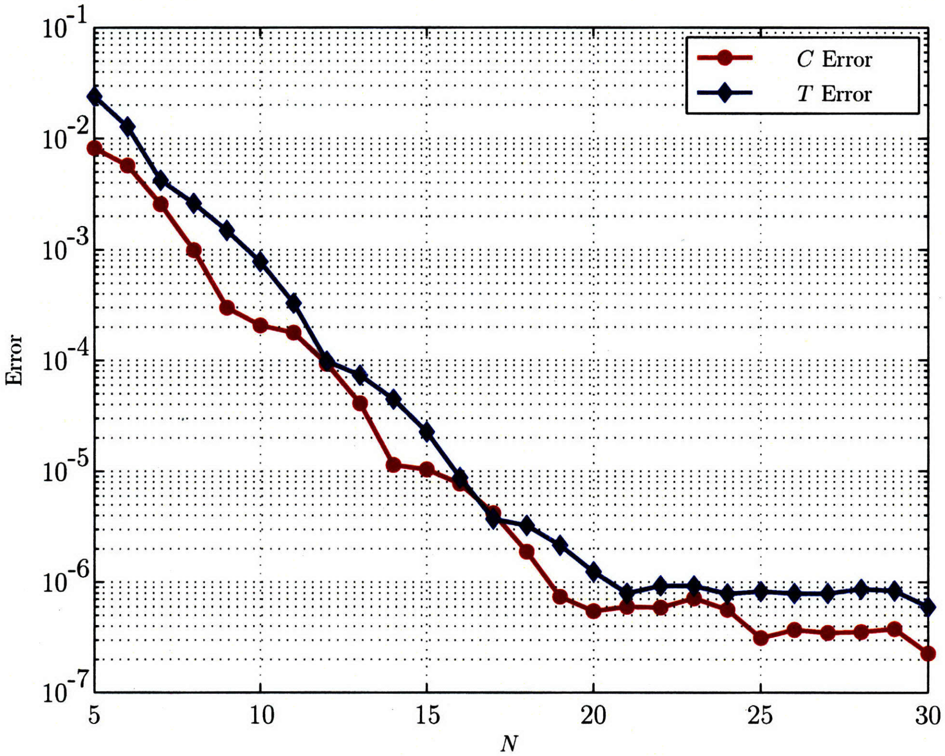


Figure 5-3: Convergence of Hicks-Ray Problem via global approach as a function of the number of nodes, N.

$0 < t_b < t_f$). Because of the discontinuity in the optimal control, it would appear at first glance that a global approximation would be much less accurate than a local approximation. However, the two methods are surprisingly comparable in their solution accuracy for this problem.

Example 2 Solved without Knowledge of Discontinuity

The discontinuity in the control can drastically affect the accuracy of each approach depending on where the discontinuity is located. In order to fully assess the accuracy of solutions, two different sets of initial conditions are considered. The first set of initial conditions is as follows:

$$x_{10} = 3, \qquad x_{20} = 2 \tag{5.18}$$

For the initial conditions of Eq. (5.18) the optimal switch time of the control is $t_b \approx 4.2361$, which is designed to lie on the interior of a segment in both the global and local approaches,

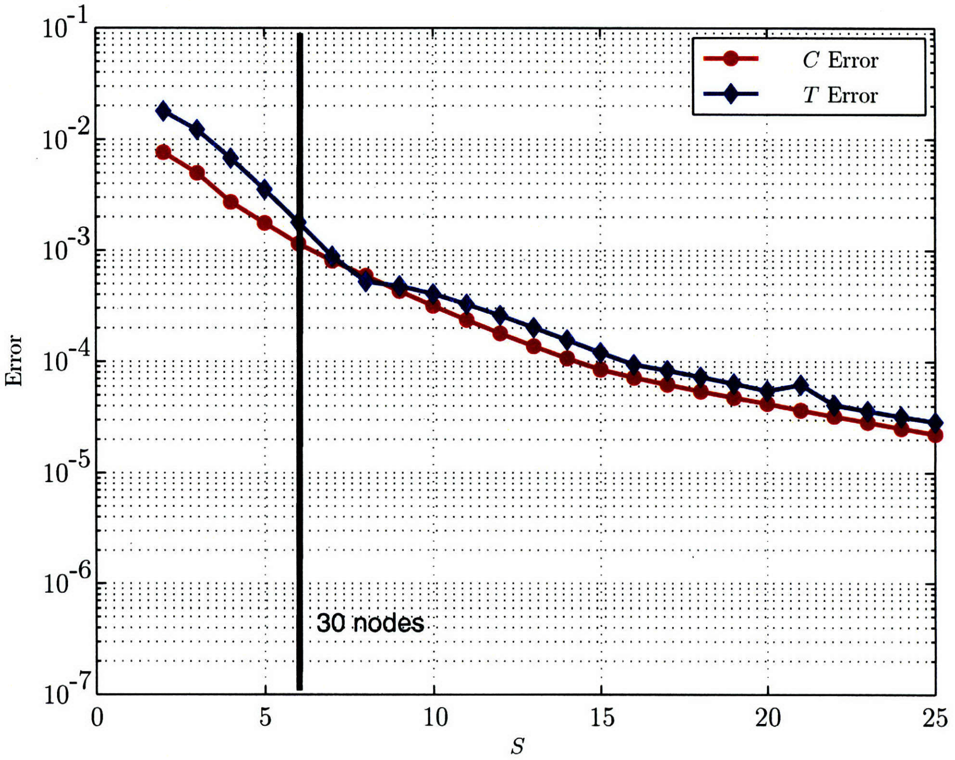


Figure 5-4: Convergence of Hicks-Ray problem via local approach as a function of the number of segments, S (5 nodes per segment).

as opposed to on a segment boundary. Figs. 5-5 and 5-6 show the solutions for these initial conditions which employ 40 nodes for the global GPM and 8 equally-spaced segments with 5 nodes per segment (40 nodes total) for the local GPM. Furthermore, Figs. 5-7 and 5-8 show the errors in the state (x_1 and x_2) and costate (λ_1 and λ_2) obtained from the global and local GPM methods, respectively. Naturally, for both approaches the largest error in the state occurs at the switch time. Note that the error in these figures is the absolute difference between the NLP solution and exact analytic solution. Figs. 5-5–5-8 show clearly that the results from both approaches are strikingly similar, and that the global approach is slightly more accurate. Critics of global methods claim that by shifting the nodes towards the boundaries, the sparsity of nodes in the middle of the trajectory degrades the accuracy of the approximation. However, despite a reduced density in the center of the interval, the maximum error is actually smaller with the global approach.

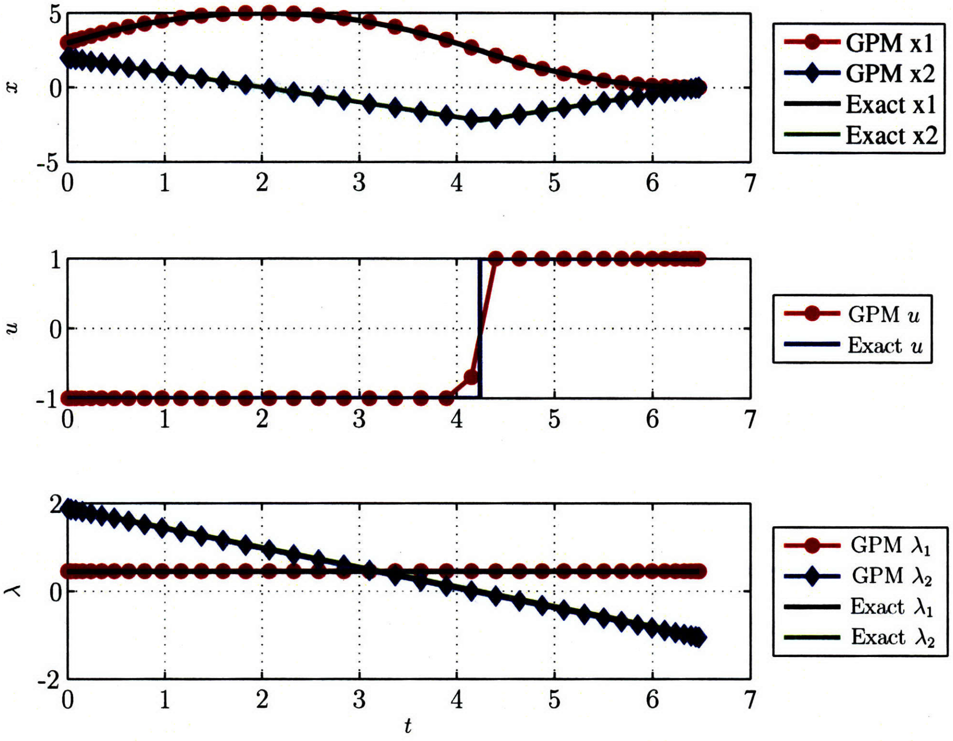


Figure 5-5: Solution to double integrator problem for initial conditions of Eq. (5.18) using global LG collocation with 40 nodes.

Example 2 Solved with Knowledge of Discontinuity

It is important to note that both the global and local applications of the GPM assumed no prior knowledge of the control switching time, t_b. As is often the case, even if a discontinuity in the solution is expected, the exact location of the discontinuity may be unknown. Therefore in general, the switch time will not coincide with a segment interface when using the local approach. However, one of the benefits of the local approach is that, if a switching time is known, a segment interface can be located to coincide with this switch time. Yet this concept can be applied to the global approach as well. The global approach then becomes a *semi-global* approach where the number of segments is one greater than the number of known discontinuities. This semi-global approach has been developed for different classes of optimal control problems using the Legendre pseudospectral method[38, 85, 92, 95] and the Gauss pseudospectral method.[66–68]

In order to assess how *a priori* knowledge of the discontinuity can improve accuracy in

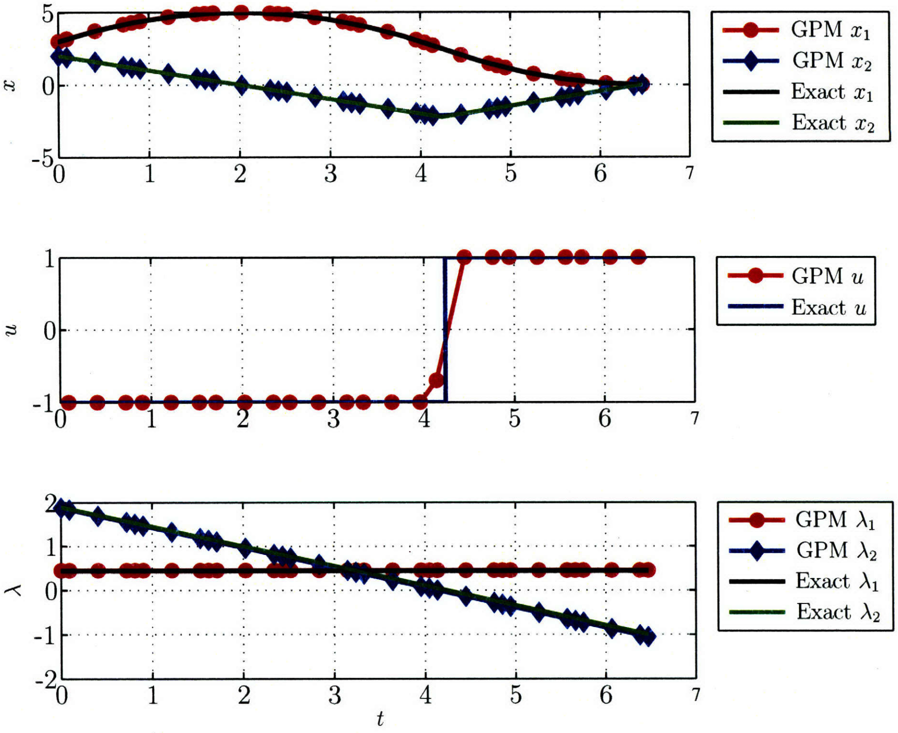


Figure 5-6: Solution to double integrator problem for initial conditions of Eq. (5.18) using local LG collocation with 8 segments of 5 nodes each.

both the global and local approaches, consider the following initial conditions:

$$x_{10} = -1 \qquad x_{20} = 2 \tag{5.19}$$

The initial conditions of Eq. (5.19) has the interesting feature that the switch time in the control is *exactly* $t_b = 3$ and the optimal final time is $t_f^* = 4$. Consequently, when a local approach is employed with 8 equally spaced segments [i.e., the same number of segments used with Eq. (5.18)], the switch time of the control lies at the boundary between the sixth and seventh segments.

Using the initial conditions of Eq. (5.19), the GPM is employed in the following two ways. First, the semi-global GPM is employed using two segments of 20 nodes each, where the segment interface occurs at $t = 3$. The local GPM is employed as before with 8 segments and 5 nodes per segment. The results of these two cases is shown in Figs. 5-9– 5-12. The accuracy of both methods is improved dramatically. Comparing Figs. 5-11 and 5-12 it is seen again that for all relevant quantities, the error using the local GPM is slightly larger

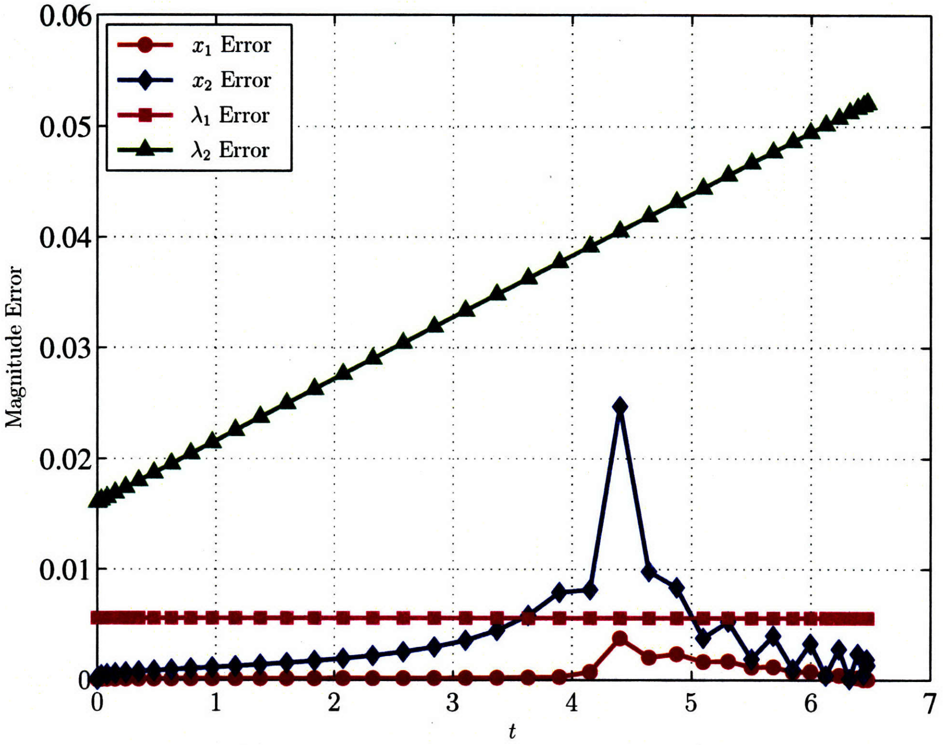


Figure 5-7: Error in state and costate for the double integrator problem using initial conditions of Eq. (5.18) and global LG collocation with 40 nodes.

than the corresponding error using global GPM. Although it isn't clear from Figs. 5-11 and 5-12 , the state is accurate to within the NLP tolerances due to the fact that the solution is a piecewise polynomial expression and can be approximated exactly with at least three collocaion points per phase.

5.3 Summary

A comparison was made between a global and a local implementation of the Gauss pseudospectral method. The purpose of the comparison was to determine the accuracy and computational efficiency of each approach. Employing the GPM as a local method, the number of collocation points within each segment was held fixed at a small number (e.g., three collocation points) while the number of segments was varied. In global collocation form, a single segment was used and the number of collocation points was varied. Analysis of the examples in this chapter suggests that for smooth problems, global collocation is sig-

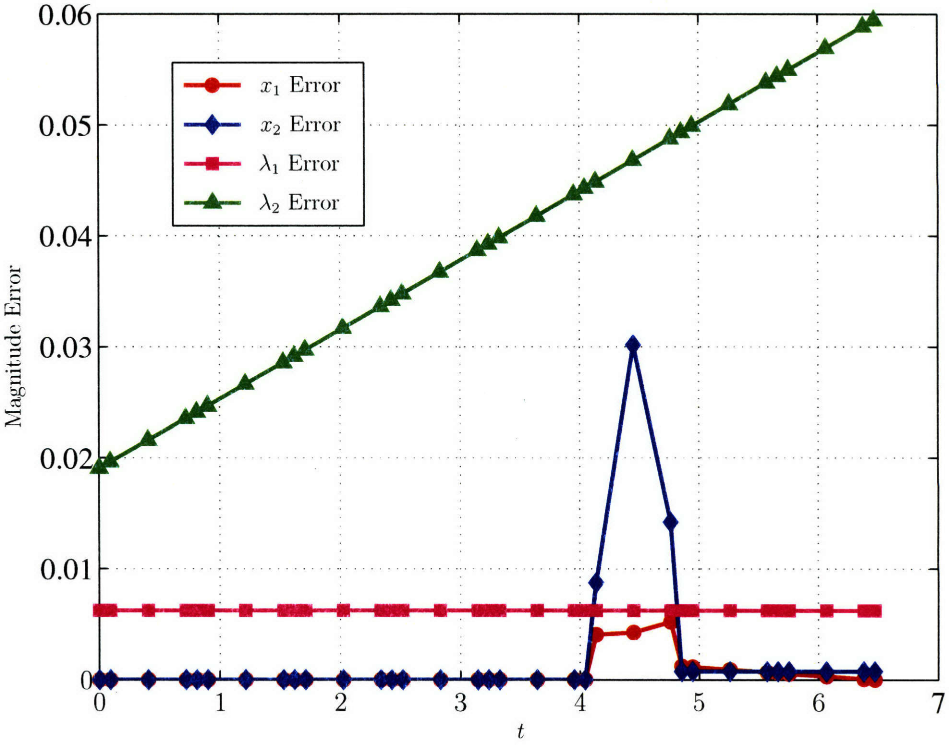


Figure 5-8: Error in state and costate for the double integrator problem using initial conditions of Eq. (5.18) and local LG collocation with 8 segments of 5 nodes each.

nificantly more accurate than local collocation for a given number of total collocation points. For nonsmooth problems, the accuracies of global and local collocation methods were found to be comparable, and if the discontinuities are known *a priori*, a semi-global approach is the most accurate approach. Furthermore it was found that, for a desired accuracy, the global approach was computationally more efficient for smooth problems . These claims were supported by two example problems which had features that might lead one to prefer the use of a local method.

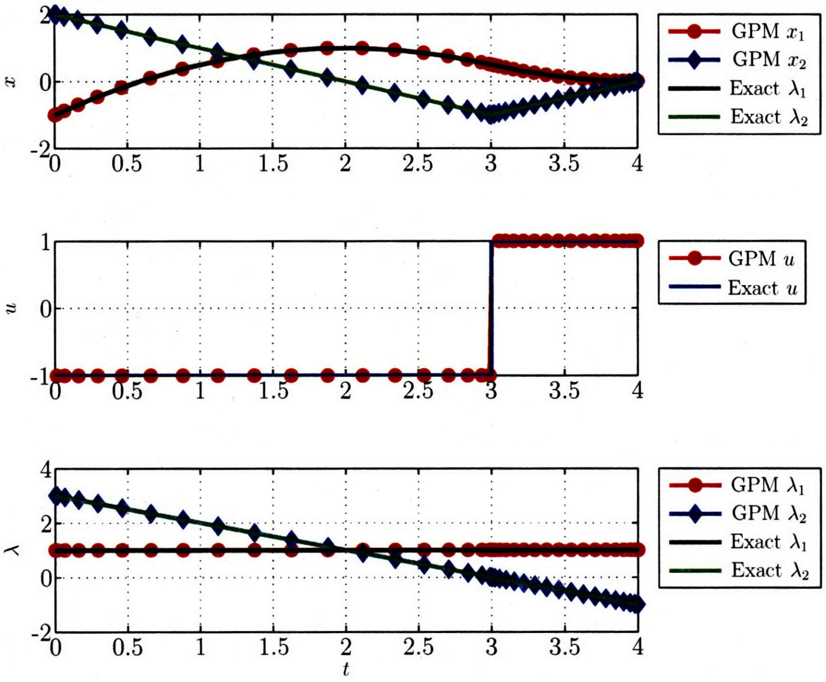


Figure 5-9: Solution to the double integrator problem with the initial conditions of Eq. (5.19) and semi-global LG collocation with two segments of 20 nodes each.

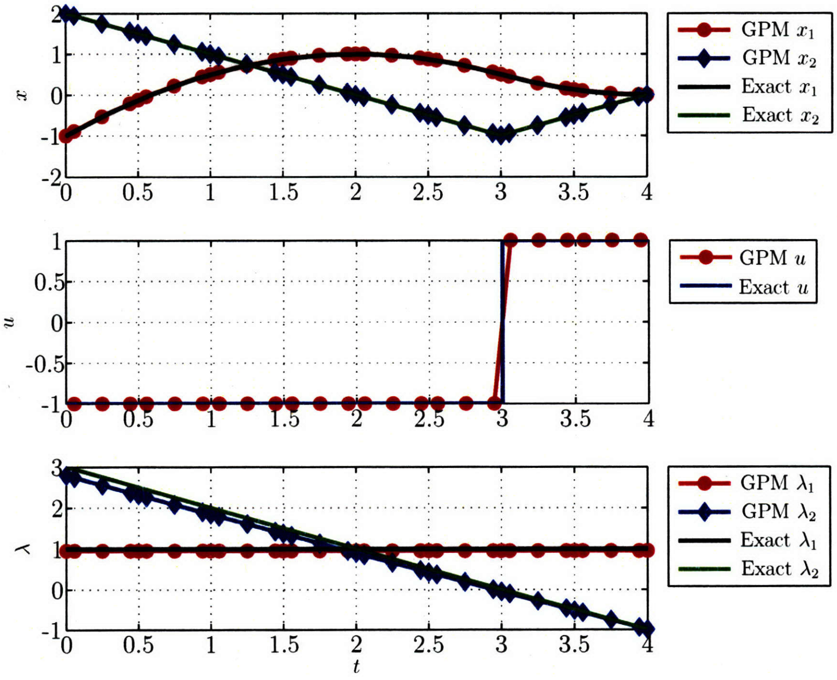


Figure 5-10: Solution to the double integrator problem with the initial conditions of Eq. (5.19) and local LG collocation.

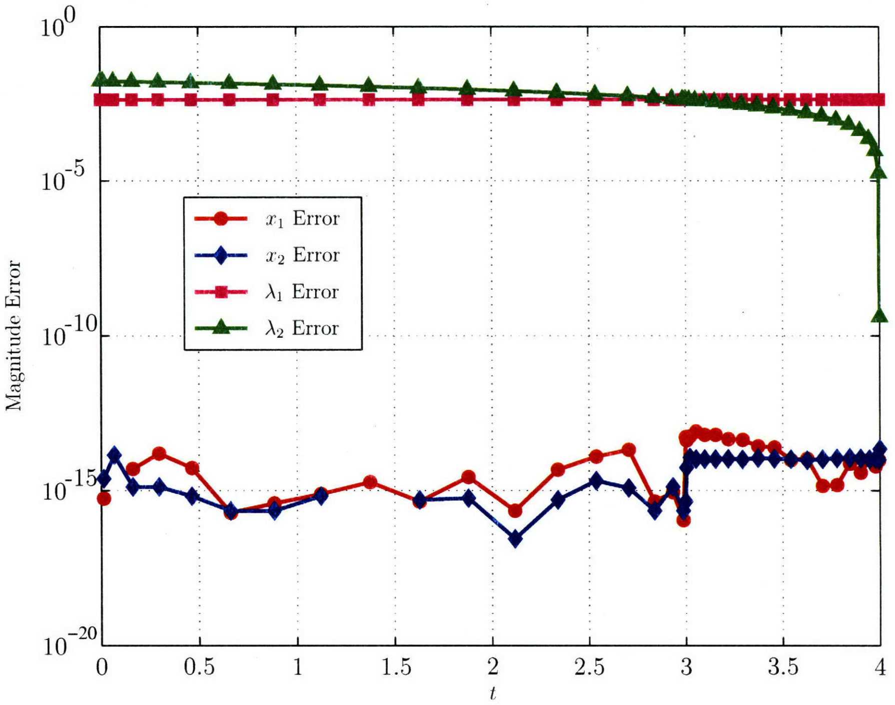


Figure 5-11: Error in state and costate for the double integrator problem using initial conditions of Eq. (5.19) and semi-global LG collocation with two segments of 20 LG points each.

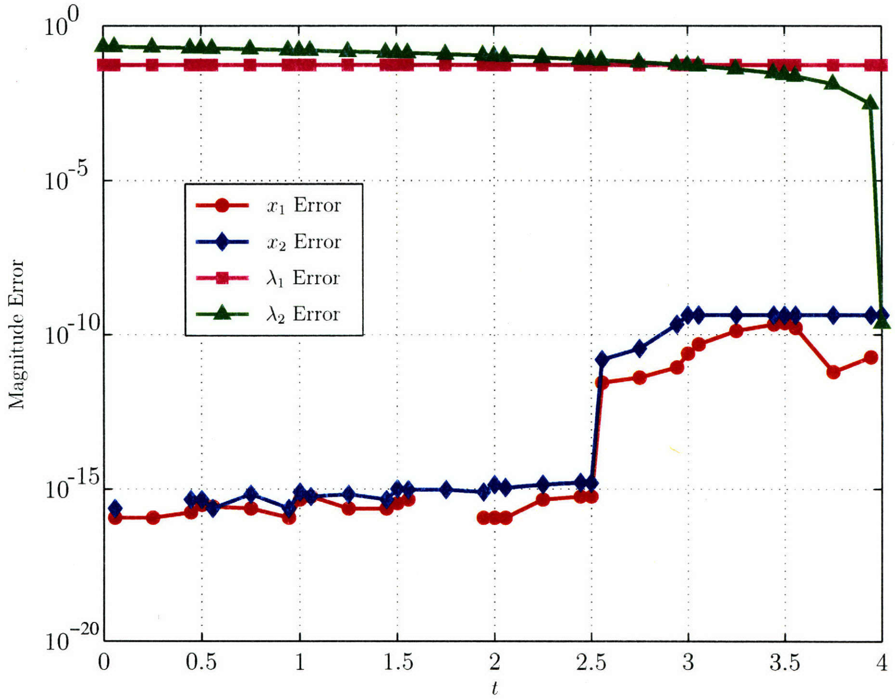


Figure 5-12: Error in state and costate for the double integrator problem using initial conditions of Eq. (5.19) and local LG collocation.

Chapter 6

Comparison between Three Pseudospectral Methods

This chapter analyzes the performance of three pseudospectral methods: the Gauss,[8] Radau,[72] and Legendre[33] pseudospectral methods. All three approaches are global in nature and implement orthogonal collocation based on Legendre polynomials. The key distinctions between these methods lie in the choice of both the discretization points and collocation points. This chapter introduces these differences, discusses the mathematical implications of each set of points, and numerically compares the three methods with respect to the accuracy and convergence rates of the resulting NLP state, control, and costate solution.

6.1 Continuous Mayer Problem

Without loss of generality, consider the following optimal control problem in Mayer form. Determine the control, $\mathbf{u}(\tau) \in \mathbb{R}^m$, that minimizes the cost functional

$$J = \Phi(\mathbf{x}(\tau_f)) \tag{6.1}$$

subject to the constraints

$$\frac{d\mathbf{x}}{d\tau} = \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau)) \in \mathbb{R}^n \tag{6.2}$$

$$\phi(\mathbf{x}(\tau_0),\mathbf{x}(\tau_f)) = \mathbf{0} \in \mathbb{R}^q \tag{6.3}$$

The optimal control problem of Eqs. (6.1)–(6.3) is referred to as the *continuous Mayer problem*. It is noted that, in order to simplify the analysis and provide a clear comparison, this chapter considers optimal control problems with fixed terminal times and *without* algebraic path constraints. As in Chapter 2, the time interval $\tau \in [-1,1]$ can be transformed to the arbitrary time interval $t \in [t_0, t_f]$ via the affine transformation of Eq. (2.6).

6.2 First-Order Necessary Conditions of Continuous Mayer Problem

Following a similar procedure to the continuous Bolza problem of Chapter 2, the indirect approach to solving the continuous Mayer problem of Eqs. (6.1)–(6.3) is to apply the calculus of variations to obtain first-order necessary conditions for optimality.[73] These variational conditions are derived using the Hamiltonian, $\mathcal{H}$, defined for the Mayer problem as

$$\mathcal{H}(\mathbf{x},\boldsymbol{\lambda},\mathbf{u}) = \boldsymbol{\lambda}^T\mathbf{f}(\mathbf{x},\mathbf{u}) \tag{6.4}$$

where $\boldsymbol{\lambda}(\tau) \in \mathbb{R}^n$ is the costate. Note that, for brevity, the explicit dependence of the state, control, and costate on τ has been dropped. Assuming that the optimal control lies within the feasible control set, the continuous-time first-order optimality conditions are

$$\begin{array}{rclcl}
\dfrac{d\mathbf{x}^T}{d\tau} & = & \dfrac{t_f-t_0}{2}\mathbf{f}^T(\mathbf{x},\mathbf{u}) & = & \dfrac{t_f-t_0}{2}\dfrac{\partial\mathcal{H}}{\partial\boldsymbol{\lambda}} \\
\dfrac{d\boldsymbol{\lambda}^T}{d\tau} & = & -\dfrac{t_f-t_0}{2}\boldsymbol{\lambda}^T\dfrac{\partial\mathbf{f}}{\partial\mathbf{x}} & = & -\dfrac{t_f-t_0}{2}\dfrac{\partial\mathcal{H}}{\partial\mathbf{x}} \\
\mathbf{0}^T & = & \boldsymbol{\lambda}^T\dfrac{\partial\mathbf{f}}{\partial\mathbf{u}} & = & \dfrac{\partial\mathcal{H}}{\partial\mathbf{u}} \\
 & & \phi(\mathbf{x}(\tau_0),\mathbf{x}(\tau_f)) & = & \mathbf{0} \\
\boldsymbol{\lambda}(\tau_0)^T & = & \boldsymbol{\nu}^T\dfrac{\partial\phi}{\partial\mathbf{x}(\tau_0)}\ ,\quad \boldsymbol{\lambda}(\tau_f)^T & = & \dfrac{\partial\Phi}{\partial\mathbf{x}(\tau_f)} - \boldsymbol{\nu}^T\dfrac{\partial\phi}{\partial\mathbf{x}(\tau_f)}
\end{array} \tag{6.5}$$

where $\boldsymbol{\nu} \in \mathbb{R}^q$ is Lagrange multiplier associated with the boundary condition $\boldsymbol{\phi}$. The optimality conditions of Eq. (6.5) define a *Hamiltonian boundary value problem* (HBVP).

6.3 Descriptions of the Pseudospectral Methods

The three pseudospectral methods in this comparison approximate the state using a finite basis of global polynomials, which are created by first discretizing the problem into N *nodes* along the trajectory (described in Chapter 2). However, the points within the node set, $\mathcal{N}$, are unique to each method, along with the choice of basis polynomials. These differences are explained in detail in this section. Furthermore, all pseudospectral methods transcribe the dynamic equations into algebraic conditions, which are evaluated at a set of *collocation points*, $\mathcal{K}$. This set of collocation points is also unique to each method and may or may not be the same as the node set. The locations of nodes and collocation points, as well as the choice of basis functions are key features that distinguish these methods from each other. There are many different types of pseudospectral methods, but this comparison is limited to three methods: the Legendre, Radau, and Gauss pseudospectral methods, since these are the three methods with published costate approximation procedures. The Gauss method has already been described in detail in Chapter 3, so this chapter provides only a brief overview. The Legendre and Radau direct transcription methods are described here in more detail and the key differences between the methods are highlighted. It is noted that the indexing in this section may be slightly different than the indexing contained in the references. This is done in order to create some commonality between the three methods in terms of their indices, and to create NLPs that are equivalent in size. Therefore, in all methods, N represents the total number of discretization points (nodes) used in the NLP, which is kept the same among all methods for the numerical comparisons in this chapter. Moreover, in general, the index j is used to represent the j-th node, while the index k is used to represent the k-th collocation point.

6.3.1 Legendre Pseudospectral Method (LPM)

The Legendre pseudospectral method (LPM), like all pseudospectral methods, is based on approximating the state using a finite set of interpolating polynomials. The state is approximated using a basis of N Lagrange interpolating polynomials, $\mathcal{L}_i(\tau)$ $(i = 1, \ldots, N)$, as defined in Eq. (2.23).

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=1}^{N} \mathcal{L}_i(\tau)\mathbf{X}(\tau_i) \tag{6.6}$$

In the LPM, the location of N nodes are described by the Legendre-Gauss-Lobatto (LGL) points on the interval $\tau \in [-1, 1]$ which were defined in Chapter 2. Note that the endpoints -1 and 1 are included in the state approximation.

Next, all pseudospectral methods transcribe the dynamic equations by collocating the derivative of this state approximation with the vector field, $\mathbf{f}(\mathbf{x}, \mathbf{u})$, at a set of points along the interval. Let us assume there are K collocation points. In the LPM, the LGL points are both the nodes and the collocation points, meaning the two sets are equivalent $(\mathcal{N} = \mathcal{K})$. Naturally, the number of points in both sets is the same as well $(N = K)$. Mathematically, the derivative of the state approximation at the k-th LGL point, τ_k, is

$$\dot{\mathbf{x}}(\tau_k) \approx \dot{\mathbf{X}}(\tau_k) = \sum_{i=1}^{N} \dot{\mathcal{L}}_i(\tau_k)\mathbf{X}(\tau_i) = \sum_{i=1}^{N} D_{ki}\mathbf{X}(\tau_k), \quad (k = 1, \ldots, K) \tag{6.7}$$

where the differentiation matrix, $D \in \mathbb{R}^{K \times N}$, is defined as

$$D_{ki} = \begin{cases} \frac{P_{N-1}(\tau_k)}{P_{N-1}(\tau_i)} \frac{1}{\tau_k - \tau_i}, & \text{if } k \neq i \\ -\frac{(N-1)N}{4}, & \text{if } k = i = 1 \\ \frac{(N-1)N}{4}, & \text{if } k = i = N \\ 0, & \text{otherwise} \end{cases} \tag{6.8}$$

The continuous dynamics of Eq. (6.2) are then transcribed into the following set of K algebraic equations via collocation described in Chapter 2 by Eq. (2.33):

$$\sum_{i=1}^{N} D_{ki}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k)) = \mathbf{0}, \quad (k = 1, \ldots, K) \tag{6.9}$$

The approximated control, $\mathbf{U}(\tau)$, is defined in a similar manner as the state:

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{i=1}^{N} \mathcal{L}_i(\tau)\mathbf{U}(\tau_i) \tag{6.10}$$

where τ_i, $i = 1, \ldots, N$ are the LGL nodes. Again, it is emphasized that in the LPM, the LGL points are both the discretization points and the collocation points. These two sets of points are not the same for the other methods presented in this thesis.

Next, the continuous cost function of Eq. (6.1) is approximated simply as

$$J = \Phi(\mathbf{X}(\tau_N)) \tag{6.11}$$

where $\tau_N \equiv 1$ for the LPM. Lastly, the boundary constraint of Eq. (6.3) is expressed using τ_N and $\tau_1 \equiv -1$ as

$$\boldsymbol{\phi}(\mathbf{X}(\tau_1), \mathbf{X}(\tau_N)) = \mathbf{0} \tag{6.12}$$

The cost function of Eq. (6.11) and the algebraic constraints of Eqs. (6.9) and (6.12) define an NLP whose solution is an approximate solution to the continuous Mayer problem.

6.3.2 Radau Pseudospectral Method (RPM)

The Radau pseudospectral method has been primarily used in the chemical engineering community.[27,71] The location of the nodes in the Radau method are based on the flipped Legendre-Gauss-Radau (LGR) points, which lie on the interval $\tau \in (-1, 1]$. Recall from Chapter 2 that this set of points includes the final point but not the initial point. When discretizing optimal control problems, it is desirable that the discretization span the entire interval, including both endpoints. Therefore, in order to complete the full discretization of the time interval, the N discretization points are found using $N-1$ flipped Radau points plus the initial point, $\tau_0 \equiv -1$. The state approximation is constructed exactly the same as the Legendre pseudospectral method, using a basis of N Lagrange polynomials:

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=0}^{N-1} \mathcal{L}_i(\tau)\mathbf{X}(\tau_i) \tag{6.13}$$

where Eq. (6.13) uses a slight index modification from $(i = 1, \ldots, N)$ in the LPM to $(i = 0, \ldots, N-1)$. This new index highlights the fact that the nodes τ_i, $i = 0, \ldots, N-1$ are the initial point plus the N-1 LGR points.

Unlike the Legendre pseudospectral method, the Radau method uses a different number of collocation points, K, than discretization points, N. Specifically, the collocation points are the $N-1$ LGR points, while the discretization points are the LGR points plus the initial point, τ_0. Therefore, $K = N-1$ and $\mathcal{K} \subset \mathcal{N}$. The K collocation equations are then described as

$$\sum_{i=0}^{N-1} D_{ki}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k)) = \mathbf{0}, \quad (k = 1, \ldots, K) \tag{6.14}$$

where τ_k are the LGR points and

$$\dot{\mathcal{L}}_i(\tau_k) = D_{ki} = \begin{cases} \dfrac{\dot{g}(\tau_k)}{(\tau_k - \tau_i)\dot{g}(\tau_i)}, & \text{if } k \neq i \\ \dfrac{\ddot{g}(\tau_i)}{2\dot{g}(\tau_i)}, & \text{if } k = i \end{cases} \tag{6.15}$$

where the function $g(\tau_i) = (1+\tau_i)[P_K(\tau_i) - P_{K-1}(\tau_i)]$ and τ_i, $(i = 0, \ldots, K)$ are the K LGR points plus the initial point. Since the collocation equations involve the control solely at the Radau points, the control is approximated using $N-1$ Lagrange polynomials, $\bar{\mathcal{L}}_k(\tau)$ $(k = 1, \ldots, N-1)$, as

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{k=1}^{N-1} \bar{\mathcal{L}}_k(\tau)\mathbf{U}(\tau_k) \tag{6.16}$$

where τ_k again are the LGR points. This control approximation uses one less point than the state approximation meaning there is no control defined at the initial point. See Table 6.1 for more details. In practice, the initial control is simply extrapolated from Eq. (6.16). The discretization of the rest of the optimal control problem is similar to the LPM. The continuous cost function of Eq. (6.1) is approximated as

$$J = \Phi(\mathbf{X}(\tau_K)) \tag{6.17}$$

Recall that in the Radau method, the final LGR point $\tau_K = 1$. Lastly, the boundary constraint of Eq. (6.3) is expressed as a function of the initial point and the final point as

$$\phi(\mathbf{X}(\tau_0), \mathbf{X}(\tau_K)) = \mathbf{0} \tag{6.18}$$

The cost function of Eq. (6.17) and the algebraic constraints of Eqs. (6.14) and (6.18) define an NLP whose solution is an approximate solution to the continuous Mayer problem.

6.3.3 Gauss Pseudospectral Method (GPM)

This section largely echoes Chapter 3, however slight differences in index notation are induced to highlight nuances between this method and the Legendre and Radau pseudospectral methods. Recall from Chapter 3 that the location of the nodes in the Gauss pseudospectral method are based on Legendre-Gauss (LG) points, which lie on the interval $\tau \in (-1, 1)$. This set of points includes neither the initial point nor the final point. Therefore, in order to fully discretize the time interval, the N discretization points are $(N-2)$ interior LG points, the initial point, $\tau_0 \equiv -1$, and the final point, $\tau_F \equiv 1$. The state is approximated using a basis of N-1 Lagrange interpolating polynomials, which is slightly smaller than the previous two methods,

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=0}^{N-2} \bar{\mathcal{L}}_i(\tau)\mathbf{X}(\tau_i) \tag{6.19}$$

where τ_i $(i = 0, \ldots, N-2)$ are the initial point plus the $(N-2)$ LG points. Note that the final point, although it is part of the NLP discretization, is not part of the state approximation. This results in a state approximation that is one order less than the previous methods, but is necessary in order to have the equivalence property between the KKT conditions and HBVP conditions. As in the Radau pseudospectral method, the Gauss pseudospectral method uses a different number of collocation points, K, than discretization points, N. Specifically, the collocation points are the LG points, while the discretization points are the LG points plus the initial point and final point. Therefore, $K = N - 2$ and $\mathcal{K} \subset \mathcal{N}$. The K collocation equations are then described as

$$\sum_{i=0}^{N-2} D_{ki}\mathbf{X}(\tau_i) - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k)) = \mathbf{0}, \quad (k = 1, \ldots, K) \tag{6.20}$$

where τ_k, $(k = 1, \ldots, K)$ are the LG points. This equation is exactly the same as Eq. (3.7) in Chapter 3 except for a slight change in indices to show the relationship of this collocation approach to the others presented in this chapter. Note that there is no collocation at either boundary. The control is approximated at the $(N-2)$ collocation points using a basis of $(N-2)$ Lagrange interpolating polynomials $\tilde{\mathcal{L}}_i(\tau)$, $(i = 1, \ldots, N-2)$ as

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{i=1}^{N-2} \tilde{\mathcal{L}}_i(\tau)\mathbf{U}(\tau_i) \tag{6.21}$$

where τ_i, $(k = 1, \ldots, N-2)$ are the LG points. Table 6.1 lists all the Lagrange polynomial expressions described in this work.

As explained in Chapter 3, an additional constraint must be added to the discretization in order to ensure that the the final state obeys the state dynamics. This is enforced by including a Gauss quadrature to approximate the integral of the dynamics across the entire interval (Eq. (3.4)):

$$\mathbf{X}(\tau_F) - \mathbf{X}(\tau_0) - \frac{t_f - t_0}{2}\sum_{k=1}^{K} w_k \mathbf{f}(\mathbf{X}(\tau_k), \mathbf{U}(\tau_k)) = \mathbf{0} \tag{6.22}$$

where w_k are the Gauss weights and τ_k are the LG points. Lastly, the cost function is approximated by

$$J = \Phi(\mathbf{X}(\tau_F)) \tag{6.23}$$

and the boundary constraint is expressed as

$$\boldsymbol{\phi}(\mathbf{X}(\tau_0), \mathbf{X}(\tau_F)) = \mathbf{0} \tag{6.24}$$

The cost function of Eq. (6.23) and the algebraic constraints of Eqs. (6.20), (6.22), and (6.24) define an NLP whose solution is an approximate solution to the continuous Mayer problem.

Table 6.1: Definitions of Lagrange interpolation polynomials used in this work

# of Basis Polynomials	Application	Symbol	Index
N	LPM ($\mathbf{x}$,$\mathbf{u}$,$\boldsymbol{\lambda}$)	$\mathcal{L}_i(\tau)$	$i = 1, \ldots, N$
N	RPM ($\mathbf{x}$)	$\mathcal{L}_i(\tau)$	$i = 0, \ldots, N-1$
N-1	RPM ($\mathbf{u}, \boldsymbol{\lambda}$)	$\bar{\mathcal{L}}_i(\tau)$	$i = 1, \ldots, N-1$
N-1	GPM ($\mathbf{x}$,$\boldsymbol{\lambda}$)	$\bar{\mathcal{L}}_i(\tau)$	$i = 0, \ldots, N-2$
N-2	GPM ($\mathbf{u}$)	$\tilde{\mathcal{L}}_i(\tau)$	$i = 1, \ldots, N-2$

6.4 Costate Estimation

All three aforementioned pseudospectral methods have established costate estimation procedures. In particular, a mapping between the KKT multipliers of the NLP and the costates of the continuous-time optimal control problem has been derived for all three methods. This section describes the costate estimation procedure for each method.

Legendre Pseudospectral Method

As explained in Chapter 3, the KKT conditions of the NLP are derived by defining an augmented cost function, which brings the NLP constraints into the cost function via Lagrange multipliers. For LPM, the augmented cost is

$$J_a = \Phi(\mathbf{X}_N) - \tilde{\boldsymbol{\nu}}^T \boldsymbol{\phi}(\mathbf{X}_1, \mathbf{X}_N) - \sum_{k=1}^{K} \tilde{\boldsymbol{\Lambda}}_k^T \left(\sum_{i=1}^{N} D_{ki} \mathbf{X}_i - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k) \right) \tag{6.25}$$

The KKT multipliers in Eq. (6.25) are $\tilde{\boldsymbol{\Lambda}}_k$, $(k = 1, \ldots, K)$, which are associated with the collocation equations of Eq. (6.9) and $\tilde{\boldsymbol{\nu}}$, which relates to the boundary condition of Eq. (6.12). Without going into detail, the mapping from the KKT multipliers to the HBVP variables is:[39]

$$\begin{aligned} \boldsymbol{\Lambda}_k &= \frac{\tilde{\boldsymbol{\Lambda}}_k}{w_k}, \quad (k = 1, \ldots, K) \\ \boldsymbol{\nu} &= \tilde{\boldsymbol{\nu}} \end{aligned} \tag{6.26}$$

The HBVP variables $\mathbf{\Lambda}_k$, $(k = 1, \ldots, K)$, and $\boldsymbol{\nu}$, now defined in terms of NLP multipliers, can be substituted into the HBVP equations to determine if the optimality conditions have been met. It has been documented in the literature[49,128] that this mapping for the LPM does not provide an exact equivalence between the KKT conditions and the HBVP conditions. Specifically, the HBVP costate differential equations are not naturally collocated at the boundaries, meaning the boundary KKT conditions result in a linear combination of two HBVP conditions: the boundary costate dynamics and the transversality conditions. Although all the KKT conditions are not derived here, the conflicting KKT condition is shown, which is found by taking the partial derivative of the augmented cost function with respect to the initial state:

$$\sum_{k=1}^{K} \mathbf{\Lambda}_k^T D_{1k} + \frac{t_f - t_0}{2} \mathbf{\Lambda}_1^T \frac{\partial \mathbf{f}_1}{\partial \mathbf{X}_1} = \frac{1}{w_1} \left(-\mathbf{\Lambda}_1^T + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}_1} \right) \tag{6.27}$$

Notice that the left-hand side of the equation resembles the discretized costate dynamics and the right hand side resembles the transversality condition from the HBVP in Section 6.2. If these conditions were to map exactly to the HBVP equations, each side of Eq. (6.27) should equal zero. However, Eq. (6.27) is inseparable in the LPM. Similarly, the KKT condition corresponding to the final state is

$$\sum_{k=1}^{K} \mathbf{\Lambda}_k^T D_{Nk} + \frac{t_f - t_0}{2} \mathbf{\Lambda}_N^T \frac{\partial \mathbf{f}_N}{\partial \mathbf{X}_N} = \frac{1}{w_N} \left(\mathbf{\Lambda}_N^T - \frac{\partial \Phi}{\partial \mathbf{X}_N} + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}_N} \right) \tag{6.28}$$

where again, the left-hand side is the discretized costate dynamics and the right-hand side is the transversality condition corresponding to the terminal state. It is impossible to get an exact mapping from the KKT conditions to the HBVP conditions without being able to separate these mixed conditions. Recent research[49] suggests that by applying relaxation techniques, linear independence constraint qualification, and an additional set of conditions called "closure conditions", one can show that there exists a costate approximation that converges to the true costate in the limit. From Ref. 49, the most recent closure conditions are

$$\begin{aligned}\left\|-\boldsymbol{\Lambda}_1^* - \frac{\partial \Phi}{\partial \mathbf{x}_1}^T(\mathbf{X}_1^*, \mathbf{X}_N^*) - \left(\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}_1}(\mathbf{X}_1^*, \mathbf{X}_N^*)\right)^T \boldsymbol{\nu}^*\right\|_\infty &\leq \delta_D \\ \left\|\boldsymbol{\Lambda}_N^* - \frac{\partial \boldsymbol{\Phi}}{\partial \mathbf{x}_N}^T(\mathbf{X}_1^*, \mathbf{X}_N^*) - \left(\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}_N}(\mathbf{X}_1^*, \mathbf{X}_N^*)\right)^T \boldsymbol{\nu}^*\right\|_\infty &\leq \delta_D\end{aligned} \tag{6.29}$$

where $\mathbf{X}^*$, $\boldsymbol{\Lambda}^*$, and $\boldsymbol{\nu}^*$ represent a solution that satisfies Eq. (6.29) and the KKT conditions. δ_D is a dual feasibility tolerance. These conditions are essentially the discretized HBVP transversality conditions.[73] The *covector mapping theorem*[49] states that there exists an N large enough so that the closure conditions and the KKT conditions are satisfied to a specified tolerance, δ. Upon close examination of Ref. 49 (and the references therein) any algorithm attempting to solve the combined KKT conditions and closure conditions must solve a mixed primal-dual feasibility problem. By including the closure conditions of Eq. (6.29) to form the "augmented optimality conditions", both the state and costate are now variables to be solved. This procedure significantly increases the computational complexity required to estimate the costate. To be thorough, both LPM costate approximations described here (Refs. 39 and 49) are considered in this chapter. The LPM costate is computed first according to Eq. (6.26), and then by formulating the primal-dual feasibility problem which attempts to satisfy the augmented KKT conditions according to some tolerance, δ. It is noted that improving the LPM costate approximation is an ongoing research topic, and better methods may be found in the future.

Similar to the LPM state approximation, the continuous-time costate approximation is represented by a basis of N Lagrange interpolating polynomials as

$$\boldsymbol{\lambda}(\tau) \approx \boldsymbol{\Lambda}(\tau) = \sum_{i=1}^{N} \boldsymbol{\Lambda}(\tau_i)\mathcal{L}_i(\tau) \tag{6.30}$$

Radau Pseudospectral Method

The augmented cost function corresponding to the Radau pseudospectral method is given as

$$J_a = \Phi(\mathbf{X}_K) - \tilde{\boldsymbol{\nu}}^T \boldsymbol{\phi}(\mathbf{X}_0, \mathbf{X}_K) - \sum_{k=1}^{K} \tilde{\boldsymbol{\Lambda}}_k^T \left(\sum_{i=0}^{K} D_{ki}\mathbf{X}_i - \frac{t_f - t_0}{2}\mathbf{f}(\mathbf{X}_k, \mathbf{U}_k)\right) \tag{6.31}$$

The KKT multipliers resulting from the NLP using the Radau pseudospectral method are $\tilde{\mathbf{\Lambda}}_k$, $(k = 1, \ldots, K)$, associated with the collocation equations of Eq. (6.14) and $\tilde{\boldsymbol{\nu}}$, associated with the boundary conditions of Eq. (6.18). The mapping from the KKT multipliers to the HBVP variables is described as follows:

$$\begin{aligned} \mathbf{\Lambda}_k &= \frac{\tilde{\mathbf{\Lambda}}_k}{w_k}, \quad (k = 1, \ldots, K) \\ \boldsymbol{\nu} &= \tilde{\boldsymbol{\nu}} \end{aligned} \tag{6.32}$$

The K discrete costates $\mathbf{\Lambda}_k$, $(k = 1, \ldots, K)$, and the HBVP multiplier $\boldsymbol{\nu}$ are now defined in terms of the KKT multipliers. As in the LPM, it is understood that there is an imperfect match between the KKT multipliers and the costate.[72] The Radau method has a similar conflict of HBVP equations when the partial derivative of the augmented cost function is taken with respect to the final state:

$$\sum_{k=1}^{K} \mathbf{\Lambda}_k^T D_{Kk} + \frac{t_f - t_0}{2} \mathbf{\Lambda}_K^T \frac{\partial \mathbf{f}_K}{\partial \mathbf{X}_K} = \frac{1}{w_K} \left(\mathbf{\Lambda}_K^T - \frac{\partial \Phi}{\partial \mathbf{X}_K} + \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \mathbf{X}_K} \right) \tag{6.33}$$

However, recent work[71, 72] has demonstrated rapid convergence rates of the costate approximation using Eq. (6.32) as the number of nodes increases. The discrete values of the costate are used to construct a continuous-time approximation to the costate, which is represented by a basis of $N - 1$ Lagrange interpolating polynomials, as in the control:[72]

$$\boldsymbol{\lambda}(\tau) \approx \mathbf{\Lambda}(\tau) = \sum_{i=1}^{K} \mathbf{\Lambda}(\tau_i) \bar{\mathcal{L}}_i(\tau) \tag{6.34}$$

where τ_k, $(k = 1, \ldots, K)$ are the LGR points. Note that since there is no collocation at the initial point, there is no initial costate estimate that is directly output from the NLP. However, the initial costate can be approximated by extrapolating Eq. (6.34).

Gauss Pseudospectral Method

The augmented cost function corresponding to the Gauss discretization of the Mayer problem is

$$
\begin{aligned}
J_a = \ & \Phi(\mathbf{X}_F) - \tilde{\boldsymbol{\nu}}^T \boldsymbol{\phi}(\mathbf{X}_0, \mathbf{X}_F) - \sum_{k=1}^{K} \tilde{\boldsymbol{\Lambda}}_k^T \left(\sum_{i=0}^{K} D_{ki} \mathbf{X}_i - \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k) \right) \\
& - \tilde{\boldsymbol{\Lambda}}_F^T \left(\mathbf{X}_F - \mathbf{X}_0 - \frac{t_f - t_0}{2} \sum_{k=1}^{K} w_k \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k) \right)
\end{aligned}
\tag{6.35}
$$

The KKT multipliers resulting from the NLP using the Gauss pseudospectral method are $\tilde{\boldsymbol{\Lambda}}_k$ $(k = 1, \ldots, K)$, associated with the collocation equations of Eq. (6.20), $\tilde{\boldsymbol{\Lambda}}_F$, associated with the quadrature constraint of Eq. (3.4), and $\tilde{\boldsymbol{\nu}}$, associated with the boundary conditions of Eq. (6.24). The mapping from the KKT multipliers to the HBVP variables is described in Chapter 3 and repeated here for clarity:

$$
\begin{aligned}
\boldsymbol{\Lambda}_k & = \frac{\tilde{\boldsymbol{\Lambda}}_k}{w_k} + \tilde{\boldsymbol{\Lambda}}_F, \quad (k = 1, \ldots, K) \\
\boldsymbol{\nu} & = \tilde{\boldsymbol{\nu}}\,, \\
\boldsymbol{\Lambda}_f & = \tilde{\boldsymbol{\Lambda}}_F
\end{aligned}
\tag{6.36}
$$

The variables $\boldsymbol{\Lambda}_k$, $\boldsymbol{\Lambda}_f$, and $\boldsymbol{\nu}$, now defined in terms of NLP multipliers, can be substituted into the HBVP equations to ensure the optimality conditions have been met. The discrete costate is then used to construct a continuous-time costate approximation, which is represented by a slightly different basis of $N-1$ Lagrange interpolating polynomials, defined as

$$
\boldsymbol{\lambda}(\tau) \approx \boldsymbol{\Lambda}(\tau) = \sum_{i=1}^{K+1} \boldsymbol{\Lambda}(\tau_i) \bar{\mathcal{L}}_i(\tau) \tag{6.37}
$$

where τ_i, $(i = 1, \ldots, K+1)$ are the LG points plus the final point, τ_F. Recall that the state is approximated using $(N-1)$ Lagrange polynomials based on the LG points and the *initial* time, while the costate, $\boldsymbol{\lambda}(\tau)$, is approximated using $(N-1)$ Lagrange polynomials consisting of the LG points and the *final* time. This discrepancy is required to preserve the mapping, as expained in Chapter 3. As in the Radau method, an approximation for the initial costate, $\boldsymbol{\Lambda}_0$, cannot be pulled directly from a corresponding multiplier in the NLP because no such multiplier exists. However, an accurate approximation to the initial costate can be determined from the following equation:

$$\boldsymbol{\Lambda}_0 \equiv \tilde{\boldsymbol{\Lambda}}_F - \sum_{k=1}^{K} D_{k0} \tilde{\boldsymbol{\Lambda}}_k \tag{6.38}$$

which is shown in Chapter 3,to satisfy the HBVP transversality condition

$$\boldsymbol{\Lambda}_0^T = \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{X}_0} \tag{6.39}$$

resulting in a perfect match between the HBVP equations and the KKT conditions. Furthermore, one could simply extrapolate Eq. (6.37) to τ_0, which would produce an equally accurate initial costate. All three methods for determining the initial costate are equivalent. The remainder of this chapter focuses on comparing these pseudospectral techniques in terms of their computational efficiency, approximation accuracy, and solution convergence rates with respect to the state, control, and especially the costate.

6.5 Comparison of Pseudospectral Methods

The three pseudospectral methods described in this chapter: GPM, RPM, and LPM, are compared on several examples of increasing complexity.

6.5.1 Single State Example

As a first example, consider the following one-dimensional optimal control problem. Minimize the cost functional

$$J = -y(t_f) \tag{6.40}$$

subject to the constraints

$$\dot{y}(t) = y(t)u(t) - y(t) - u^2(t) \tag{6.41}$$

and the initial condition

$$y(t_0) = 1 \tag{6.42}$$

where $y(t)$, is the state, $u(t)$ is the control, and $t_f = 5$ is the final time. The exact solution is

$$\begin{aligned} y^*(t) &= \frac{4}{1+3e^t} && (6.43)\\ \lambda^*(t) &= ae^{(2\ln(1+3e^t)-t)} && (6.44)\\ u^*(t) &= .5y^*(t) && (6.45)\\ \text{where } a &= \frac{-1}{(e^{-5}+6+9e^5)} && (6.46)\\ & && (6.47) \end{aligned}$$

$\lambda^*(t)$ is the costate associated with the optimal solution. The exact solution is depicted in Fig. 6-1.

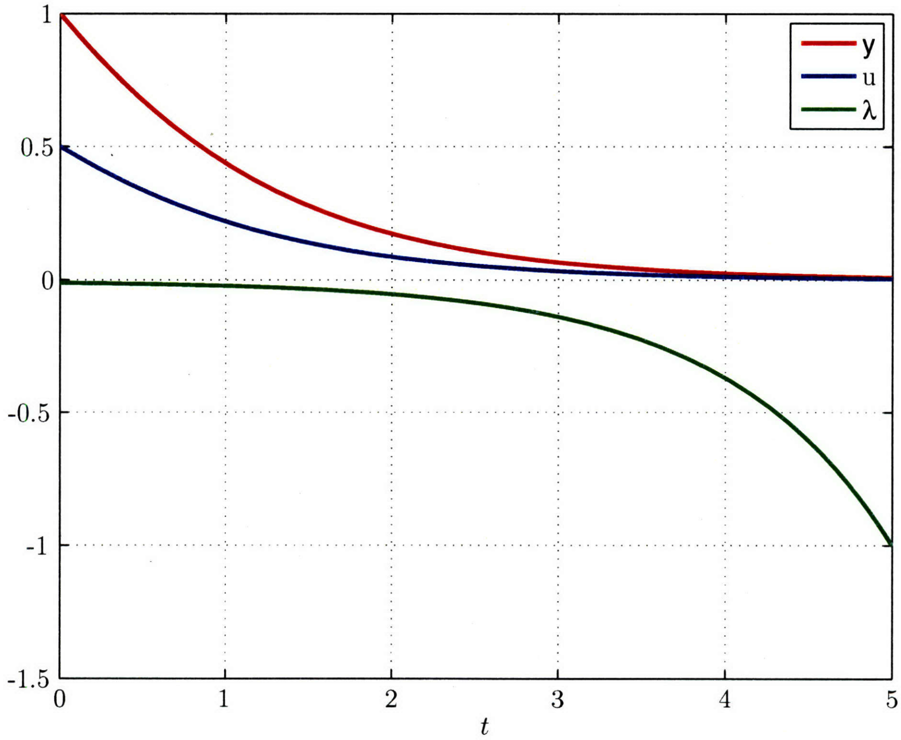


Figure 6-1: Exact solution for the single state example

This problem is solved by all three pseudospectral techniques mentioned previously. Each method transcribes the optimal control problem into an NLP that consists of 10 nodes ($N = 10$). The resulting NLP is then solved with SNOPT[45] using the default feasibility and optimality tolerances (10^{-6}). In Fig. 6-2, the absolute value of the error between the NLP state and the exact solution ($|Y(\tau_i) - y^*(\tau_i)|$, $i = 1, \ldots, N$) is plotted for each node and all three approaches. Note that the error in initial state is not included since the SNOPT satisfies state equality bounds (i.e., the initial conditions) to double precision (10^{-16}). When comparing the state accuracy of the three methods in this example, there is no clear method that outperforms the other two across the entire interval. In fact, the only noticeable difference in Fig. 6-2 is that both the GPM and RPM formulations produce an extremely accurate terminal state. It is hypothesized that the highly accurate terminal state for the GPM is due to the additional quadrature constraint in the NLP.

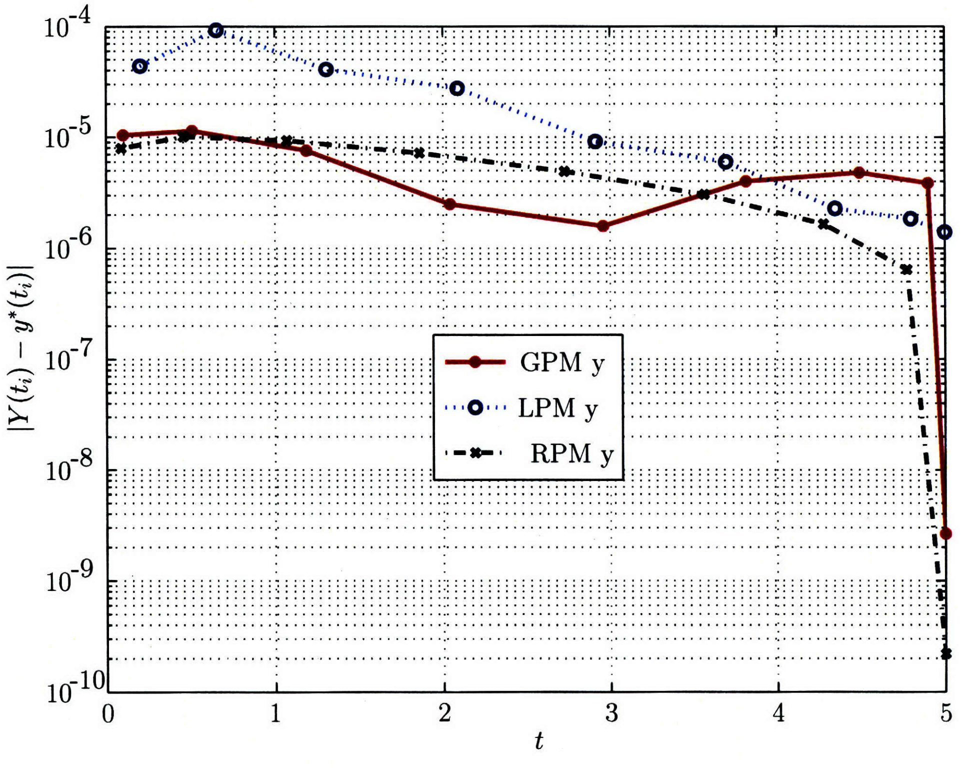


Figure 6-2: State error for the single state example.

The control error is plotted in Fig. 6-3. As described in Section 6.3, each method approximates the control using a different number of points. The LPM discretizes the control at all N discretization points, while Radau discretizes the control at the (N-1) Radau points, and the Gauss method discretizes the control at the (N-2) Gauss points. In the Radau discretization, the initial control is missing. Similarly, in the Gauss discretization, the control at both the initial and final time are missing. These missing control values can be computed by a variety of techniques. The approach often outlined in the literature[33,71] is to extrapolate the control approximation equations, Eqs. (6.10), (6.16) and (6.21). However, in practice it is more common to use a simple spline extrapolation.[14] Thirdly, a technique involving the HBVP equations[67] at the boundaries has been developed for the Gauss pseudospectral method in Chapter 4, but this technique is not used in this section. Since the extrapolated controls are not explicit variables in the NLP, one would assume that they would be less accurate than methods that include the boundary controls in the formulation of the NLP, such as the LPM. In Fig. 6-3, the boundary control estimate for the Gauss method is indeed the least accurate. Likewise, the accuracy of the initial control estimate for the Radau method is similar to the GPM value. However, it is interesting that while the largest error for Gauss and Radau are at the boundaries, the LPM has a larger error at one of the interior points. In this example, the benefit of an accurate initial control for the LPM is offset by the inaccurate interior control.

A more appropriate way to determine the effect of the control error is to actually propagate the state dynamics according to the control estimate from the NLP, as done in Chapter 4. In this comparison, the state dynamics are propagated using the Matlab function ODE45, where the control approximations are represented using Eqs. (6.10), (6.16), and (6.21). The resulting propagated state error is shown in Fig. 6-4. Clearly, the LPM propagation error is the largest despite the relatively accurate boundary controls.

The largest discrepancy between the methods is seen in the costate comparison of Fig. 6-5. The LPM costate approximation is several orders of magnitude worse than the other two methods across all values of N. This example highlights a common problem[49] with the original LPM costate approximation[39] where the approximation tends to "wiggle" about the true solution, shown clearly in Fig. 6-6. Furthermore, note that the largest error in the LPM costate occurs at the boundaries due to the conflicting constraints (explained in Section 6.4). The GPM, which has no constraint conflict, provides extremely accurate boundary costates, while the RPM produces an accurate initial costate (no conflict) but a less accurate final costate (which has a conflict explained in Section 6.4).

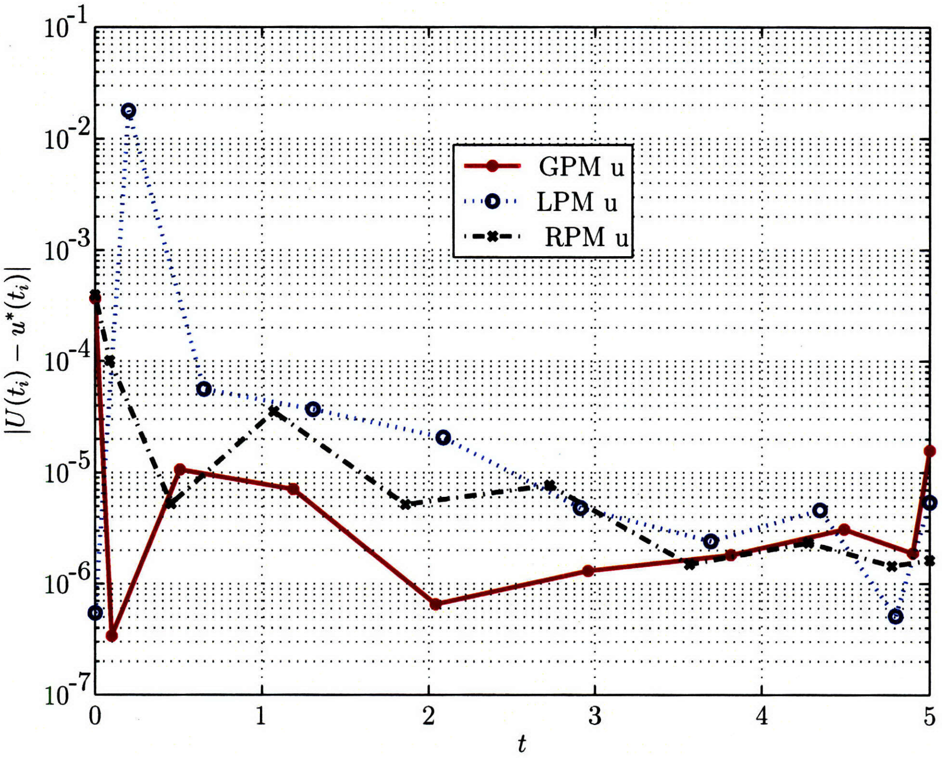


Figure 6-3: Control error for the single state example.

In pseudospectral methods, like most direct methods, the solution accuracy can be improved by increasing the number of nodes used to discretize the problem.[112] Furthermore, the rate of convergence to the true optimal solution is extremely important as it can help determine how many nodes are needed in order to adequately approximate the solution. The convergence rates of the state and costate are shown in Figs. 6-7 and 6-8, respectively. In these figures, each method was solved repeatedly while the number of nodes in the problem was increased from 5 to 50. The error shown in these figures is the maximum absolute error over all the nodes in the problem ($||X(\tau_i) - x^*(\tau_i)||_\infty$, $\forall \tau_i \in [-1, 1]$). As seen in Figure 6-7, the state convergence rate for all three methods is quite similar. The steep convergence rate depicts the "spectral" convergence that is characteristic of pseudospectral methods.[43] Naturally, once the error drops below the tolerances of the NLP, the error stops improving. The convergence rate for the costate is shown in Fig. 6-8. The Gauss and Radau methods show rapid convergence rates for the costate, which even outperform the state. Fig. 6-8 also shows the apparent lack of convergence for the costate using the LPM. It is clear that increasing the number of nodes does not improve the costate error by any significant amount

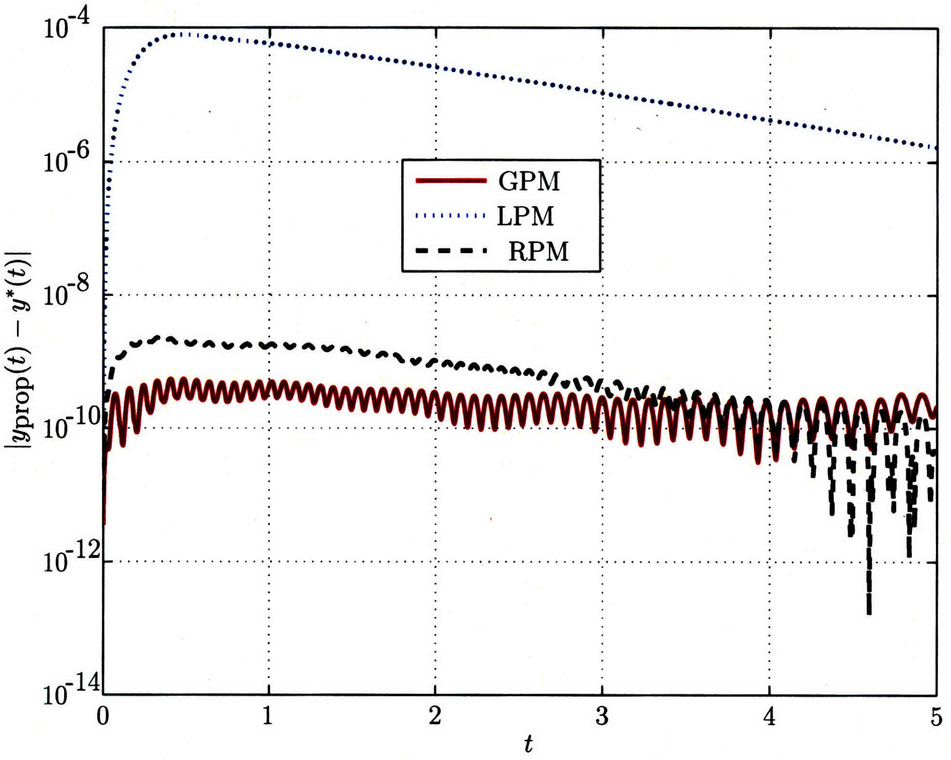


Figure 6-4: Error in the propagated state for the single state example.

for this example. As mentioned previously, Gong and Ross have devised a modified covector mapping theorem to improve the costate approximation for the LPM. This covector mapping theorem includes a set of "closure conditions" that is assumed to be implemented in a mixed primal-dual feasibility problem where the variables are the state, control, and costate. The constraints in this primal-dual feasibility problem are the KKT conditions of the NLP and the additional closure conditions. When this alternate NLP is posed and solved, the resulting costate convergence rate in Fig. 6-9 is indeed much better, although not as rapid as either the Gauss or Radau methods. As mentioned previously, this increased accuracy comes at a significant computational burden in post-processing. Due to the large number of extra steps necessary to produce this result, further computation of LPM costates involves only the original costate estimation procedure of Eq. (6.26), since this is the likely method to be used in practice.

Lastly, Fig. 6-10 depicts the convergence rate of the error in the final control, $u(t_f)$. For this figure, the final control in the GPM was computed following the approach described in Chapter 4, and the final control for the RPM and LPM methods are simply output from the

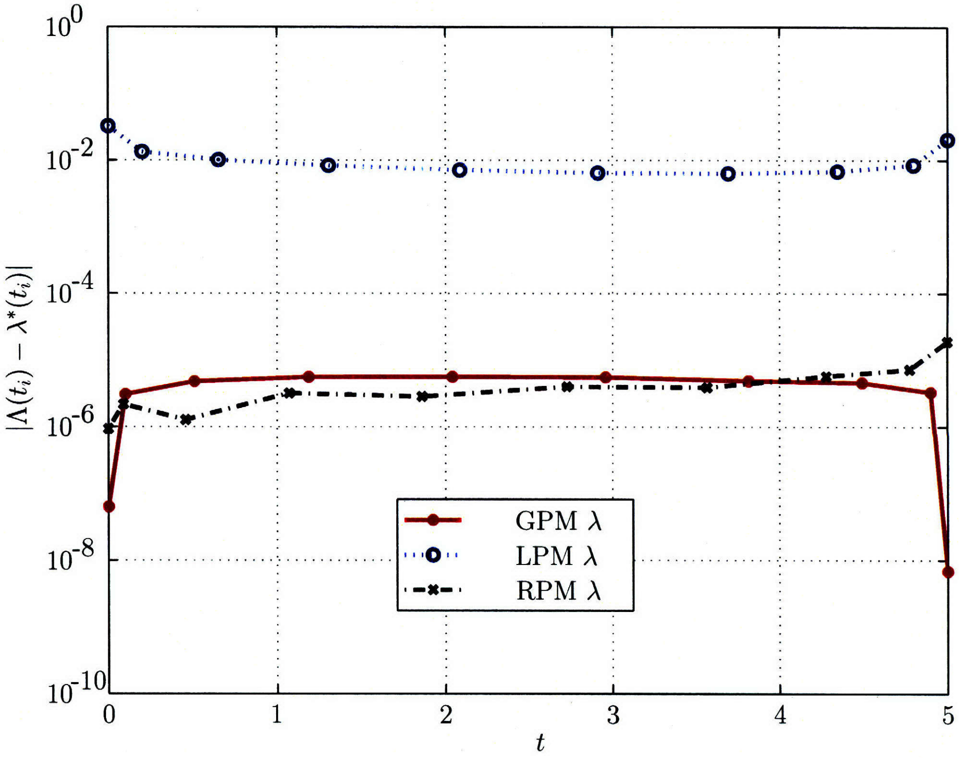


Figure 6-5: Costate error for the single state example.

NLP. This graph shows the improved accuracy in the boundary control that one can achieve when using the approach in Chapter 4, even when compared to methods that explicitly solve for the boundary control in the NLP.

6.5.2 Two State Example

As a second example, consider the following two-dimensional optimal control problem. Minimize the cost functional

$$J = y_2(t_f) \tag{6.48}$$

subject to the dynamic constraints

$$\dot{y}_1(t) = 0.5y_1(t) + u(t) \tag{6.49}$$

$$\dot{y}_2(t) = y_1^2(t) + 0.5u^2(t) \tag{6.50}$$

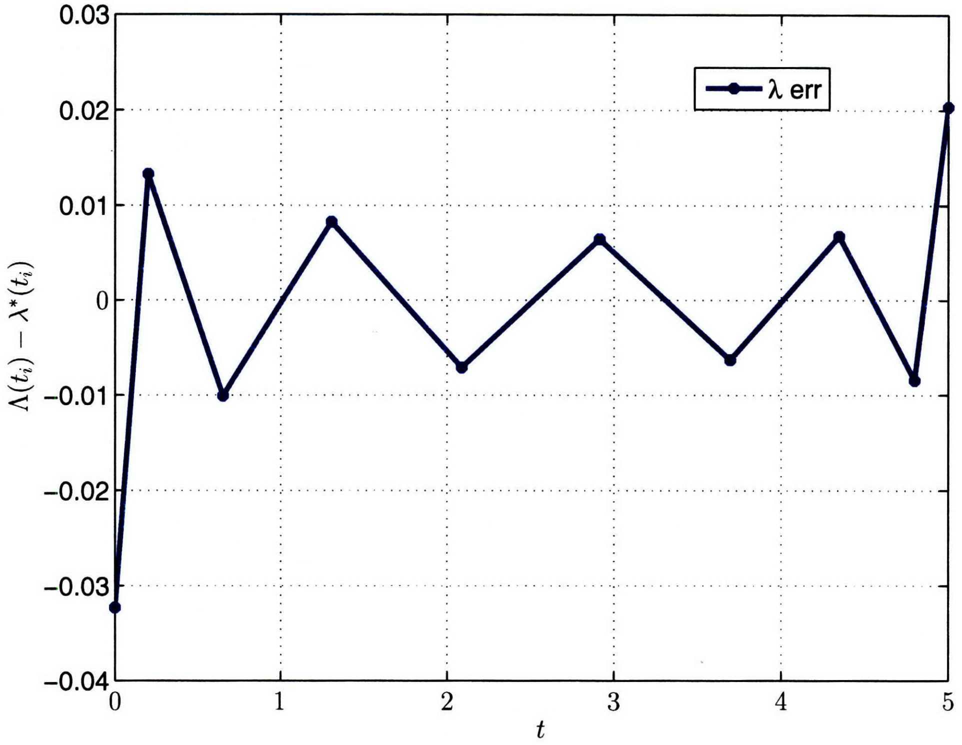


Figure 6-6: LPM costate error for the single state example.

and the boundary conditions,

$$y_1(t_0) = 1 \quad y_1(t_f) = 0.5 \tag{6.51}$$

$$y_2(t_0) = 0 \tag{6.52}$$

where $t_f = 5$. Note that this problem contains a terminal bound on the first state, $y_1(t_f)$. The exact solution is of the form

$$\begin{aligned} y_1^*(t) &= a_1 e^{1.5t} + a_2 e^{-1.5t}, && (6.53)\\ y_2^*(t) &= a_3 (e^{1.5t})^2 - a_4 (e^{-1.5t})^2 + c_1, && (6.54)\\ \lambda_1^*(t) &= a_5 e^{1.5t} + a_6 e^{-1.5t}, && (6.55)\\ \lambda_2^*(t) &= 1, && (6.56)\\ u^*(t) &= -\lambda_1^*(t) && (6.57) \end{aligned}$$

The exact solution can be seen in Fig. 6-11. Again, all three methods are compared using 10

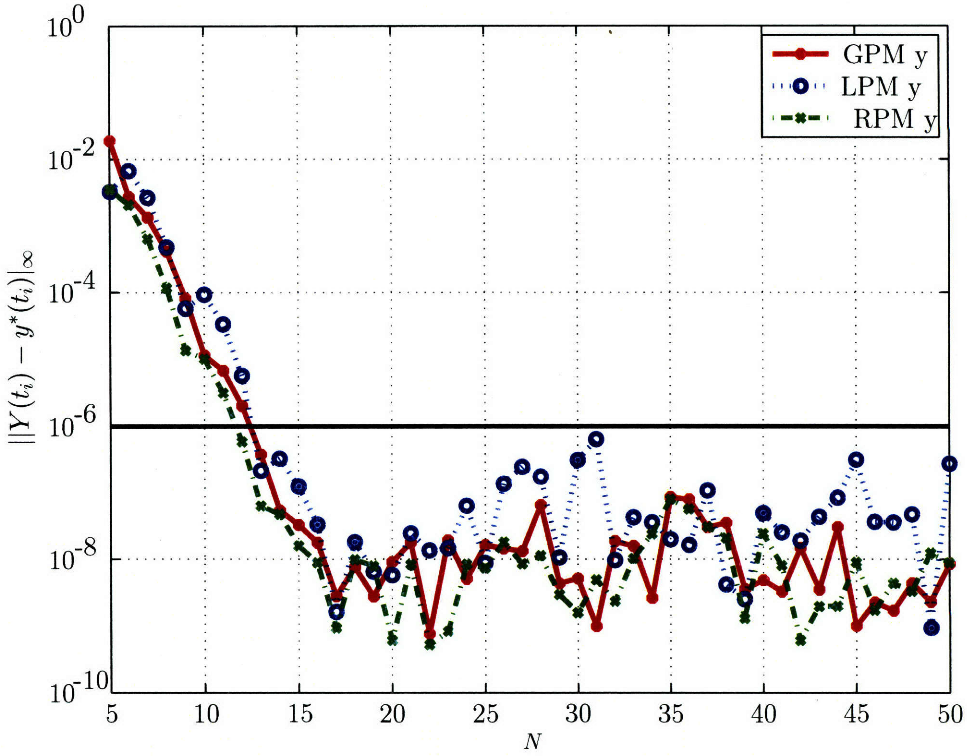


Figure 6-7: State convergence for the single state example

nodes. The error plots are formulated in the same manner as the previous example problem. Fig. 6-12 shows the state error at each node for all three methods. Recall that the error for the states involving the boundary conditions is $O(10^{-16})$, and therefore outside the range of the plot. As in the single-state example, all three methods are relatively comparable in state accuracy. Moreover, the GPM and Radau formulations again show an increase in accuracy for the unconstrained terminal state.

The control error is plotted in Fig. 6-13. Interestingly, for this problem, the LPM control estimate is worse than both GPM and RPM, even at the boundaries. The NLP control for each method is used to propagate the state equations, and their resulting state accuracy is compared in Fig. 6-14 for both states. In this example, the RPM produces the most accurate propagated solution, followed by the GPM, and lastly LPM.

Reconfirming the results in the first example, the largest discrepancy between the methods is seen in the costate comparison, and specifically with the second costate, shown in Fig. 6-15. The LPM method produces the worst costate approximation for both λ_1 and

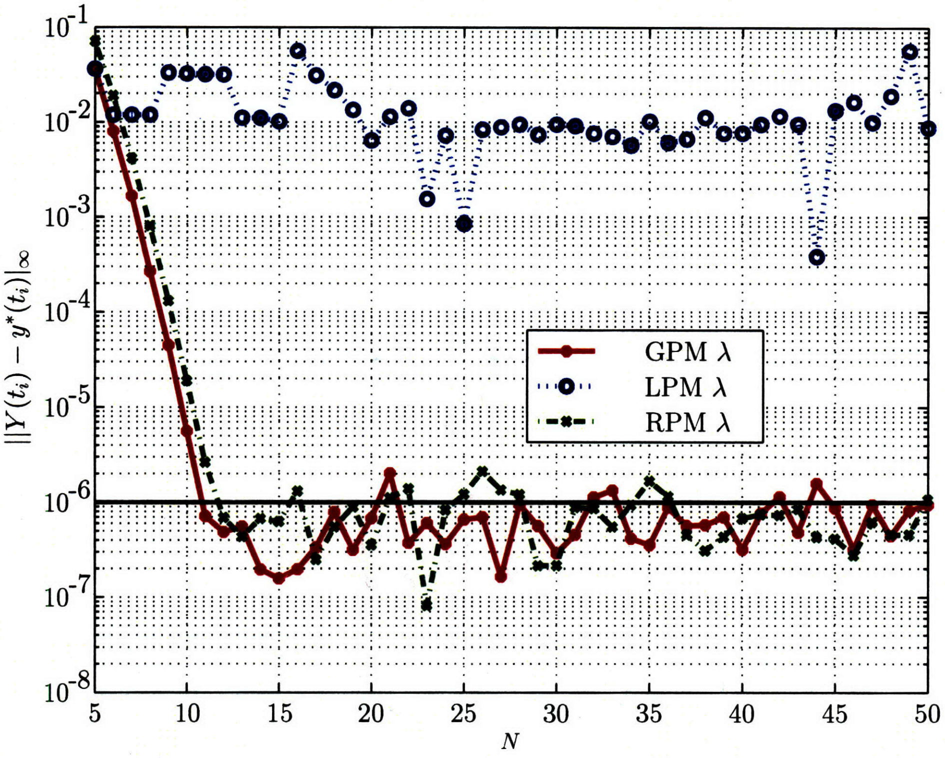


Figure 6-8: Costate convergence for the single state example

λ_2. Although it is not shown, this large error in λ_2 exhibits the same "wiggle" phenomenon around the true costate as seen in the first example. In terms of boundary costates, the GPM method provides the most accurate boundary costates, and the Radau method provides a relatively accurate initial costate, but an inaccurate final costate, as expected from the theoretical analysis. In the Radau method, the final costate is consistently the least accurate costate, suggesting that the conflicting HBVP equations described in Section 6.4 degrades the performance of this method.

The convergence rates of the state and costate are shown in Figs. 6-16 and 6-17, respectively. Each approach was solved repeatedly while the number of nodes in the problem was increased from 5 to 30. The error shown in these figures is computed in the same manner as the previous example. Fig. 6-16 displays both states, y_1 and y_2, and there is a noticeable difference between the convergence rate of y_1 and y_2 for the GPM and RPM. However, for the LPM approach, both states converge at the same (slower) rate. The state convergence rate of the RPM is the fastest for this example. It is hypothesized that this slight improvement

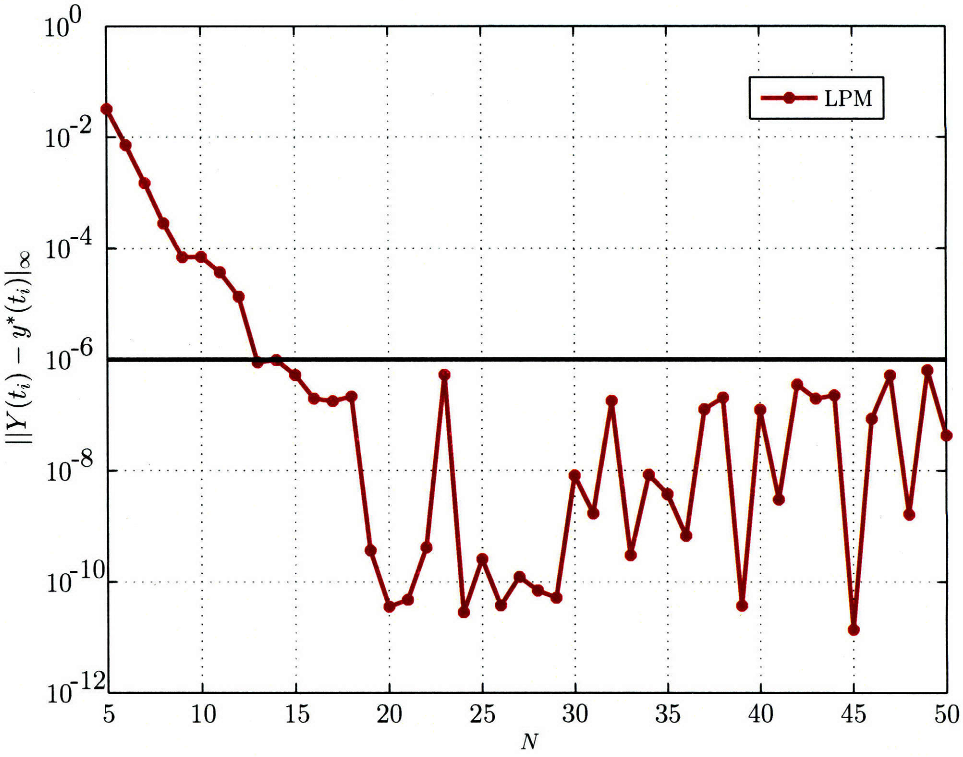


Figure 6-9: Improved LPM costate convergence via closure conditions for the single state example.

over the GPM state approximation is due to the one order reduction in the state approximation for the GPM. Similar results are seen in the costate convergence plot of Fig. 6-17. The second costate is constant in this example, and both the GPM and RPM approaches can approximate this first order polynomial quite well, even using only a few nodes, as expected with pseudospectral methods. However, the LPM approach displays the wiggle phenomenon and thus has a much slower convergence rate for λ_2. In terms of the first costate, the GPM has the fastest convergence rate.

6.5.3 Orbit-Raising Problem

Consider the problem of transferring a spacecraft from an initial circular orbit to the largest possible circular orbit in a fixed time using a low-thrust engine. The state for this problem is composed of the spacecraft radial position, $r(t)$, radial velocity, $u(t)$, and tangential velocity, $v(t)$. The control, $\phi(t)$, is the angle between the thrust vector and the tangential velocity.

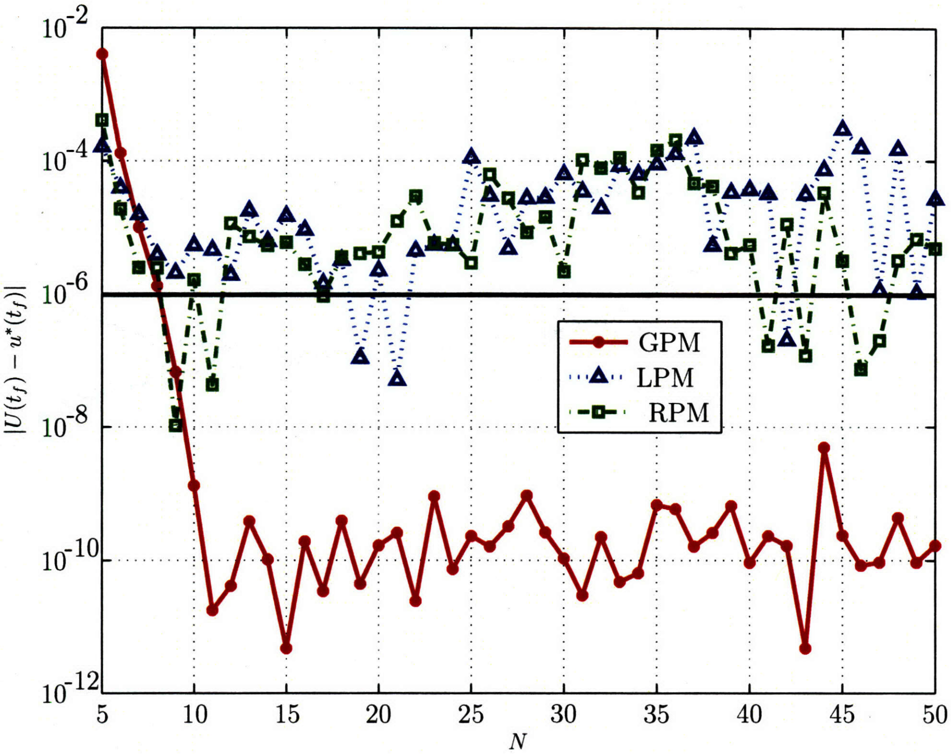


Figure 6-10: Convergence of the final control, $u(t_f)$, for the single state example.

The optimal control problem is stated as follows. Minimize the cost functional

$$J = -r(t_f) \tag{6.58}$$

subject to the dynamic constraints

$$\begin{aligned} \dot{r} &= u \\ \dot{u} &= \frac{v^2}{r} - \frac{\mu}{r^2} + \frac{T\sin\phi}{m_0 - |\dot{m}|t} \\ \dot{v} &= -\frac{uv}{r} + \frac{T\cos\phi}{m_0 - |\dot{m}|t} \end{aligned} \tag{6.59}$$

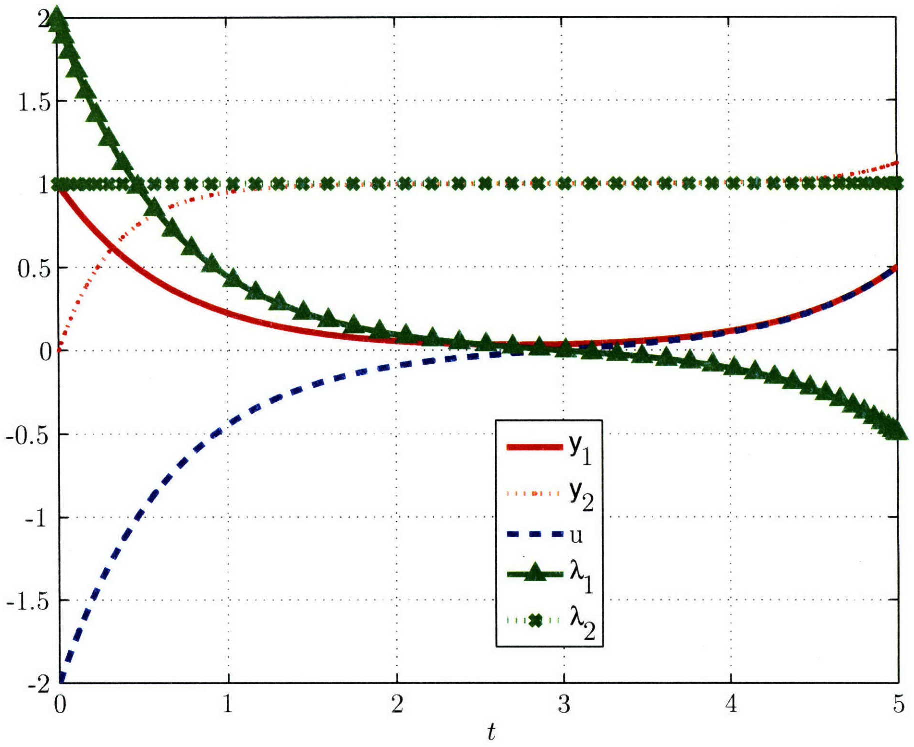


Figure 6-11: Exact solution for the two state example.

and the boundary conditions

$$
\begin{aligned}
r(0) &= 1, & r(t_f) &= \text{free} \\
u(0) &= 0, & u(t_f) &= 0, \\
v(0) &= \sqrt{\frac{\mu}{r(0)}}, & v(t_f) &= \sqrt{\frac{\mu}{r(t_f)}},
\end{aligned}
\tag{6.60}
$$

where μ is the gravitational parameter, T is the thrust magnitude, m_0 is the initial mass, and $\dot{m}$ is the fuel consumption rate. These parameters are given as follows in normalized units:

$$
T = 0.1405, \quad \dot{m} = 0.0749, \quad m_0 = \mu = 1, \quad t_f = 3.32 \tag{6.61}
$$

This problem does not have an analytic solution, but it has been solved numerically many times[6, 20] so it's solution is well known. The results of each method was compared against the solution to a boundary value problem solution using the Matlab function BVP4C with a tolerance of 10^{-9}. For brevity, an error analysis is presented for only the first state and costate, $r(t)$ and $\lambda_r(t)$. The remaining state and costate errors look very similar to the

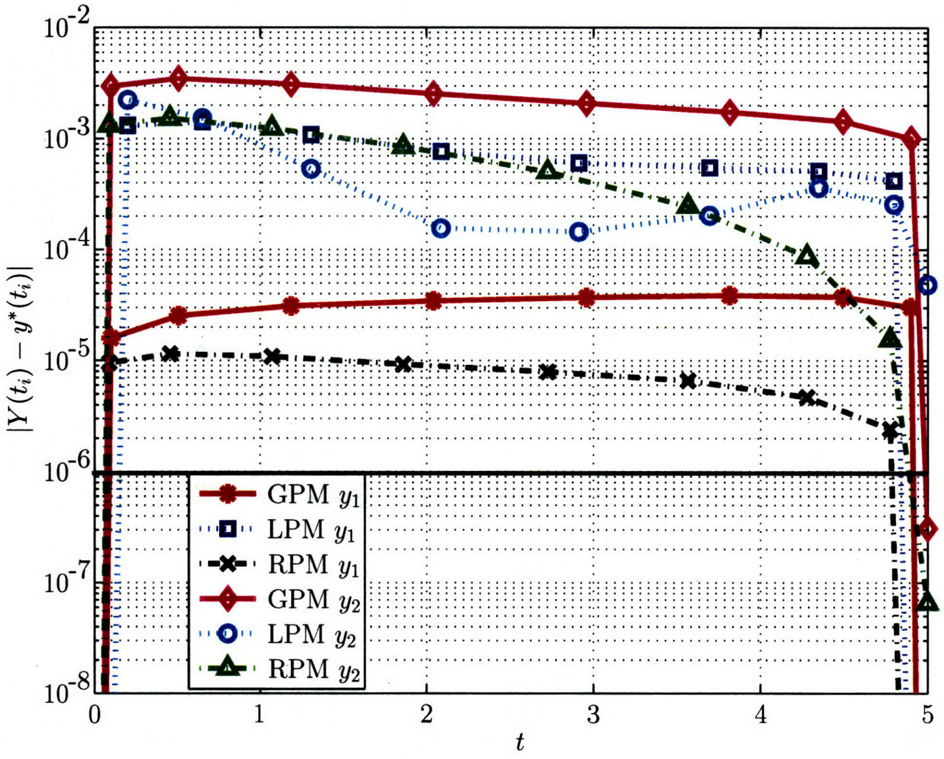


Figure 6-12: State error for the two state example

one presented. Fig. 6-18 shows the state error throughout the trajectory. The RPM has a slightly better state accuracy than the other two methods, although they all have very similar maximum state errors. Interestingly, in Fig. 6-19, the RPM also has a slightly better costate approximation than the other two methods and does not exhibit the usual reduced costate accuracy at the terminal time. As in the previous examples, the LPM costate convergence rate is significantly worse than the other two methods, shown in Fig. 6-20. Note that the convergence rates for the Radau and Gauss methods are slower than in the previous examples due to a discontinuity in the optimal control profile.

6.6 Rationale for Choosing a Method

Based on the results of all three example problems and a detailed analysis of the mathematics of each method, there may be certain circumstances under which each method should be chosen. Inaccurate boundary costates in the LPM method are attributed to boundary costs or constraints in the problem. However, if the original problem has no boundary constraints

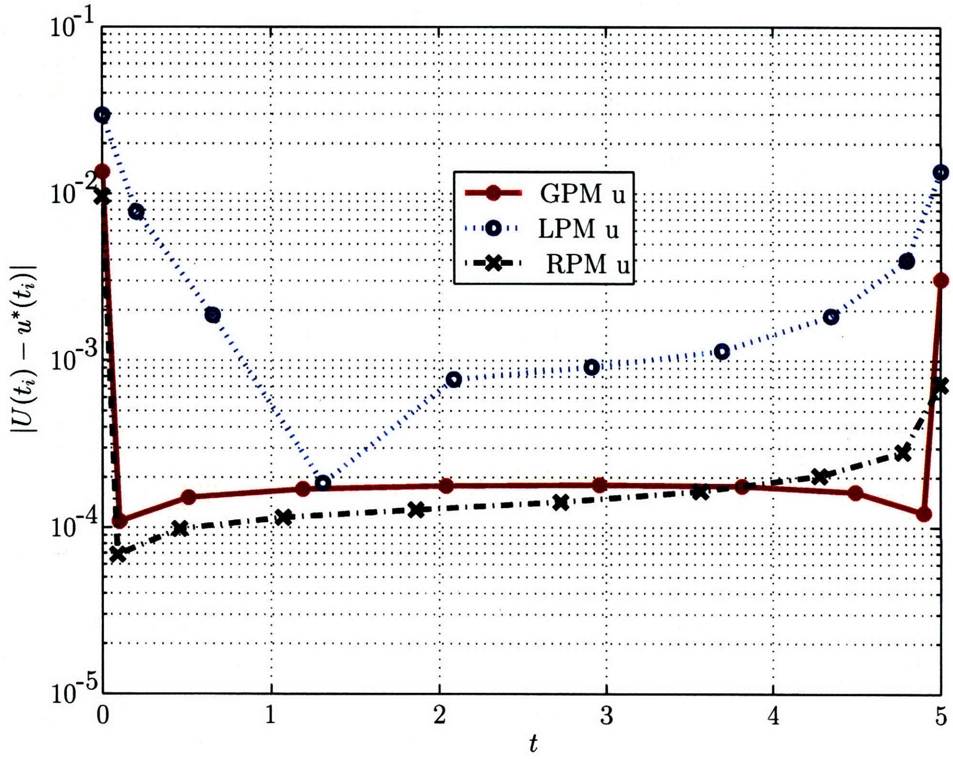


Figure 6-13: Control error for the two state example.

or costs, then the conflicting HBVP equations disappear and LPM would be an appropriate method. The RPM costate inaccuracy only occurs at the terminal time, so problems without terminal constraints would likely mitigate the errors in the final costate. The GPM has a perfect mapping with the HBVP equations, and therefore creates accurate costate estimates for general problems with both initial and final constraints.

Lastly, Table 6.2 presents NLP computation times for all three examples and all three methods. The results listed correspond to a Pentium 4, 3.2 GHz machine, where the constraint Jacobian was computed using numerical derivatives. All three methods have very similar computation times, which is expected since the methods are very similar in terms of their problem density. Note that the codes used in this analysis are not optimized for computational efficiency. Streamlined coding, analytic derivatives (or automatic differentiation), and a transition to faster programming languages like C or Fortran would significantly improve the computation times of all three methods.

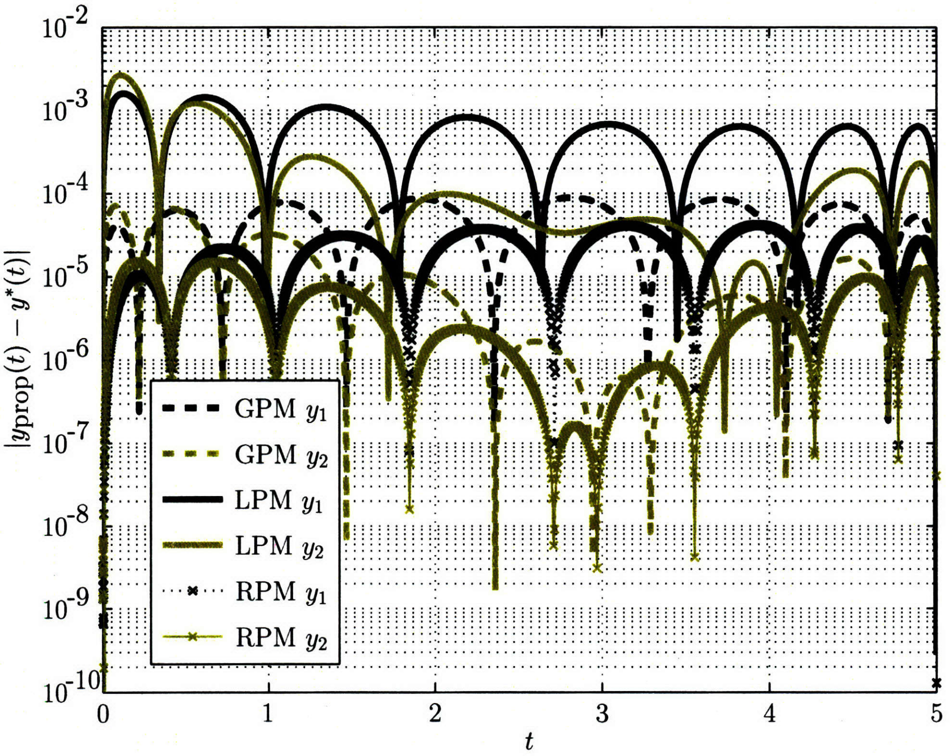


Figure 6-14: Error in the propagated state for the two state example

Table 6.2: CPU Times for each example and each method, in seconds

	GPM	RPM	LPM
Ex 1 ($N = 50$)	4.265	4.203	4.750
Ex 2 ($N = 30$)	2.203	2.296	2.688
Ex 3 ($N = 40$)	3.969	4.046	5.156

6.7 Summary

This chapter compared three established pseudospectral methods that are used to numerically solve optimal control problems. In particular, the Legendre, Radau, and Gauss pseudospectral methods have been compared in terms of their accuracy in the state, control, and costate. Three examples are used in the study to identify the key differences between the three methods. The results of this comparison indicate that the accuracy of the Radau and Gauss methods are very similar in accuracy, and both of these methods significantly outperform the Legendre method in terms of costate accuracy. Furthermore, it is found that the computational efficiency of all three methods is comparable.

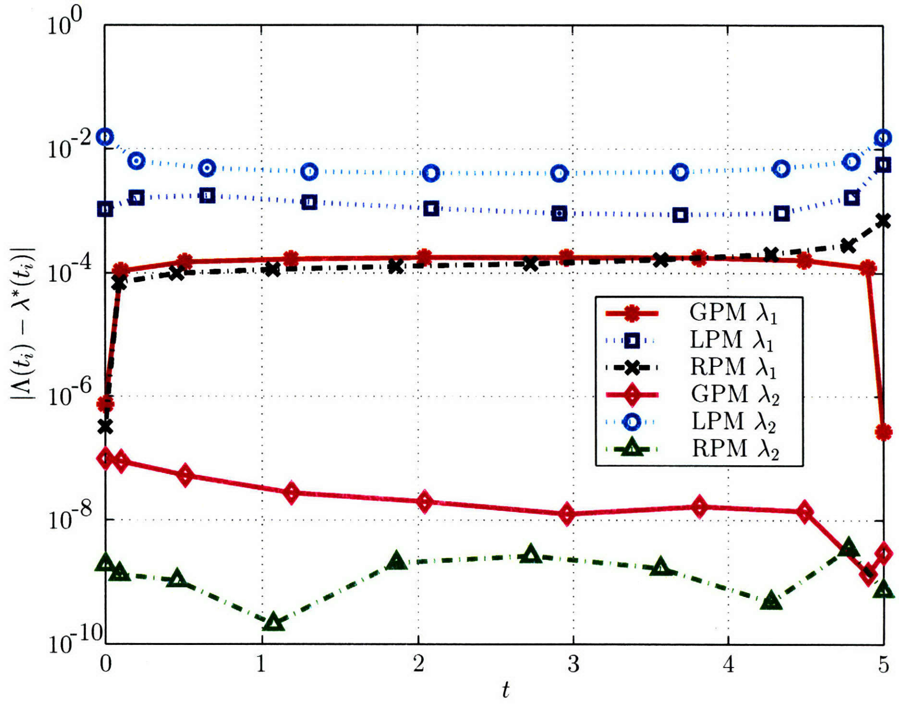


Figure 6-15: Costate error for the two state example

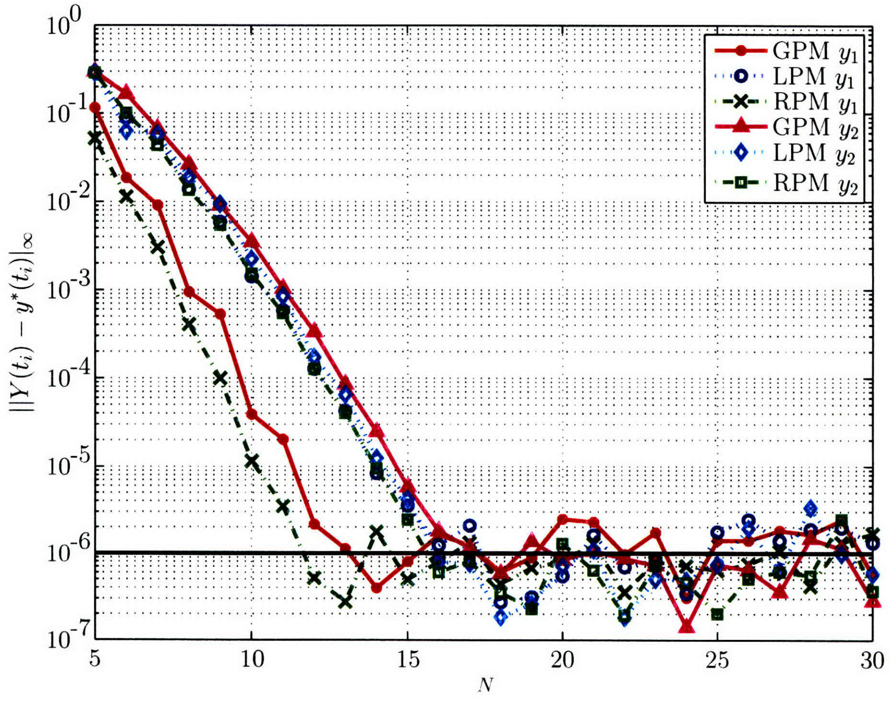


Figure 6-16: State convergence for the two state example

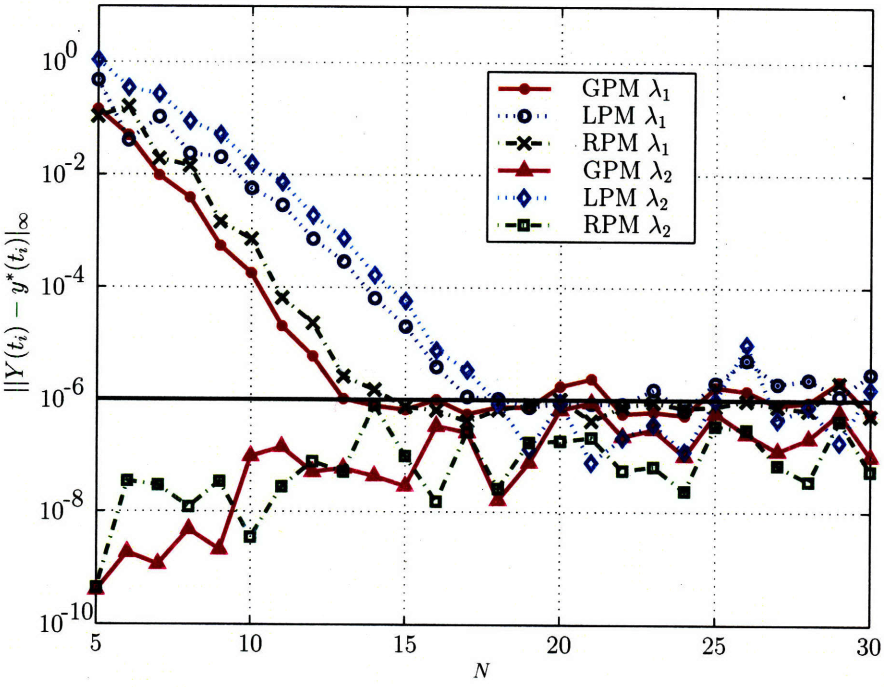


Figure 6-17: Costate convergence for the two state example

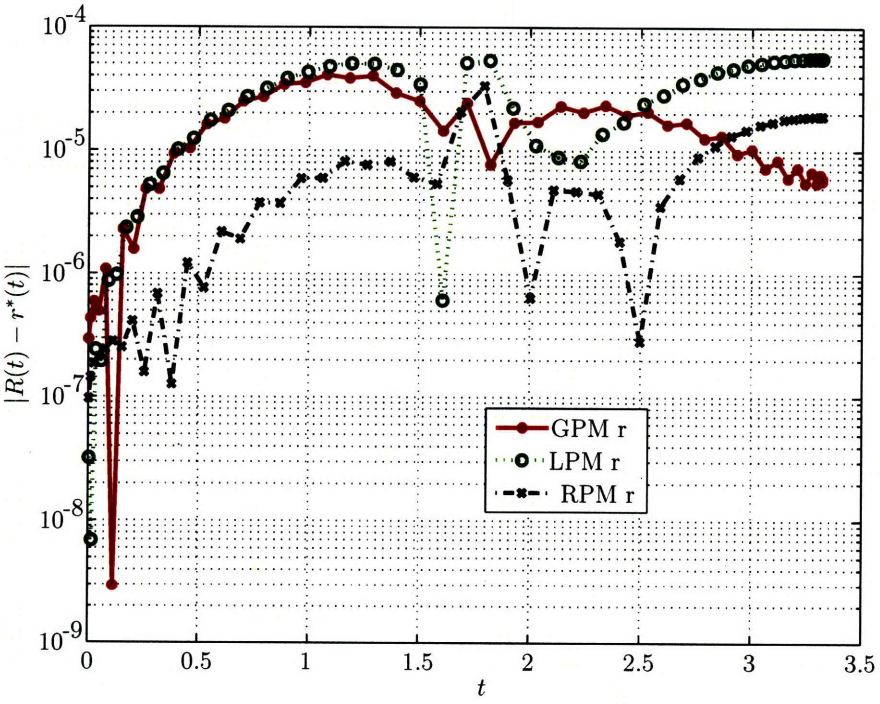


Figure 6-18: Error in state $r(t)$ for the orbit raising problem

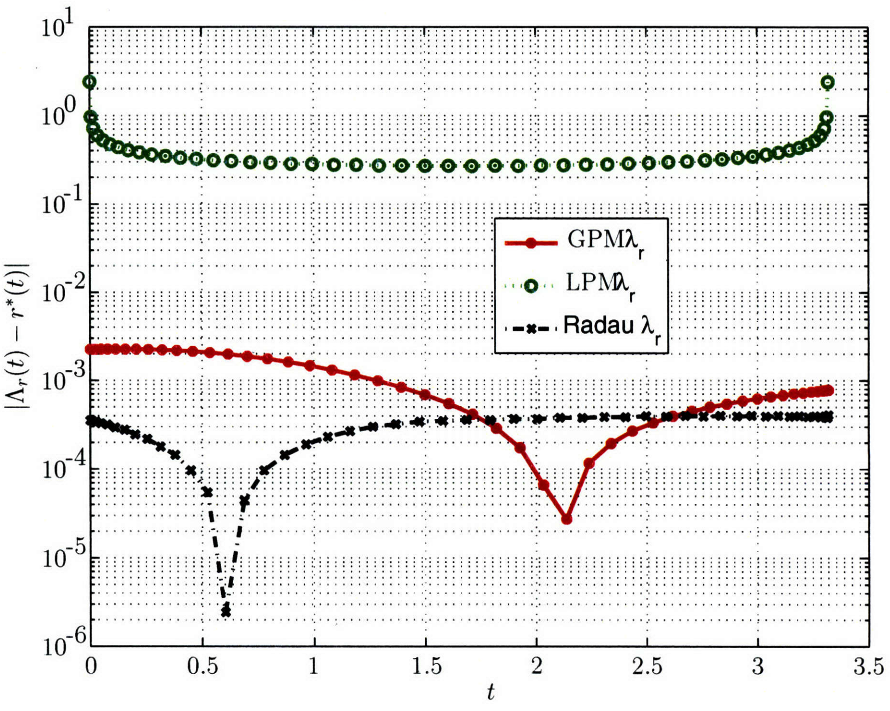


Figure 6-19: Error in the costate $\lambda_r(t)$ for the orbit raising problem

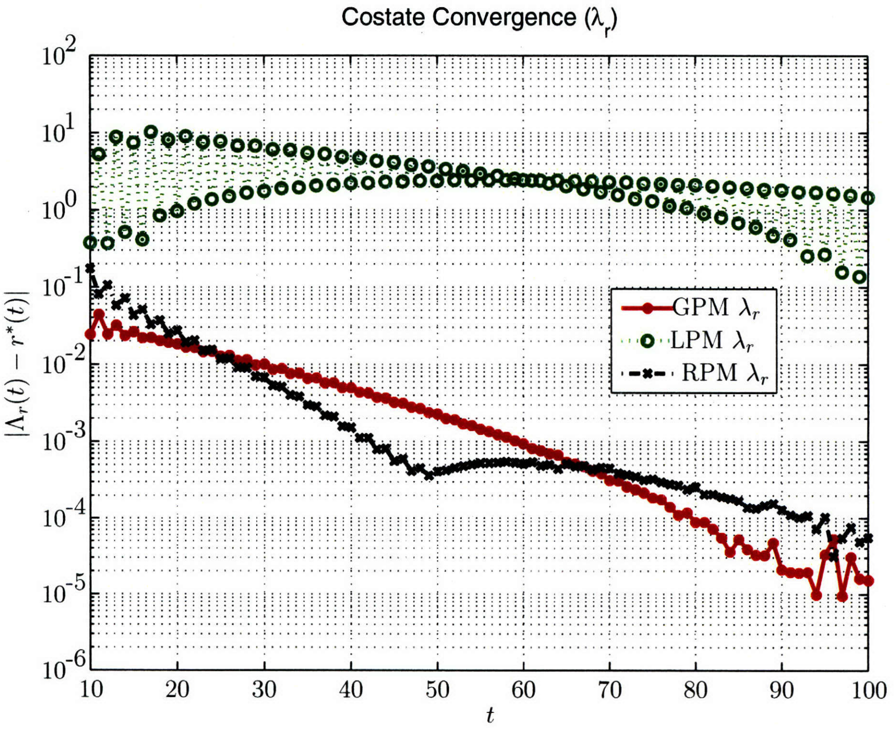


Figure 6-20: Convergence of costate λ_r for the orbit raising problem

Chapter 7

Tetrahedral Formation Flying

This chapter applies the Gauss pseudospectral method to a novel research area within aerospace engineering: spacecraft formation flying. Relatively little is known about the trajectories of satellites flying in formation, and even less is known about *fuel optimal* trajectories. The problems considered in this chapter are in reference to the Magnetospheric Multiscale (MMS) misson, which is an Earth-orbiting formation of four spacecraft working in close proximity of each other in order to study the Earth's magnetosphere. Two formation flying problems are posed in this chapter: an orbit insertion problem, simulating the fuel-optimal transfer from a parking orbit to the desired mission orbit, and a reconfiguration problem, where the formation has been significantly degraded and must optimally maneuver in order to meet the formation constraints created by the science users and minimize the total fuel used. The problems are solved using the Gauss pseudospectral method and the results are analyzed that provide new insight into the relative motion of tetrahedral formations.

7.1 Overview of Spacecraft Formation Configuration Problem

Consider the problem of maneuvering a fleet of four spacecraft from an initial parking orbit to a terminal reference orbit where the formation must attain a desired tetrahedral shape at a specified point in the terminal reference orbit. For simplicity, assume that each spacecraft

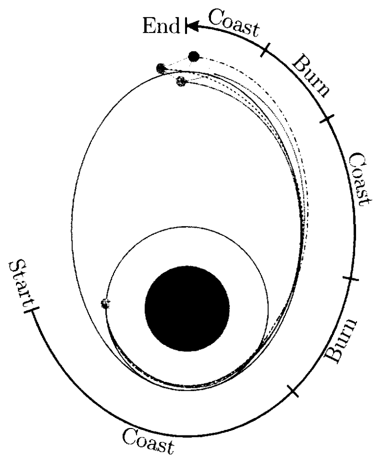


Figure 7-1: Schematic of trajectory event sequence for the orbit insertion problem

is initialized at the same point, (i.e., deployed simultaneously) on a circular parking orbit of altitude $h_0 = 600$ km and inclination $i_0 = 28$ deg. The desired terminal reference orbit is 600 km by 7000 km elliptic orbit with an inclination of 28 deg. Scientific data is taken at apogee of this reference orbit, meaning that the spacecraft must be in the desired tetrahedral formation upon reaching apogee. Constraints are placed on both the size and shape of this tetrahedral formation. Furthermore, it is desired that the spacecraft return to this tetrahedral formation repeatedly at every apogee.

The trajectory is divided into five *phases* per spacecraft with the following phase sequence: coast, burn, coast, burn, and coast. Naturally, the dynamics are different for the coast and burn phases: there is no thrust during the coast phases, and in the burn phases the thrust is constant at its maximum value $T_{\max}$. At each phase interface the state must be continuous while the control can be discontinuous. Note that the initial and terminal times of all phases are free (with the exception, of course, of the first phase where the initial time is fixed to zero). Fig. 7-1 shows a schematic of the trajectory event sequence. This five-phase trajectory clearly has two maneuver opportunities. The specifics of the configuration problem are explained in the following three sections.

7.2 Spacecraft Model and Equations of Motion

7.2.1 Spacecraft Model

This problem considers four identical spacecraft each with a dry mass of 200 kg and a fuel mass of 300 kg. The maneuvers are assumed to be non-impulsive, with the maximum thrust level of the engine set to 7.015 kN and the engine specific impulse set to 285.7 s. It is noted that these physical parameters are typical of a standard apogee kick motor.[123]

7.2.2 Dynamic Model During Thrust Phases

During all thrust phases the state of each spacecraft is modeled using modified equinoctial elements. The three degree-of-freedom equations of motion for a spacecraft moving over a spherical nonrotating Earth are given in modified equinoctial elements as[5]

$$
\begin{aligned}
\dot{p} &= \frac{2p}{w}\sqrt{\frac{p}{\mu}}a_\theta \\
\dot{P}_1 &= \sqrt{\frac{p}{\mu}}\left\{-a_r\cos L + [(w+1)\sin L + P_1]\frac{a_\theta}{w} + [Q_2\sin L - Q_1\cos L]\frac{P_2 a_n}{w}\right\} \\
\dot{P}_2 &= \sqrt{\frac{p}{\mu}}\left\{a_r\sin L + [(w+1)\cos L + P_2]\frac{a_\theta}{w} - [Q_2\sin(L) - Q_1\cos L]\frac{P_1 a_n}{w}\right\} \\
\dot{Q}_1 &= \sqrt{\frac{p}{\mu}}\left(\frac{s^2}{2w}\right)a_n\sin L \\
\dot{Q}_2 &= \sqrt{\frac{p}{\mu}}\left(\frac{s^2}{2w}\right)a_n\cos L \\
\dot{L} &= \sqrt{\mu p}\left(\frac{w}{p}\right)^2 + \sqrt{\frac{p}{\mu}}\frac{Q_2\sin L - Q_1\cos L}{w}a_n
\end{aligned}
\tag{7.1}
$$

where p is the semi-latus rectum, P_1, P_2, Q_1, and Q_2 have no geometric definition, and L is true longitude. Additionally, $w = 1 + P_1\sin L + P_2\cos L$, $s^2 = 1 + Q_1^2 + Q_2^2$, and a_r, a_θ, and a_n are the perturbing accelerations in the directions of $\mathbf{e}_r$, $\mathbf{e}_\theta$, and $\mathbf{e}_n$ where $\mathbf{e}_r$ is the unit vector in the radial direction, $\mathbf{e}_n$ is the unit vector in the direction normal to the orbital plane, and $\mathbf{e}_\theta = \mathbf{e}_n \times \mathbf{e}_r$ (thereby completing the right-handed system $\{\mathbf{e}_r, \mathbf{e}_\theta, \mathbf{e}_n\}$). For the application under consideration here, the perturbing accelerations are due entirely to thrust and can be written as

$$a_r = \frac{T}{m}u_r, \qquad a_\theta = \frac{T}{m}u_\theta, \qquad a_n = \frac{T}{m}u_n, \tag{7.2}$$

where T is the thrust magnitude, m is the spacecraft mass, and u_r, u_θ, and u_n are the $\mathbf{e}_r$, $\mathbf{e}_\theta$, and $\mathbf{e}_n$ components, respectively, of the thrust direction. Finally, the mass flow rate of the engine is governed by the equation

$$\dot{m} = -\frac{T}{g_0 I_{sp}} \tag{7.3}$$

where g_0 is the sea level acceleration due to gravity and I_{sp} is the specific impulse of the engine.

7.2.3 Dynamic Model for Coast Phases

By assuming a spherical Earth gravity model, the only equinoctial element that changes with time during a coast phase is the true longitude, L. In particular, the differential equation for the true longitude can be written as

$$\dot{L} = \frac{dL}{dt} = \frac{\sqrt{\mu p}}{p^2}(1 + P_1 \sin L + P_2 \cos L)^2 = f(L, p, P_1, P_2) \tag{7.4}$$

Observing that all quantities except L in Eq. (7.4) are constant results in a separation of variables in Eq. (7.4) to give

$$dt = \frac{dL}{f(L, p, P_1, P_2)} \tag{7.5}$$

Integrating both sides of Eq. (7.5) produces

$$t_f - t_0 = \int_{L_0}^{L_f} \frac{dL}{f(L, p, P_1, P_2)} \tag{7.6}$$

where t_0 and t_f are the initial and terminal time, respectively, of the coast phase while $L(t_0) = L_0$ and $L(t_f) = L_f$ are the initial and terminal true longitude. Since all other states are constant during a coast phase, the dynamic model for each spacecraft is given as

$$t_f - t_0 - \int_{L_0}^{L_f} \frac{dL}{f(L, p, P_1, P_2)} = 0, \qquad i = 1, \ldots, 4 \tag{7.7}$$

All other components of the state (i.e., p, P_1, P_2, Q_1, Q_2, and m) are treated as constant optimization parameters. It is noted that, due to additional terminal constraints, the above model is not used for the final coast phase.

7.2.4 Dynamic Model for Terminal Phase

The terminal constraints in this problem focus largely on the relative position of the spacecraft, and are easily expressed in Cartesian Earth-centered inertial (ECI) coordinates. Therefore it makes sense to use a dynamic model that is consistent with the mathematical form of the constraints. Consequently, as opposed to modified equinoctial elements, the dynamics of each spacecraft in the final coast phase are described using ECI coordinates. This transition is by no means necessary, but it both reduces the complexity of the analytic derivatives of the equations of motion and exemplifies the versatility inherent in the Gauss pseudospectral method. The new dynamic model is given below as

$$\dot{\mathbf{r}} = \mathbf{v}, \qquad \dot{\mathbf{v}} = -\mu \frac{\mathbf{r}}{||\mathbf{r}||^3} \tag{7.8}$$

where $\mathbf{r}$ is the position vector measured from the center of the Earth, $\mathbf{v}$ is the inertial velocity, and μ is the gravitational parameter. In order to maintain continuity between the phases, the Cartesian variables are transformed to modified equinoctial elements at the start of the final phase and are set equal to the elements at the final time of the previous phase via linkage conditions,[14,57] explained in the next section.

7.3 Constraints

This section describes the many constraints imposed in the orbit insertion problem. Some constraints are a direct result of the posed problem, while others are specific to the application of the GPM to this problem.

7.3.1 Initial Conditions

All four spacecraft start in the same circular orbit at time $t = 0$. The initial conditions corresponding to this orbit are given in orbital elements as

$$\begin{aligned} a(0) &= R_e + h_0 \quad , \quad & e(0) &= 0 \\ i(0) &= 28 \text{ deg} \quad , \quad & \omega(0) &= 270 \text{ deg} \\ \Omega(0) &= 0 \quad , \quad & \nu(0) &= 270 \text{ deg} \end{aligned} \tag{7.9}$$

where a is the semi-major axis, e is the eccentricity, i is the inclination, ω is the argument of perigee, Ω is the longitude of the ascending node, ν is the true anomaly, R_e is the radius of the Earth, and $h_0 = 600$ km is the initial altitude. It is noted that the initial argument of perigee, $\omega(0)$, is chosen to be the same as that of the terminal reference orbit while the initial true anomaly, $\nu(0)$, is arbitrary. The orbital elements in Eq. (7.9) are then converted to modified equinoctial elements using a transformation T_{o2e} (see for example Ref. 5) to obtain the initial state in modified equinoctial elements as

$$\left.\begin{aligned} p^{(i)}(t_0) &= p_0 \\ P_1^{(i)}(t_0) &= P_{1,0} \\ P_2^{(i)}(t_0) &= P_{2,0} \\ Q_1^{(i)}(t_0) &= Q_{1,0} \\ Q_2^{(i)}(t_0) &= Q_{2,0} \\ L^{(i)}(t_0) &= L_0 \end{aligned}\right. \quad , \quad (i = 1, \ldots, 4) \tag{7.10}$$

where i is the i^{th} spacecraft. Furthermore, the initial mass of each spacecraft is equal to its maximum value, i.e.,

$$m^{(i)}(t_0) = m_{\max} \quad , \quad (i = 1, \ldots, 4) \tag{7.11}$$

where $m_{\max} = 500$ kg. Lastly it is noted that for this preliminary analysis, it is reasonable to assume that all four spacecraft can start at the same initial point without any conflict. However, if more realism is added to the problem, issues such as collision avoidance must be addressed.

7.3.2 Interior Point Constraints

In order for the state to be continuous at each phase interface, it is necessary to enforce linkage conditions at the phase boundaries. These linkage conditions are enforced on the modified equinoctial elements, mass, and time as

$$\begin{array}{rcl} p^{(i)}(t_f^{(j)}) & = & p^{(i)}(t_0^{(j+1)}) \\ P_1^{(i)}(t_f^{(j)}) & = & P_1^{(i)}(t_0^{(j+1)}) \\ P_2^{(i)}(t_f^{(j)}) & = & P_2^{(i)}(t_0^{(j+1)}) \\ Q_1^{(i)}(t_f^{(j)}) & = & Q_1^{(i)}(t_0^{(j+1)}) \\ Q_2^{(i)}(t_f^{(j)}) & = & Q_2^{(i)}(t_0^{(j+1)}) \\ L^{(i)}(t_f^{(j)}) & = & L^{(i)}(t_0^{(j+1)}) \\ m^{(i)}(t_f^{(j)}) & = & m^{(i)}(t_0^{(j+1)}) \\ t_f^{(i)^{(j)}} & = & t_0^{(i)^{(j+1)}} \end{array} \quad , \quad \begin{array}{c} (i = 1, \ldots, 4) \\ (j = 1, \ldots, P-1) \end{array} \tag{7.12}$$

where j is the j^{th} phase and P is the number of phases in the problem (in this case $P =$ 5). Finally, in order to ensure that time is increasing during the trajectory, the following inequality constraint is placed on time during each phase:

$$t_f^{(i)^{(j)}} - t_0^{(i)^{(j)}} \geq 0, \qquad (i = 1, \ldots, 4), \; (j = 1, \ldots, P) \tag{7.13}$$

7.3.3 Path Constraints

The following two path constraints are imposed on the four spacecraft. First, during the thrust phases of the trajectory it is necessary to constrain the thrust direction to be unit length. Defining the thrust direction as $\mathbf{u}_T = [u_r \; u_\theta \; u_n]^T$, the following constraint is imposed on $\mathbf{u}_T$:

$$\mathbf{u}_T \cdot \mathbf{u}_T = 1 \tag{7.14}$$

Second, during flight the mass of each spacecraft cannot fall below the vehicle dry mass. Defining the dry mass of each vehicle as m_{dry}, the following inequality constraint is imposed on the mass of each spacecraft during the each phase of the trajectory:

$$m^{(i)}(t) \geq m_{\text{dry}} \quad , \quad (i = 1, \ldots, 4) \tag{7.15}$$

7.3.4 Terminal Constraints

The position of the *mesocenter* of the formation is defined as

$$\bar{\mathbf{r}} = \frac{1}{4} \sum_{i=1}^{4} \mathbf{r}^{(i)} \tag{7.16}$$

where $\mathbf{r}^{(i)}$ is the position of the i^{th} spacecraft as measured from the center of the Earth. At time t_f, the mesocenter must coincide with the apogee of the terminal orbit. Defining the position of the terminal orbit apogee as $\mathbf{r}_a$, this constraint is given as

$$\bar{\mathbf{r}}(t_f) = \mathbf{r}_a \tag{7.17}$$

Similarly, velocity of the mesocenter of the formation is defined as

$$\bar{\mathbf{v}} = \frac{1}{4} \sum_{i=1}^{4} \mathbf{v}^{(i)} \tag{7.18}$$

where $\mathbf{v}^{(i)}$ is the velocity of the i^{th} spacecraft. Since six orbital elements are necessary to completely define an orbit's characteristics, constraining both the position and velocity of the mesocenter at $t = t_f$ ensures that the mesocenter is on the reference orbit for that instant in time. Defining the velocity of the reference orbit apogee as $\mathbf{v}_a$, the constraint imposed on the velocity of the formation mesocenter is given as

$$\bar{\mathbf{v}}(t_f) = \mathbf{v}_a \tag{7.19}$$

Next, the formation must form a tetrahedron upon reaching apogee of the final orbit. To ensure the quality of this tetrahedron, a set of constraints are formed that take into account both the formation *size* and *shape*. First, it is noted that the spacecraft must be within a certain distance of each other to take useful measurements. Although an average inter-spacecraft separation of 10 km is considered ideal, acceptable science return is still possible for average separations ranging from 4 km to 18 km.[62,63] For this application, the formation size is constrained by placing bounds on the formation's average side length, $\bar{L}$.

$$4 \leq \bar{L} \leq 18 \text{ km}. \tag{7.20}$$

In addition to the formation size, the formation shape must meet certain performance criteria. The metric used to determine the quality of the shape of the formation is called the Q_{R8} Geometric Factor,[88] and is given as

$$Q_{R8} = V_a / V^* \tag{7.21}$$

where V_a is the actual volume of the tetrahedron, and V^* is the (ideal) volume of a regular tetrahedron whose side length is $\bar{L}$. It is seen that the Q_{R8} metric contains no sensitivity to the size of the tetrahedron. For this work, values of Q_{R8} greater than 0.9 are considered acceptable. Consequently, the following terminal constraint is enforced:

$$Q_{R8} \geq 0.9 \tag{7.22}$$

See Ref. 88 for details on how to compute the Q_{R8} Geometric Factor. Next, in the absence of perturbations, a spacecraft's orbit period is solely a function of semi-major axis. In this case, spacecraft with equal semi-major axes (and therefore equal periods) will return to their relative positions exactly one period later. Consequently, the entire formation will be periodic if all spacecraft involved have the same semi-major axis.

$$a^{(1)}(t_f^{(P)}) = a^{(2)}(t_f^{(P)}) = a^{(3)}(t_f^{(P)}) = a^{(4)}(t_f^{(P)}) \tag{7.23}$$

Finally, it is required that the trajectories of all four spacecraft terminate at the same time, i.e.,

$$t_f^{(1)^{(P)}} = t_f^{(2)^{(P)}} = t_f^{(3)^{(P)}} = t_f^{(4)^{(P)}} \tag{7.24}$$

where t_f is *free.*

7.4 Spacecraft Orbit Insertion Optimal Control Problem

The spacecraft configuration optimal control problem is now stated formally. Using the aforementioned trajectory event sequence, determine the thrust profile that maximizes the sum of the terminal masses of each spacecraft, i.e., maximize the cost functional

$$J = \sum_{i=1}^{4} m^{(i)}(t_f^{(P)}) \tag{7.25}$$

subject to the differential equation constraints of Eq. (7.1) and Eq. (7.3), the initial constraints of Eq. (7.10) and Eq. (7.11), the interior point constraints of Eq. (7.12) and Eq. (7.13), the path constraints of Eqs. (7.14) and (7.15), and the terminal constraints of Eq. (7.17), Eq. (7.19), Eq. (7.20), and Eqs. (7.22)–(7.24).

7.5 Numerical Solution via Gauss Pseudospectral Method

The spacecraft configuration problem as described in Sections 7.1 – 7.4 is solved using the aforementioned Gauss pseudospectral method described in Chapter 3. The optimization was implemented using the MATLAB mex interface of the NLP solver SNOPT[46] with analytic first-order derivatives for both the constraint Jacobian and the gradient of the objective function. Furthermore, the optimal control problem is scaled canonically from SI units to a set of units such that the length scale is the radius of the Earth, the time scale is one Schuler period, and the mass scale is equal to the initial spacecraft mass. For this problem, 15 nodes (i.e., 13 LG points) were used to approximate the trajectory in the burn phases. Since the first two coast phases utilize either static variables or a simple quadrature constraint, a pseudospectral approximation is not necessary for those phases. The quadrature can be computed numerically from the initial and final state; therefore only two nodes were needed in these coast phases. The final coast phase, which is calculated in ECI coordinates instead of equinoctial elements, requires a pseudospectral approximation. Because the duration of the final coast phase is unknown *a priori*, 40 nodes (i.e., 38 LG points) were used in the final coast phase. Note that these node amounts were chosen in order to accurately approximate the solution yet still maintain a reasonably sized NLP. In the future it is likely that the number of nodes per phase could become an optimization variable itself. In fact, some more mature optimization software codes such as SOCS[14] do have this feature, but this ability has not been developed for the GPM.

7.6 Results

The key results of the configuration problem are discussed in this section. The first subsection summarizes the results for the two-maneuver configuration problem, and the second subsection summarizes the results for a single-maneuver configuration problem, where each spacecraft is given only one maneuver opportunity. Lastly, the third subsection provides a detailed post-optimality analysis of the single-maneuver solution.

7.6.1 Two-Maneuver Solution

Recall that the formation must attain a geometry that is within 10% of a regular tetrahedron (measured by the quality factor) and have an average inter-spacecraft distance between 4 and 18km at the terminal time. Furthermore, the spacecraft mesocenter must have a position and velocity that corresponds to the apogee of the reference orbit, and the spacecraft must have equal semi-major axes and terminal times. The solution from the NLP does indeed satisfy all these constraints, and interestingly, the minimum-fuel formation resides at the lower bound for both the volume constraint and average inter-spacecraft distance. This result is largely intuitive, since maneuvering the spacecraft to a larger separation distance would naturally require more fuel, and is counter-productive to the given cost function. Fig. 7-2 provides a three-dimensional perspective of the terminal tetrahedron where x, y, and z represent the standard Earth-centered inertial coordinate system.[24]

Notice in Table 7.1 that, despite the presence of two maneuver opportunities, only one spacecraft actually thrusts twice. Three of the four spacecraft are able to complete the

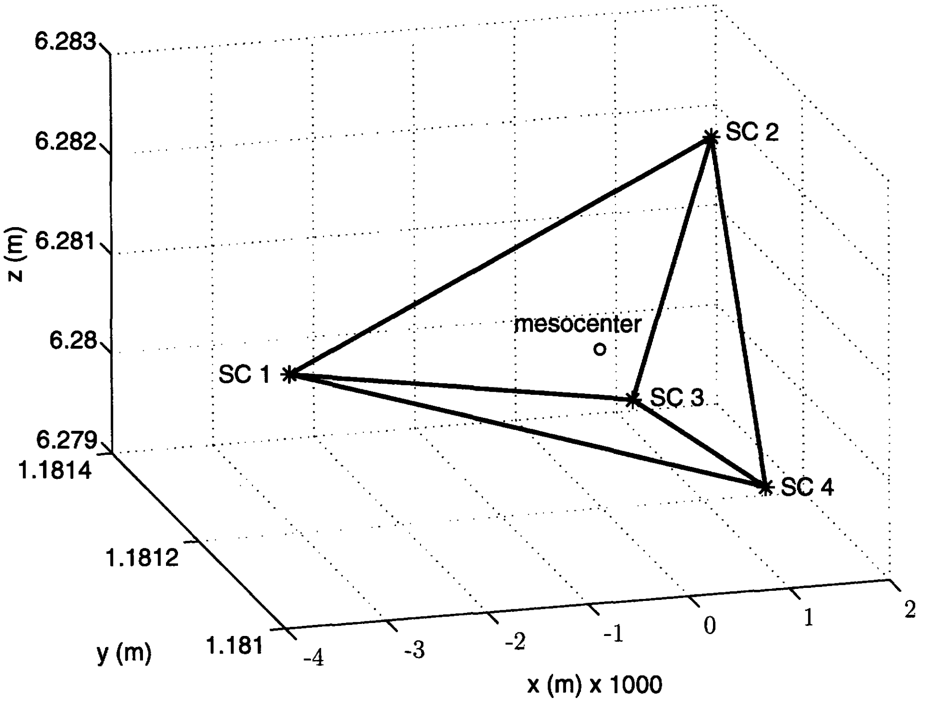


Figure 7-2: Three-dimensional view of optimal terminal tetrahedron for two-maneuver problem

Table 7.1: Maneuver durations for the two-maneuver problem

Burn Duration (s) \ Spacecraft Number	1	2	3	4
First Burn	65.1965	65.2958	65.2960	65.1584
Second Burn	0	0	0	0.0356

transfer and maintain equal semi-major axes with only the first maneuver. The fourth spacecraft, however, uses a slight corrective maneuver later in the orbit, at a true anomaly of approximately 90 deg. Interestingly, as seen in Fig. 7-3, the fourth spacecraft is also the spacecraft with the lowest final altitude. The small second maneuver is predominantly in the tangential direction (see Fig. 7-4) and raises the semi-major axis of that spacecraft to match the other three. With such a small second maneuver, one wonders if two maneuvers is even necessary. The next section shows that this problem can be solved with only one maneuver allowed per spacecraft, but the one-maneuver solution uses more fuel than the two-maneuver trajectory. Further discussion comparing the two scenarios is presented in the next subsection.

As seen in Fig. 7-5, the optimal locations of the first maneuver for each of the spacecraft

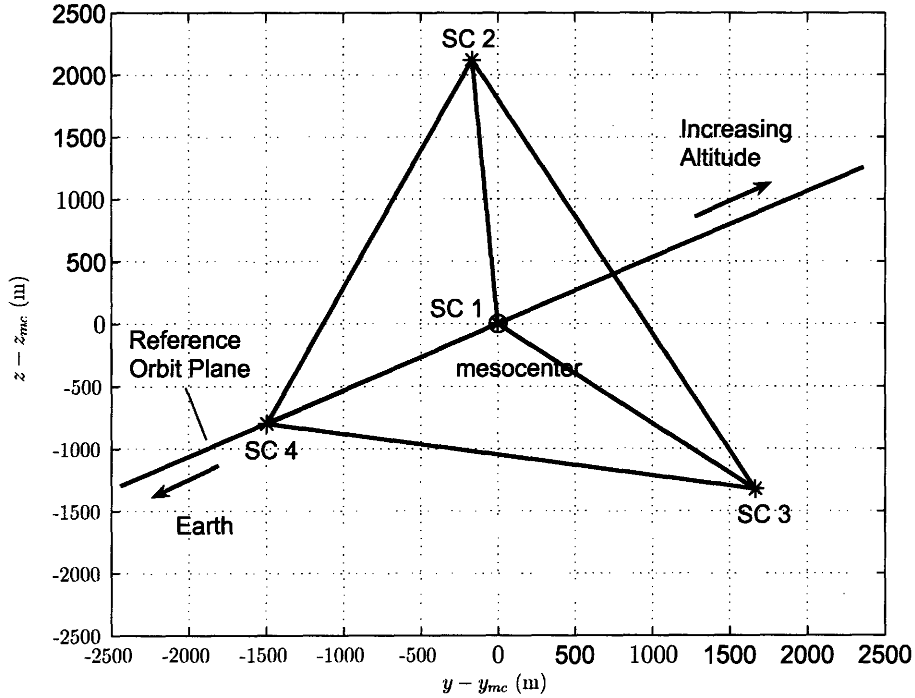


Figure 7-3: Optimal terminal tetrahedron viewed along the orbital plane for the two-maneuver problem

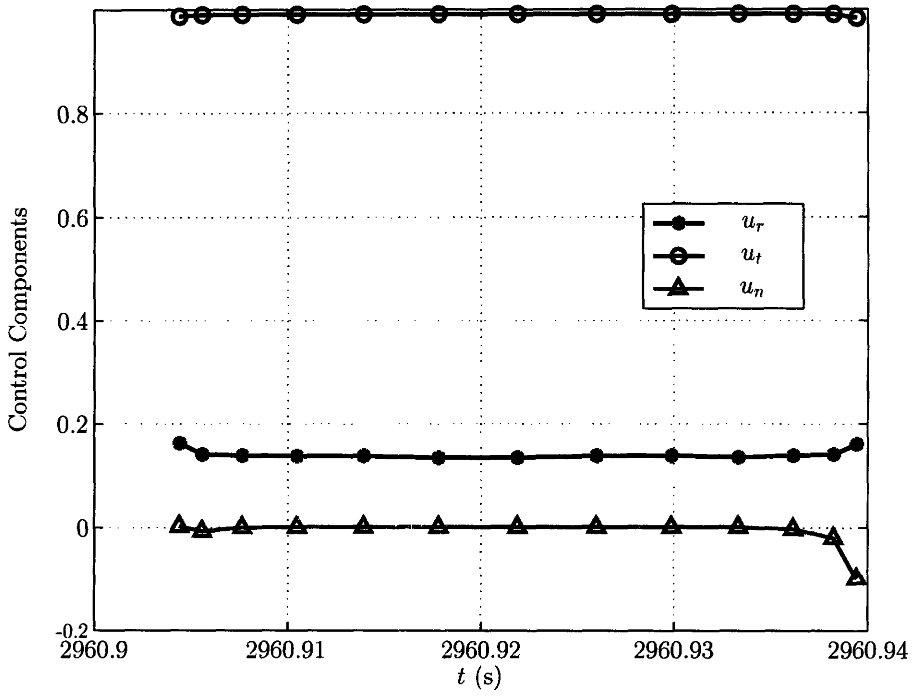


Figure 7-4: SC #4 control for the 2^{nd} maneuver in the two-maneuver problem

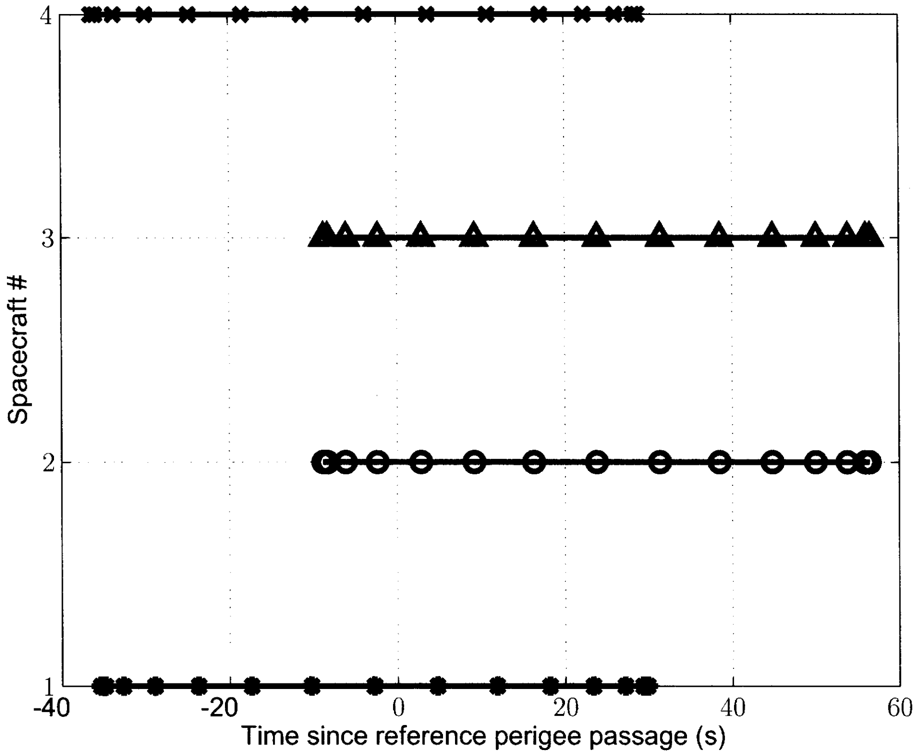


Figure 7-5: Spacecraft burn durations relative to time of reference perigee passage, $t - t_{\text{perigee}}$, for spacecraft $i = 1, \ldots, 4$ during the first burn phase for the two-maneuver problem

are near the reference perigee, resembling a Hohmann transfer. Upon closer inspection of Fig. 7-5, it is seen that all four burns are offset by varying amounts from the reference perigee. By staggering the burns, the three spacecraft reach the desired geometric positions but attain the same semi-major axis upon completion of the maneuver. These maneuvers do use slightly different amounts of fuel, and when comparing Fig. 7-6 and Fig. 7-3 it is seen that the amount of fuel burned by each spacecraft increases with increasing terminal altitude.

One interesting aspect of the formation is the discovery of the fuel-optimal *orientation* of the tetrahedron. Recall that no constraints are placed on the orientation of the formation, yet one can see a definite structure to the solution by looking at Figs. 7-3 and 7-7. In Fig. 7-3, it is seen that two of the spacecraft remain in the orbit plane, while the other two spacecraft are located symmetrically out of the orbit plane. Figure 7-7 shows the view of the formation from a direction normal to the orbit plane. In particular, Fig. 7-7 shows the terminal position of each spacecraft relative to the mesocenter where $x_{r,mc}, x_{\theta,mc}, x_{n,mc}$

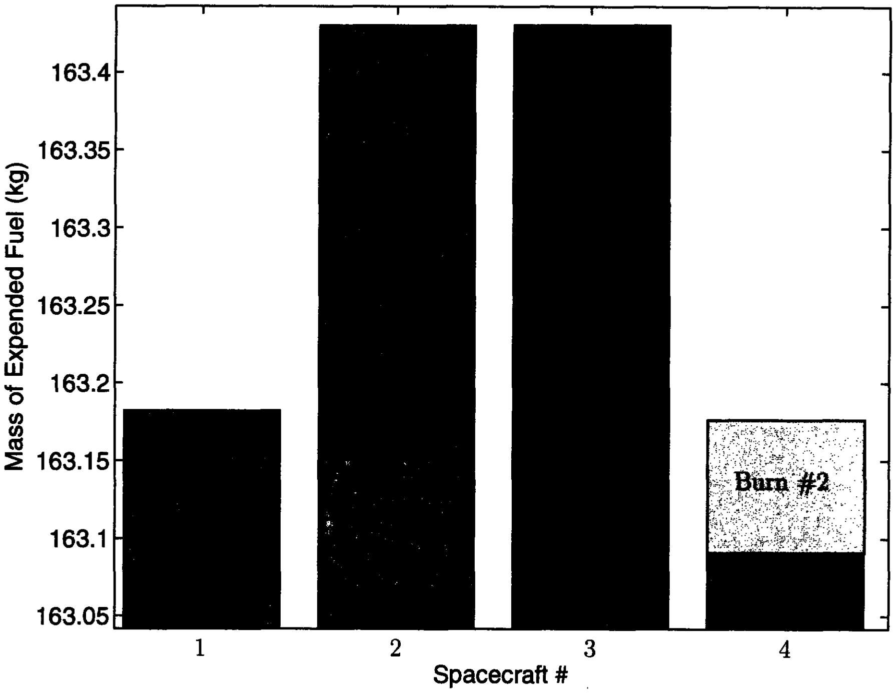


Figure 7-6: Optimal fuel consumption $m_0^{(i)} - m_f^{(i)}$ of spacecraft $i = 1, \ldots, 4$ for the two-maneuver problem

represent the instantaneous radial, transverse, and normal components of the position of the mesocenter. It is seen from Fig. 7-7 the two out-of-plane spacecraft (2 and 3) have the same projection into the orbit plane. Furthermore, the altitude of spacecraft 4, which burns twice, is significantly lower than the altitude of the other three spacecraft. Although the solution is not guaranteed to be globally optimal due to the nonconvexity of the problem, it was found that several different initial guesses all converged to the solution obtained in this analysis.

Figs. 7-8, 7-9, and 7-10 show the maneuver profile for the three components of thrust for all four spacecraft during the first maneuver. The four arcs in each figure represent the four spacecraft. Note that Figs. 7-8–7-10 include the control at the boundaries, which was calculated using the method described in Chapter 4. The overwhelming majority of the thrust is in the transverse direction (Fig. 7-9), due to the orbit transfer. Consistent with the relative geometry in Fig. 7-3, only two spacecraft have nonzero normal components of thrust (Fig. 7-10), since two spacecraft remain in the reference orbit plane. Interestingly, the small radial component of thrust in all four spacecraft (Fig. 7-8) is negative at the start

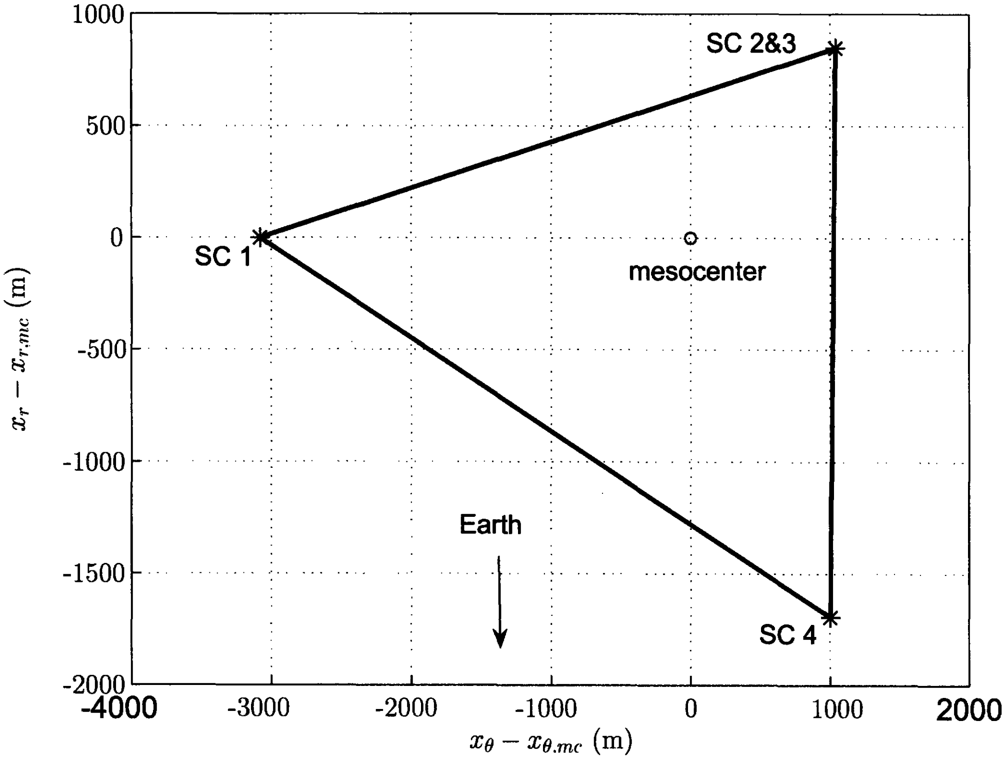


Figure 7-7: Optimal terminal tetrahedron viewed from normal to the orbital plane for the two-maneuver problem

of the maneuver and positive at the end of the maneuver. By examining Gauss's variational equations,[5] it is seen that a change in semi-major axis, a, is related to the radial acceleration, a_r by

$$\dot{a} = \frac{2a^2}{h}\left(e\sin(\nu)a_r + \frac{p}{r}a_\theta\right) \tag{7.26}$$

where ν is the true anomaly, h is the angular momentum, p is the semilatus rectum, e is the eccentricity, and r is the radial distance to the spacecraft. From Eq. (7.26) it is seen that when ν is slightly negative (i.e., just before perigee), the radial acceleration must be negative in order to increase the semi-major axis. Similarly, when ν is slightly positive (i.e., just after perigee), the radial acceleration must be positive to increase the semi-major axis. Therefore starting the maneuver in the negative radial direction before perigee aids in raising the semi-major axis of the spacecraft. Note that the crossover point for each spacecraft is not exactly at perigee, but this slight difference is most likely attributed to the need to satisfy the terminal conditions.

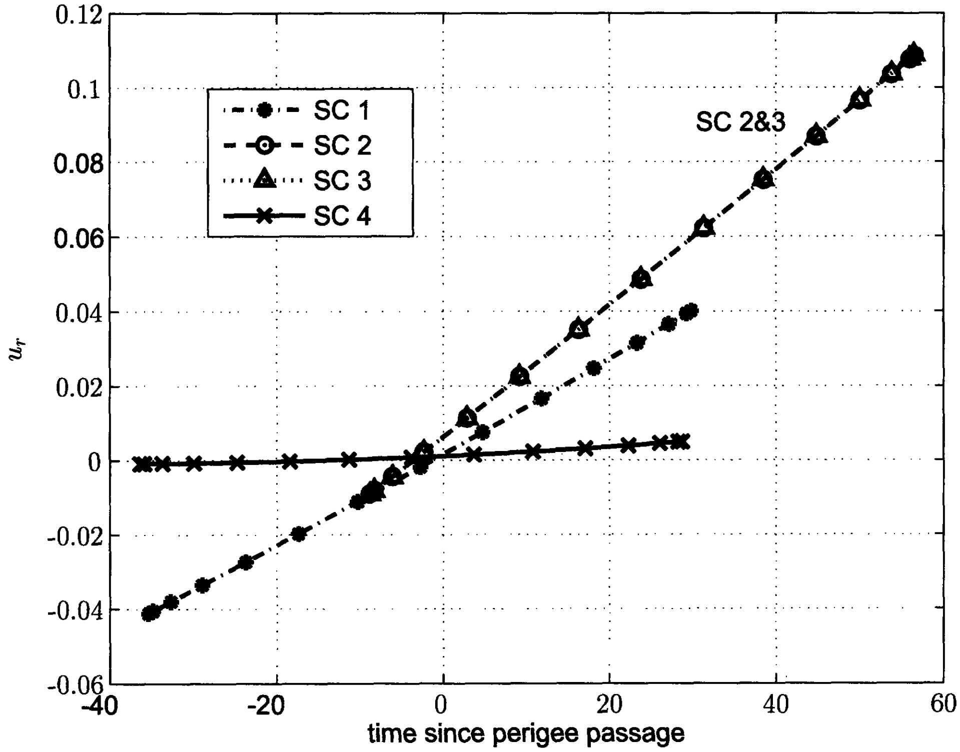


Figure 7-8: Radial control for the two-maneuver problem

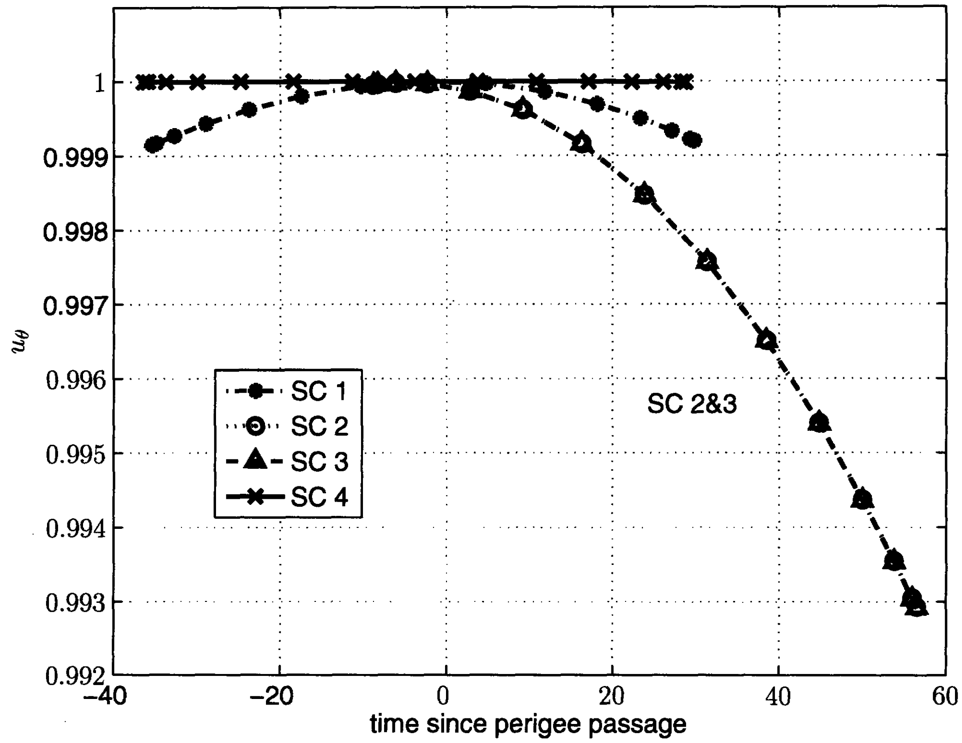


Figure 7-9: Transverse control for the two-maneuver problem

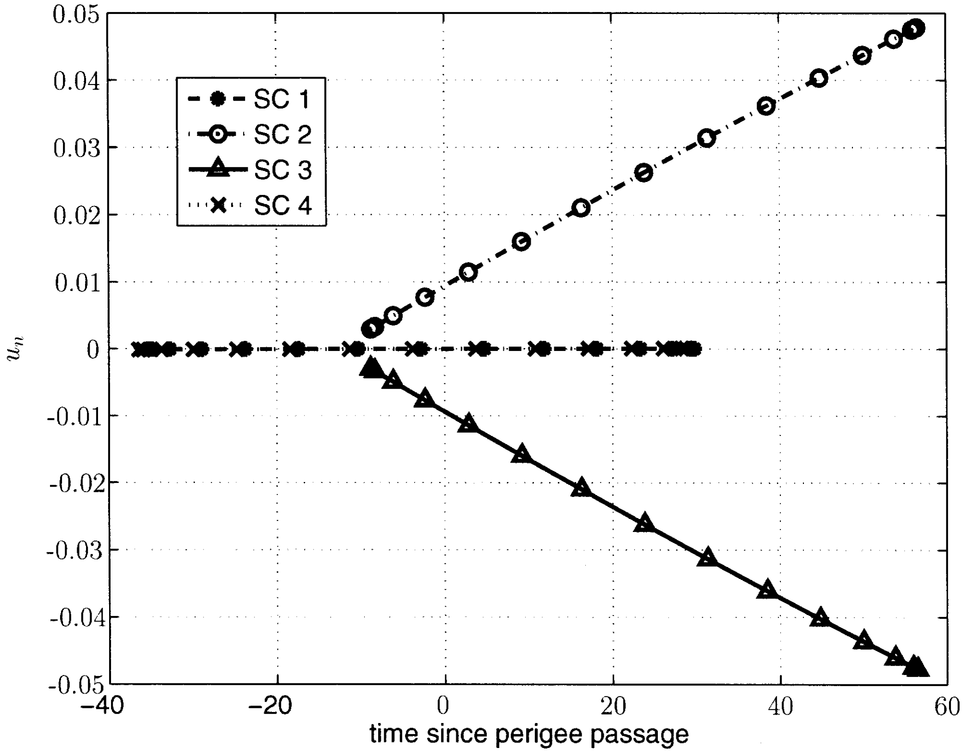


Figure 7-10: Normal control for the two-maneuver problem

7.6.2 Single-Maneuver Solution

In the two-maneuver solution, only one of the four spacecraft used both maneuver opportunities. Moreover, the spacecraft that did use two maneuvers had an extremely small second maneuver. A natural question that arises from the two-maneuver result is if the problem can be solved using only one maneuver opportunity per spacecraft. The one maneuver problem was posed by setting the duration of the second burn phase to zero.

As expected, slightly more fuel is used in the single-maneuver solution: 653.375 kg as opposed to 653.220 kg in the two-maneuver solution. Interestingly, allowing only one maneuver per spacecraft changes the orientation of the tetrahedron at the terminal time. The orientation of the tetrahedron for the one maneuver solution is seen in Figs. 7-11 and 7-12. By allowing only one maneuver per spacecraft, the single burn must simultaneously achieve two conflicting goals. On one hand, at least one spacecraft must terminate at a different altitude than the other three in order to satisfy the tetrahedral constraints. With the periodicity constraint, the only way to achieve a different altitude at the final time is by staggering the

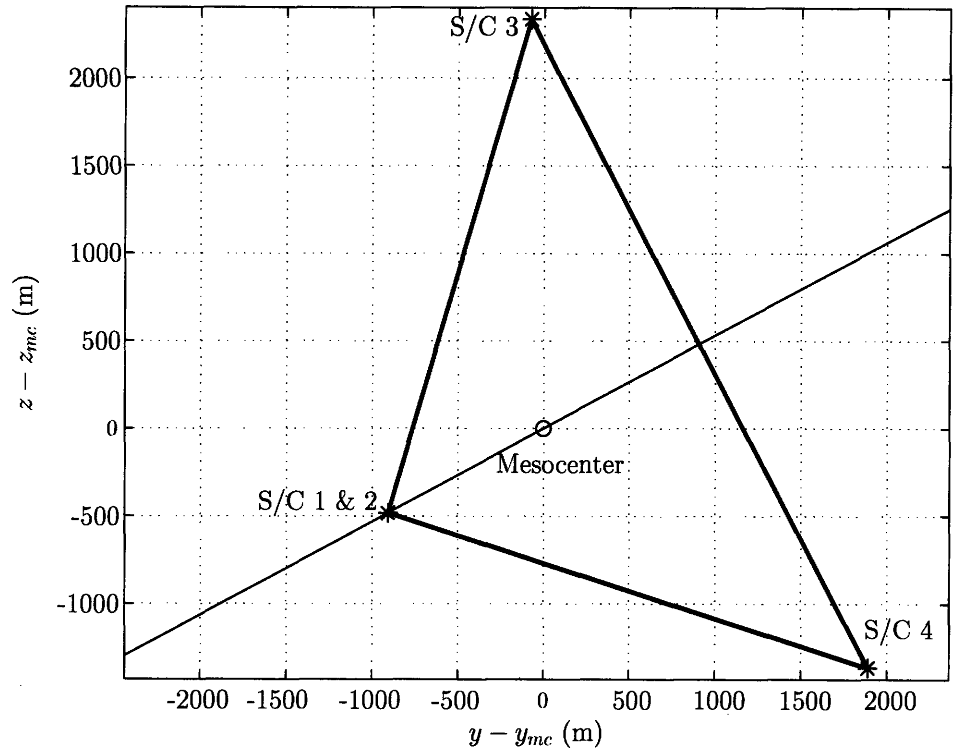


Figure 7-11: Optimal terminal tetrahedron viewed along orbital plane for the single-maneuver problem

maneuvers to occur at different times. On the other hand, each spacecraft should burn at perigee in order to efficiently complete the orbit transfer. To minimize fuel, each spacecraft should burn as close as possible to perigee, yet still be staggered enough to produce sufficient separation in the final geometry. By balancing these two goals, the resulting effect is to minimize the maximum altitude separation. This result can be seen in Fig. 7-12, where the difference between all four spacecraft and the altitude of the mesocenter is the same. Furthermore, from Fig. 7-13, it can be seen that the maneuvers must be staggered further away from perigee than in the two-maneuver solution, which is the likely cause of the larger fuel expenditure than the two-burn solution. Two of the spacecraft burn almost completely before perigee while the other two are approximately centered around perigee. From a fuel balancing perspective, the difference in expended fuel between the four spacecraft is 0.25 kg. In fact, the difference in expended fuel between the four spacecraft for the two-burn solution is also 0.25 kg, (seen in Fig. 7-6), implying that there is no significant advantage in choosing one scenario over the other with respect to fuel balancing.

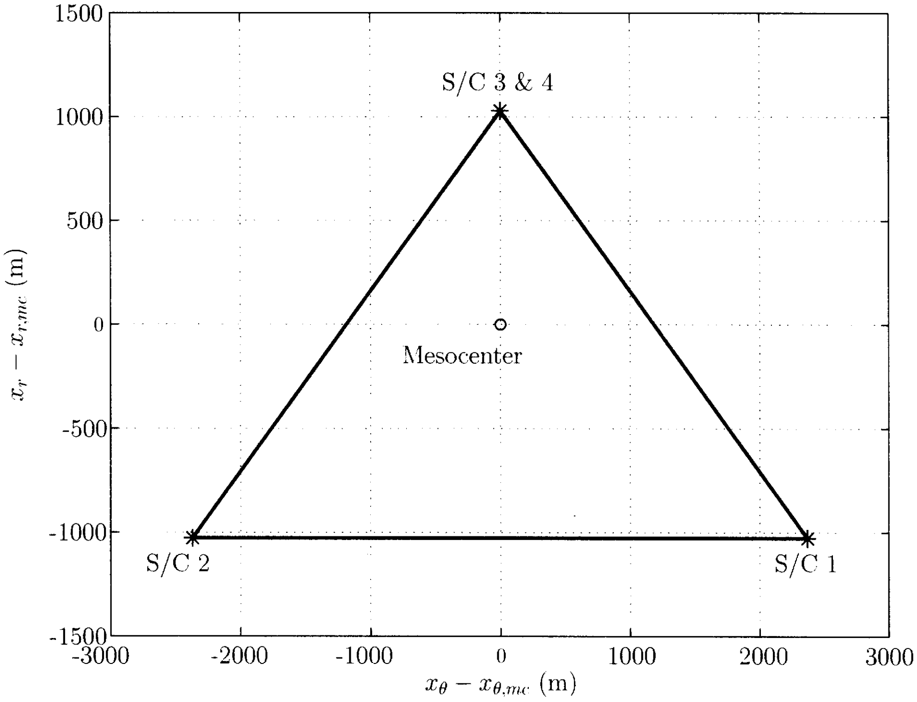


Figure 7-12: Optimal terminal tetrahedron viewed from normal to the orbital plane for the single-maneuver problem

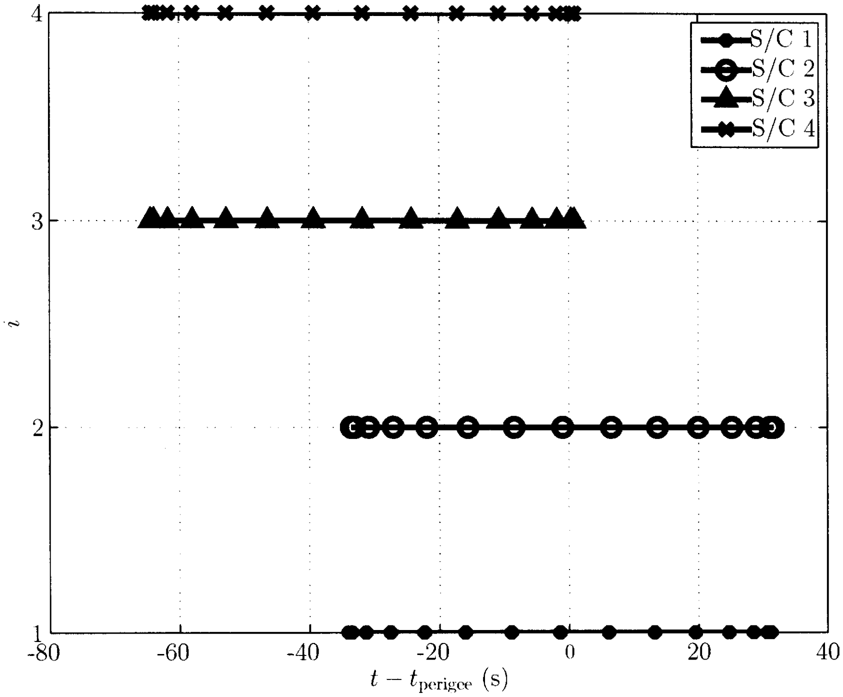


Figure 7-13: Spacecraft burn durations relative to time of reference perigee passage, $t - t_{\mathrm{perigee}}$, for spacecraft $i = 1, \ldots, 4$ during the burn phase for the single-maneuver problem

7.6.3 Analysis of Optimality for the 1-Maneuver Problem

As mentioned in Section 6.4, a primary benefit of the Gauss Pseudospectral Method is the equivalence that exists between the direct and indirect approaches.[8] By using the costate mapping shown in this thesis, one can verify that the control from the NLP matches the analytic control determined from the first-order HBVP optimality conditions at the LG points (Section 6.2). Confidence in the NLP solution is increased if the NLP solution also satisfies the HBVP necessary conditions for optimality.

This analysis was done for both the single-maneuver and two-maneuver cases, but only the results from the single-maneuver case are shown here. Figs. 7-14–7-16 show the magnitude of the differences between the NLP control and the estimated optimal control derived from the HBVP conditions for the three components of control, $|\Delta u_r|$, $|\Delta u_\theta|$, and $|\Delta u_n|$. It is seen from Figs. 7-14–7-16 that the largest magnitude difference is order 10^{-6}, verifying that the NLP control is in excellent agreement with the estimated control from the first-order optimality conditions. This error can potentially be reduced further by increasing the number of LG points or by tightening the tolerances of the NLP optimizer. Recall that there are no control variables in the NLP at the boundaries, and therefore only the control at the LG points are compared.

7.7 Overview of the Spacecraft Reconfiguration Problem

Now consider a fleet of four spacecraft in a 1.2x12 Earth radii (R_e) reference orbit with a 10 degree inclination. In this scenario, science measurements are taken in a region of the orbit, called the *region of interest* (ROI), rather than at a single point as in the orbit insertion problem. Furthermore, assume that the initial state of each spacecraft is such that the formation tetrahedron is highly degraded and is incapable of being used to take science measurements anywhere in this region of interest.[62,63] The problem is then to reconfigure the four spacecraft into a formation such that the formation satisfies a set of geometric constraints for a portion of the orbit near apogee. This ROI is defined as a portion of the orbit such that

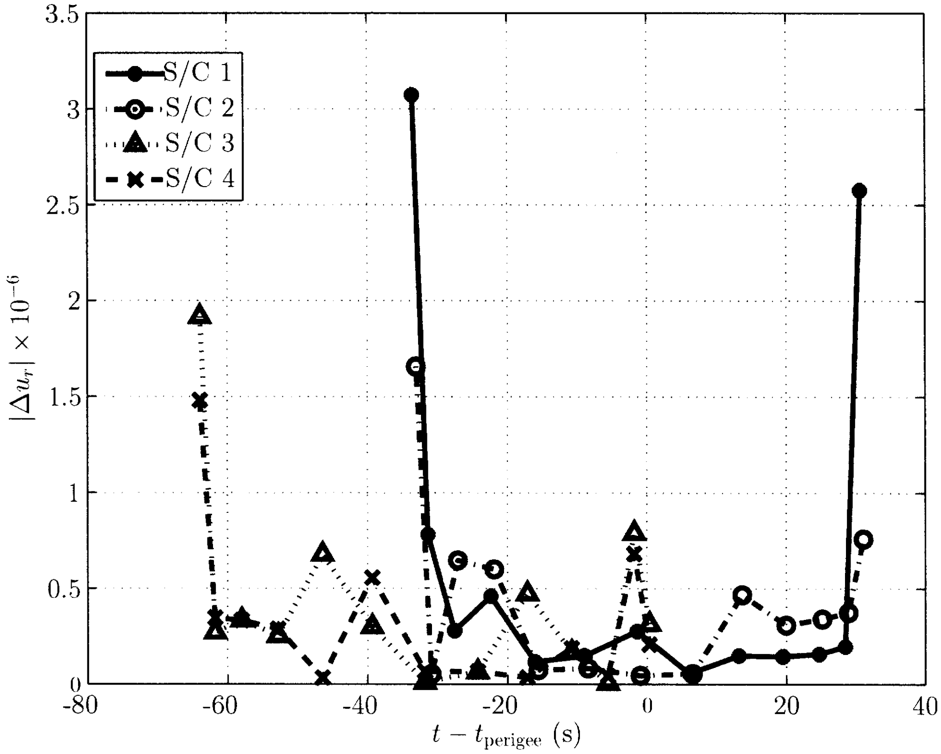


Figure 7-14: Error in radial control for the single-maneuver problem

the mesocenter (i.e., the geometric center of the formation) is located a distance greater than $9R_e$ from the center of the Earth (roughly ±20 deg in true anomaly on either side of apogee).[1] It is known that the initial (degraded) formation does not satisfy the geometric constraints in the region of interest. The problem considered in this section is to design a single-orbit, minimum-fuel trajectory and control profile that reconfigures the spacecraft to satisfy both the required geometric constraints in the region of interest. Furthermore, in order to reduce interference with the scientific measurements, it is desired that the spacecraft perform no maneuvers for at least three weeks after the final reconfiguration maneuver. Consequently, the tetrahedron must be able to maintain the specified quality in the region of interest for at least three weeks without additional maneuvers. To assess the quality of the tetrahedron well beyond the final reconfiguration maneuver, the spacecraft are propagated from the terminal state, $\mathbf{x}(t_f)$, of the single orbit minimum-fuel trajectory for a period of three weeks under the influence of several perturbations. The resulting ephemeris of each spacecraft is then

[1] The region of interest represents the plasma sheet region of the magnetosphere, in which scientific measurements are taken.

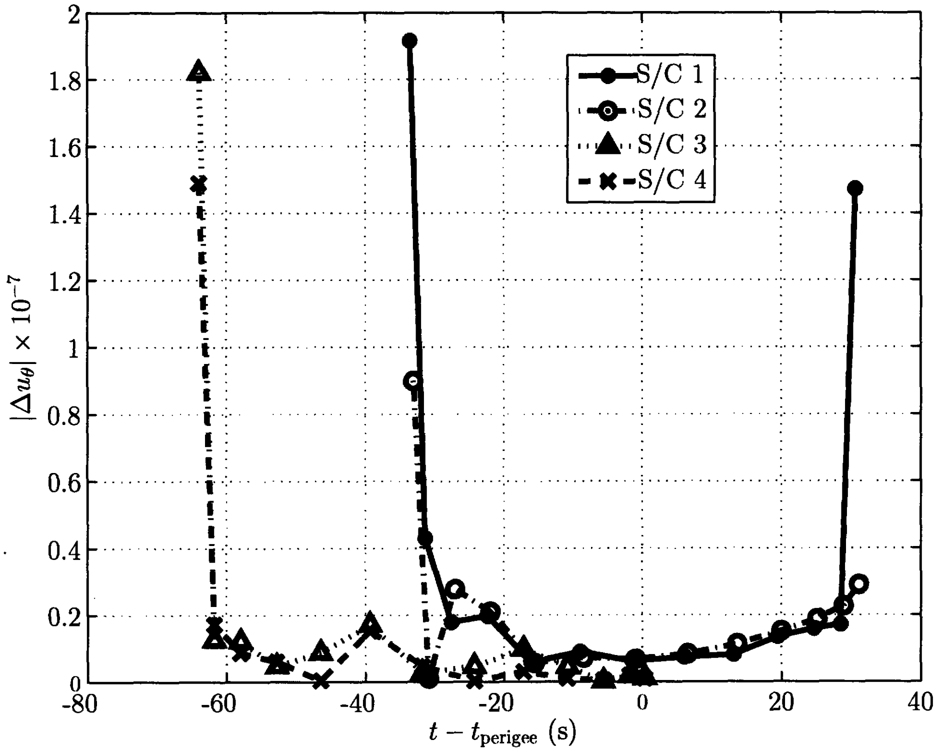


Figure 7-15: Error in transverse control for the single-maneuver problem

used to analyze the tetrahedron during this three-week period to see if the region of interest geometric constraints have been violated. This section addresses and compares minimum-fuel trajectories that both violate and adhere to this constraint.

Trajectory Event Sequence

The trajectory event sequence for this problem is given as follows. First, the trajectory is divided into seven phases with the following ordered event sequence:

[Burn Coast Burn Coast Burn Coast ROI]

During the coast phases the thrust is zero, while during the burn phase the thrust is constant at a maximum value T_{max}. Furthermore, at each phase interface it is assumed that the trajectories and mass of each spacecraft are continuous, but that the control is discontinuous. It is important to note that, with the exception of the initial time of each spacecraft in the first phase, the initial and terminal times of *each spacecraft* and *each phase* are free. Also,

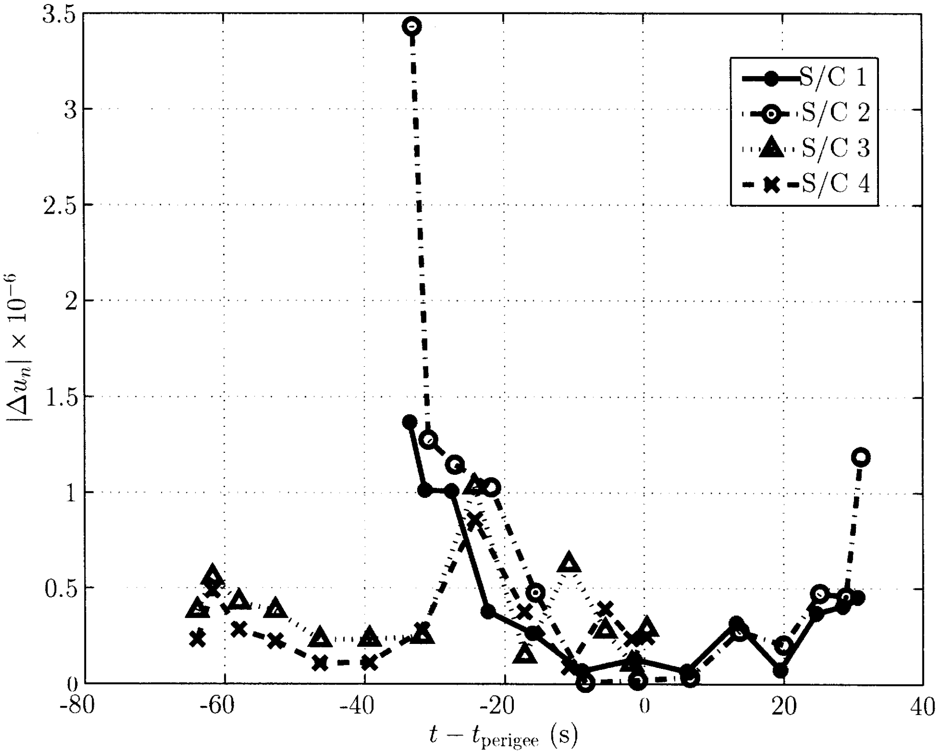


Figure 7-16: Error in normal control for the single-maneuver problem

while no maneuvers are allowed in the region of interest phase (and thus the region of interest phase is a coast phase), several constraints on the formation are applied only in the region of interest. Therefore, the region of interest is specified separately from the other coast phases in the problem. Fig. 7-17 shows a schematic of the reconfiguration problem.

7.8 Spacecraft Model and Equations of Motion

7.8.1 Spacecraft Model

In this application four identical spacecraft are considered, each with a dry mass of 150 kg and a fuel mass of 200 kg. Furthermore, the maximum thrust level of the engine is 0.22 N with an engine specific impulse of 110 s. These physical parameters are typical of a standard monopropellant hydrazine orbit maintenance thruster.[123] Consequently, all thrusting maneuvers are assumed to be non-impulsive. Lastly, there are a few parameters that are not used in the single-orbit optimal control problem, but are used when propagating the

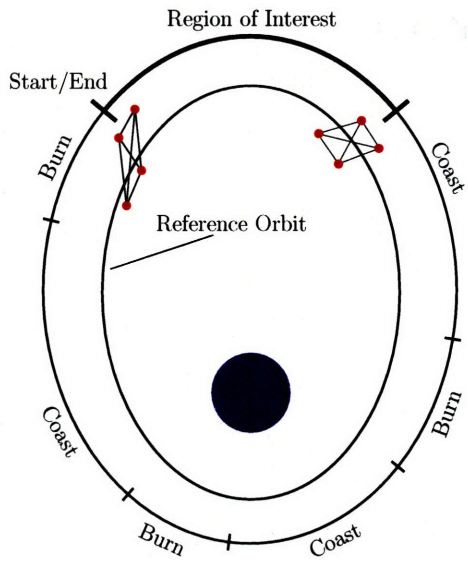


Figure 7-17: Schematic of the event sequence for the reconfiguration problem

spacecraft beyond t_f to analyze the long term effects of the tetrahedral formation under certain perturbations. The drag coefficient, C_d, is 2.2 and the drag area, A_d, is 1 m^2 for all spacecraft.

7.8.2 Equations of Motion

The equations of motion for the reconfiguration problem are exactly the same as those defined for the orbit insertion problem. Consequently, the dynamic model during thrust phases is defined in Section 7.2.2, the dynamic model during coast phases is defined in Section 7.2.3, and the dynamic model for the region of interest is defined in Section 7.2.4.

7.9 Constraints

This section describes the many constraints imposed in the reconfiguration problem. Some constraints are a direct result of the posed problem, while others are specific to the application of the GPM to this problem.

7.9.1 Initial Conditions

As mentioned earlier, the four spacecraft begin in a significantly degraded tetrahedron, shortly after having departed from the region of interest. Table 7.9.1 shows the initial conditions for each spacecraft in orbital elements. To be clear, a is the semi-major axis in m, e is the eccentricity, i is the inclination, Ω is the longitude of the ascending node, ω is the argument of perigee, and ν is the true anomaly. All angle units are in radians. Figure 7-18 depicts this initial tetrahedron, which clearly shows that this configuration is not the desired regular tetrahedron.

Table 7.2: Spacecraft initial conditions

SC num	a_0	e_0	i_0	Ω_0	ω_0	ν_0
1	4.2255510e7	.8025552	.1753710	6.259796	1.6225498	3.5779493
2	4.2255523e7	.8024699	.1753933	6.259694	1.6227984	3.5779235
3	4.2259621e7	.8024827	.1754062	6.260461	1.6217395	3.5781680
4	4.2259623e7	.8254586	.1753016	6.260448	1.6218584	3.5779821

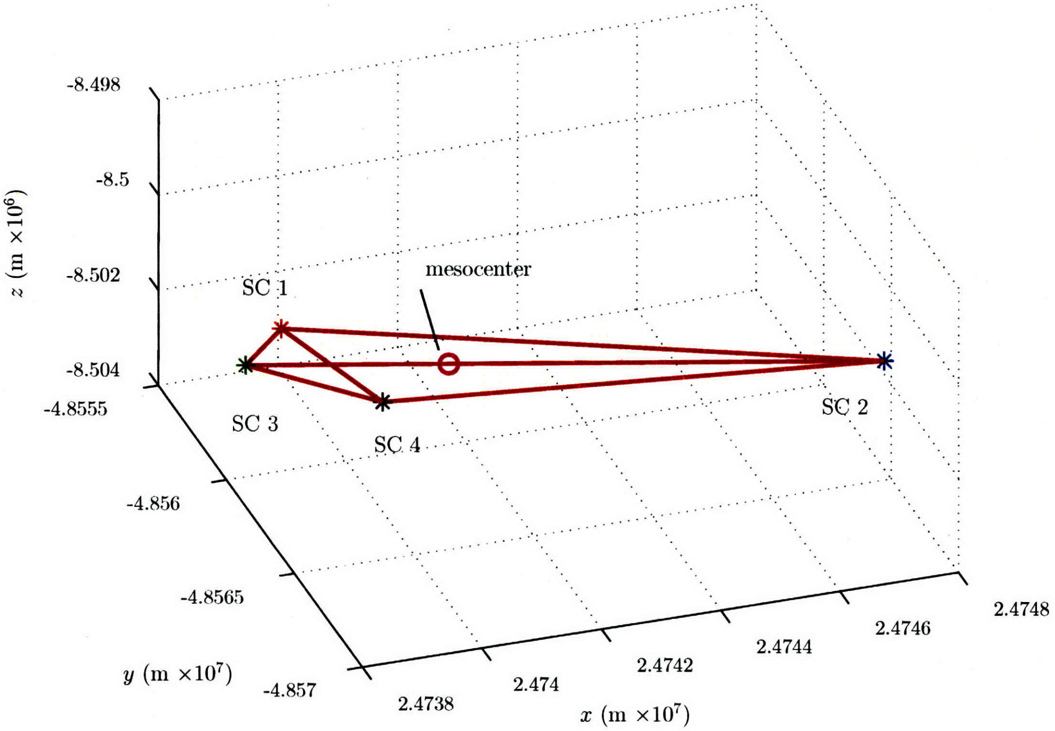


Figure 7-18: Initial tetrahedral configuration

The orbital elements in Table 7.9.1 are converted to modified equinoctial elements using a transformation T_{o2e} (see for example Ref. 5) to obtain the initial state in modified equinoctial elements. Furthermore, the initial mass of each spacecraft is set to its maximum value, i.e.,

$$m^{(i)}(t_0) = m_{max}, \qquad i = 1, \ldots, 4 \tag{7.27}$$

where for this problem, $m_{max} = 350$ kg.

7.9.2 Interior Point Constraints

The interior point constraints are exactly the same as those defined for the orbit insertion problem in Section 7.3.2, where the number of phases, $P = 7$.

7.9.3 Trajectory Path Constraints during Thrust Phases

The path constraints are exactly the same as those defined for the orbit insertion problem in Section 7.3.3.

7.9.4 Constraints in the Region of Interest Phase

The final phase of the trajectory spans the region of interest. In the region of interest, several geometric constraints are placed on the formation. As noted in Section 7.7, this phase is a coast phase. Consequently, the thrust is zero in the region of interest. Next, let the spacecraft *mesocenter* (i.e., the mean position) be defined as

$$\bar{\mathbf{r}} = \frac{1}{4}\sum_{i=1}^{4} \mathbf{r}^{(i)} \tag{7.28}$$

where $\mathbf{r}^{(i)}$, $i = 1, \ldots, 4$ are the positions of the four spacecraft as measured from the center of the Earth. At the start of the region of interest phase, the mesocenter must attain a particular position on the reference orbit, $\mathbf{r}_{ROI}$, where $\|\mathbf{r}_{ROI}\| = 9R_e$ and is located at a point prior to apogee. Consequently, a constraint on the position of the mesocenter at the start of the region of interest is given as

$$\bar{\mathbf{r}}(t_0^{(P)}) = \mathbf{r}_{ROI} \tag{7.29}$$

There is a second constraint enforced at the entrance of the region of interest. In the absence of perturbations, a spacecraft's orbit period is solely a function of semi-major axis. Consequently, a formation in which all spacecraft have equal semi-major axes will be periodic in its motion. When perturbations are introduced, there will be some degradation to the tetrahedron over time, but in general the degradation rate is reduced if the orbit periods are equal at the start. Therefore, in order for the formation to maintain its quality in the ROI for a period of time into the future (i.e., beyond t_f), it is beneficial to require that the semi-major axes of each spacecraft at the start of the region of interest. Applying this condition, a second constraint on the formation in the region of interest is given as

$$a^{(1)}(t_0^{(P)}) = a^{(2)}(t_0^{(P)}) = a^{(3)}(t_0^{(P)}) = a^{(4)}(t_0^{(P)}) \tag{7.30}$$

Now, the region of interest terminates when the position of the mesocenter has again attained a magnitude of $9R_e$. Consequently, a constraint at the terminus of the region of interest is given as

$$||\bar{\mathbf{r}}(t_f^{(P)})|| = 9R_e \tag{7.31}$$

Next, several path constraints are enforced during the region of interest. First, as the formation evolves in the region of interest, it must maintain a certain geometry in order to be able to take useful scientific measurements. In particular, the two aspects to the formation geometry that are important in order to fulfill the science requirements are the *shape* and *size* of the formation. The metric used to determine the quality of the shape of the formation is the Glassmeier metric (Q_{gm}).[88] The Glassmeier metric is given as

$$Q_{gm} = \frac{V_a}{V^*} + \frac{SA_a}{SA^*} + 1 \tag{7.32}$$

where V_a is the actual volume of the tetrahedron, SA_a is the actual surface area of the tetrahedron, and V^* and SA^* are the volume and surface area, respectively, of a regular tetrahedron whose average side length is $\bar{L}$. It is noted that the largest possible value of each ratio in Eq. (7.32) is unity. Consequently, the maximum value of Q_{gm} is 3. It is seen that the Glassmeier quality metric is a function of both volume and surface area of the tetrahedron. However, because of the functional form of Q_{gm}, the Glassmeier metric has no sensitivity to the size of the tetrahedron. The ideal geometry for this mission is a regular tetrahedron with 10 km spacing between each satellite, but science investigators note that there is considerable flexibility here.[62,63] In particular, values of Q_{gm} greater than 2.7 are acceptable in the region of interest. Consequently, the following path constraint on Q_{gm} is enforced in the region of interest

$$Q_{gm} \geq 2.7 \tag{7.33}$$

See Appendix for the computation of the Glassmeier quality factor.

Because the Glassmeier metric is insensitive to the size of the tetrahedron, an additional constraint is required to constrain the size of the tetrahedron. Although an average inter-spacecraft separation of 10 km is considered ideal, acceptable science return is still possible for average separations ranging from 4 km to 18 km.[62,63] Consequently, the following path constraint is enforced during the region of interest that keeps the average side length, $\bar{L}$, bounded between its acceptable lower and upper limits:

$$4 \leq \bar{L} \leq 18. \tag{7.34}$$

Next, in order to evaluate the aforementioned path constraints in the region of interest at the same time for all four spacecraft, it is necessary that both the initial and terminal time for all four spacecraft be the same. Constraints on the initial and terminal time in the region of interest are then given as

$$\begin{aligned} t_0^{(1)^{(P)}} &= t_0^{(2)^{(P)}} = t_0^{(3)^{(P)}} = t_0^{(4)^{(P)}} \\ t_f^{(1)^{(P)}} &= t_f^{(2)^{(P)}} = t_f^{(3)^{(P)}} = t_f^{(4)^{(P)}} \end{aligned} \tag{7.35}$$

It is noted that t_0 and t_f in these equations are *free.*

7.10 Tetrahedral Reconfiguration Optimal Control Problem

The optimal control tetrahedral reconfiguration problem is now stated formally. Using the aforementioned trajectory event sequence, determine the thrust profile that maximizes the sum of the terminal masses of each spacecraft, i.e., maximize the objective functional

$$J = \sum_{i=1}^{4} m^{(i)}(t_f^{(P)}) \tag{7.36}$$

subject to the dynamic constraints of Eq. (7.1), (7.3), and (7.7), the initial constraints of Table 7.9.1 and Eq. (7.27), the interior point constraints of Eq. (7.12) and (7.13), the path constraints of Eq. (7.14) and (7.15), the terminal constraints of Eq. (7.29), (7.30), (7.31), (7.35), (7.32), and (7.34).

7.11 Numerical Solution via Gauss Pseudospectral Method

The spacecraft configuration problem as described in Section 7.7 - Section 7.10 is solved using the aforementioned Gauss pseudospectral method.[6] The optimization was carried out with the MATLAB mex interface of the NLP solver SNOPT[46] using analytic first-order derivatives for both the constraint Jacobian and the gradient of the objective function. Furthermore, the optimal control problem was scaled canonically from SI units to a non-dimensional set of units such that the length scale is the radius of the Earth, the time scale is one Schuler period, and the mass scale is equal to the initial spacecraft mass. For this numerical reconfiguration problem, 10 nodes (i.e., 8 Gauss points) were used for the burn phases, and 50 nodes (i.e., 48 Gauss points) were used in the region of interest coast phase. The coast phases (other than the region of interest phase) were not discretized, but were computed via numerical quadrature of Eq. (7.7). Consequently, each coast phase had two nodes representing the initial and final points of the phase.

Ideally, science investigators would like the formation geometry constraints to hold for a period of three weeks. Yet, the problem described in the previous sections has a duration of only one orbit, or roughly one day. Because of this, any constraints on the formation after this first pass through the ROI are not explicitly coded into the problem. However, the solution to the single-orbit problem was propagated for three weeks using an orbit propagation tool called FreeFlyer® to determine the effects of perturbations on the formation. These results are presented in the following section. The problem was then solved with a tighter requirement on the tetrahedral formation, which, when propagated for three weeks, met the tetrahedral requirements for the entire duration. The results of this case are presented in the following section as well.

7.12 Results

7.12.1 Minimum-Fuel Solution for $Q_{\min} = 2.7$

A minimum-fuel solution to the problem posed in Section 7.10 with $Q_{\min} = 2.7$ is presented here. The total amount of fuel burned is $\sum_{i=1}^{4} m_f^i = 0.1289$ kg. The propellant used by each spacecraft is shown in Fig. 7-19. Furthermore, Table 7.3 describes in detail the phase durations for each spacecraft. Although three maneuver opportunities were included in the trajectory design, it is seen in the optimal trajectory that at least one maneuver was of zero duration for each spacecraft. In fact, as seen in Table 7.3, only spacecraft #4 used more than one of the three provided maneuver opportunities. In addition, all of the thrust maneuvers were relatively short in duration (the longest burn being roughly six minutes in duration). It is also interesting to note that spacecraft #1 burned over half of the total fuel expended and that this maneuver resulted in a significant change in the position of spacecraft #1 relative to to the other three spacecraft. By examining the tetrahedral formation at the initial time (Fig. 7-20) and the tetrahedral formation at the terminal time (Fig. 7-21), it can be seen that the optimal solution essentially re-positioned spacecraft #1 according to the arrow in Fig. 7-20, with only minor changes in the other three spacecraft. It is important to note that this optimal trajectory design is unique to the initial conditions given, meaning different initial conditions could result in very different maneuver profiles.

Fig. 7-22 shows the evolution of the Glassmeier quality factor,Q_{gm}, in the region of

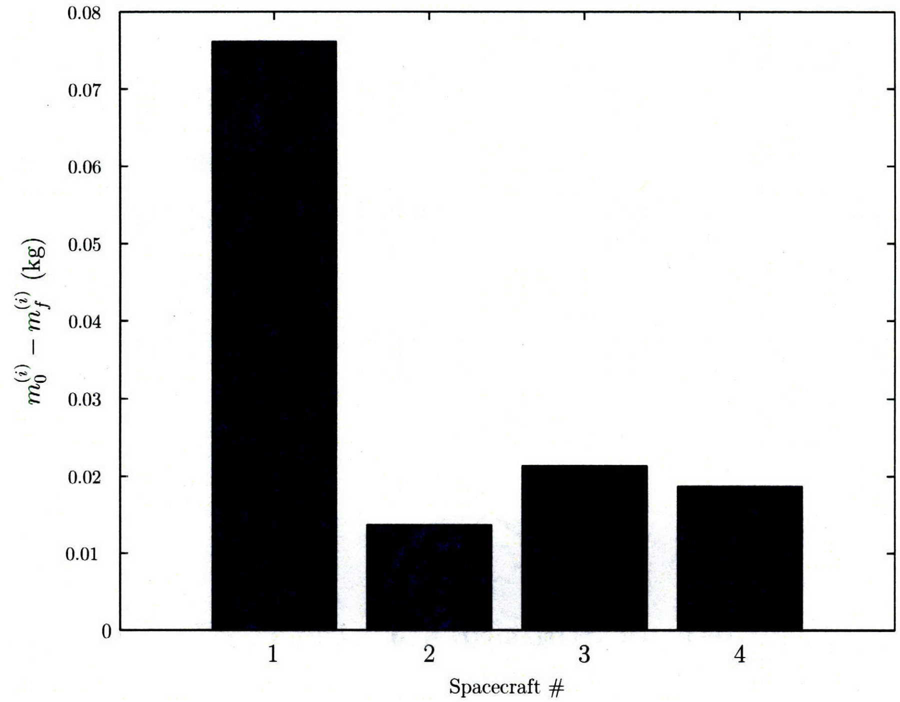


Figure 7-19: Fuel expenditure of each spacecraft for $Q_{\min} = 2.7$.

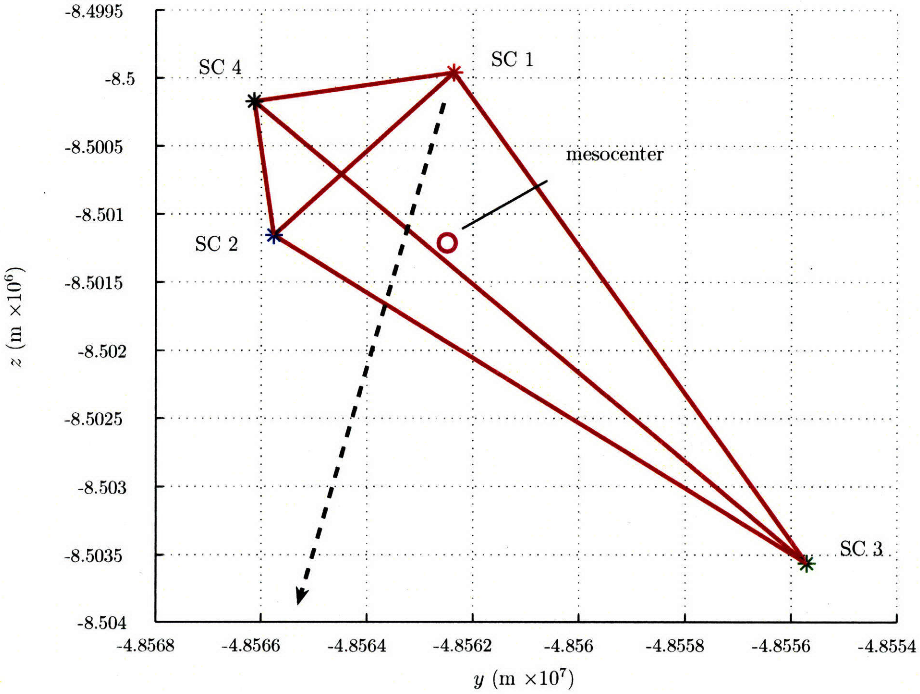


Figure 7-20: Projection of initial tetrahedron into ECI YZ-Plane for $Q_{\min} = 2.7$

Table 7.3: Phase durations (in minutes) for $Q_{\min} = 2.7$

SC num	Burn1	Coast1	Burn2	Coast2	Burn3	Coast3	Coast ROI
1	0	158.483	0	355.017	6.224	61.533	837.300
2	1.118	226.667	0	164.500	0	188.967	837.300
3	0	157.483	0	84.083	1.746	337.950	837.300
4	0.383	272.717	1.143	75.000	0	232.017	837.300

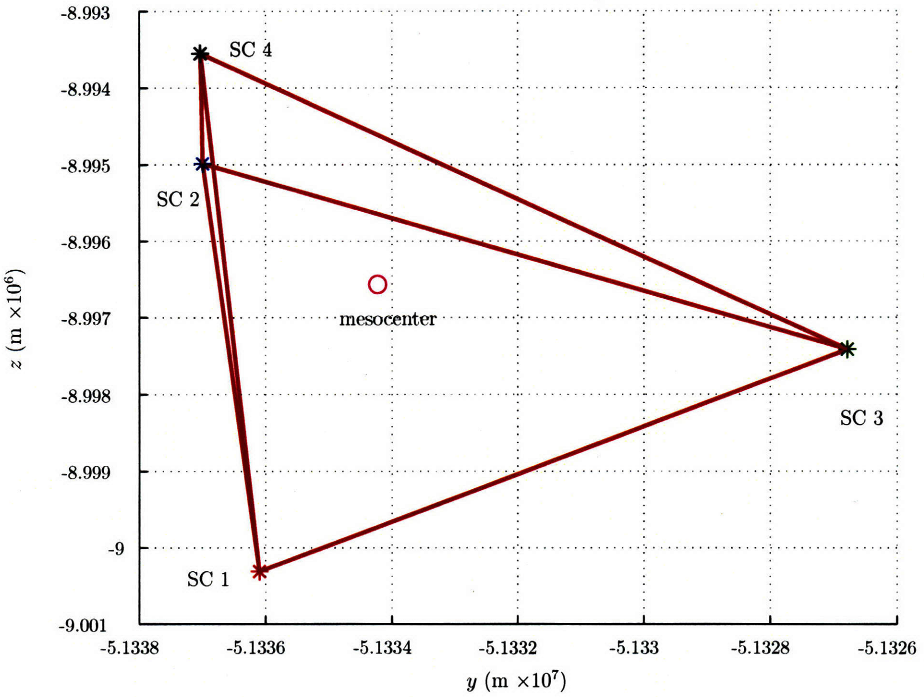


Figure 7-21: Projection of terminal tetrahedron into ECI YZ-Plane for $Q_{\min} = 2.7$

interest. Note that the constraint on the quality factor becomes active twice during the region of interest, once at the terminal point and once in the middle of the region of interest. Interestingly, as seen in Figs. 7-22 and 7-23, the instant of time near the middle of the region of interest where the quality factor attains is lower bound occurs when the average length has simultaneously attained its minimum value. When the spacecraft are in close proximity, small changes in relative position can lead to large changes in the relative geometry. As a result, the sensitivity of the shape of the tetrahedron to changes in average length is highest when the spacecraft are more closely spaced. In particular, this sensitivity can result in a rapid degradation of the quality factor. Fig. 7-24 supports this by showing that the sensitivity of the quality factor with respect to the average length, $\partial Q_{gm}/\partial \bar{L}$, is indeed largest at the

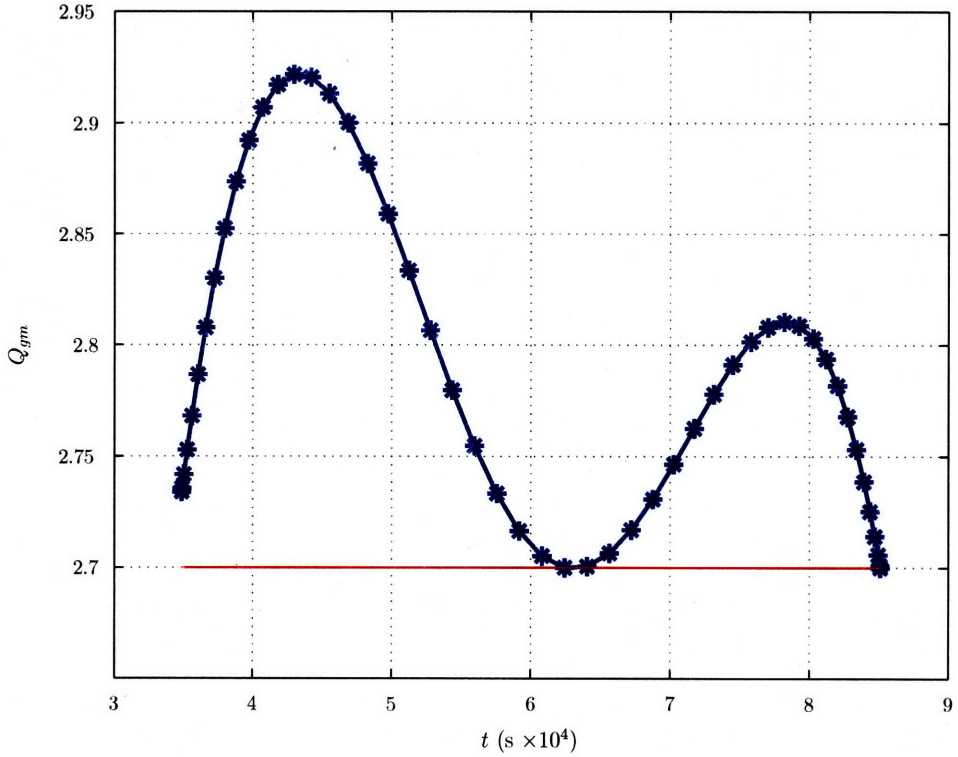


Figure 7-22: Quality factor in region of interest for $Q_{\min} = 2.7$

time that $\bar{L}$ is at a minimum.

Finally, it is observed from Fig. 7-23 that the average length always remains between its upper and lower bound in the region of interest. A possible explanation why the average length never attains its lower bound is because of the shape of the tetrahedron is highly sensitive to changes in $\bar{L}$ when the average length is small. As a result, it may be difficult to maintain the quality factor if $\bar{L}$ is too small. A possible explanation as to why the average length never attains its upper bound is that an exceedingly large value of $\bar{L}$ may result in an increase in fuel consumption (since the initial tetrahedron, while deformed, has an acceptable average side length only of 9.054 km). As a result, a tetrahedron with a large average length would not be fuel-optimal.

7.12.2 Ensuring Future Satisfaction of Quality Factor Constraint

It is important to note that the fuel-optimal solution described in Section 7.12.1 does not consider any potential degradation in the quality factor beyond a single passage though the

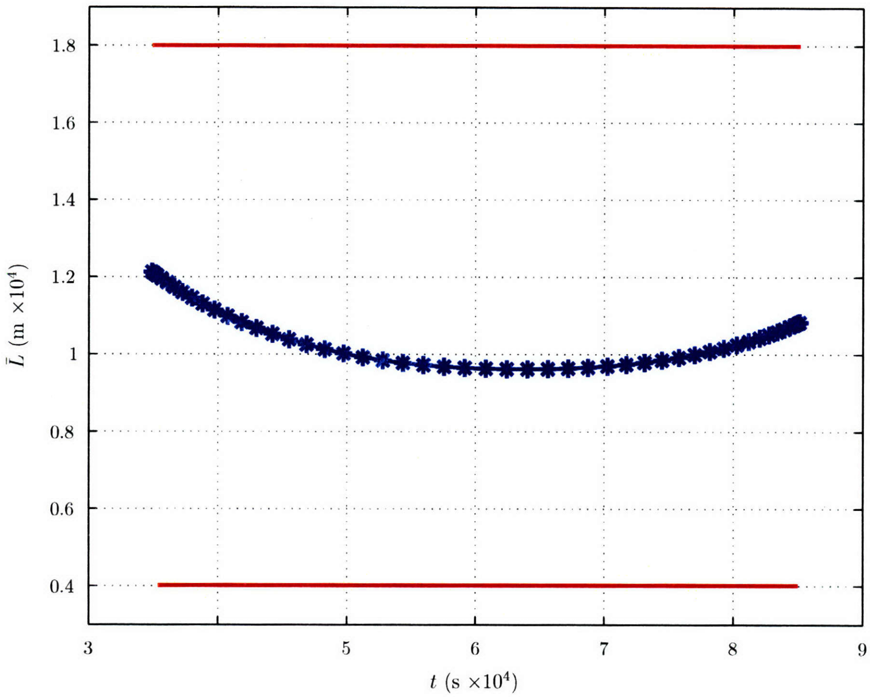


Figure 7-23: Average length, $\bar{L}$, in region of interest for $Q_{\min} = 2.7$

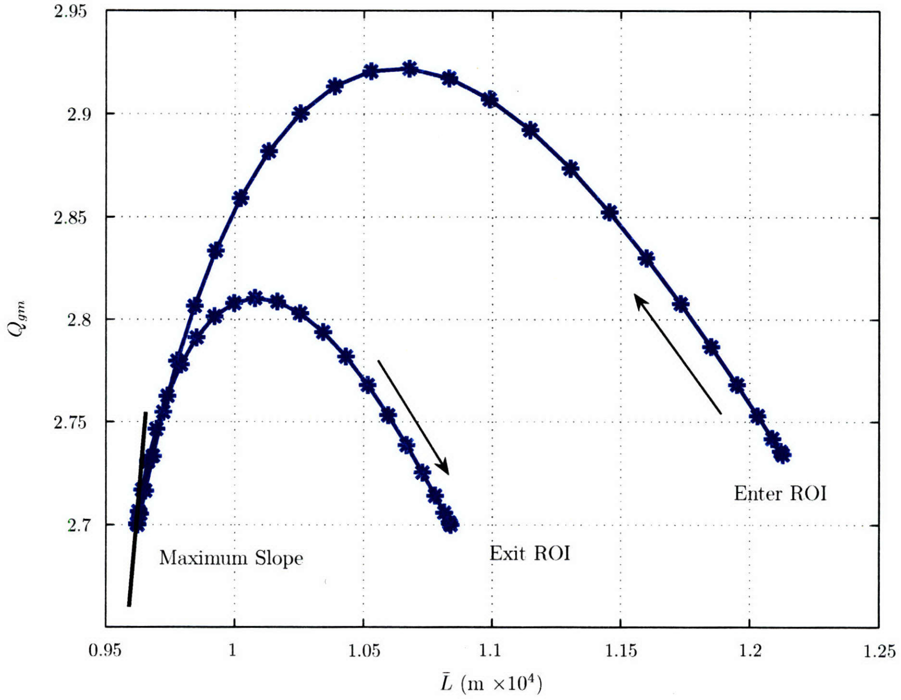


Figure 7-24: Quality factor, Q_{gm}, as a function of average length, $\bar{L}$, in region of interest for $Q_{\min} = 2.7$. Note: slope of Q_{gm} attains a maximum when $\bar{L}$ attains a minimum.

region of interest. However, if no more maneuvers were performed after the spacecraft have exited the region of interest, it is clear that perturbations (e.g., oblateness, solar, lunar, and drag) could potentially degrade the quality of the tetrahedron. As a result, at a future time, the quality of the formation would be unacceptable.

As an example of potentially unacceptable degradation in the quality of the formation, suppose that, upon the first exit from the region of interest, it is required that *no maneuvers* be performed for a period of three weeks in order to take meausurements without interruption. Naturally, the tetrahedron must be able to maintain a quality factor of greater than 2.7 during every passage through the region of interest for this three-week period. A question that arises then is whether an optimal trajectory (such as that obtained in Section 7.12.1) meets these requirements. Fig. 7-25 shows the trajectory obtained when the final state of the optimal trajectory of Section 7.12.1 is propagated forward over a period of three weeks.[2] It is seen from Fig. 7-25 that the quality of the tetrahedron begins to degrade in the region of interest on subsequent revolutions. More specifically, it is seen in Fig. 7-25 that, within 20 revolutions (1 revolution $\approx$ 1 day), the quality of the tetrahedral formation no longer satisfies the requirement on the quality factor in the region of interest. However, it is noted that the average side length between the spacecraft changes very little (Fig. 7-26) over this three-week period, due in large part to the constraint of Eq. 7.30 on the semi-major axis.

Ideally, the way to ensure that the quality factor constraint is met for a specified time into the future would be to pose an optimal control problem that includes a sufficient number of revolutions of the orbit and enforces the quality factor constraint during the region of interest in every revolution. However, this is computationally cumbersome since the size of the optimal control problem would increase substantially. An alternative to solving such a complex optimal control problem is to include some margin in the quality factor constraint for the single revolution optimal control problem (i.e., the problem considered in this section). In particular, it may be useful to set the lower bound on Q_{gm} to a value that is higher than the desired value of 2.7 and see if the tetrahedron is of acceptable quality for three-weeks into the future.

[2] The three-week propagation was performed using the orbit propagation tool *FreeFlyer* using oblateness, solar, lunar, and drag perturbations.

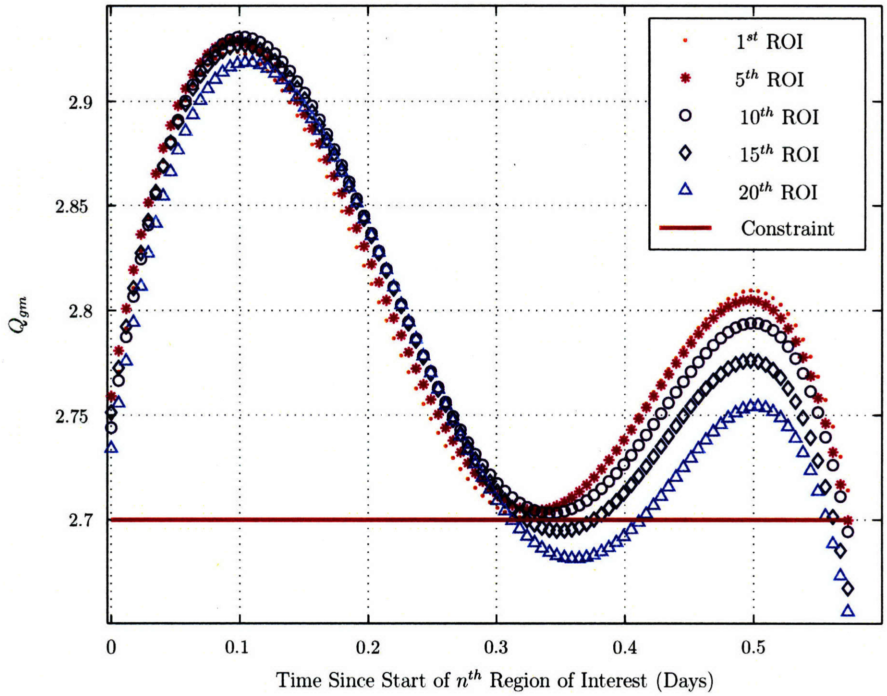


Figure 7-25: Evolution of tetrahedral quality (Q_{gm}) over three-week period beyond t_f in region of interest for $Q_{\min} = 2.7$

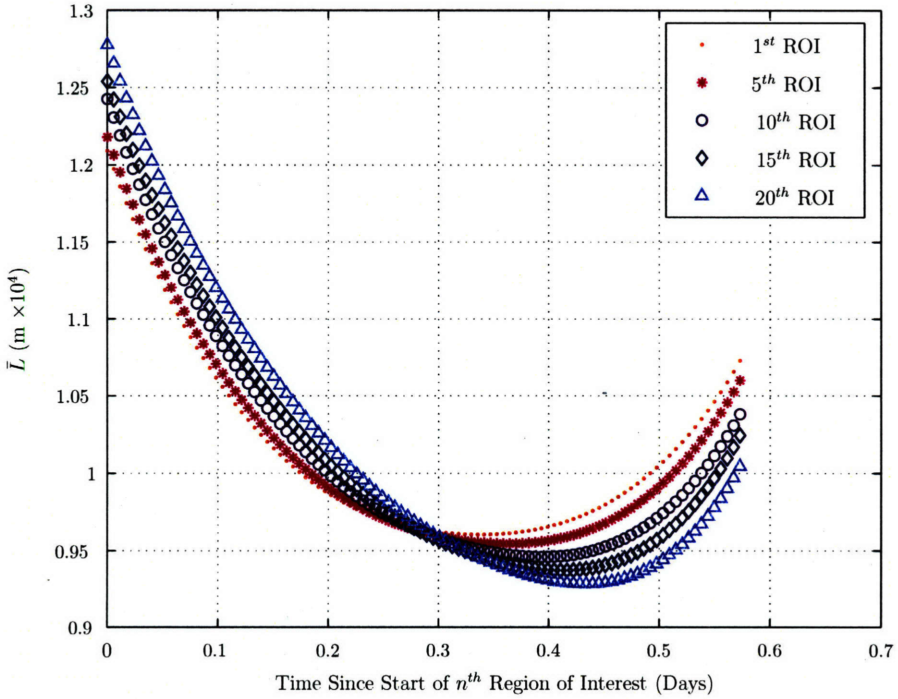


Figure 7-26: Evolution of average length, $\bar{L}$, over three-week period after t_f in region of interest for $Q_{\min} = 2.7$

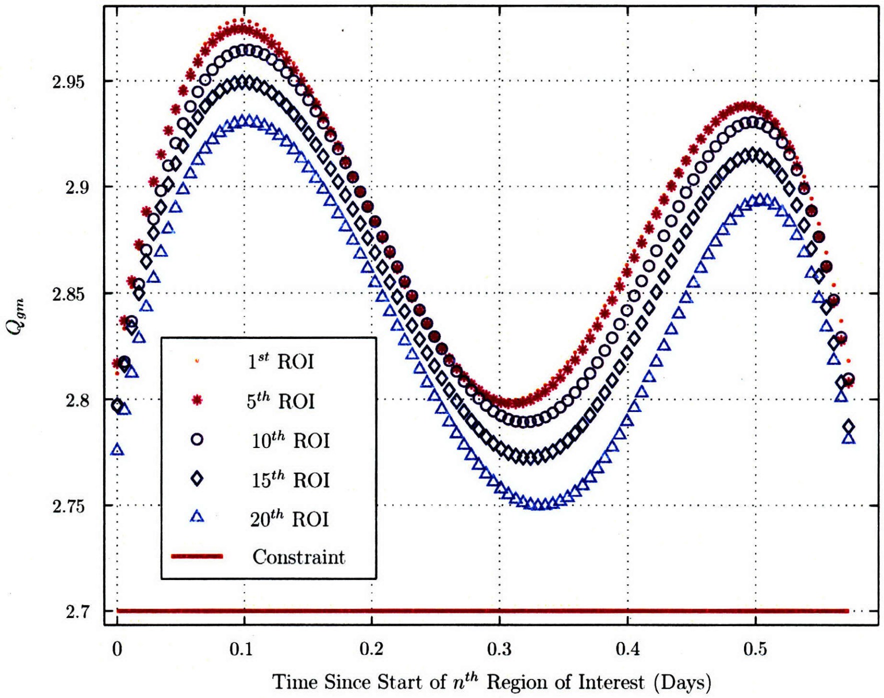


Figure 7-27: Evolution of quality factor, Q_{gm}, over three-week period after t_f in region of interest for $Q_{\min} = 2.8$

In order to demonstrate the effectiveness of providing margin in the quality factor, the original optimal control problem was modified so that the lower limit on Q_{gm} in the first region of interest was 2.8 instead of 2.7. The final state of this modified problem was then propagated for three weeks. The results of this modified problem are shown in Fig. 7-27. Although the quality factor also degrades (from 2.8) over the three-week period, this time the value of Q_{gm} remains above 2.7 for the entire three week period. However, because of the tighter constraint on Q_{gm}, the amount of fuel burned is significantly greater than the $Q_{min} = 2.7$ case. In particular, the total fuel burned in the solution to the modified problem was 0.1674 kg (an increase roughly 29% over the 2.7 case). The propellant used by each spacecraft is shown in Fig. 7-28. In the optimal solution of the modified problem, all four spacecraft utilize two maneuver opportunities, as opposed to the 2.7 case where only one spacecraft burned twice. The phase durations (and consequently the maneuver durations) are shown in Table 7.12.2. In terms of the terminal tetrahedral formation, the geometry is similar, with some small changes in the relative positions. Fig. 7-29 directly compares the geometries of the solutions obtained from the 2.7 and 2.8 cases.

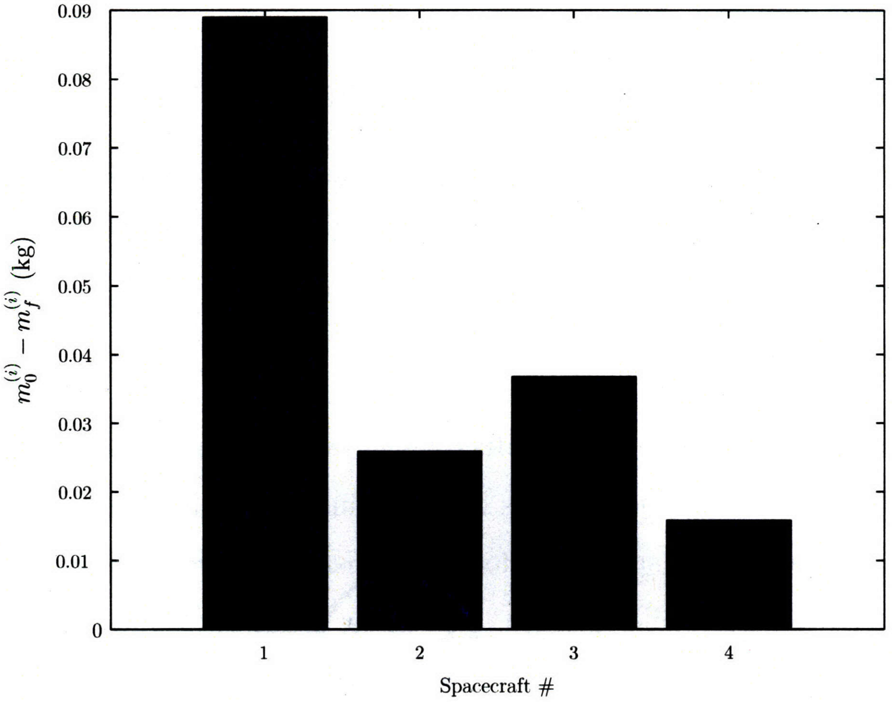


Figure 7-28: Fuel expenditure of each spacecraft for $Q_{\min} = 2.8$

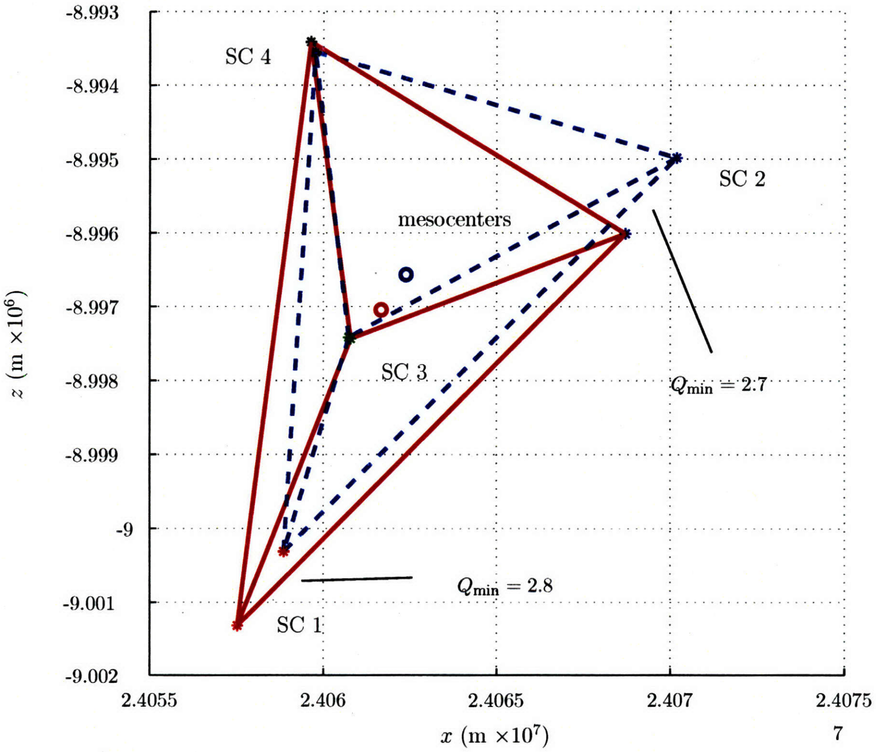


Figure 7-29: Comparison of projection of terminal tetrahedra into ECI XZ-Plane for $Q_{\min} = 2.7$ and $Q_{\min} = 2.8$

Table 7.4: Phase durations (in minutes) for $Q_{\text{min}} = 2.8$ case

SC num	Burn1	Coast1	Burn2	Coast2	Burn3	Coast3	Coast ROI
1	0	149.150	0.065	424.850	7.209	0	837.300
2	0	184.517	1.008	241.767	1.107	152.867	837.300
3	1.448	155.767	0	88.200	1.556	334.300	837.300
4	0	273.000	1.268	87.35	0.024	219.617	837.300

7.13 Summary

The problem of determining minimum-fuel maneuver sequences for a four-spacecraft formation was considered. The objective was to determine fuel-optimal configuration trajectories that would transfer a four spacecraft formation from an initial parking orbit to a desired terminal reference orbit while satisfying particular tetrahedral formation constraints. The configuration problem was solved numerically using the *Gauss pseudospectral method.* In particular, two versions of the minimum-fuel configuration problem were considered. In the first scenario each spacecraft was given two maneuver opportunities to complete the transfer. In the second scenario each spacecraft was given only one maneuver opportunity. Optimal solutions, and the discovery of the fuel-optimal *orientation* was found for each scenario. In addition, the first-order optimality conditions (or KKT conditions) obtained from the Gauss pseudospectral method were shown for a multiple-phase optimal control problem. These KKT conditions were compared with the optimality conditions obtained by discretizing the calculus of variations problem via the Gauss pseudospectral method. A mapping between the KKT multipliers and the HBVP costates was used to determine an estimate of the optimal control for a four spacecraft formation flying problem. It was found that the control obtained from the NLP was in excellent agreement with the control estimated by the discretized HBVP.

In addition, a minimum-fuel trajectory was designed to transfer four spacecraft from a degraded tetrahedral formation into a tetrahedron that satisfied certain geometric quality constraints for a portion of its orbit. This problem was posed as an optimal control problem and was solved via direct transcription using the Gauss pseudospectral method. The final state of the solution was propagated three weeks into the future to examine the potential degradation of the formation. Observing that the formation degraded in an unacceptably short period of time, a modified optimal control problem was posed that provided margin in the quality factor. It was found that the solution to this modified problem satisfied the tetrahedral quality constraints for this entire three week period beyond the final time of the trajectory design. The results in this chapter provide insight into the relative motion of tetrahedral formations and demonstrate the applicability of the Gauss pseudospectral method to spacecraft formation flying problems.

[This page intentionally left blank.]

Appendix: Computation of Glassmeier Quality Factor

In order to compute the Glassmeier quality factor at a given point, a few quantities must be calculated. First, using the mesocenter in Eq. (7.28), the following volumetric tensor is calculated:

$$\bar{\bar{R}}^{I/mc} = \frac{1}{4}\sum_{i=1}^{4} \mathbf{r}_{mci}\mathbf{r}_{mci}^{T} \tag{7.37}$$

where

$$\mathbf{r}_{mci} = \mathbf{r}^{(i)} - \bar{\mathbf{r}}, \qquad i = 1, \ldots, 4 \tag{7.38}$$

Using $\bar{\bar{R}}^{I/mc}$, the volume of the actual tetrahedron is computed as

$$V_a = \frac{8}{3}\sqrt{|\bar{\bar{R}}^{I/mc}|} \tag{7.39}$$

where $|\bar{\bar{R}}^{I/mc}|$ is the determinant of the volumetric tensor. Next, the surface area of the actual tetrahedron is computed as follows. First, let

$$\begin{gathered} S_1 = \tfrac{1}{2}|\mathbf{r}^{(2)} \times \mathbf{r}^{(3)}|, \quad S_2 = \tfrac{1}{2}|\mathbf{r}^{(1)} \times \mathbf{r}^{(3)}|, \quad S_3 = \tfrac{1}{2}|\mathbf{r}^{(1)} \times \mathbf{r}^{(2)}| \\ S_4 = \tfrac{1}{2}|(\mathbf{r}^{(2)} - \mathbf{r}^{(1)}) \times (\mathbf{r}^{(3)} - \mathbf{r}^{(1)})| \end{gathered} \tag{7.40}$$

Then the volume of the actual tetrahedron is given as

$$SA_a = \sum_{i=1}^{4} S_i \tag{7.41}$$

Lastly, the ideal volume and surface area equations are solely a function of the average side length, and are expressed below.

$$V^* = \frac{\sqrt{2}}{12}(\bar{L})^3, \quad SA^* = \sqrt{3}(\bar{L})^2 \tag{7.42}$$

Chapter 8

Conclusions

This chapter provides a brief recap of the key contributions of this thesis, and expounds on potential areas for future work.

8.1 Thesis Summary

The Gauss pseudospectral method, a recently developed direct transcription method for solving optimal control problems, is presented in its most current form. This formulation has been modified from the original formulation in Ref. 6 by increasing its applicability to problems with path constraints, explicitly including the terminal state in the NLP, and improving the control estimate at the boundaries. The Gauss pseudospectral method is implemented as a global approach, meaning the trajectory is approximated using a global interpolating polynomial across the entire interval, as opposed to using piecewise polynomials to locally approximate segments of the trajectory. However, since both implementations can be found in the literature for pseudospectral methods, a rigorous analytic and numerical comparison is conducted where the GPM is implemented in both a global and local fashion on several problems. The results validate the use of a global approach (or semi-global approach in the presence of known discontinuities).

A comparison is made between three commonly used pseudospectral methods: the Legendre, Radau, and Gauss pseudospectral methods. In order to provide a fair comparison, the NLP solver SNOPT[45] is used for each of the discretizations. Furthermore, all three methods are implemented using the same version of MATLAB® and the initial guesses provided for all examples are identical. The accuracy and and computational performance between all the three methods is compared, and a guideline is created to help determine under what circumstances each method is appropriate.

This thesis also provides a discussion on past and present attempts to prove the convergence of the NLP solution towards the exact optimal solution as the number of discretization points increases towards infinity.

Lastly, this thesis applies the Gauss pseudospectral method to tetrahedral spacecraft formation flying problems. Both orbit insertion problems and reconfiguration problems are formulated, solved, and analyzed, thus creating a greater understanding for the relative motion and maneuvers of minimum-fuel spacecraft formations.

8.2 Future Work

This work primarily focuses on the viability of the Gauss pseudospectral method as a preferred numerical method for solving optimal control problems. Specifically, the GPM's capability to obtain accurate, efficient solutions to very complex, non-intuitive problems is scrutinized, improved, and compared against several other competitive methods. Yet more advancements can be made to advance the GPM even further. This section will highlight some of the potential areas where the GPM can be improved.

Real-Time Path Planning

Real-time path planning is one of the hottest topics in this field right now. Simply determining an optimal trajectory *a priori*, and then implementing a controller that will bring the vehicle back to this optimal trajectory in the presence of disturbances has been done several times and in fact is sub-optimal. Researchers are now looking at control algorithms that attempt to re-optimize the trajectory once a vehicle deviates from the original optimal trajectory. Thus the ability to formulate and solve an optimal control problem in real-time is highly desirable. Simplified optimal control problems, like those that form linear programs when discretized, have already been used in real-time control. However, there is still much work to be done in terms of implementing nonlinear programs that can be consistently and reliably solved in real-time. That said, pseudospectral methods have the potential to be used in real-time due to the extremely high solution accuracy obtained with relatively few nodes (meaning a smaller problem size). Ross[98] has initiated work in real-time optimization using pseudospectral methods and it is likely others will follow his example. Along similar lines, research is underway[1] to try and reduce the computational burden of pseudospectral methods. Many real-time applications have a nonconvex feasible region, meaning it may be quite difficult (or computationally expensive) to start from an arbitrary initial guess and converge to the globally optimal solution. Research on algorithms that help determine the initial guess for the GPM transcription may significantly reduce overall computation times and bring pseudospectral methods closer to real-time applications.

Convergence

As mentioned in Chapter 3, the issue of convergence for the Gauss pseudospectral methods, and pseudospectral methods in general, is currently unresolved. Due to the global nature of pseudospectral methods, this proof is inherently different than many convergence proofs of direct transcription methods. Moreover, the inclusion of a cost function makes this convergence proof more complicated than those of boundary value problems. Yet the evidence suggests that a convergence proof is tractable, and that the convergence rate for smooth problems using the GPM is extremely good.

Mesh Refinement

The high accuracy and convergence rate for the GPM requires the assumption that the problem is sufficiently smooth. However, many real world problems have either state discontinuities, control discontinuities, or both. Often the number and location of these discontinuities are unknown. A mesh refinement algorithm that could detect discontinuities (and insert a phase interface in response), and also determine an appropriate number of discretization points to use would be extremely helpful, and would significantly increase the effectiveness of pseudospectral methods.

Improving the Control

As was seen in Chapter 4, Pontryagin's minimum principle was used to create a more accurate estimate of the boundary control. This approach works because the state and costate approximation are more accurate than the control approximation, not only at the boundaries, but at the interior points as well. If this approach was able to improve the control at the boundaries, it may also be able to improve the control at the interior LG points as well, or possibly even at arbitrary points between the LG points. This result was also noticed by Hager,[56] who first suggested *a posteriori* computation of the control. More research must be done in order to determine to what extent this algorithm should be used when determining the most accurate control profile.

Bibliography

[1] Aoude, G., and How, J., "Two-Stage path Planning Approach for Designing Multiple Spacecraft Reconfiguration Maneuvers," to appear at the 20^{th} International Symposium on Space Flight Dynamics, Sept, 2007.

[2] Ascher, U., Mattheij, R., Russel, R., *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM-Philadelphia, 1995.

[3] Athans, M., and Falb, P. L., *Optimal Control*, McGraw-Hill, New York, 1966, pp. 284-351.

[4] Axelsson, O., "Global Integration of Differential Equations Through Lobatto Quadrature," *BIT*, Vol. 4 (1964) pp. 69-86

[5] Battin, R.H., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Press, New York, 1987, pp. 490-493 and p. 119.

[6] Benson, D., *A Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. Dissertation, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, November 2004.

[7] Benson, D.A., Huntington, G.T., Thorvaldsen, T.P., and Rao, A.V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2006–6358, Keystone, Colorado, August 21–24, 2006.

[8] 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.

[9] Bertsekas, D., *Nonlinear Programming*, Athena Scientific, Belmont, MA,1995.

[10] Betts, J.T. and Frank, P.D., "A Sparse Nonlinear Optimization Algorithm," *Journal of Optimization Theory and Applications*, Vol. 82, pp. 519-541, 1994.

[11] Betts, J.T. and Huffman, W.P., "Sparse Optimal Control Software - SOCS," Mathematics and Engineering Analysis Library Report, MEA-LR-085, Boeing Information and Support Services, P. O. Box 3797, Seattle, WA, 98124-2297, 15 July 1997.

[12] Betts, J.T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[13] Betts, J.T., "A Direct Approach to Solving Optimal Control Problems," *Computing in Science and Engineering*, May-June 1999, pp. 73-75

[14] Betts, J.T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Society for Industrial and Applied Mathematics Press, 2001.

[15] Betts, J., Biehn, N., and Campbell, S., "Convergence of Nonconvergent IRK Discretizations of Optimal Control Problems with State Inequality Constraints," *SIAM Journal of Scientific Computing*, Vol. 23, No. 6, 2002, pp. 1981–2007.

[16] Bock, H.G. and Plitt, K.J., " A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems," *IFAC 9th World Congress*, Budapest, Hungary, 1984.

[17] Breger, L.,et al., "Distributed Control of Formation Flying Spacecraft Build on OA," AIAA Guidance, Nav, & Control Conference, Austin, TX, Aug 2003.

[18] Breger, L. and How, J., "Formation Flying Control for the MMS Mission using GVE-based MPC," *Proceedings of the 2005 IEEE Conference on Control Applications*, Aug, 2005.

[19] Breger, L., How., "J2-Modified GVE-Based MPC for Formation Flying Spacecraft," *AIAA Guidance, Navigation, and Control Conference*, San Francisco, CA, Aug. 15-18, 2005.

[20] Bryson, A., Ho, Y., *Applied Optimal Control*, Hemisphere Publishing Co., New York, 1975.

[21] Bryson, A., "Optimal Control–1950 to 1985", *IEEE Control Systems Magazine*, Vol 16, Issue 3, June 1996, pp. 26-33

[22] Bryson, A.E., *Dynamic Optimization*, Addison Wesley Longman, Menlo Park, CA, 1999, pp. 105–106.

[23] Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A., *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, 1988.

[24] Chobotov, V.A., Ed., *Orbital Mechanics*, AIAA Press, New York, 1991, p. 18.

[25] Clemente, D., Atkins, E., "Optimization of a Tetrahedral Satellite Formation," AIAA Guidance, Nav, & Control Conference, Aug 2004.

[26] Cuthrell, J.E. , and Biegler, L.T., "On the Optimization of DIfferential-Algebraic Process Systems," *AIChE Journal*, Vol.33, No.8, pp.1257-1270, Aug. 1987.

[27] Cuthrell, J.E. and Biegler, L.T., "Simultaneous Optimization and Solution Methods for Batch Reactor Control Profiles," *Computers and Chemical Engineering*, Vol. 13, Nos. 1/2, 1989, pp.49–62.

[28] Davis, P., *Interpolation & Approximation*, Dover Publications, 1975.

[29] Davis, P., Rabinowitz, P., *Methods of Numerical Integration*, Academic Press, 1984.

[30] DeBoor, C., *A Practical Guide to Splines*, Springer-Verlag, New York, NY, 1978.

[31] Dontchev, A. L., Hager, W., Veliov, V., "Second-Order Runge-Kutta Approximations in Control Constrained Optimal Control," *SIAM Journal on Numerical Analysis*, Vol. 38, No. 1, June 2000, pp. 202-226.

[32] Euler, L., *Elementa Calculi variationum*, Originally published in *Novi Commentarii academiae scientiarum Petropolitanae 10*, 1766, pp. 51-93

[33] Elnagar, G., Kazemi, M., Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, October 1995.

[34] Elnagar, G.E., and Kazemi, M.A., "Pseudospectral Legendre-based optimal computation of nonlinear constrained variational problems," *Journal of Computational and Applied Mathematics*, Vol. 88, pp.363-375, 1997.

[35] Elnagar, G. and Kazemi, M., "Pseudospectral Chebyshev Optimal Control of Constrained Nonlinear Dynamical Systems," *Computational Optimization and Applications*, Vol. 11, 1998, pp. 195–217.

[36] Engelsone, A., Campbell, S., Betts, J., "Order of Convergence in the Direct Transcription Solution of Optimal Control Problems," *Proceedings of the 44th Conference on Decision and Control*, Seville, Spain, Dec., 12-15, 2005.

[37] Enright, P.J. and Conway, B.A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002.

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

[39] Fahroo, F. and Ross, I.M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, March-April 2002, pp. 270-277.

[40] Fahroo, F. and Ross, I.M., "Direct Trajectory Optimization by a Chebyshev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, January-February 2002, pp. 160–166.

[41] Fahroo, F. and Ross, I., "Pseudospectral Methods for Infinite Horizon Nonlinear Optimal Control Problems," *2005 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2005-6076, San Francisco, CA, August 15–18, 2005.

[42] Fahroo, F., Ross, I.M., "On Discrete-Time Optimality Conditions for Pseudospectral Methods," AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, CO, Aug 2006.

[43] Fornberg, B., *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, 1998.

[44] Gear, G., *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, New Jersey, 1971.

[45] Gill, P.E., Murray, W., and Saunders, M.A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[46] Gill, P.E., *Users Guide for SNOPT Version 7, A Fortran Package for Large-Scale Nonlinear Programming,* University of California, San Diego, La Jolla, CA, Sept 04.

[47] Gim, D. , Alfriend, K., "Criteria for Best Configuration and Sub-optimal Reconfiguration for MMS Mission," AAS Spaceflight Mechanics Meeting, Maui, HI, Feb. ,2004.

[48] Gong, Q., Kang, W., and Ross, I., "A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems", IEEE Transactions on Automatic Control, Vol. 51, No. 7, July 2006.

[49] Gong, Q., Ross, I., Kang, W., and Fahroo, F., "On the Pseudospectral Covector Mapping Theorem for Nonlinear Optimal Control," IEEE Conference on Decision and Control, Dec 13-15, 2006.

[50] Gottlieb, D. and Orszag, S., *Numerical Analysis of Spectral Methods: Theory and Applications,* Society for Industrial and Applied Mathematics, 1977.

[51] Gottlieb, D., "The Stability of Pseudospectral-Chebyshev Methods," *Mathematics of Computation*, Vol. 36, No. 153, Jan 1981, pp.107-118.

[52] Gottlieb, D., Lustman, L., Tadmor, E., "Convergence of Spectral Methods for Hyperbolic Initial-Boundary Value Systems", *SIAM Journal on Numerical Analysis*, Vol. 24, No. 3, June 1987, pp. 532–537.

[53] Guzman, J., Schiff, C., "A Preliminary Study for a Tetrahedron Formation: Quality Factors and Visualization," AIAA Guid, Nav, & Contr, Conf., 2002

[54] Guzman, J.J., "Tetrahedron Formation Control," *Journal of the Astronautical Sciences*, Vol. 51, No. 4, 2003, pp. 419–431.

[55] Hager, W.W., "Rate of Convergence for Discrete Approximations to Unconstrained Control Problems," *SIAM Journal on Numerical Analysis*, Vol. 13, 1976, pp. 449–471.

[56] Hager, W.W., "Runge-Kutta Methods in Optimal Control and the Transformed Adjoint System," *Numer. Math.*, Vol. 87, 2000, pp. 247–282.

[57] Hargraves, C.R. and Paris, S.W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338-342.

[58] Herman, A.L. and Conway, B.A., "Direct Optimization Using Collocation Based on High Order Gauss-Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599.

[59] Holmstrom, K., Goran, A.O., and Edvall, M.M., "User's Guide for TOMLAB/SNOPT," TOMLAB Optimization, 5 July 2005.

[60] Hoskins, A. and Atkins, E., "Spacecraft Formation Optimization with a Multi-Impulse Design," *Proceedings of the 2005 AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2005–5835, August 15–18, 2005, San Francisco, CA.

[61] Haung, W., Heping, M., Sun, W., "Convergence Analysis of Spectral Collocation Methods for a Singular Differential Equation," *SIAM Journal of Numerical Analysis*, Vol. 41, No. 6, pp. 2333-2349.

[62] Hughes, S.P., “Orbit Design for Phase I and Phase II of the Magnetospheric Multiscale Mission,” NASA Goddard Flight Center. 2004.

[63] Hughes, S. ”Formation Tetrahedron Design for Phase 1 of the Magnetospheric Multiscale Mission,” Flight Dynamics Analysis Branch, NASA Goddard Space Flight Center, 2004.

[64] Hull, D., “Conversion of Optimal Control Problems into Parameter Optimization Problems,” AIAA Guidance, Navigation, and Control Conference, San Diego, CA, July 29-31, 1996

[65] Hull, D.G., *Optimal Control Theory for Applications*, Springer-Verlag, New York, 2003.

[66] Huntington, G.T., and Rao, A.V., “Optimal Configuration of Spacecraft Formations via a Gauss Pseudospectral Method,” *AAS Spaceflight Mechanics Meeting*, AAS 05-103, Copper Mountain, CO, Jan 2005.

[67] Huntington, G.T., Benson, D.A., and Rao, A. V., “Post-Optimality Evaluation and Analysis of a Formation Flying Problem via a Gauss Pseudospectral Method,” *Proceedings of the 2005 AAS/AIAA Astrodynamics Specialist Conference*, AAS Paper 05-339, Lake Tahoe, California, August 7–11, 2005.

[68] Huntington, G.T. and Rao, A.V., “Optimal Reconfiguration of a Spacecraft Formation via a Gauss Pseudospectral Method,” *Proceedings of the 2005 AAS/AIAA Astrodynamics Specialist Conference*, AAS Paper 05–338, Lake Tahoe, California, August 7–11, 2005.

[69] Huntington, G. T. , Benson, D. A., How, J. P., Kanizay, N., Darby, C. L., and Rao, A. V., “Computation of Boundary Controls Using a Gauss Pseudospectral Method,” *2007 Astrodynamics Specialist Conference*, Mackinac Island, Michigan, August 21–24, to appear.

[70] Huntington, G. T., Benson, D. A, and Rao, A. V., “Design of Optimal Tetrahedral Spacecraft Formations,” *The Journal of the Astronautical Sciences*, Vol. 55, No. 2, April–June 2007, to appear.

[71] Kameswaran, S. and Biegler, L.T., "Convergence Rates for Dynamic Optimization Using Radau Collocation," *SIAM Conference on Optimization*, Stockholm, Sweden, 2005.

[72] Kameswaran, S. and Biegler, L.T., "Convergence Rates for Direct Transcription of Optimal Control Problems at Radau Points," *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, June 2006.

[73] Kirk, D.E., *Optimal Control Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1970.

[74] Lewis, F. and Syrmos, V., *Optimal Control*, John Wiley & Sons, New York, 1995.

[75] Mailhe, L.M. and Guzman, J.J., "Initialization and Resizing of Formation Flying using Global and Local Optimization Methods," *Proceedings of the 2004 IEEE Aerospace Conference*, Vol. 1, March 6–13, 2004, pp. 556.

[76] Matrell, C.A. and Lawton, J.A., "Adjoint Variable Solutions via an Auxiliary Optimization Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 6, 1995, pp. 1267–1272.

[77] *MATALB, version 2006a: The Language of Scientific Computing*, Mathworks, Natick, MA, 2006.

[78] Milam, M., Mushambi, K., Murray, M., "A New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems," *Proceedings of the* 39th *IEEE Conference on Decision and Control*, Sydney, Australia, Dec., 2000.

[79] Milam, M., "Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems," Ph.D. Thesis, California Institute of Technology, Pasadena, CA, 2003.

[80] Oberle, H.J. and Grimm, W., "BNDSCO: A proggram for the numerical solution of optimal control problems," Technical Report 515, DFVLR, Hamburg, October 2001.

[81] Paris, S.W., Riehl, J.P., Sjuaw, W..K., "Enhanced Procedures for Direct Trajectory Optimization Using Nonlinear Programming and Implicit Integration," AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone CO, Aug 2006

[82]Pontryagin, L.S., Boltyanskii, V., Gamkrelidze, R., Mischenko, E., *The Mathematical Theory of Optimal Processes.* Wiley Interscience, New York City, NY, 1962.

[83]Quarteroni, A., Zampeiri, E., "Finite Element Preconditioning for Legendre Spectral Collocation Approximations to Elliptic Equations and Systems", *SIAM Journal on Numerical Analysis*, Vol. 29, No. 4, Aug. 1992, pp.917-936.

[84]Rao, A.V. and Clarke, K.A., "Performance Optimization of a Maneuvering Re-entry Vehicle Using a Legendre Pseudospectral Method," *2002 AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2002-4885, Monterey, CA August 5–8, 2002.

[85]Rao, A.V., "Extension of a Pseudospectral Legendre Method to Non-Sequential Multiple-Phase Optimal Control Problems", AIAA Guidance, Navigation, and Control Conference, AIAA Paper No. 2003-5634, Austin, TX, Aug. 11-14, 2003.

[86]Rao, A.V., *GPOCS: A MATLAB Implementation of the Gauss Pseudospectral Method for Solving Non-Sequential Multiple-Phase Optimal Control Problems*, University of Florida, 2007.

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

[88]Robert, P., et.al., "Tetrahedron Geometric Factors," Analysis Methods for Multi-Spacecraft Data (G. Paschmann and P. Daly, eds.), pp. 323-348, Nordwijk, The Netherlands: ISSI Report SR-001, ESA Publications Division, 1998.

[89]Ross, I.M. and Fahroo, F., "A Pseudospectral Transformation of the Covectors of Optimal Control Systems," *Proceedings of the 1st IFAC/IEEE Symposium on Structure and Control*, Prague, Czech Republic, August 2001.

[90]Ross, I.M., Fahroo, F., "Convergence of Pseudospectral Discretizations of Optimal Control Problems," Proceedings of the 40^{th} IEEE Conference on Decision and Control, Orlando, FL, Dec. 2001.

[91] Ross, I., King, J., and Fahroo, F., "Designing Optimal Spacecraft Formations," *2002 AIAA/AAS Astrodynamics Specialist Conference*, AIAA Paper 2002-4635, Monterey, CA, August 5–8, 2002.

[92] Ross, I.M., and Fahroo, F. , "A Direct Method for Solving Nonsmooth Optimal Control Problems", Proceedings of the 2002 World Congress of the International Federation on Automatic Control, IIAC, Barcelona, July 2002.

[93] Ross, I. M. and Fahroo, F., "A Perspective on Methods for Trajectory Optimization," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*, Montery, CA, August 2002. Invited Paper AIAA 2002-4727.

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

[95] 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.

[96] Ross, I.M. and D'Souza,C., "Hybrid Optimal Control Framework for Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 28,No. 4,July-August 2005.

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

[98] Ross, I.M., Fahroo, F., "Issues in real-time computation of optimal control," *Journal of Mantematical and Computer Modeling*, Vol. 43, 2006, pp. 1172–1188.

[99] Rossiter, J.A., *Model Predictive Control: A Practical Approach*, CRC Press, Boca Raton, FL, 2003

[100] Scharf, D.P., Hadaegh, F.Y., and Ploen, S.R., "A Survey of Spacecraft Formation Flying Guidance and Control (Part I): Guidance," *Proceedings of the American Control Conference*, Denver, CO, pp. 1733–1739, 2003.

[101] Scharf, D.P., Hadaegh, F.Y., and Ploen, S.R., "A Survey of Spacecraft Formation Flying Guidance and Control (Part II): Control," *Proceedings of the American Control Conference*, Boston, MA, pp. 2976–2985, 2004.

[102] Schwartz, A., “Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems,” Ph. D. Thesis, University of California, Berkeley, 1996.

[103] Seywald, H. and Kumar, R.R., “Method for Automatic Costate Calculation,” *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, 1996, pp. 1252–1261.

[104] Silva-Beard, A., Flores-Tlacuahuac, A., and Arrieta-Camacho, J.J., "Dynamic Trajectory Optimization Between Unstable Steady-States of a Class of CSTRs," *Proceedings of the XII European Symposium on Computer Aided Process Engineering*, Jan Van Shijndel (ed.), Elsevier Sccience Publishing Co., The Hague, Ned, 2002.

[105] Stanton, S. and Proulx, R., “Optimal Orbital Transfer Using a Legendre Pseudospectral Method,” *AAS/AIAA Astrodynamics Specialist Conference*, AAS-03-574, Big Sky, Montana, August 3–7, 2003.

[106] Stoer, J., Bulirsch, R., *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1993.

[107] Stryk, O., “Numerical Solution of Optimal Control Problems by Direct Collocation,” *Optimal Control*, ed. R. Bulirsch, A. Miele and J. Stoer, Birkhauser Verlag, Basel, 1993, pp. 129–143.

[108] Tadmor, E., "The Exponential Accuracy of Fourier and Chebyshev Differencing Methods," *SIAM Journal of Numerical Analysis*, Vol. 23, No. 1, Feb 1986, pp. 1–10.

[109] Teo, K.L., Goh C.J., and Wong, K.H., *A Unified Computational Approach to Optimal Control Problems*, Longman Scientific & Technical, 1991.

[110] Thorvaldsen, T., Proulx, R., and Ross, I. M., “A Fast Accurate Method for Low-Thrust Trajectory Optimization,” *Proceedings of the 16th Space Flight Dynamics Conference,* Pasadena, CA, December 2001.

[111] Tillerson, M., Breger, L., How, J.P., "Distributed Coordination and Control of Formation Flying Spacecraft," *American Control Conference*, Vol. 2, June 4-6, 2003, pp. 1740-1745.

[112] Trefethen, L.N., *Spectral Methods in MATLAB*, SIAM Press, Philadelphia, 2000.

[113] Tsuda, Y., "Global Optimization of Maneuver Schedule for Multiple Spacecrafts Flying in Formation," *55th International Astronautical Congress*, Vancouver, Canada, IAC-04-A.2.01, 2004.

[114] Vallado, D., *Fundamentals of Astrodynamics and Applications*, McGraw-Hill, 1997.

[115] Vasantharajan, S., and Biegler, L.T., "Simultaneous Strategies for Optimization of Differential-Algebraic Systems with Enforcement of Error Criteria," *Computers and Chemical Engineering*, Vol. 14, No. 10, pp.1083-1100, 1990.

[116] Villadsen, J., and Michelsen, M., "Solution of differential equation models by polynomial approximation," Prentice Hall, New Jersey, 1978.

[117] Vlassenbroeck, J. and Van Doreen, R., "A Chebyshev Technique for Solving Nonlinear Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 33, No. 4, 1988, pp. 333–340.

[118] Vlassenbroeck, J., "A Chebyshev Polynomial Method for Optimal Control with State Constraints," *Automatica*, Vol. 24, 1988, pp. 499–506.

[119] Vlases, W.G., Paris, S.W., Lajoie, R.M., Martens, M.J., and Hargraves, C.R., "Optimal Trajectories by Implicit Simulation," Boeing Aerospace and Electronics, Technical Report WRDC-TR-90-3056, Wright-Patterson Air Force Base, 1990.

[120] Voigt, R.G., Gottlieb, D., Hussaini, M.Y., (eds.) *Spectral Methods for Partial Differential Equations,* SIAM-CBMS, Philadelphia, 1984.

[121] Von Stryk, O., "User's Guide for DIRCOL 2.1: A Direct Collocation Method for the Numerical Solution of Optimal Control Problems," Technische Universitat Darmstadt, 1999.

[122] Well, K., *Graphical Environment for Simulation and Optimization*, Department of Optimization, Guidance, and Control, Stuttgart, Germany, 2002.

[123] Wertz, J.R. and Larsen, W.J., *Space Mission Analysis and Design*, 3rd Edition, Microcosm Press, El Segundo, CA and Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999, p. 694.

[124] Wiesel, W.E., *Spaceflight Dynamics*, McGraw-Hill Companies, Inc., New York, 1997, pp. 75-76.

[125] Willems, J.C., "1696: The Birth of Optimal Control", *Proceedings of the 35th Conference on Decision and Control*, Kobe, Japan, December 1996.

[126] Williams, P., "Jacobi Pseudospectral Method for Solving Optimal Control Problems", *Journal of Guidance*, Vol. 27, No. 2,2003

[127] Williams, P., "A Comparison of Differentiation and Integration Based Direct Transcription Methods", AAS/AIAA Space Flight Mechanics Conference, Copper Mountain, CO, Jan. 23-27, 2005. Paper AAS 05-128.

[128] Williams, P., "Hermite-Legendre-Gauss-Lobatto Direct Transcription Methods In Trajectory Optimization," Advances in the Astronautical Sciences. Vol. 120, Part I, pp. 465-484. 2005

[129] Williams, P., "A Gauss-Lobatto Quadrature Method For Solving Optimal Control Problems," Australia Mathematical Society, ANZIAM J.47, pp.C101-C115, July, 2006.

[130] Williamson, W.E., "Use of Polynomial Approximations to Calculate Suboptimal Controls," AIAA J. 9 (1971) 2271 - 2273.